

Chapter 2

Flow Monitoring

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.

This feature guide covers the following topics:

- Overview on page 68

 - Passive Flow Monitoring on page 69

 - Active Flow Monitoring on page 70

- System Requirements on page 70

- Terms and Acronyms on page 73

- Configure Passive Flow Monitoring on page 74

 - Hardware and Software Considerations on page 93

 - Example: Passive Flow Monitoring Configuration on page 95

 - Check Your Work on page 102

 - Example: Flow Collector Interface Configuration on page 111

 - Check Your Work on page 118

- Configure Active Flow Monitoring on page 123

 - Example: Sampling Configuration on page 134

 - Check Your Work on page 136

 - Example: Sampling and Discard Accounting Configuration on page 138

 - Check Your Work on page 141

 - Example: Multiple Port Mirroring with Next-Hop Groups Configuration on page 143

[cflowd Output Formats on page 146](#)

[For More Information on page 156](#)

[Revision History on page 157](#)

Overview

Using a Juniper Networks M-series router, a selection of Physical Interface Cards (PICs)—including the Monitoring Services PIC, Monitoring Services II PIC, or Adaptive Services PIC—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 IPv4 traffic flows between source and destination nodes in your network.

- Sample all incoming IPv4 traffic on the monitoring interface and present the data in cflowd record format.

- Encrypt or tunnel outgoing cflowd records, intercepted IPv4 traffic, or both.

- Direct filtered traffic to different packet analyzers and present the data in its original format.

- Intercept unwanted traffic, discard it, and perform accounting on the discarded packets.

There are two main types of flow monitoring:

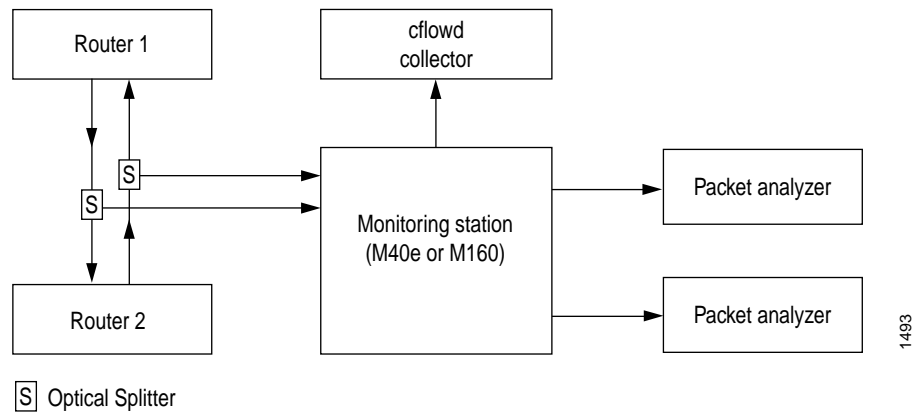
- [Passive Flow Monitoring on page 69](#)

- [Active Flow Monitoring on page 70](#)

Passive Flow Monitoring

The M40e or M160 router 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 7 shows a typical topology for the passive flow monitoring application.

Figure 7: 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, which is an M40e or M160 router. The optical cable connects only the receive port on the monitoring station, never the transmit port. This configuration allows the monitoring station to receive traffic 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 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 cflowd version 5 format, and the records are then exported to the cflowd collector.

If you are performing lawful interception of packets transiting between the two routers, the Internet Processor II ASIC filters the incoming traffic and forwards it to the Tunnel Services PIC. Filter-based forwarding is then applied to direct the traffic to the packet analyzers. Optionally, the intercepted traffic or the cflowd records can be encrypted by the ES PIC and then sent to their destination. Also, cflowd records can be processed by a flow collector.

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.

1493

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.

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.

System Requirements

Passive and active flow monitoring applications are supported on the PICs shown in Table 5.

Table 5: Passive and Active Flow Monitoring PIC Support

PIC Type and Service	M5 / M7i	M10 / M10i	M20	M40e	M160
Monitoring Services PIC: passive flow monitoring	No	No	No	Yes	Yes
Monitoring Services PIC: active flow monitoring	Yes	Yes	Yes	Yes	Yes
Monitoring Services II PIC: passive flow monitoring	No	No	No	Yes	Yes
Monitoring Services II PIC: flow collection services	No	No	No	Yes	Yes
Adaptive Services PIC: active flow monitoring	Yes	Yes	Yes	Yes	Yes

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

JUNOS Release 6.4 or later for support of the next-hop IP address field in cflowd version 5 records

JUNOS Release 6.2 or later for ATM2 intelligent queuing (IQ) interface passive monitoring, flow collection services, and MPLS label stripping

JUNOS Release 6.1 or later for MPLS passive monitoring

JUNOS Release 6.0 or later for the Monitoring Services II PIC

JUNOS Release 5.7 or later for the automatic insertion of autonomous system (AS) numbers and SNMP index values for input and output interfaces into cflowd records

JUNOS Release 5.4 or later for the Monitoring Services PIC

M40e or M160 router with an Internet Processor II ASIC

Type 1 enhanced FPCs

Two optical splitters

One Monitoring Services-related PIC for every OC-3 interface worth of monitored traffic

A Tunnel Services PIC (required if you wish to send traffic to more than one analyzer)

A SONET/SDH PIC (OC-3, OC-12, or OC-48) or ATM2 IQ PIC (OC-3 or OC-12) for the input interface

Outgoing PICs to connect to the cflowd collector or packet analyzer

cflowd version 5 collector

ES PIC and packet analyzers (optional)

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

JUNOS Release 6.0 or later for the Adaptive Services PIC

JUNOS Release 5.7 or later for the automatic insertion of autonomous system (AS) numbers and SNMP index values for input and output interfaces into cflowd records, port mirroring to multiple ports, and discard accounting

JUNOS Release 5.6 or later for the Monitoring Services PIC

M5, M7i, M10, M10i, M20, M40e, or M160 router with an Internet Processor II ASIC

Type 1 enhanced FPCs

Two PICs of your choice: One to receive incoming traffic and one to forward outgoing traffic (the second PIC is not necessary for discard accounting)

Export PICs to connect to the cflowd collector or packet analyzer

Tunnel Services PIC (required for multiple port mirroring or multi-interface load balancing)

cflowd version 5 or 8 collector

ES PIC and packet analyzers (optional)

Table 6, Table 7, and Table 8 on page 73 describe the specifications for the Monitoring Services PIC, Monitoring Services II PIC, and Adaptive Services PIC.

