

Chapter 27

Flow Monitoring and Discard Accounting Configuration Guidelines

This chapter describes the following tasks for configuring traffic sampling and flow-monitoring properties:

Minimum Traffic Sampling or Forwarding Configuration on page 315

Configuring Traffic Sampling on page 316

Configuring Flow Monitoring on page 323

Configuring cflowd on page 327

Configuring Port Mirroring on page 330

Load Balancing Among Multiple Monitoring Interfaces on page 335

Configuring Discard Accounting on page 339

Enabling Passive Flow Monitoring on page 341

Minimum Traffic Sampling or Forwarding Configuration

To configure traffic sampling on a logical interface, you must perform at least the following tasks:

Create a firewall filter to apply to the logical interfaces being sampled by including the filter statement at the [edit firewall family *family-name*] hierarchy level. In the filter then statement, you must specify the action modifier `sample` and the action `accept`.

```
[edit firewall family family-name]  
filter filter-name {  
  term term-name {  
    then {  
      sample;  
      accept;  
    }  
  }  
}
```

For more information about firewall filter actions and action modifiers, see the *JUNOS Policy Framework Configuration Guide*.

Apply the filter to the interfaces on which you want to sample traffic by including the address and filter statements at the [edit interfaces interface-name unit logical-unit-number family family-name] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family family-name]
address address {
    destination destination-address;
}
filter {
    input filter-name;
}
```

Enable sampling and specify a nonzero sampling rate by including the sampling statement at the [edit forwarding-options] hierarchy level:

```
[edit forwarding-options]
sampling {
    input {
        family inet {
            max-packets-per-second number;
            rate number;
        }
    }
}
```

Configuring Traffic Sampling

Traffic sampling enables you to copy traffic to a Physical Interface Card (PIC) that performs flow accounting while the router forwards the packet to its original destination. You can configure the router to perform sampling in either of two locations:

On the Routing Engine, using the sampled process. To select this method, use a filter (input or output) with a matching term that contains the then sample statement.

On the Monitoring Services PIC.

The following sections provide information about traffic sampling configuration:

Configuring Traffic Sampling Properties on page 317

Disabling Traffic Sampling on page 318

Configuring Traffic Sampling Output on page 318

Tracing Traffic Sampling Operations on page 320

Examples: Configuring Traffic Sampling on page 320

Configuring Traffic Sampling Properties

To configure traffic sampling on any logical interface, include the input statement at the [edit forwarding-options sampling] hierarchy level:

```
[edit forwarding-options sampling]
input {
  family inet {
    max-packets-per-second number;
    rate number;
    run-length number;
  }
}
```

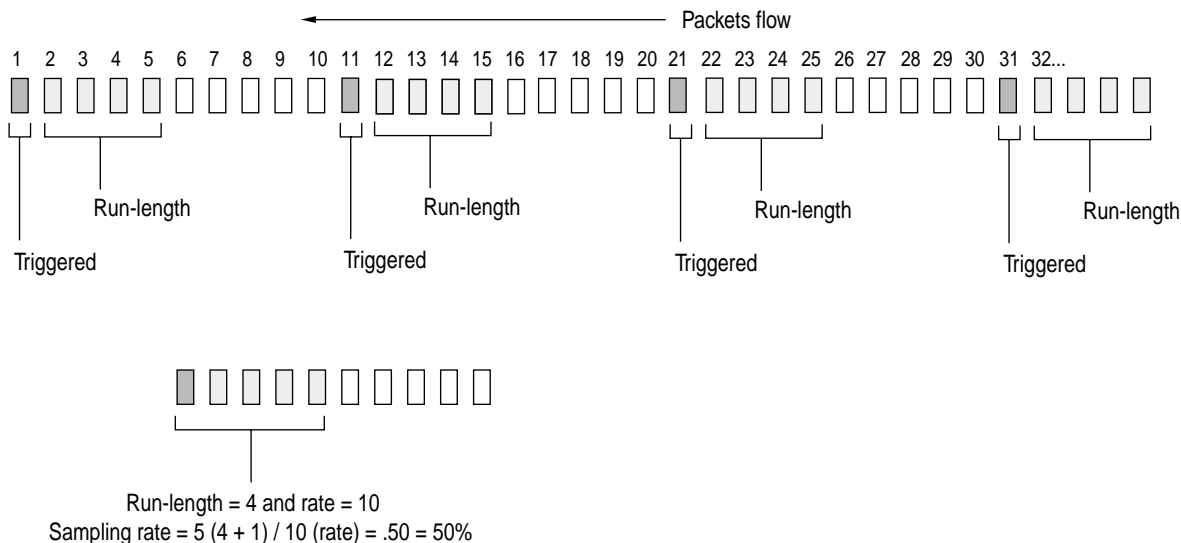
Specify the threshold traffic value by including the max-packets-per-second statement. The value is the maximum number of packets to be sampled, beyond which the sampling mechanism begins dropping packets. The range is from 0 through 65,535. A value of 0 instructs the Packet Forwarding Engine not to sample any packets. The default value is 1000.

Specify the sampling rate by setting the values for rate and run-length (see Figure 5).

Figure 5: Configure Sampling Rate

Rate and Run-length

Sampling rate = (run-length + 1) / rate



1746

The rate statement specifies the ratio of packets to be sampled. For example, if you configure a rate of 10, x number of packets out of every 10 is sampled, where $x = \text{run-length} + 1$. By default, the rate is 0, which means that no traffic is sampled.

The run-length statement specifies the number of matching packets to sample following the initial one-packet trigger event. By default, the run-length is 0, which means that no more traffic is sampled after the trigger event. The range is from 0 through 20. Configuring a run length greater than 0 allows you to sample packets following those already being sampled.

If you do not include the input statement, sampling is disabled.

To collect the sampled packets in a file, include the file statement at the [edit forwarding-options sampling output] hierarchy level. For more information about the output file formats, see “Configuring Traffic Sampling Output” on page 318.

Disabling Traffic Sampling

To explicitly disable traffic sampling on the router, include the disable statement at the [edit forwarding-options] hierarchy level:

