

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 60

- System Requirements on page 61

- Terms and Acronyms on page 64

- Configure Passive Flow Monitoring on page 65

 - Hardware and Software Considerations on page 77

 - Example: Passive Flow Monitoring Configuration on page 79

 - Check Your Work on page 85

- Configure Active Flow Monitoring on page 91

 - Example: Sampling Configuration on page 100

 - Check Your Work on page 102

 - Example: Sampling and Discard Accounting Configuration on page 103

 - Check Your Work on page 106

 - Example: Multiple Port Mirroring with Next-Hop Groups Configuration on page 108

- Flowd Output Formats on page 111

- For More Information on page 121

- Revision History on page 121

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.

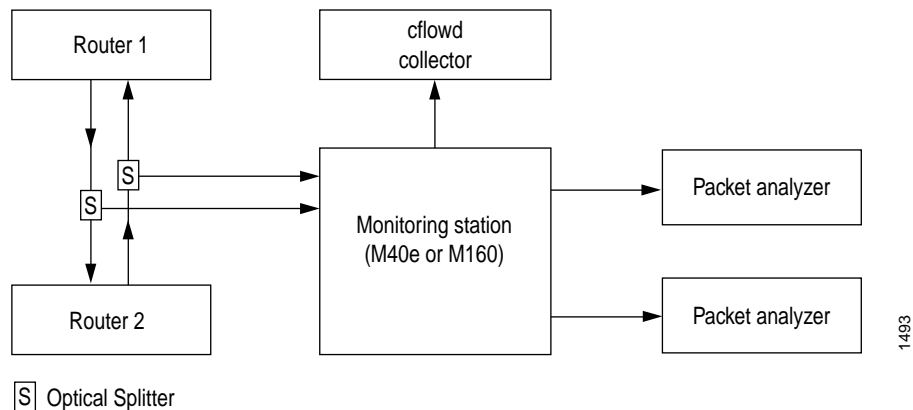
- Encrypt or tunnel outgoing cflowd records, intercepted IPv4 traffic, or both.

- 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.

Passive Flow Monitoring The M40e or M160 router used for passive flow monitoring does not route packets from the monitored interface, 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.

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 the router to a destination. Instead, the traffic is quarantined and deleted.

System Requirements

Passive and active flow monitoring are supported on the PICs shown in Table 6:

Table 6: Passive and Active Flow Monitoring PIC Support

PIC Type and Service	M5	M10	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
Adaptive Services PIC: active flow monitoring	Yes	Yes	Yes	Yes	Yes

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

JUNOS Release 5.4 or later for the Monitoring Services PIC

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

JUNOS Release 6.0 or later for the Monitoring Services II PIC

M40e or M160 router with an Internet Processor II ASIC

Type 1 enhanced FPCs

Two optical splitters

One Monitoring Services or Monitoring Services II PIC for every OC-3 worth of monitored traffic

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

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

Outgoing PICs to connect to the cflowd collector or packet analyzer

ES PIC (optional)

cflowd version 5 collector

Packet analyzers (optional)

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

JUNOS Release 5.6 or later for the Monitoring Services PIC

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

JUNOS Release 6.0 or later for the Adaptive Services PIC

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

Type 1 enhanced FPCs

A Monitoring Services PIC or Adaptive Services PIC

A PIC of your choice to receive incoming traffic

A PIC of your choice to forward outgoing traffic (not necessary for discard accounting)

Export PICs to connect to the cflowd collector or packet analyzer

Tunnel Services PIC (required for multiple port mirroring, otherwise optional)

ES PIC (optional)

cflowd version 5 or 8 collector

Packet analyzers (optional)

Table 7, Table 8, and Table 9 describe the specifications for the Monitoring Services PIC, Monitoring Services II PIC, and Adaptive Services PIC.

Table 7: 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 8: 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 9: 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

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

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>.

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

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

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.

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 Internet Software Configuration Guide: Services Interfaces*.

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 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. 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 even encrypt cflowd output before it is sent to a cflowd server for processing.