Table 6: 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—Service is not running Green—Service is running under acceptable load Amber—Service is overloaded

Table 7: 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—cflowd collector is not running Green—cflowd collector is running under acceptable load Amber—cflowd collector is overloaded

Table 8: 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—cflowd collector is not running Green—cflowd collector is running under acceptable load Amber—cflowd collector is overloaded

Terms and Acronyms

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*.

cflowd—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>.

ES PIC—PIC that handles encryption and security services (such as IPSec).

flow collector interface—Converted Monitoring Services II PIC that processes multiple cflowd records into compressed ASCII data files and exports these files to a File Transfer Protocol (FTP) server.

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

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

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

Configure Passive Flow Monitoring

When you want to monitor traffic passively in an M40e or M160 router, you can use either the Monitoring Services PIC or the Monitoring Services II PIC. The PICs receive passively monitored network traffic from a SONET/SDH or ATM2 IQ input interface, convert the received packets into cflowd records, and export them to a cflowd server for further analysis.

The key configuration hierarchy statement for passive flow monitoring is the monitoring statement found at the [edit forwarding-options] hierarchy level. At minimum, you must configure a VRF routing instance to direct the traffic to a monitoring services interface for cflowd 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 encrypt cflowd output before it is sent to a cflowd server for processing, or send cflowd records to a flow collector.

The following sections explain the variety of passive flow monitoring topics:

Monitor Traffic with a VRF Instance and a Monitoring Group on page 74

Copy and Redirect Traffic with Port Mirroring and Filter-Based Forwarding on page 81

Hardware and Software Considerations on page 93

Example: Passive Flow Monitoring Configuration on page 95

Check Your Work on page 102

Example: Flow Collector Interface Configuration on page 111

Check Your Work on page 118

Monitor 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 cflowd server for analysis. Complete the following tasks:

Specify a Firewall Filter to Select Traffic to Monitor on page 75

Configure Input Interfaces, Monitoring Services Interfaces, and Export Interfaces on page 75

Establish a VRF Instance for the Monitored Traffic on page 78

Configure a Monitoring Group to Send Traffic to the cflowd Server on page 78

Configure Policy Options on page 80

Option: Strip MPLS Labels on ATM and SONET/SDH Interfaces on page 80

Specify 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;
        }
      }
    }
  }
}
```

Configure Input Interfaces, Monitoring Services Interfaces, and Export Interfaces

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. For 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`).

In addition to passive monitor mode, apply the previously defined firewall filter to the interface with the filter statement at the [edit interfaces *so-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-0/1/0 {
    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;
        }
      }
    }
  }
}
```

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 cflowd 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 cflowd packets exit the monitoring station and are sent to the cflowd server:

```
[edit]
interfaces
  fe-3/0/0 {
    description "export interface to cflowd server";
    unit 0 {
      family inet;
      address 192.168.245.1/30;
    }
  }
}
```

Establish 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];
      }
    }
  }
}
```

Configure a Monitoring Group to Send Traffic to the cflowd Server

You collect cflowd 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 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 cflowd 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 cflowd 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 *output-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 cflowd server IP address and port number, include the cflowd `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 cflowd servers in a monitoring group and the IP address for each server must be unique. cflowd records are exported and load-balanced between all active cflowd 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 cflowd 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 “Copy and Redirect Traffic with Port Mirroring and Filter-Based Forwarding” on page 81.



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 “Configure Input Interfaces, Monitoring Services Interfaces, and Export Interfaces” on page 75.

```
[edit]
forwarding-options {
  monitoring group1 {
    family inet {
      output {
        export-format cflowd-version-5;
        flow-active-timeout 60;
        flow-inactive-timeout 30;
        cflowd 192.168.245.1 port 2055;
        cflowd 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;
        }
      }
    }
  }
}
```

Configure 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: Strip MPLS Labels on ATM 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 a SONET/SDH, ATM1, or ATM2 IQ interface, include the pop-all-labels statement at the [edit interfaces *so-fpc/pic/port* sonet-options mpls] or [edit interfaces *at-fpc/pic/port* atm-options mpls] hierarchy level.

To remove a specified number of labels from selected packets with MPLS labels, include the required-depth statement at the [edit interfaces *so-fpc/pic/port* sonet-options mpls pop-all-labels] or [edit interfaces *at-fpc/pic/port* atm-options mpls pop-all-labels] hierarchy level. A required-depth value of 1 removes labels from all packets containing only 1 MPLS label, a value of 2 removes labels from all packets containing only 2 MPLS labels, and a value of [1 2] removes labels from all packets containing either 1 or 2 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 the MPLS label stripping described here.

```
[edit]
interfaces {
  at-fpc/pic/port {
    atm-options {
      mpls {
        pop-all-labels {
          required-depth 1;
        }
      }
    }
  }
  so-fpc/pic/port {
    sonet-options {
      mpls {
        pop-all-labels {
          required-depth 2;
        }
      }
    }
  }
}
```

Copy and Redirect 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 the cflowd 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.

You can use a flow collector interface to process cflowd records into compressed ASCII data files and export these files to a File Transfer Protocol (FTP) server. Flow collection requires converting a Monitoring Services II PIC into a flow collector.

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

Specify Port Mirroring Input and Output on page 83

Create a Firewall Filter to Split the Port-Mirrored Traffic into Different Instances on page 84

Apply the Firewall Filter to a Tunnel PIC Interface on page 85

Use Filter-Based Forwarding to Export Monitored Traffic to Multiple Destinations on page 85

Configure a Routing Table Group to Add Interface Routes into the Forwarding Instance on page 86

Option: Use an ES PIC to Send Traffic to a Packet Analyzer on page 86

Option: Use a Flow Collector Interface to Process and Export Multiple cflowd Records on page 88

Specify 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 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 packets are sent. To configure, include the input and output statements at the [edit forwarding-options port-mirroring] hierarchy level. The traffic to be monitored is copied, port-mirrored, and sent to the packet analyzer for analysis.

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 input family inet] hierarchy level. For your output requirements, specify the target interface with the interface statement at the [edit forwarding-options sampling 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 output] hierarchy level.