```
[edit forwarding-options]
sampling {
  disable;
}
```

Configuring Traffic Sampling Output

You can configure the following traffic sampling output statements:

```
[edit forwarding-options sampling output]
aggregate-export-interval seconds;
cflowd hostname {
  aggregation {
    autonomous-system;
    destination-prefix;
    protocol-port;
    source-destination-prefix {
      caida-compliant;
    }
    source-prefix;
  }
  autonomous-system-type (origin | peer);
  (local-dump | no-local-dump);
  port port-number;
  source-address address;
  version format;
}
file {
  disable;
  filename filename;
  files number;
  size bytes;
  (stamp | no-stamp);
  (world-readable | no-world-readable);
}
flow-active-timeout seconds;
flow-inactive-timeout seconds;
interface interface-name {
  engine-id number;
  engine-type number;
  source-address address;
}
```

To direct sampled traffic to a flow-monitoring interface, include the interface statement. The engine-id and engine-type statements specify the identity and type numbers of the interface; they are dynamically generated based on the Flexible PIC Concentrator (FPC), PIC, and slot numbers and the chassis type. The source-address statement specifies the traffic source.

For information on cflowd, see “Configuring cflowd” on page 327. The aggregate-export-interval statement is described in “Configuring Discard Accounting” on page 339, and the flow-active-timeout and flow-inactive-timeout statements are described in “Configuring Flow Monitoring” on page 323.

Traffic sampling results are automatically saved to a file in the /var/tmp directory. To collect the sampled packets in a file, include the file statement at the [edit forwarding-options sampling output] hierarchy level:

```
[edit forwarding-options sampling output]
file {
  disable;
  filename filename;
  files number;
  size bytes;
  (stamp | no-stamp);
  (world-readable | no-world-readable);
}
```

Traffic Sampling Output Files

Traffic sampling output is saved to an ASCII text file. The following is an example of the traffic sampling output that is saved to a file in the /var/tmp directory. Each line in the output file contains information for one sampled packet. You can optionally display a timestamp for each line.

The column headers are repeated after each group of 1000 packets.

```
# Apr 7 15:48:50
Time          Dest      Src Dest Src Proto TOS Pkt Intf IP  TCP
              addr      addr port port   len num frag flags
Apr 7 15:48:54 192.168.9.194 192.168.9.195 0 0 1 0x0 84 8 0x0 0x0
Apr 7 15:48:55 192.168.9.194 192.168.9.195 0 0 1 0x0 84 8 0x0 0x0
Apr 7 15:48:56 192.168.9.194 192.168.9.195 0 0 1 0x0 84 8 0x0 0x0
Apr 7 15:48:57 192.168.9.194 192.168.9.195 0 0 1 0x0 84 8 0x0 0x0
Apr 7 15:48:58 192.168.9.194 192.168.9.195 0 0 1 0x0 84 8 0x0 0x0
```

To set the timestamp option for the file my-sample, enter the following:

```
[edit forwarding-options sampling output file]
user@host# set filename my-sample files 5 size 2m world-readable stamp;
```

Whenever you toggle the timestamp option, a new header is included in the file. If you set the stamp option, the Time field is displayed.