The following section explains the myriad of passive flow monitoring configuration topics:

Monitor Traffic with a VRF Instance and a Monitoring Group on page 65

Copy and Redirect Traffic with Port Mirroring and Filter-Based Forwarding on page 71

Hardware and Software Considerations on page 77

Example: Passive Flow Monitoring Configuration on page 79

Check Your Work on page 85

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 66

Configure Input Interfaces, Monitoring Services Interfaces, and Export Interfaces on page 67

Establish a VRF Instance for the Monitored Traffic on page 69

Configure a Monitoring Group to Send Traffic to the cflowd Server on page 69

Configure Policy Options on page 71

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 input interfaces, include the `passive-monitor-mode` statement at the `[edit interfaces so-fpc/pic/port unit unit-number]` hierarchy level. This mode suppresses SONET keepalives and disables the router from participating in the network as an active device. Apply the previously defined firewall filter 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 "input interface";
    encapsulation ppp;
    unit 0 {
      passive-monitor-mode;
      family inet {
        filter {
          input input-monitoring-filter;
        }
      }
    }
  }
  so-0/1/0 {
    description "input interface";
    encapsulation ppp;
    unit 0 {
      passive-monitor-mode;
      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 SONET 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 specify a source address for transmission of cflowd information. The router ID IP address is the default source address, but you can configure this statement manually. If you provide a different source-address statement for each Monitoring Services output interface, you can track which interface processes a particular cflowd record.

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 SONET 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 71.



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 67.

```
[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;
    }
  }
}
```

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.

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

Specify Port Mirroring Input and Output on page 72

Create a Firewall Filter to Split the Port-Mirrored Traffic into Different Instances on page 73

Apply the Firewall Filter to a Tunnel PIC Interface on page 74

Use Filter-Based Forwarding to Export Monitored Traffic to Multiple Destinations on page 74

Configure a Routing Table Group to Add Interface Routes into the Forwarding Instance on page 75

Option: Use an ES PIC to Send Traffic to a Packet Analyzer on page 75

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 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 Internet Software Configuration Guide: Getting Started*.

```
[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";
  }
}
}
}

```

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 interface on the monitoring station must be a SONET OC-3, OC-12, or OC-48 interface.

To monitor the flows on both directions of an interface, the monitoring station must have two SONET receive ports, one for each direction of flow. In Figure 7 on page 60, 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 OC-3 interface, the monitoring station must have two Monitoring Services PICs.

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

To monitor a fully loaded bidirectional SONET 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 this inactivity timer, include the `flow-inactive-timeout` statement at the [edit forwarding-options monitoring *group-name* family inet output] hierarchy level. The timer value ranges from 15 seconds through 1800 seconds and has a default setting of 60 seconds.

The monitoring station also has an activity timeout for aging flows. You can configure the activity timer for the monitoring station 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 has a default setting of 180 seconds.

When you use cflowd version 8 records, 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 setting of 90 seconds.

Multiple expired flows are exported together, if possible. A UDP packet is sent when one of the following conditions is met:

When thirty flows are contained in the current packet, the flows are exported.

If there are less than thirty 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 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.

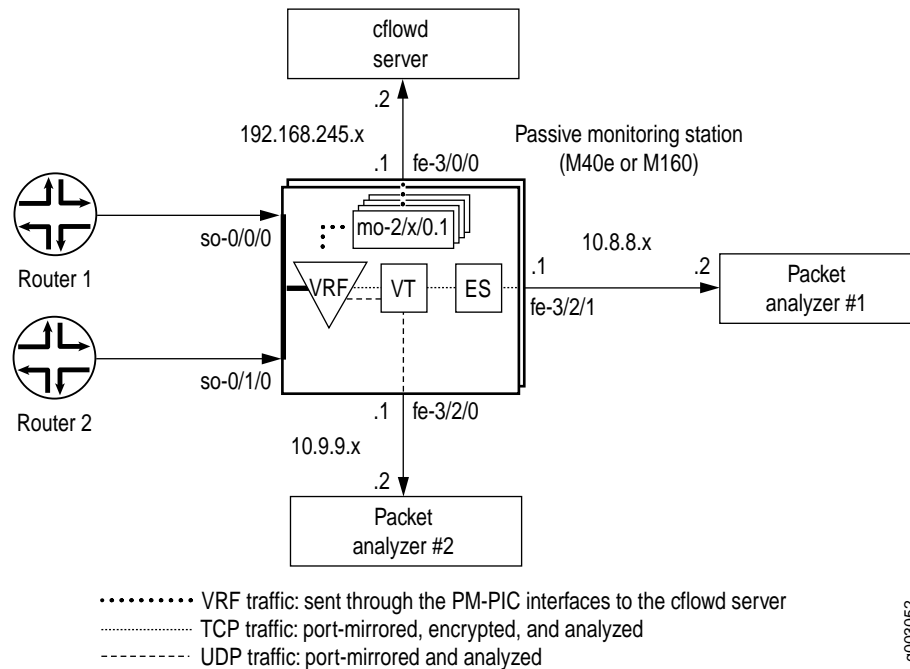
The interfaces on the monitoring station that collect intercepted traffic must be configured with Cisco HDLC or PPP encapsulation.