```
[edit]
forwarding-options {
  port-mirroring {
    input {
      family inet {
        rate 1;
      }
    }
    output {
      interface vt-0/2/0.0;
      no-filter-check;
    }
  }
}
```

Create 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;
        }
      }
    }
  }
}
```

Apply the Firewall Filter to a Tunnel PIC Interface

Once the firewall filter is defined, apply it to 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;
        }
      }
    }
  }
}
```

Use 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;
      }
    }
  }
}
```

Configure 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 enables 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: Use 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, see 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 3.3.3.1/32 {
          destination 3.3.3.2;
        }
      }
    }
  }
}
```

```

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 "$9$qmQnuORrIMBlds2oiHOBIESe";
  }
}
}

```

Option: Use a Flow Collector Interface to Process and Export Multiple cflowd Records

You can manage multiple cflowd records with a flow collector interface. You create a flow collector interface from a Monitoring Services II PIC. The flow collector interface combines multiple cflowd 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 cflowd 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 cflowd 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 cflowd records are collected and processed. To specify the destination FTP server, include the destinations statement at the [edit services flow-collector] hierarchy level. You can specify up to two FTP server destinations and include the password for each configured server. If two FTP servers are configured, the first server in the configuration is the primary server and the second is a backup server.

File specifications—Presets 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 file name format, include the name-format statement at the [edit services flow-collector file-specification *file-name*] hierarchy level. 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.

Input interface-to-flow collector interface mappings—Matches an input interface with a flow collector interface and applies 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—Allows 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—Allows 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 cflowd 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 cflowd server as the destination for cflowd 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 System Basics Configuration Guide* or view the enterprise-specific Juniper Networks MIBs at <http://www.juniper.net/techpubs/software/junos/mibs.html>.

The following is the set of hierarchy levels ([edit chassis], [edit interfaces], [edit forwarding-options], and [edit services]) used for the flow collector service:

```
[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-or-ATM-interface-fpc/pic/port {
    description "input_interface";
    encapsulation ppp;
    unit 0 {
      passive-monitor-mode;
    }
  }
}

```

```

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 {
      name-format "name-0-date_time-IfAlias_generation-number.bcp.bi.gz";
      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";
      }
    }
  }
}

```

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 OC-3, OC-12, or OC-48 interfaces or ATM2 IQ OC-3 or OC-12 interfaces.

To monitor the flows in both directions for a single interface, the monitoring station must have two SONET/SDH or ATM2 IQ receive ports, one for each direction of flow. In Figure 7 on page 69, 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.

Each Monitoring Services or Monitoring Services II PIC can handle the volume of traffic that one OC-3 PIC can accommodate.

To monitor a fully loaded bidirectional SONET/SDH or ATM2 IQ OC-3 interface, the monitoring station must have two Monitoring Services PICs.

To monitor a fully loaded bidirectional SONET/SDH or ATM2 IQ OC-12 interface, the monitoring station must have four Monitoring Services PICs.

To monitor a fully loaded bidirectional SONET/SDH OC-48 interface, the monitoring station must have 16 Monitoring Services PICs.

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 cflowd 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 configure an inactivity timer for the monitoring station on a per-monitoring-group basis. The timer sets the length of time in seconds that the monitoring station allows a flow to be inactive before terminating the flow and exporting 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 a timeout for aging active flows on a per-monitoring-group basis. 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.

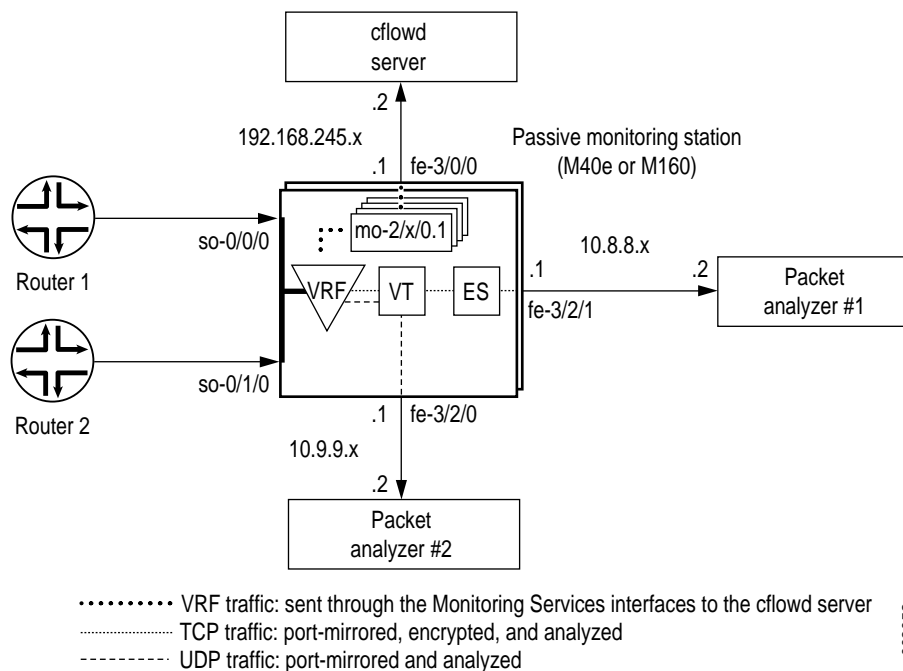
The interfaces on the monitoring station that collect intercepted traffic must be configured with Cisco HDLC or PPP encapsulation.

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 cflowd 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 cflowd version 5 records to multiple cflowd servers. You can configure up to eight servers and cflowd traffic is load-balanced between the servers in a round-robin fashion. If one of the servers ceases operation, cflowd traffic load-balances automatically between the remaining active servers. To configure, include up to eight cflowd statements at the [edit forwarding-options monitoring *group-name* output] hierarchy level.

Example: Passive Flow Monitoring Configuration

Figure 8: Passive Flow Monitoring—Topology Diagram



In Figure 8, 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 cflowd processing. The final cflowd packets are sent from the monitoring services interfaces out the fe-3/0/0 interface to a cflowd 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 cflowd 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 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;#This statement 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;#This statement 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 3.3.3.1/32 {
        destination 3.3.3.2;
      }
    }
  }
}
fe-3/0/0 { # Flow records exit here and travel to the cflowd server.
  description "export interface to the cflowd 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
        # packet groups.
      }
    }
  }
}
}
forwarding-options {
  monitoring group1 {# Monitored traffic is processed by the monitoring services
    family inet { # interfaces and cflowd records are sent to the cflowd server.
      output {
        export-format cflowd-version-5;
        flow-active-timeout 60;
        flow-inactive-timeout 30;
        cflowd 192.168.245.2 port 2055; # IP address and port for the 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.
input {
    family inet {
        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 "$9$qmQnuORrIMBlds2oiHOBIESe";
    }
  }
}
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;
        }
      }
    }
  }
}

```



```

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;
    }
  }
}
}

```

Check 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

```

You can also view passive flow monitoring status with 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@mon-station> 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)
+ = Active Route, - = Last Active, * = Both

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      *[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      *[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 9: 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@mon-station> 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 10: 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 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.