```
# Apr 7 15:48:50
# Time          Dest      Src Dest Src Proto TOS Pkt Intf IP  TCP
#              addr      addr port port   len num frag flags
# Feb 1 20:31:21
#              Dest      Src Dest Src Proto TOS Pkt Intf IP  TCP
#              addr      addr port port   len num frag flags
```

Tracing Traffic Sampling Operations

Tracing operations track all traffic sampling operations and record them in a log file in the /var/log directory. By default, this file is named /var/log/sampled. The default file size is 128K, and 10 files are created before the first one gets overwritten.

To trace traffic sampling operations, include the traceoptions statement at the [edit forwarding-options sampling] hierarchy level:

```
[edit forwarding-options sampling]
traceoptions {
  file filename {
    files number;
    size bytes;
    (world-readable | no-world-readable);
  }
}
```

Examples: Configuring Traffic Sampling

The following sections provide examples of configuring traffic sampling:

Sampling a Single SONET Interface on page 320

Sampling All Traffic from a Single IP Address on page 321

Sampling All FTP Traffic on page 322

Sampling a Single SONET Interface

The following configuration gathers statistical sampling information from a small percentage of all traffic on a single SONET interface and collects it in a file named sonet-samples.txt.

Create the filter:

```
[edit firewall family inet]
filter {
  input sample-sonet {
    then {
      sample;
      accept;
    }
  }
}
```

Apply the filter to the SONET interface:

```
[edit interfaces]
so-0/0/1 {
  unit 0 {
    family inet {
      filter {
        input sample-sonet;
      }
      address 216.127.68.254/32 {
        destination 216.127.74.7;
      }
    }
  }
}
```

```

    }
  }
}

```

Finally, configure traffic sampling:

```

[edit forwarding-options]
sampling {
  input {
    family inet {
      rate 100;
      run-length 2;
    }
  }
  output {
    file {
      filename sonet-samples-txt;
      files 40;
      size 5m;
    }
  }
}

```

Sampling All Traffic from a Single IP Address

The following configuration gathers statistical information about every packet entering the router on a specific Gigabit Ethernet port originating from a single source IP address of 168.45.92.31, and collects it in a file named samples-168-45-92-31.txt.

Create the filter:

```

[edit firewall family inet]
filter one-ip {
  term get-ip {
    from {
      source-address 168.45.92.31;
    }
    then {
      sample;
      accept;
    }
  }
}

```

Apply the filter to the Gigabit Ethernet interface:

```

[edit interfaces]
ge-4/1/1 {
  unit 0 {
    family inet {
      filter {
        input one-ip;
      }
      address 215.45.92.254;
    }
  }
}

```

```
    }
}
```

Finally, gather statistics on all the candidate samples; in this case, gather all statistics:

```
[edit forwarding-options]
sampling {
  input {
    family inet {
      rate 1;
    }
  }
  output {
    file {
      filename samples-215-45-92-31.txt;
      files 100;
      size 100k;
    }
  }
}
```

Sampling All FTP Traffic

The following configuration gathers statistical information about a moderate percentage of packets using the FTP data transfer protocol in the output path of a specific T3 interface, and collects the information in a file named t3-ftp-traffic.txt.

Create a filter:

```
[edit firewall family inet]
filter ftp-stats {
  term ftp-usage {
    from {
      destination-port [ftp ftp-data];
    }
    then {
      sample;
      accept;
    }
  }
}
```

Apply the filter to the T3 interface:

```
[edit interfaces]
t3-7/0/2 {
  unit 0 {
    family inet {
      filter {
        input ftp-stats;
      }
      address 141.35.78.254/32 {
        destination 141.35.78.4;
      }
    }
  }
}
```

Finally, gather statistics on 10 percent of the candidate samples:

```
[edit forwarding-options]
sampling {
  input {
    family inet {
      rate 10;
    }
  }
  output {
    file {
      filename t3-ftp-traffic.txt;
      files 50;
      size 1m;
    }
  }
}
```

Configuring Flow Monitoring

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

To configure flow monitoring you need to do the following:

- Configuring the Flow Monitoring Interface on page 324

- Configuring Flow Monitoring Properties on page 325

Configuring the Flow Monitoring Interface

To enable flow monitoring on the Monitoring Services PIC, include the following statements at the [edit interfaces] hierarchy level:

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

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

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

Include the boot-command statement to specify a boot image for the monitoring services interface.

Include the core-dump statement to enable storage of core files in /var/tmp.

Include the syslog statement to enable storage of system logging information in /var/log.

Configuring Flow Monitoring Properties

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

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

A monitoring instance is a named entity that specifies collector information under the monitoring *name* statement.

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

engine-id—Monitoring interface location.

engine-type—Platform-specific monitoring interface type.

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

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

To configure time periods for active flow monitoring and intervals of inactivity, include the flow-active-timeout and flow-inactive-timeout statements at the [edit forwarding-options monitoring *name* output] hierarchy level. The flow-active-timeout statement specifies the duration of an active flow; when it expires, the flow is exported. The flow-inactive-timeout statement specifies the interval of inactivity between active flows.

To configure the cflowd version number, include the export-format statement at the [edit forwarding-options monitoring *name* output] hierarchy level. For information on cflowd properties, see “Configuring cflowd” on page 327.

Example: Configuring Flow Monitoring

The following is an example of flow-monitoring properties configured to support input SONET interfaces, output monitoring services interfaces, and export to cflowd for flow analysis. To complete the configuration, you also need to configure the interfaces and set up a VPN routing and forwarding (VRF) instance. For a complete example, see the *JUNOS Feature Guide*. For information on cflowd, see “Configuring cflowd” on page 327.