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 these servers in a round-robin fashion. If one of the servers ceases operation, cflowd traffic will load balance 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 interfaces and apply the firewall filter that you just defined. The passive-monitor-mode statement disables SONET keepalives on the SONET 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 routing instances. Remember to apply the filter to the Tunnel interface to enable the separation of TCP traffic from UDP traffic. Also, import the interface routes into the forwarding instances with a routing table group.

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

```
[edit]
interfaces {
  so-0/0/0 {          # Traffic enters the router on this interface.
    description "input interface";
    encapsulation ppp;
    unit 0 {
      passive-monitor-mode; #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 the monitoring station 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 another 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 configured in the VRF instance.
    family inet;
  }
}
mo-4/1/0 {          # Unit 0 is part of the inet.0 routing table and generates flow records.
  unit 0 {
    family inet;
  }
  unit 1 {          # Unit 1 receives monitored traffic and is configured in the VRF instance.
    family inet;
  }
}
mo-4/2/0 {          # Unit 0 is part of the inet.0 routing table and generates flow records.
  unit 0 {
    family inet;
  }
  unit 1 {          # Unit 1 receives monitored traffic and is configured in the VRF instance.
    family inet;
  }
}
mo-4/3/0 {          # Unit 0 is part of the inet.0 routing table and generates flow records.
  unit 0 {
    family inet;
  }
  unit 1 {          # Unit 1 receives monitored traffic and is configured in 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 { # This allows monitored traffic to be processed by Monitoring Services
    family inet { # interfaces and cflowd records to be 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; # This is the cflowd server's IP address and port.
        interface mo-4/0/0.1 { # All Monitoring Services interfaces are output interfaces.
          engine-id 1; # engine statements 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 interface fe-3/0/0.
        }
        interface mo-4/1/0.1 {
          engine-id 2; # engine statements 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 interface fe-3/0/0.
        }
        interface mo-4/2/0.1 {
          engine-id 3; # engine statements 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 interface fe-3/0/0.
        }
        interface mo-4/3/0.1 {
          engine-id 4; # engine statements 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 interface fe-3/0/0.
        }
      }
    }
  }
}

port-mirroring { # This copies the monitored 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; # This applies a per-packet load-balancing policy 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 instance and
                                     # prepares the traffic for port mirroring.
      term 1 {
        from {
          destination-address {
            10.7.0.0/16;
          }
        }
        then {
          port-mirror;
          accept;
        }
      }
      term 2 {
        from {
          destination-address {
            10.6.0.0/16;
          }
        }
        then accept;
      }
    }
    filter tunnel-interface-filter {# This filter breaks the port-mirrored traffic into two filter-based
                                     # forwarding routing instances: TCP packets and UDP packets
      term tcp { # forwarding routing instances: TCP packets and UDP packets
        from {
          protocol tcp;
        }
        then { # This counts TCP packets and sends them into a TCP instance.
          count tcp;
          routing-instance tcp-routing-table;
        }
      }
      term udp {
        from {
          protocol udp;
        }
        then { # This counts UDP packets and sends them into a UDP instance.
          count udp;
          routing-instance udp-routing-table;
        }
      }
      term rest {
        then {
          count rest;
          discard;
        }
      }
    }
  }
}

```

```

routing-instances {
  monitoring-vrf { # This is the VRF instance where you send the original traffic. It contains
    instance-type vrf; # the input interface and the Monitoring Services interfaces.
    interface so-0/0/0.0; # These are input interfaces where traffic enters the router.
    interface so-0/1/0.0;
    interface mo-4/0/0.1;
    interface mo-4/1/0.1; # These are output interfaces (duplicate them as output
    interface mo-4/2/0.1; # interfaces in your monitoring group).
    interface mo-4/3/0.1;
    route-distinguisher 69:1;
    vrf-import monitoring-vrf-import;
    vrf-export monitoring-vrf-export;
    routing-options { # This sends the traffic to the group of Monitoring Services interfaces.
      static {
        route 0.0.0.0/0 next-hop [ mo-4/0/0.1 mo-4/1/0.1 mo-4/2/0.1 mo-4/3/0.1 ];
      }
    }
  }
  tcp-routing-table { #This is the filter-based forwarding instance for TCP traffic.
    instance-type forwarding;
    routing-options { # The next hop is the ES PIC.
      static {
        route 0.0.0.0/0 next-hop es-3/1/0.0;
      }
    }
  }
  udp-routing-table { #This is the filter-based forwarding instance for UDP traffic.
    instance-type forwarding;
    routing-options { # The next hop is the second packet analyzer.
      static {
        route 0.0.0.0/0 next-hop 10.9.1.2;
      }
    }
  }
}