```

user@mon-station> 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 11: Output Fields for the show passive-monitoring memory Command

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

```

user@mon-station> 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 12: Output Fields for the show passive-monitoring status Command

Field	Explanation
Interface state	Indicates if 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 cflowd records, in seconds.
Export format	Configured export format (only cflowd v5 is currently supported).
Protocol	Protocol the PIC is configured to monitor (only IPv4 is currently supported).
Engine type	Configured engine type that is inserted in output cflowd packets.
Engine ID	Configured engine ID that is inserted in output cflowd 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 if the time stamp is in place.
Configuration set	Indicates if the monitoring configuration is set.
Route record set	Indicates if routes are being recorded
IFL SNMP map set	Indicates if logical interfaces are being mapped to an SNMP index.

```
user@mon-station> 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 13: 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 five 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 one minute. The number is a percentage obtained by dividing the time spent on active tasks by the total elapsed time.

```

user@mon-station> 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%

```

Example: Flow Collector Interface Configuration

Figure 9: Flow Collector Interface Topology Diagram

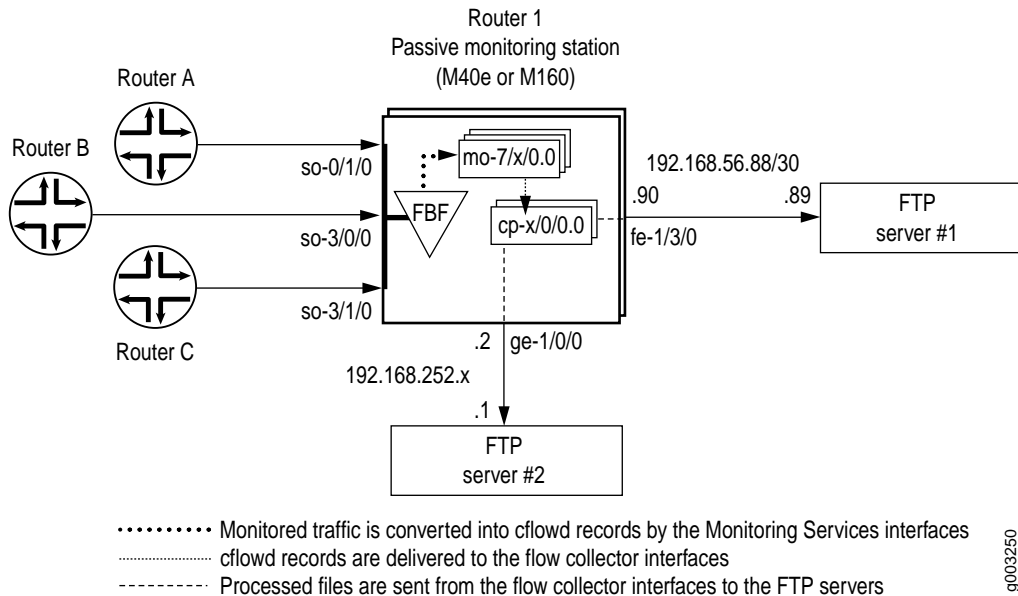


Figure 9 shows the path travelled by monitored traffic as it passes through the router. Packets arrive at input interfaces so-0/1/0, so-3/0/0, and so-3/1/0. The raw packets are directed into a filter-based forwarding routing instance and processed into cflowd records by the monitoring services interfaces mo-7/1/0, mo-7/2/0, and mo-7/3/0. The cflowd records are compressed into files at the flow collector interfaces cp-6/0/0 and cp-7/0/0 and sent to the FTP server for analysis. Finally, a mandatory class-of-service (CoS) configuration is applied to export channels 0 and 1 on the flow collector interfaces to manage the outgoing processed files.

```

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 cflowd records.
  unit 0 {
    family inet;
  }
}
mo-7/2/0 { # This is the second interface that creates cflowd records.
  unit 0 {
    family inet;
  }
}

```

```

mo-7/3/0 { # This is the third interface that creates cflowd records.
  unit 0 {
    family inet;
  }
}
so-0/1/0 { # This is the first input interface that receives traffic to be monitored.
  encapsulation ppp;
  unit 0 {
    passive-monitor-mode; # This allows the interface to be passively monitored.
    family inet {
      filter {
        input catch;      # The filter-based forwarding filter is applied here.
      }
    }
  }
}
so-3/0/0 { # This is the second interface that receives traffic to be monitored.
  encapsulation ppp;
  unit 0 {
    passive-monitor-mode; # This allows the interface to be passively monitored.
    family inet {
      filter {
        input catch;      # The filter-based forwarding filter is applied here.
      }
    }
  }
}
so-3/1/0 { # This is the third interface that receives traffic to be monitored.
  encapsulation ppp;
  unit 0 {
    passive-monitor-mode; # This allows the interface to be passively monitored.
    family inet {
      filter {
        input catch;      # The filter-based forwarding filter is applied here.
      }
    }
  }
}
}

```

```

forwarding-options {
  monitoring group1 {                                     # Always define your monitoring group here.
    family inet {
      output {
        export-format cflowd-version-5;
        flow-active-timeout 60;
        flow-inactive-timeout 15;
        flow-export-destination collector-pic; # Sends records to the flow collector.
      }
      interface mo-7/1/0.0 {
        source-address 192.168.252.2;
      }
      interface mo-7/2/0.0 {
        source-address 192.168.252.2;
      }
      interface mo-7/3/0.0 {
        source-address 192.168.252.2;
      }
    }
  }
}
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 {
  fb_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 "$9$IXJK8xN-w2oZdbZDHmF3001";# SECRET-DATA
      }
      "ftp://user@192.168.252.1//tmp/collect2/" {# The secondary FTP server.
        password "$9$elbvL7-dsgaGVwGjkP3nOBI"; # SECRET-DATA
      }
    }
  }
  file-specification {# Define sets of flow collector characteristics here.
    def-spec {
      name-format "default-allInt-0-%D_%T-%L_%N.bcp.bi.gz";
      data-format flow-compressed; The default compressed output format.
    } # When no overrides are specified, a collector uses default transfer values.
    f1 {
      name-format "cFlowd-py69Ni69-0-%D_%T-%L_%N.bcp.bi.gz";
      data-format flow-compressed; The default compressed output format.
      transfer timeout 1800 record-level 1000000; # Here are configured values.
    }
  }
}

