

Chapter 11

Policer Configuration

To configure policers, you include statements at the [edit firewall] hierarchy level of the configuration:

```
[edit firewall]
policer policer-name {
  filter-specific;
  if-exceeding {
    bandwidth-limit bps;
    bandwidth-percent number;
    burst-size-limit bytes;
  }
  then {
    policer-action;
  }
}
interface-set interface-set-name {
  [ interface-names ];
}
family family-name {
  filter filter-name {
    accounting-profile name;
    interface-specific;
  }
  prefix-action name {
    count;
    destination-prefix-length prefix-length;
    policer policer-name;
    source-prefix-length prefix-length;
    subnet-prefix-length prefix-length;
  }
}
load-balance-group group-name {
  next-hop-group [ group-names ];
}
```

The following sections describe the tasks required for configuring policers and provide configuration examples:

Minimum Policer Configuration on page 204

Configuring Policers on page 205

Configuring an Interface Set on page 213

Applying an Interface Policer on page 214

Configuring a Load-Balance Group on page 215

Examples: Configuring Policing on page 215

Minimum Policer Configuration

To configure a policer, you must perform at least the following tasks:

Configure policers—To configure policers, include the policer statement at the [edit firewall] hierarchy level. After policers are defined, you reference them in the then clause of a term:

```
[edit firewall]
policer policer-name {
  filter-specific;
  if-exceeding {
    bandwidth-limit bps;
    bandwidth-percent number;
    burst-size-limit bytes;
  }
  then {
    policer-action;
  }
}
family family-name {
  filter filter-name {
```

Add actions, such as accept, discard, or next term, or action modifiers, such as count or log.

Apply the policers to an interface to activate them.

The policer is applied to the packet first, and if the packet exceeds the defined limits, the actions of the then clause of the policer are applied. If the result of the policing action is not a discard, the remaining components of the then clause of the term are applied.

To display statistics about a filter statement policer configuration, use the show policers command.

Configuring Policers

To configure term-specific policers, include the policer statement at the [edit firewall] hierarchy level:

```
[edit firewall]
policer policer-name {
  filter-specific;
  if-exceeding {
    bandwidth-limit rate;
    bandwidth-percent number;
    burst-size-limit bytes;
  }
  then {
    policer-action;
  }
}
```

The following sections describe the components of the policer statement and provide policer configuration examples:

Configuring Rate Limiting on page 205

Configuring a Policer Action on page 206

Configuring Multifield Classification and Policing on page 207

Configuring Filter-Specific Policers on page 207

Configuring Prefix-Specific Actions on page 208

Examples: Classifying Traffic on page 212

Configuring Rate Limiting

To specify the rate limiting part of a policer, include an if-exceeding statement at the [edit firewall policer *policer-name*] hierarchy level:

```
[edit firewall policer policer-name]
if-exceeding {
  bandwidth-limit bps;
  bandwidth-percent number;
  burst-size-limit bytes;
}
```

You specify the bandwidth limit in bits per second (bps). You can specify the value as a complete decimal number or as a decimal number followed by the abbreviation k (1000), m (1,000,000), or g (1,000,000,000). There is no absolute minimum value for bandwidth limit, but any value below 61,040 bps will result in an effective rate of 30,520 bps. The maximum bandwidth limit is 4.29 gigabits per second (Gbps).

You can rate-limit based upon port speed. This port speed can be specified by a bandwidth percentage in a policer. You must specify the percentage as a complete decimal number between 1 and 100.



NOTE: You cannot rate-limit based on bandwidth percentage for aggregate, tunnel, and software interfaces. The bandwidth percentage policer cannot be used for forwarding table filters. This can only be used for interface specific filters.