```

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

0.0.0.0/0          *[Static/5] 5d 17:34:57
                   via mo-4/0/0.1
                   > via mo-4/1/0.1
                   via mo-4/2/0.1
                   via mo-4/3/0.1

tcp-rt.inet.0: 13 destinations, 13 routes (12 active, 0 holddown, 1
hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 19:24:39
                   > via es-3/1/0.0
                   : <other interface routes>

udp-rt.inet.0: 13 destinations, 13 routes (12 active, 0 holddown, 1
hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 19:24:39
                   > to 10.9.1.2 via fe-3/2/0.0
                   : <other interface routes>
```



Note

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

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

Table 10: 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 version check.
Packets dropped (header too small)	Number of packets dropped because the packet length or IP header length was too small.
Memory allocation failures	Number of flow record memory allocation failures. A small number reflects failures to replenish the free list. A large number indicates the monitoring station is almost out of memory space.
Memory free failures	Number of flow record memory 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 overload	Whether the memory has been overloaded. The response can be 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 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 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 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 overload: No, PPS overload: No, BPS overload: No

```

Table 11: 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 12: 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 13: Output Fields for the show passive-monitoring status Command

Field	Explanation
Group index	Integer that represents the monitoring group that the PIC is a member of.
Export interval (in seconds)	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.

```

user@mon-station> show passive-monitoring status all
Passive Monitoring interface: mo-4/0/0, Local interface index: 44
  Group index: 0
  Export interval (in seconds): 15, Export format: cflowd v5
  Protocol: IPv4, Engine type: 1, Engine ID: 1

Passive Monitoring interface: mo-4/1/0, Local interface index: 45
  Group index: 0
  Export interval (in seconds): 15, Export format: cflowd v5
  Protocol: IPv4, Engine type: 1, Engine ID: 2

Passive Monitoring interface: mo-4/2/0, Local interface index: 46
  Group index: 0
  Export interval (in seconds): 15, Export format: cflowd v5
  Protocol: IPv4, Engine type: 1, Engine ID: 3

Passive Monitoring interface: mo-4/3/0, Local interface index: 47
  Group index: 0
  Export interval (in seconds): 15, Export format: cflowd v5
  Protocol: IPv4, Engine type: 1, Engine ID: 4

```

**Note**

Group index is a mapping from the group name to an index. It is not related to the number of monitoring groups.

Table 14: 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%

```

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, M10, 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 some of the packets in a flow.

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 93

Configure the Interfaces That Will Be Actively Monitored on page 94

Enable the Monitoring Services or Adaptive Services Interfaces and the Export Interface on page 94

Collect cflowd Records on page 95

Option: Configure Port-Mirroring Statements on page 98

Option: Send Traffic to Multiple Export Interfaces with Next-Hop Groups on page 98

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

Example: Sampling Configuration on page 100

Example: Sampling and Discard Accounting Configuration on page 103

Example: Multiple Port Mirroring with Next-Hop Groups Configuration on page 108

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. 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 111.

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 96

Collect cflowd Records with an Accounting Group on page 97

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.

A Monitoring Services or Adaptive Services interface is usually the target for the output of the sampling process. When you specify one of these interfaces as an output interface, you perform PIC-based sampling. Also, be sure to specify the IP address and port number of your cflowd server.

The interface-level statements of engine-id, engine-type, and source-address are all added automatically. However, you can override these values with manually configured statements to track different flows with a single cflowd collector, as needed. Note that SNMP input and output interface index information is captured in cflowd records by default when you configure sampling.

```
[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 found 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.