```


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 input interface cp-6/0/0 extensive
Interface          Packets  Bytes
mo-7/1/0.0         6260    9074096

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 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 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 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 destination interface cp-6/0/0 secondary
Flow collector interface: cp-6/0/0
Interface state: Collecting flows
Destination change successful

```

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@router> 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 14 explains the various fields available in the transfer log.

Table 14: Flow Collector Interface Transfer Log Fields

Field	Explanation
fn	File name
sz	File size
nr	Number of records
ts	Time stamp 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=32
80
```

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=32
90
```

As the flow collector interface receives and processes cflowd 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 file naming 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 15 explains the various fields available in the flow collector interface file.

Table 15: Flow Collector Interface File Fields in Order of Appearance

Field	Explanation
linkDir	Link directory—a randomly generated number used to identify the record
analyzer-address	Analyzer address
analyzer-ID	Analyzer identifier
ifAlias	Interface identifier
source-address	Source address
destination-address	Destination address
packets	Number of packets
bytes	Number of bytes
start-time	Start time
end-time	End time
source-port	Source port
destination-port	Destination port
tcp_flag	TCP flag
protocol	IP protocol number
src_AS_number	Source AS number
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
```

Configure Active Flow Monitoring

Although the Monitoring Services PIC was designed initially for use as an offline passive flow monitoring tool, it can also be used in an active flow monitoring topology. In contrast, the Adaptive Services PIC is designed exclusively for active flow monitoring. To use either the Monitoring Services PIC or Adaptive Services PIC for active flow monitoring purposes, you must install the PIC in an M5, M7i, M10, M10i, M20, M40e, or M160 router. The router participates in both the monitoring application and in the normal routing functionality of the network.

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 PIC, the interface name contains the sp- prefix.



NOTE: If you upgrade from the Monitoring Services PIC to the Adaptive Services 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.

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

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

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

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

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

The router can perform sampling OR port mirroring at any one time.

The router can perform forwarding OR discard accounting at any one time.

Because the Monitoring Services PIC and Adaptive Services 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:

- Define a Firewall Filter to Select Traffic for Active Flow Monitoring on page 125

- Configure the Interfaces That Will Be Actively Monitored on page 126

- Enable the Monitoring Services or Adaptive Services Interfaces and the Export Interface on page 127

- Collect cflowd Records on page 128

- Option: Configure Port Mirroring on page 131

- Option: Configure Port Mirroring with Filter-Based Forwarding and a Monitoring Group on page 132

- Option: Send Traffic to Multiple Export Interfaces with Next-Hop Groups on page 133

To view examples of active flow monitoring, see the following sections:

- Example: Sampling Configuration on page 134

- Example: Sampling and Discard Accounting Configuration on page 138

- Example: Multiple Port Mirroring with Next-Hop Groups Configuration on page 143

Define 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 cflowd 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 the same traffic at the same time, but not perform both actions 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;
        }
      }
    }
  }
}
```

Configure 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 to SONET/SDH or ATM2 IQ interfaces, 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]);
        }
      }
    }
  }
}
```

Enable the Monitoring Services or Adaptive Services Interfaces and the Export Interface

You configure the monitoring services or adaptive services 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 an adaptive services interface uses an sp- prefix.

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

cflowd records leave the router through an export interface to reach the cflowd server.

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

Collect cflowd Records

Traffic flows can be exported in cflowd version 5 and 8 formats for active flow monitoring. The default export format for cflowd records is version 5. To change the export format to cflowd version 8, include the version 8 statement either at the [edit forwarding-options accounting *name* output cflowd *cflowd-server-address*] or the [edit forwarding-options sampling output cflowd *cflowd-server-address*] hierarchy level. For more information on cflowd record formats, see “cflowd Output Formats” on page 146.

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

Collect cflowd Records with a Sampling Group on page 128

Collect cflowd Records with an Accounting Group on page 129

Option: Configure an Aggregate Export Timer on page 130

Collect cflowd Records with a Sampling Group

If your needs for active flow monitoring are simple, you can collect cflowd 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 cflowd 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 cflowd server in a sampling configuration, include the cflowd statement at the [edit forwarding-options sampling output] hierarchy level. You must specify the IP address, port number, and cflowd version of the destination cflowd server. Routing Engine-based sampling can use only one cflowd version 5 server or one version 8 server at a time. In contrast, PIC-based sampling allows you to specify one cflowd version 5 server and one version 8 server simultaneously. However, the two cflowd servers must have different IP addresses.

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 `cflowd` collector, as needed. When you configure sampling, SNMP input and output interface index information is captured in `cflowd` records by default.

```
[edit]
forwarding-options {
  sampling {
    input {
      family inet {
        rate 1;
      }
    }
    output {
      cflowd 10.60.2.1 {
        port 2055;
        version 5;
      }
      flow-inactive-timeout 15;
      flow-active-timeout 60;
      interface sp-2/0/0 {
        engine-id 5;
        engine-type 55;
        source-address 10.60.2.2;
      }
    }
  }
}
```

Collect `cflowd` Records with an Accounting Group

To perform discard accounting on specified traffic, you can collect `cflowd` 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 `cflowd` 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 `cflowd` 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 `cflowd` server and the services interface you plan to use for processing `cflowd` 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 cflowd collector. SNMP input and output interface index information is captured in cflowd records by default when you configure discard accounting.

```
[edit]
forwarding-options {
  accounting counter1 {
    output {
      flow-inactive-timeout 65;
      flow-active-timeout 65;
      cflowd 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;
    }
  }
}
```

Option: Configure an Aggregate Export Timer

When you use cflowd version 8 records for active flow monitoring, you can configure an aggregate export timer. To configure, 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: Configure 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 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 cflowd processing.