```
[edit forwarding-options]
monitoring group1 {
  family inet {
    output {
      cflowd 192.168.245.2 port 2055;
      export-format cflowd-version-5;
      flow-active-timeout 60;
      flow-inactive-timeout 30;
      interface mo-4/0/0.1 {
        engine-id 1;
        engine-type 1;
        input-interface-index 44;
        output-interface-index 54;
        source-address 192.168.245.1;
      }
      interface mo-4/1/0.1 {
        engine-id 2;
        engine-type 1;
        input-interface-index 45;
        output-interface-index 55;
        source-address 192.168.245.1;
      }
      interface mo-4/2/0.1 {
        engine-id 3;
        engine-type 1;
        input-interface-index 46;
        output-interface-index 56;
        source-address 192.168.245.1;
      }
      interface mo-4/3/0.1 {
        engine-id 4;
        engine-type 1;
        input-interface-index 47;
        output-interface-index 57;
        source-address 192.168.245.1;
      }
    }
  }
}
```

Configuring cflowd

You can collect an aggregate of sampled flows and send the aggregate to a specified host that runs the cflowd application available from CAIDA (<http://www.caida.org>). Before you can perform flow aggregation, the routing protocol process must export the autonomous system (AS) path and routing information to the sampling process. To do this, include the route-record statement at the [edit routing-options] hierarchy level (for routing instances, include the statement at the [edit routing-instances routing-instance-name routing-options] hierarchy level):

```
[edit]
routing-options {
    route-record;
}
```

By default, flow aggregation is disabled.

By using cflowd, you can obtain various types of byte and packet counts of flows through a router. The cflowd application collects the sampled flows over a period of 1 minute. At the end of the minute, the number of samples to be exported are divided over the period of another minute and are exported over the course of the same minute.

To enable the collection of cflowd flow formats, include the cflowd statement at the [edit forwarding-options sampling output] or [edit forwarding-options accounting *name* output cflowd *hostname*] hierarchy level:

```
cflowd hostname {
    aggregation {
        autonomous-system;
        destination-prefix;
        protocol-port;
        source-destination-prefix {
            caida-compliant;
        }
        source-prefix;
    }
    autonomous-system-type (origin | peer);
    (local-dump | no-local-dump);
    port port-number;
    version format;
}
```

You can also configure cflowd version 5 for flow-monitoring applications by including the cflowd statement at the [edit forwarding-options monitoring *name* family inet output] hierarchy level:

```
cflowd hostname {
    port port-number;
}
```

The following restrictions apply to cflowd flow formats:

You can configure up to one version 5 and one version 8 flow format at the [edit forwarding-options accounting *name* output] hierarchy level.

You can configure only one version 5 or one version 8 flow format at the [edit forwarding-options sampling output] hierarchy level for Routing Engine-based sampling. In contrast, PIC-based sampling allows you to specify one cflowd version 5 server and one version 8 server simultaneously. However, the two cflowd servers must have different IP addresses.

You can configure up to eight version 5 flow formats at the [edit forwarding-options monitoring *name* output] hierarchy level. Version 8 flow formats and aggregation are not supported for flow-monitoring applications.

In the cflowd statement, specify the name or identifier of the host that collects the flow aggregates. You must also include the User Datagram Protocol (UDP) port number on the host and the version, which gives the format of the exported cflowd aggregates. To collect cflowd records in a log file before exporting, include the local-dump statement.



NOTE: You can specify both host (cflowd) sampling and port mirroring in the same configuration; however, only one action takes effect at any one time. Port mirroring takes precedence. For more information, see “Configuring Port Mirroring” on page 330.

For cflowd version 8 only, you can specify aggregation of specific types of traffic by including the aggregation statement. This conserves memory and bandwidth by enabling cflowd to export targeted flows rather than all aggregated traffic. To specify a flow type, include the aggregation statement at the [edit forwarding-options sampling output cflowd *hostname*] or [edit forwarding-options accounting *name* output cflowd *hostname*] hierarchy level:

```
aggregation {
  autonomous-system;
  destination-prefix;
  protocol-port;
  source-destination-prefix {
    caida-compliant;
  }
  source-prefix;
}
```

The autonomous-system statement configures aggregation by the AS number; this statement might require setting the separate cflowd autonomous-system-type statement to include either origin or peer AS numbers. The origin option specifies to use the origin AS of the packet source address in the Source Autonomous System cflowd field. The peer option specifies to use the peer AS through which the packet passed in the Source Autonomous System cflowd field. By default, cflowd exports the origin AS number.

The destination-prefix statement configures aggregation by the destination prefix only.

The protocol-port statement configures aggregation by the protocol and port number; requires setting the separate cflowd port statement.

The source-destination-prefix statement configures aggregation by the source and destination prefix. Version 2.1b1 of CAIDA's cflowd application does not record source and destination mask length values in compliance with CAIDA's *cflowd Configuration Guide*, dated August 30, 1999. If you configure the caida-compliant statement, the JUNOS software complies with Version 2.1b1 of cflowd. If you do not include the caida-compliant statement in the configuration, the JUNOS software records source and destination mask length values in compliance with the *cflowd Configuration Guide*.

The source-prefix statement configures aggregation by the source prefix only.

Collection of sampled packets in a local ASCII file is not affected by the cflowd statement.

Debugging cflowd Flow Aggregation

To collect the cflowd flows in a log file before they are exported, include the local-dump statement at the [edit forwarding-options sampling output cflowd *hostname*] hierarchy level:

```
[edit forwarding-options sampling output cflowd hostname]
local-dump;
```

By default, the flows are collected in /var/log/sampled; to change the filename, include the filename statement at the [edit forwarding-options sampling traceoptions] hierarchy level. For more information about changing the filename, see “Configuring Traffic Sampling Output” on page 318.



NOTE: Because the local-dump statement adds extra overhead, you should use it only while debugging cflowd problems, not during normal operation.

The following is an example of the flow information. The AS number exported is the origin AS number. All flows that belong under a cflowd header are dumped, followed by the header itself:

```
Jun 27 18:35:43 v5 flow entry
Jun 27 18:35:43 Src addr: 192.53.127.1
Jun 27 18:35:43 Dst addr: 192.6.255.15
Jun 27 18:35:43 Nhop addr: 192.6.255.240
Jun 27 18:35:43 Input interface: 5
Jun 27 18:35:43 Output interface: 3
Jun 27 18:35:43 Pkts in flow: 15
Jun 27 18:35:43 Bytes in flow: 600
Jun 27 18:35:43 Start time of flow: 7230
Jun 27 18:35:43 End time of flow: 7271
Jun 27 18:35:43 Src port: 26629
Jun 27 18:35:43 Dst port: 179
Jun 27 18:35:43 TCP flags: 0x10
Jun 27 18:35:43 IP proto num: 6
Jun 27 18:35:43 TOS: 0xc0
Jun 27 18:35:43 Src AS: 7018
Jun 27 18:35:43 Dst AS: 11111
Jun 27 18:35:43 Src netmask len: 16
```

```
Jun 27 18:35:43 Dst netmask len: 0
```

[... 41 more version 5 flow entries; then the following header:]

```
Jun 27 18:35:43 cflowd header:
Jun 27 18:35:43 Num-records: 42
Jun 27 18:35:43 Version: 5
Jun 27 18:35:43 Flow seq num: 118
Jun 27 18:35:43 Engine id: 0
Jun 27 18:35:43 Engine type: 3
```

Configuring Port Mirroring

On routers containing an Internet Processor II ASIC or T-series Internet Processor, you can send a copy of an IPv4 packet from the router to an external host address or a packet analyzer for analysis. This is known as *port mirroring*.

Port mirroring is different from traffic sampling. In traffic sampling, a sampling key based on the IPv4 header is sent to the Routing Engine. There, the key can be placed in a file, or cflowd packets based on the key can be sent to a cflowd server. In port mirroring, the entire packet is copied and sent out through a next-hop interface.

You can configure simultaneous use of sampling and port mirroring, and set an independent sampling rate and run-length for port-mirrored packets. However, if a packet is selected for both sampling and port mirroring, only one action can be performed and port mirroring takes precedence. For example, if you configure an interface to sample every packet input to the interface and a filter also selects the packet to be port mirrored to another interface, only the port mirroring would take effect. All other packets not matching the explicit filter port-mirroring criteria continue to be sampled when forwarded to their final destination.

To prepare traffic for port mirroring, include the filter statement at the [edit firewall family inet] hierarchy level:

```
[edit firewall family inet]
filter filter-name;
```

This filter at the [edit firewall family inet] hierarchy level selects traffic to be port-mirrored:

```
filter filter-name {
  term term-name {
    then {
      port-mirror;
      accept;
    }
  }
}
```

To configure port mirroring on a logical interface, configure the following statements at the [edit forwarding-options port-mirroring] hierarchy level:

```
[edit forwarding-options port-mirroring]
input {
  family inet {
    rate rate;
    run-length number;
  }
}
output {
  interface interface-name {
    next-hop address;
  }
  no-filter-check;
}
traceoptions {
  file filename {
    files number;
    size bytes;
    (world-readable | no-world-readable);
  }
}
```

Specify the port-mirroring destination by including the next-hop statement at the [edit forwarding-options port-mirroring output] hierarchy level:

```
[edit forwarding-options port-mirroring output]
interface interface-name {
  next-hop address;
}
```

The no-filter-check statement is required when you send port-mirrored traffic to a Tunnel PIC that has a filter applied to it.

The interface used to send the packets to the analyzer is the output interface configured above at the [edit forwarding-options port-mirroring output] hierarchy level. You can use any physical interface type, including generic routing encapsulation (GRE) tunnel interfaces. The next-hop address specifies the destination address; this statement is mandatory for non-point-to-point interfaces, such as Ethernet interfaces.

If your router is equipped with a Tunnel PIC, you can forward duplicate packets to multiple interfaces by configuring a next-hop group. To configure a next-hop group, include the next-hop-group statement at the [edit forwarding-options] hierarchy level:

```
[edit forwarding-options]
next-hop-group [ group-names ] {
  interface interface-name {
    next-hop [ addresses ];
  }
}
```

The interface statement specifies the interface that sends out sampled information. The next-hop statement specifies the next-hop addresses to which to send the sampled information.

Next-hop groups have the following restrictions:

Next-hop groups are supported for IPv4 addresses only.

Next-hop groups are supported on M-series routers only, except the M320.

Next-hop groups support up to 16 next-hop addresses.

Up to 30 next-hop groups are supported.

Each next-hop group must have at least two next-hop addresses.

To configure the sampling rate or duration, include the rate or run-length statements at the [edit forwarding-options port-mirroring input family inet] hierarchy level.

In typical applications, you send the sampled packets to an analyzer or a workstation for analysis, rather than another router. If you must send this traffic over a network, you should use tunnels. For more information about tunnel interfaces, see “Tunnel Interfaces Configuration Guidelines” on page 491.

You can trace port-mirroring operations in the same way as you trace sampling operations. For more information, see “Tracing Traffic Sampling Operations” on page 320.

The following restrictions apply to port-mirroring configurations:

You cannot configure firewall filters on the port-mirroring interface.

The interface you configure for port mirroring should not participate in any kind of routing activity.

The destination address you specify should not have a route to the ultimate traffic destination. For example, if the sampled IPv4 packets have a destination address of 190.68.9.10 and the port-mirrored traffic is sent to 190.68.20.15 for analysis, the device associated with the latter address should not know a route to 190.68.9.10. Also, it should not send the sampled packets back to the source address.

Only IPv4 traffic is supported.

Only transit data is supported.

You can configure only one port-mirroring interface per router. If you include more than one interface in the port-mirroring statement, the previous one is overwritten.

You must include a firewall filter with both the accept action and the sample action modifier on the inbound interface. Do not include the discard action, or port mirroring will not work.

If the port-mirroring interface is a non-point-to-point interface, you must include an IP address under the port-mirroring statement to identify the other end of the link. This IP address must be reachable for you to see the sampled traffic. If the port-mirroring interface is an Ethernet interface, the router should have an Address Resolution Protocol (ARP) entry for it. The following sample configuration sets up a static ARP entry.

You do not need to configure firewall filters on both inbound and outbound interfaces, but at least one is necessary on the inbound interface to provide the copies of the packets to send to an analyzer.

Examples: Configuring Port Mirroring

Send port-mirrored traffic to multiple cflowd servers or packet analyzers:

```
[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.
      }
    }
  }
}
[edit forwarding-options]
port-mirroring { # This is required when you configure next-hop groups.
  input {
    family inet {
      rate 1; # This rate port mirrors one packet for every one received (1:1 = all
              # packets).
    }
  }
  output { # This sends traffic to a tunnel interface to prepare for multipoint
           # mirroring.
    interface vt-3/3/0.1;
    no-filter-check;
  }
}
next-hop-group ftp-traffic { # Point-to-point interfaces require you to specify the interface
                             # name only.
  interface so-4/3/0.0;
  interface so-0/3/0.0;
}
next-hop-group http-traffic { # You need to configure a next hop for 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;
}
[edit 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.
        accept;
      }
    }
  }
  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;
    }
  }
}

```

Load Balancing Among Multiple Monitoring Interfaces

The active monitoring application was initially intended for port-mirroring packets on an interface on a normal network router to single or multiple destinations. By port-mirroring these packets to a tunnel interface and using filter-based forwarding on the tunnel interface, port-mirrored packets can be load-balanced across set of interfaces. This method employs existing configuration statements for passive monitoring.

The configuration consists of the following parts:

Firewall filter configuration—Firewall filter PORT-MIRROR-TO-VT is used to port-mirror the packet to a Tunnel PIC, and filter catch, applied on the virtual tunnel (vt) interface, is used to send traffic to a filter-based routing instance.

```
[edit firewall]
filter PORT-MIRROR-TO-VT {
  term a {
    then {
      port-mirror;
      accept;
    }
  }
}
filter catch {
  term def {
    then {
      count counter;
      routing-instance fbf_instance;
    }
  }
}
```

For more information about firewall filters, see the *JUNOS Policy Framework Configuration Guide*.

Interface configuration—Apply filter PORT-MIRROR-TO-VT to the interface on which traffic is to be monitored actively.

```
[edit interfaces]
ge-1/3/0 {
  unit 0 {
    family inet {
      filter {
        input PORT-MIRROR-TO-VT;
      }
      address 10.38.0.2/30;
    }
  }
}
vt-3/2/0 {
  unit 0 {
    family inet {
      filter {
        input catch;
      }
    }
  }
}
mo-6/1/0 {
  unit 0 {
    family inet;
  }
}
```

```

mo-6/2/0 {
  unit 0 {
    family inet;
  }
}
mo-6/3/0 {
  unit 0 {
    family inet;
  }
}
mo-7/1/0 {
  unit 0 {
    family inet;
  }
}
mo-7/2/0 {
  unit 0 {
    family inet;
  }
}
mo-7/3/0 {
  unit 0 {
    family inet;
  }
}

```

For more information on configuring interface properties, see the *JUNOS Network Interfaces and Class of Service Configuration Guide*.

Routing instance configuration for filter-based forwarding:

```

[edit routing-instances fbf_instance]
instance-type forwarding;
routing-options {
  static {
    route 0.0.0.0/0 next-hop [ mo-7/1/0.0 mo-7/2/0.0 mo-7/3/0.0
mo-6/3/0.0 mo-6/2/0.0 mo-6/1/0.0 ];
  }
}

```

For more information on routing instance configuration, see the *JUNOS Routing Protocols Configuration Guide*.

Routing table groups—Configure the routing table group to resolve the routes installed in the routing instances to directly connected next hops on the interface:

```
[edit routing-options]
interface-routes {
  rib-group inet common;
}
rib-groups {
  common {
    import-rib [ inet.0 fbf_instance.inet.0 ];
  }
}
forwarding-table {
  export pplb;
}
```

For more information on routing table groups, see the *JUNOS Routing Protocols Configuration Guide*.

Policy for per-packet load balancing:

```
[edit policy-options]
policy-statement pplb {
  then {
    load-balance per-packet;
  }
}
```

For more information on routing policy groups, see the *JUNOS Policy Framework Configuration Guide*.

Port mirroring and monitoring groups—Configure the monitoring services options, and also define hash-based load balancing:

```
[edit forwarding-options]
port-mirroring {
  input {
    family inet {
      rate 1;
    }
  }
  output {
    interface vt-3/2/0.0;
    no-filter-check;
  }
}
```

```

monitoring group1 {
  family inet {
    output {
      export-format cflowd-version-5;
      flow-active-timeout 60;
      flow-inactive-timeout 15;
      cflowd 10.36.252.1 port 2055;
      interface mo-6/1/0.0 {
        source-address 10.36.252.2;
      }
      interface mo-6/2/0.0 {
        source-address 10.36.252.2;
      }
      interface mo-6/3/0.0 {
        source-address 10.36.252.2;
      }
      interface mo-7/1/0.0 {
        source-address 10.36.252.2;
      }
      interface mo-7/2/0.0 {
        source-address 10.36.252.2;
      }
      interface mo-7/3/0.0 {
        source-address 10.36.252.2;
      }
    }
  }
}
hash-key {
  family inet {
    layer-3;
  }
}

```

For more information on hash keys, see the *JUNOS Policy Framework Configuration Guide*.

Configuring Discard Accounting

Discard accounting is similar to traffic sampling, but varies from it in two ways:

In discard accounting, the packet is intercepted by the Monitoring PIC and is not forwarded to its destination.

Traffic sampling allows you to limit the number of packets sampled by configuring the `max-packets-per-second`, `rate`, and `run-length` statements. Discard accounting does not provide these options and a high packet count can potentially overwhelm the Monitoring PIC.

To configure discard accounting, include the following statements at the [edit forwarding-options] hierarchy level:

```
[edit forwarding-options]
accounting name {
  output {
    aggregate-export-interval seconds;
    cflowd hostname {
      aggregation {
        autonomous-system;
        destination-prefix;
        protocol-port;
        source-destination-prefix {
          caida-compliant;
        }
        source-prefix;
      }
      autonomous-system-type (origin | peer);
      port port-number;
      version format;
    }
    flow-active-timeout seconds;
    flow-inactive-timeout seconds;
    interface interface-name {
      engine-id number;
      engine-type number;
      source-address address;
    }
  }
}
```

A discard instance is a named entity that specifies collector information under the accounting *name* statement. Discard instances are referenced in firewall filter term statements by including the then discard accounting *name* statement.

Most of the other statements are also found at the [edit forwarding-options sampling] hierarchy level. For information on cflowd, see “Configuring cflowd” on page 327. The flow-active-timeout and flow-inactive-timeout statements are described in “Configuring Flow Monitoring” on page 323.

To direct sampled traffic to a flow-monitoring interface, include the interface statement. The engine-id and engine-type statements specify the accounting interface used on the traffic, and the source-address statement specifies the traffic source.

You cannot use rate-limiting with discard accounting; however, you can specify the duration of the interval for exporting aggregated accounting information by including the aggregate-export-interval statement in the configuration. This enables you to put a boundary on the amount of traffic exported to a flow-monitoring interface.

Enabling Passive Flow Monitoring

The Monitoring Services PIC is one of a group of multiservice PICs designed to enable IP services. If you have Monitoring Services PICs or Adaptive Services (AS) PICs and SONET/SDH PICs installed in an M40e, M160, M320, or T-series routing platform, you can monitor IPv4 traffic from another router.

On Asynchronous Transfer Mode (ATM) or SONET/SDH interfaces, you enable passive flow monitoring by including the `passive-monitor-mode` statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

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

If you include the `passive-monitor-mode` statement in the configuration:

The ATM interface is always up, and the interface does not receive or transmit incoming control packets, such as Operation, Administration, and Maintenance (OAM) and Interim Local Management Interface (ILMI) cells.

The SONET/SDH interface does not send keepalives or alarms, and does not participate actively on the network.

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

```
[edit interfaces mo-fpc/pic/port unit logical-unit-number]
family inet;
```

For the monitoring services interface, you can configure multiservice physical interface properties. For more information, see the *JUNOS Network Interfaces and Class of Service Configuration Guide*.

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

```
[edit interfaces mo-fpc/pic/port unit logical-unit-number family inet]
receive-options-packets;
receive-ttl-exceeded;
```

Passive Flow Monitoring for MPLS Encapsulated Packets

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

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

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

For more information about static labels, see the *JUNOS MPLS Applications Configuration Guide*.

Removing MPLS Labels from Incoming Packets

The JUNOS software can forward only IPv4 packets to a Monitoring Services PIC. IPv4 packets with MPLS labels cannot be forwarded to a Monitoring PIC. By default, if packets with MPLS labels are forwarded to the Monitoring PIC, they are discarded. To monitor packets with MPLS labels, you must remove the MPLS labels as the packets arrive on the interface.

You can remove up to two MPLS labels from an incoming packet by including the `pop-all-labels` statement at the `[edit interfaces interface-name atm-options mpls]` hierarchy level:

```
[edit interfaces interface-name atm-options mpls]
pop-all-labels {
  required-depth number;
}
```

By default, the `pop-all-labels` statement takes effect for incoming packets with one or two labels. You can specify the number of MPLS labels that an incoming packet must have for the `pop-all-labels` statement to take effect by including the `required-depth` statement at the `[edit interfaces interface-name atm-options mpls pop-all-labels]` hierarchy level:

```
[edit interfaces interface-name atm-options mpls pop-all-labels]
required-depth number;
```

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

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

The pop-all-labels statement has no effect on IP packets with three or more MPLS labels.

When you enable MPLS label removal, you must configure all ports on a PIC with the same label popping mode and required depth.

You use the pop-all-labels statement to enable passive monitoring applications, not active monitoring.

You cannot apply MPLS filters or accounting to the MPLS labels because the labels are removed as soon as the packet arrives on the interface.

The following ATM encapsulation types are not supported on interfaces with MPLS label removal:

atm-ccc-cell-relay

atm-ccc-vc-mux

atm-mlppp-llc

atm-tcc-snap

atm-tcc-vc-mux

ether-over-atm-llc

ether-vpls-over-atm-llc