The interface-level statements of engine-id, engine-type, and source-address are all added automatically. However, you can override these values with manually configured statements to track different flows with a single cflowd collector, as needed. Note that 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 Port-Mirroring Statements

You can copy entire packets and reroute them to another interface by using port mirroring. To send port-mirrored traffic 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 the 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: Send Traffic to Multiple Export Interfaces with Next-Hop Groups

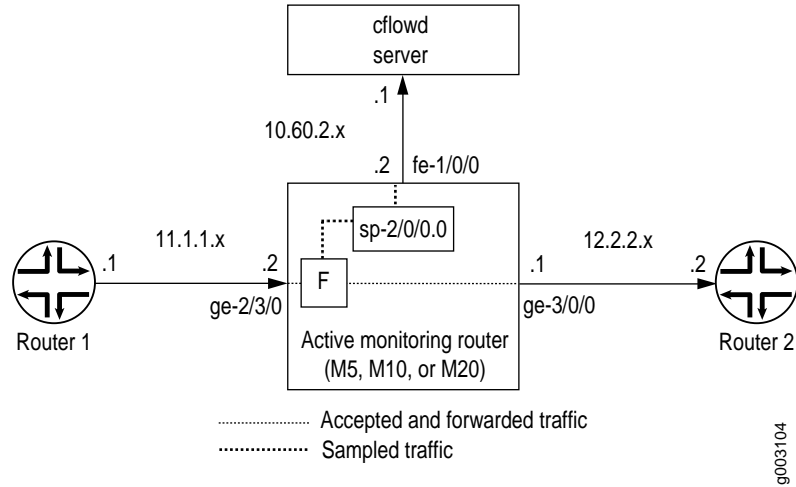
To send port-mirrored traffic to multiple cflowd servers or packet analyzers, you can use the next-hop-group statement. The router can make up to sixteen copies of traffic per group and send the traffic to the next-hop group members you configure. A maximum of 30 groups can be configured on an M-series router at any given time. The port-mirrored traffic can be sent to any interface, except aggregated SONET, 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 9: Active Flow Monitoring—Sampling Configuration Topology Diagram



In Figure 10, traffic from Router 1 arrives on the monitoring router’s Gigabit Ethernet ge-2/3/0 interface. The exit interface on the monitoring router leading to destination Router 2 is ge-3/0/0, but this could be any interface type (such as SONET, Gigabit Ethernet, and so on). The export interface leading to the cflowd server is fe-1/0/0.

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 export 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 exit 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; # The router samples 1 out of x packets (a rate of 1 samples every packet).
      }
    }
    output {
      cflowd 10.60.2.1 { # Here you configure the IP address and port of the cflowd server.
        port 2055;
        version 5; # The records are sent to the cflowd server using version 5 format.
      }
      flow-inactive-timeout 15;
      flow-active-timeout 60;
      interface sp-2/0/0 { # By adding an interface here, you enable PIC-based sampling.
        engine-id 5; # Engine statements are generated dynamically, but can be configured.
        engine-type 55;
        source-address 10.60.2.2;# If you do not configure this statement, the router ID is the
        # default source address.
      }
    }
  }
}
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)
Group index: 0
Export interval (in seconds): 60, Export format: cflowd v5
Protocol: IPv4, Engine type: 55, Engine ID: 5

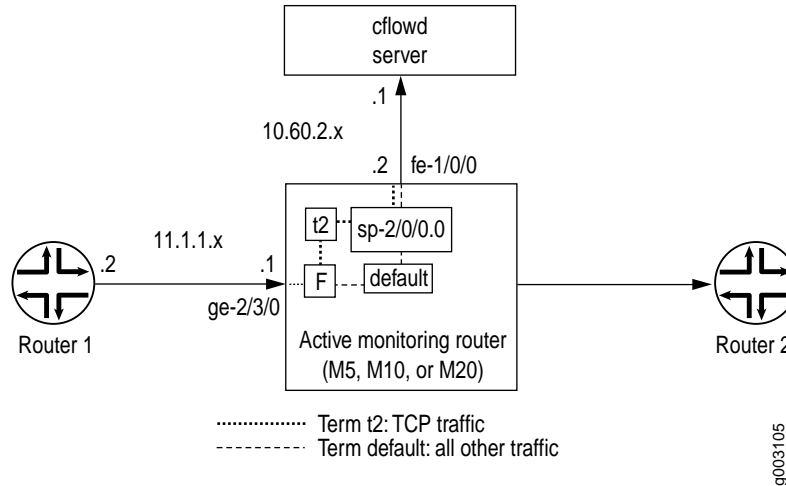
user@router> show services accounting usage
Service Accounting interface: sp-2/0/0, Local interface index: 468
Service name: (default sampling)
CPU utilization
Uptime: 4790345 milliseconds, Interrupt time: 1668537848 microseconds
Load (5 second): 71%, Load (1 minute): 63%

```

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



In Figure 10, 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 export 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;
    }
  }
} # Note: 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 the cflowd 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 generated dynamically, but can be configured.
        engine-type 55;
        source-address 10.60.2.2;# Configure manually if you wish to differentiate records.
      }
    }
  }
}
accounting counter1 { # The first 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 cflowd 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 generated dynamically, but can be configured.
    engine-type 11;
    source-address 10.60.2.3;# Configure manually if you wish to differentiate records.
  }
}
}