To configure how often packets are copied from the monitored traffic, include the rate statement at the [edit forwarding-options port-mirroring input family inet] 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 {
    input {
      family inet {
        rate 1;
      }
    }
    output {
      interface sp-2/0/0.0;
    }
  }
}
```

Option: Configure 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 cflowd records from the monitoring services interfaces to a cflowd 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 “Copy and Redirect Traffic with Port Mirroring and Filter-Based Forwarding” on page 81.

Option: Send Traffic to Multiple Export Interfaces with Next-Hop Groups

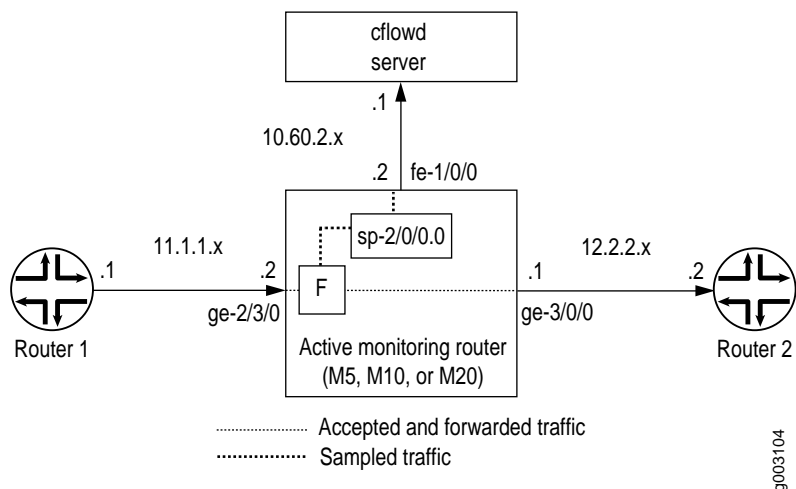
To send port-mirrored traffic to multiple cflowd servers or packet analyzers, you can use the next-hop-group statement. The router can make up to 16 copies of traffic per group and send the traffic to the next-hop group members you configure. A maximum of 30 groups can be configured on an M-series router 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 {
    input {
      family inet {
        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 11.12.0.2;
    }
    interface ge-1/2/0.0 {
      next-hop 11.13.0.2;
    }
  }
  next-hop-group default-collect {
    interface so-7/0/0.0;
    interface so-7/0/1.0;
  }
}
```

Example: Sampling Configuration

Figure 10: Active Flow Monitoring—Sampling Configuration Topology Diagram



In Figure 11, 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 cflowd 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 cflowd processing), and the export interface (for exporting cflowd records).

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

```
[edit]
interfaces {
  sp-2/0/0 { # This adaptive services interface creates the cflowd records.
    unit 0 {
      family inet {
        address 10.1.1.1/32 {
          destination 10.1.1.2;
        }
      }
    }
  }
  fe-1/0/0 { # This is the interface where records are sent to the cflowd 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 11.1.1.1/20;
    }
  }
}
ge-3/0/0 { # This is the interface where the original traffic is forwarded.
  unit 0 {
    family inet {
      address 12.2.2.1/24;
    }
  }
}
}
forwarding-options {
  sampling { # Traffic is sampled and sent to a cflowd server.
    input {
      family inet {
        rate 1; # Samples 1 out of x packets (here, a rate of 1 sample per packet).
      }
    }
    output {
      cflowd 10.60.2.1 {# The IP address and port of the cflowd server.
        port 2055;
        version 5; # Records are sent to the cflowd 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;
        }
      }
    }
  }
}
}

```

Check 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
```



NOTE: 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@router> 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@router> show services accounting flow-detail limit 10
Service Accounting interface: sp-2/0/0, Local interface index: 468
Service name: (default sampling)

```

Protocol	Source Address	Source Port	Destination Address	Destination Port	Packet count	Byte count
udp(17)	11.1.1.2	53	40.0.0.1	53	4329	3386035
ip(0)	11.1.1.2	0	50.0.0.2	0	4785	3719654
ip(0)	11.1.1.2	0	20.0.0.2	0	4530	3518769
udp(17)	11.1.1.2	0	70.0.0.1	0	5011	3916767
tcp(6)	11.1.1.2	20	45.3.0.1	20	1	1494
tcp(6)	11.1.1.2	20	44.168.80.1	20	1	677
tcp(6)	11.1.1.2	20	44.69.192.1	20	1	446
tcp(6)	11.1.1.2	20	43.239.240.1	20	1	1426
tcp(6)	11.1.1.2	20	35.126.160.1	20	1	889
tcp(6)	11.1.1.2	20	43.71.224.1	20	1	1046

```

user@router> 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@router> 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@router> 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

```

```

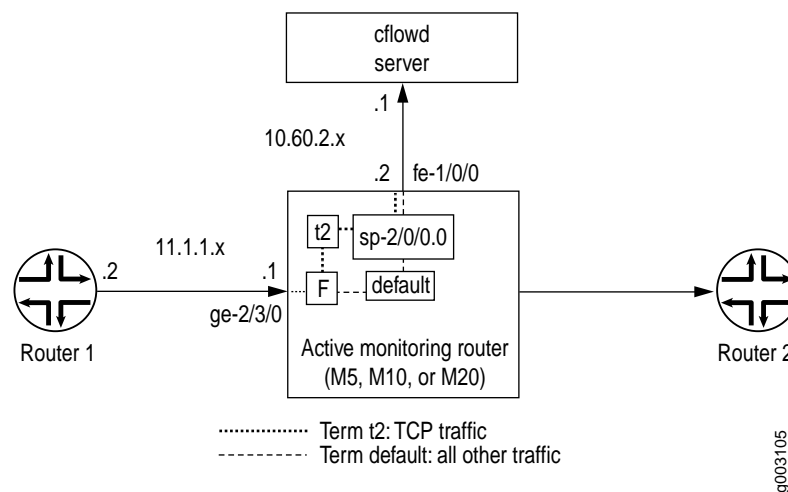
user@router> show services accounting usage
Service Accounting interface: sp-2/0/0, Local interface index: 468
Service name: (default sampling)
CPU utilization
Uptime: 4790345 milliseconds, Interrupt time: 1668537848 microseconds
Load (5 second): 71%, Load (1 minute): 63%

```

Example: Sampling and Discard Accounting Configuration

Discard accounting allows you to sample traffic, send it to a cflowd 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 11: Active Flow Monitoring—Sampling and Discard Accounting Topology Diagram