```



```
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      Destination  Input SNMP  Output SNMP  Flow  Packet  Byte
  Prefix      Prefix      Index      Index      count count  count
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, 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 rate port-mirrors all packets (one copy for every packet received).
      }
    }
    output { # This sends traffic to a Tunnel interface to prepare for multiport mirroring.
      interface vt-3/3/0.1;
      no-filter-check;
    }
  }
}

```

```

next-hop-group ftp-traffic {# Point-to-point interfaces require you to specify the interface name.
  interface so-4/3/0.0;
  interface so-0/3/0.0;
}
next-hop-group http-traffic {# Configure a next hop for all multipoint interfaces (Ethernet).
  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 term 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

The JUNOS software does *not* collect information about the following cflowd fields:

- Next-hop router's IP address

Detailed descriptions of the formats are available as follows:

- cflowd Version 5 Formats and Fields on page 111

- cflowd Version 8 Formats and Fields on page 114

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 11, “cflowd Version 5 Packet Header Format” on page 112

- Table 15, “cflowd Export Version 5 Packet Header Fields” on page 112

- Figure 12, “cflowd Version 5 Flow-Export Flow Header Format” on page 113

- Table 16, “cflowd Export Version 5 Flow-Export Flow Header Fields” on page 113

Figure 11: 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	

9003132

Table 15: 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 12: 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 16: 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	Set to zero; not implemented currently
Input ifIndex	SNMP index value for the input interface where the router receives flows	JUNOS Release 5.7—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—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	–
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 114)
TCP flags	TCP flags set in the flow	–

Field	Description	Comments
IP protocol	IP protocol number	–
TOS	IP type of service	–
Source AS	AS number of the source address	JUNOS Release 5.7—Dynamically inserted if AS information is available
Destination AS	AS number of the destination address	JUNOS Release 5.7—Dynamically inserted if AS information is available
Source mask length	Source address network mask length	Set to zero; not implemented currently
Dest. mask length	Destination address network mask length	Set to zero; not implemented currently
Padding	Bytes available to ensure a minimum packet length	

Useful formulas for cflowd are:

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

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



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 13, “cflowd Version 8 Flow Header Format” on page 115

Table 17, “cflowd Version 8 Flow Header Fields” on page 115

Figure 14, “cflowd Version 8 AS Aggregation Flow Entry Format” on page 116

Table 18, “cflowd Version 8 AS Aggregation Flow Entry Fields” on page 116

Figure 15, “cflowd Version 8 Protocol/Port Aggregation Flow Entry Format” on page 117

Table 19, “cflowd Version 8 Protocol/Port Aggregation Flow Entry Fields” on page 117

Figure 16, “cflowd Version 8 Prefix Aggregation Flow Entry Format” on page 118

Table 20, “cflowd Version 8 Prefix Aggregation Flow Entry Fields” on page 118

Figure 17, “cflowd Version 8 Source Prefix Aggregation Flow Entry Format” on page 119

Table 21, “cflowd Version 8 Source Prefix Aggregation Flow Entry Fields” on page 119

Figure 18, “cflowd Version 8 Destination Prefix Aggregation Flow Entry Format” on page 120

Table 22, “cflowd Version 8 Destination Prefix Aggregation Flow Entry Fields” on page 120

Figure 13: 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 17: 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 14: 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	

9003077

Table 18: 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 15: 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 19: 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 16: 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	

g003079

Table 20: 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 17: 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 21: 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 18: 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	

9003081

Table 22: 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) website at <http://www.caida.org>.

JUNOS Internet Software Configuration Guide: Policy Framework

JUNOS Internet Software Configuration Guide: Services Interfaces

For more information on IPSec and the ES PIC, see the *JUNOS Internet Software Configuration Guide: Getting Started*.

Revision History

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, 5.6B1 Release. 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.