In Figure 11, traffic from Router 1 arrives on the monitoring router's Gigabit Ethernet ge-2/3/0 interface. The export interface leading to the cflowd 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 cflowd records and send the records to the cflowd version 8 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 cflowd server.

```
[edit]
interfaces {
  sp-2/0/0 { # This adaptive services interface creates the cflowd records.
    unit 0 {
      family inet {
        address 10.1.1.1/32 {
          destination 10.1.1.2;
        }
      }
    }
  }
}
```

```

fe-1/0/0 { # This is the interface where records are sent to the cflowd 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 11.1.1.1/20;
    }
  }
} #There is no exit interface, because all traffic is processed and discarded.
}
forwarding-options {
  sampling { # The router samples the traffic.
    input {
      family inet {
        rate 100; # One out of every 100 packets is sampled.
      }
    }
  }
  output { # The sampling process creates and exports cflowd records.
    cflowd 10.60.2.1 { # You can configure a variety of settings for this server.
      port 2055;
      version 8;
      aggregation { # Aggregation is unique to cflowd 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 cflowd records.
    flow-inactive-timeout 65;
    flow-active-timeout 65;
    cflowd 10.60.2.1 { # You can configure a variety of settings for the server.
      port 2055;
      version 8;
      aggregation { # Aggregation is unique to cflowd 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 cflowd records.
    aggregate-export-interval 90;
    flow-inactive-timeout 65;
    flow-active-timeout 65;
    cflowd 10.60.2.1 { # You can configure a variety of settings for the server.
      port 2055;
      version 8;
      aggregation { # Aggregation is unique to cflowd 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.
      }
    }
  }
}

```

Check 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 cflowd 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@router> show services accounting
Service Name:
  (default sampling)
  counter1
  t2

```

```

user@router> 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@router> 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@router> 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 Prefix	Destination Prefix	Input SNMP Index	Output SNMP Index	Flow count	Packet count	Byte
11.1.1.2/20	40.225.0.1/0	24	26	0	13	9650
11.1.1.2/20	44.143.80.1/0	24	26	0	13	10061
11.1.1.2/20	44.59.176.1/0	24	26	0	13	10426
11.1.1.2/20	38.5.32.1/0	24	26	0	13	12225
11.1.1.2/20	40.36.16.1/0	24	26	0	13	9116
11.1.1.2/20	35.1.96.1/0	24	26	0	12	11050
11.1.1.2/20	43.14.48.1/0	24	26	0	13	10812
11.1.1.2/20	36.31.192.1/0	24	26	0	13	11473
11.1.1.2/20	37.129.144.1/0	24	26	0	13	7647
11.1.1.2/20	35.188.160.1/0	24	26	0	13	10056

```

user@router> 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: 11.1.1.2, Source prefix length: 20
Destination address: 44.200.176.1, 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: 11.1.1.2, Source prefix length: 20
Destination address: 45.243.160.1, 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: 11.1.1.2, Source prefix length: 20
Destination address: 45.162.160.1, 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

```

Example: Multiple Port Mirroring with Next-Hop Groups Configuration

To send port-mirrored traffic to multiple cflowd servers or packet analyzers, you can use the next-hop-group statement available in JUNOS Release 5.7 and later. 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 an M-series 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. This example shows how to configure multiple port mirroring with next-hop groups:

```
[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 11.11.0.1/24;
      }
    }
  }
  ge-1/1/0 {          # This is an exit interface for HTTP packets.
    unit 0 {
      family inet {
        address 11.12.0.1/24;
      }
    }
  }
  ge-1/2/0 {          # This is an exit interface for HTTP packets.
    unit 0 {
      family inet {
        address 11.13.0.1/24;
      }
    }
  }
  so-0/3/0 {          # This is an exit interface for FTP packets.
    unit 0 {
      family inet {
        address 1.1.1.1/30;
      }
    }
  }
  so-4/3/0 {          # This is an exit interface for FTP packets.
    unit 0 {
      family inet {
        address 2.2.2.2/30;
      }
    }
  }
}
```

```

so-7/0/0 {          # This is an exit interface for all remaining packets.
  unit 0 {
    family inet {
      address 5.5.5.5/30;
    }
  }
}
so-7/0/1 {          # This is an exit interface for all remaining packets.
  unit 0 {
    family inet {
      address 6.6.6.6/30;
    }
  }
}
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.
    input {
      family inet {
        rate 1; # This port-mirrors all packets (one copy for every packet received).
      }
    }
    output { # Sends traffic to a tunnel interface to prepare for multipoint 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 11.12.0.2;
    }
    interface ge-1/2/0.0 {
      next-hop 11.13.0.2;
    }
  }
  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;
    }
  }
}

```

cflowd Output Formats

When you implement passive flow monitoring and active flow monitoring, you should be familiar with cflowd formats and fields. cflowd version 5 is used for both active and passive flow monitoring, while cflowd version 8 is available only for active flow monitoring.

The monitoring station monitors the traffic flow and exports the data in cflowd format to an external server. The JUNOS software collects information about the following cflowd 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

Detailed descriptions of the formats are available as follows:

- cflowd Version 5 Formats and Fields on page 146
- cflowd Version 8 Formats and Fields on page 150

cflowd Version 5 Formats and Fields

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

- Figure 12, “cflowd Version 5 Packet Header Format” on page 147
- Figure 16, “cflowd Version 5 Packet Header Format” on page 147
- Figure 13, “cflowd Version 5 Flow-Export Flow Header Format” on page 148
- Figure 17, “cflowd Version 5 Flow-Export Flow Header Format” on page 148

Figure 12: cflowd Version 5 Packet Header Format

Byte 3	Byte 2	Byte 1	Byte 0
Version		Count	
sysUptime			
UNIX seconds			
UNIX nanoseconds			
Flow sequence number			
Engine type	Engine ID	Reserved	

g003132

Table 16: cflowd 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
Engine ID	User-configured 8-bit value	–

Figure 13: cflowd Version 5 Flow-Export Flow Header Format

Byte 3	Byte 2	Byte 1	Byte 0
Source IP address			
Destination IP address			
Next-hop IP address			
Input ifIndex		Output ifIndex	
Packets			
Bytes			
Start time of flow			
End time of flow			
Source port		Destination port	
Padding	TCP flags	IP protocol	TOS
Source AS		Destination AS	
Source mask length	Dest. mask length	Padding	

g003133

Table 17: cflowd 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 Release 5.7 and later—Dynamically inserted, but overridden by manual configuration JUNOS Release 5.5—Manually set JUNOS Release 5.4—Set to zero
Output ifIndex	SNMP index value for the output interface where the router forwards flows	JUNOS Release 5.7 and later—Dynamically inserted, but overridden by manual configuration JUNOS Release 5.5—Manually set JUNOS Release 5.4—Set to zero
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	–

Field	Description	Comments
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 (see Note on page 149)
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 Release 5.7 and later—Dynamically inserted if AS information is available
Destination AS	AS number of the destination address	JUNOS 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 cflowd are:

$$\text{start flow timestamp absolute} = \text{unixTime} \times 1000 - (\text{sysUptime} - \text{start flow timestamp})$$

$$\text{end flow timestamp absolute} = \text{unixTime} \times 1000 - (\text{sysUptime} - \text{end flow timestamp})$$


NOTE: In the two-byte destination port field of the cflowd 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>.

cflowd Version 8 Formats and Fields

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

Figure 14, “cflowd Version 8 Flow Header Format” on page 151

Figure 18, “cflowd Version 8 Flow Header Format” on page 151

Figure 15, “cflowd Version 8 AS Aggregation Flow Entry Format” on page 152

Figure 19, “cflowd Version 8 AS Aggregation Flow Entry Format” on page 152

Figure 16, “cflowd Version 8 Protocol/Port Aggregation Flow Entry Format” on page 153

Figure 20, “cflowd Version 8 Protocol/Port Aggregation Flow Entry Format” on page 153

Figure 17, “cflowd Version 8 Prefix Aggregation Flow Entry Format” on page 154

Figure 21, “cflowd Version 8 Prefix Aggregation Flow Entry Format” on page 154

Figure 18, “cflowd Version 8 Source Prefix Aggregation Flow Entry Format” on page 155

Figure 22, “cflowd Version 8 Source Prefix Aggregation Flow Entry Format” on page 155

Figure 19, “cflowd Version 8 Destination Prefix Aggregation Flow Entry Format” on page 156

Figure 23, “cflowd Version 8 Destination Prefix Aggregation Flow Entry Format” on page 156

Figure 14: cflowd Version 8 Flow Header Format

Byte 3	Byte 2	Byte 1	Byte 0
Version		Count	
sysUptime			
UNIX seconds			
UNIX nanoseconds			
Flow sequence number			
Engine type	Engine ID	Aggregation method	Aggregation version
Reserved			

9003076

Table 18: cflowd Version 8 Flow Header 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 15: cflowd Version 8 AS Aggregation Flow Entry Format

Byte 3	Byte 2	Byte 1	Byte 0
Flows			
Packets			
Bytes			
Start time of flow			
End time of flow			
Source AS		Destination AS	
Input interface		Output interface	

g003077

Table 19: cflowd 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 16: cflowd Version 8 Protocol/Port Aggregation Flow Entry Format

Byte 3	Byte 2	Byte 1	Byte 0
Flows			
Packets			
Bytes			
Start time of flow			
End time of flow			
IP protocol	Padding	Reserved	
Source port		Destination port	

g003078

Table 20: cflowd 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 17: cflowd Version 8 Prefix Aggregation Flow Entry Format

Byte 3	Byte 2	Byte 1	Byte 0
Flows			
Packets			
Bytes			
Start time of flow			
End time of flow			
Source prefix			
Destination prefix			
Source mask length	Dest. mask length	Reserved	
Source AS		Destination AS	
Input interface		Output interface	

9003079

Table 21: cflowd 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
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 18: cflowd Version 8 Source Prefix Aggregation Flow Entry Format

Byte 3	Byte 2	Byte 1	Byte 0
Flows			
Packets			
Bytes			
Start time of flow			
End time of flow			
Source prefix			
Source mask length	Padding	Source AS	
Input interface		Reserved	

g003080

Table 22: cflowd Version 8 Source Prefix Aggregation Flow Entry Fields

Field	Description
Flows	Total number of flows
Packets	Total number of packets received in a flow
Bytes	Total number of bytes received in a flow
Start time of flow	System up time in seconds at the start of the flow
End time of flow	System up time in seconds at the end of the flow
Source prefix	Source IP address prefix
Source mask length	Source address network mask length
Padding	Bytes available to ensure a minimum packet length
Source AS	AS number of the source address
Input interface	SNMP index value for the input interface where the router receives flows
Reserved	Empty field reserved for future usage

Figure 19: cflowd Version 8 Destination Prefix Aggregation Flow Entry Format

Byte 3	Byte 2	Byte 1	Byte 0
Flows			
Packets			
Bytes			
Start time of flow			
End time of flow			
Destination prefix			
Dest. mask length	Padding	Destination AS	
Output interface		Reserved	

g003081

Table 23: cflowd 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 cflowd packet formats and fields, see <http://www.caida.org>.

For More Information

To learn more about passive flow monitoring, active flow monitoring, and cflowd, see the following:

CAIDA (Cooperative Association for Internet Data Analysis) Web site at <http://www.caida.org>.

JUNOS Policy Framework Configuration Guide

JUNOS Services Interfaces Configuration Guide

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

Revision History

- 6 July 2004—Added support for the next-hop IP address field in cflowd version 5 records, 6.4R1 Release. Richard Hendricks.
- 5 April 2004—Added an explanation of how to load-balance traffic across multiple Monitoring Services I PICs for active flow monitoring and provided more information about flow collection services, 6.3R1 Release. Richard Hendricks.
- 21 January 2004—Added additional flow collector interface information. Richard Hendricks.
- 22 December 2003—Added passive flow monitoring support for ATM2 IQ interfaces, MPLS label removal, and flow collector interface configuration, 6.2R1 Release. Richard Hendricks.
- 22 September 2003—6.1R1 Release. Richard Hendricks.
- 30 June 2003—Added Monitoring Services II PIC, Adaptive Services PIC, and rearranged existing content, 6.0R1 Release. Richard Hendricks.
- 2 April 2003—Added new active flow monitoring content, 5.7R1 Release. Richard Hendricks.
- 27 December 2002—Revised the entire chapter for the 5.6R1 Release. Richard Hendricks.
- 22 October 2002—Added active flow monitoring section. Richard Hendricks.
- 30 September 2002—5.5R1 Release. Richard Hendricks.
- 27 August 2002—Added 5.5 show commands and expanded the cflowd packet, header, and field descriptions. Richard Hendricks.
- 19 July 2002—5.4R1 Release. Richard Hendricks.
- 28 June 2002—Reformatted the document, edited content, and added several new sections. Richard Hendricks.
- 6 May 2002—Initial document written. Renu Bhargava.