The maximum burst size controls the amount of traffic bursting allowed. To determine the value for the burst-size limit, the preferred method is to multiply the bandwidth of the interface on which you are applying the filter by the amount of time you allow a burst of traffic at that bandwidth to occur; for example, 5 milliseconds.

$$\text{burst size} = \text{bandwidth} \times \text{allowable time for burst traffic}$$

If you do not know the interface bandwidth, you can multiply the maximum transmission unit (MTU) of the traffic on the interface by 10 to obtain a value. For example, the burst size for an MTU of 4700 would be 47,000 bytes. At minimum, burst size should be at least 10 interface MTUs. The maximum value for the burst-size limit is 100 megabits per second (Mbps).

For a sample filter configuration for rate limiting, see “Examples: Configuring Policing” on page 215.

Configuring a Policer Action

If a packet does not exceed its rate limits, it is processed further without being affected. If the packet exceeds its limits, it is handled in one of two ways, depending on what you specify:

Discarded

Marked for subsequent processing based on its loss priority and forwarding class

To configure a policer action, include the then statement at the [edit firewall policer *policer-name*] hierarchy level:

```
[edit firewall policer policer-name]  
then {  
    policer-action;  
}
```

Policer actions include one or more of the following:

discard—Discard a packet that exceeds the rate limits.

loss-priority level—Set the loss priority level to low or high.

forwarding-class class name—Specify the forwarding class to any class name already configured for the forwarding class.

Example: Configuring a Policer Action

Discard any packet that exceeds a bandwidth of 300 kilobits per second (Kbps) and a burst-size limit of 500 kilobytes (KB):

```
[edit firewall]
policer p1 {
  if-exceeding {
    bandwidth-limit 300k;
    burst-size-limit 500k;
  }
  then {
    discard;
  }
}
```

Configuring Multifield Classification and Policing

You can configure *multifield classifiers* within a firewall filter to set the packet's forwarding class and packet loss priority. You can also apply policers to packets matching some classification term. The policing action might affect the resulting forwarding class, packet loss priority, and accept or drop status. For more information, see the *JUNOS Network Interfaces and Class of Service Configuration Guide*.

To configure the forwarding class and loss priority, include the then statement:

```
then {
  loss-priority level;
  forwarding-class class-name;
}
```

You can include the statement at the following hierarchy levels:

```
[edit firewall filter filter-name term term-name]
```

```
[edit firewall policer policer-name]
```

You can specify one or both of the following actions:

loss-priority—Set the loss priority level to low or high.

forwarding-class—Specify the forwarding class to any class name already configured for the forwarding class.

For more information about forwarding class and loss priority, see the *JUNOS Network Interfaces and Class of Service Configuration Guide*.

Configuring Filter-Specific Policers

You can configure filter-specific policers within the firewall configuration. Filter-specific policers allow you to configure policers and counters for a specific filter name.

To configure filter-specific policers, include the filter-specific statement at the [edit firewall policer *policer-name*] hierarchy level:

```
[edit firewall policer policer-name]  
filter-specific;
```

If the filter-specific statement is not configured, then the policer defaults to a term-specific policer.

You can apply the filter-specific policers to the family inet.

Configuring Prefix-Specific Actions

You can configure prefix-specific actions within the firewall configuration. Prefix-specific actions allow you to configure policers and counters for specific addresses or ranges of addresses. This allows you to essentially create policers and counters on a per-prefix level.

To configure prefix-specific actions, include the prefix-action *name* statement at the [edit firewall family inet] hierarchy level:

```
[edit firewall family inet]  
prefix-action name {  
  count;  
  destination-prefix-length prefix-length;  
  policer policer-name;  
  source-prefix-length prefix-length;  
  subnet-prefix-length prefix-length;  
}
```

The following formula determines the number of prefix-specific actions created:

$$\text{Number} = 2 ^ {(\text{source/destination-prefix-length} - \text{subnet-prefix-length})}$$

The subnet-prefix-length statement allows for more control for the flexibility offered by prefix-specific actions, allowing the policers to be more applicable and powerful. For example, if you want to filter all Transmission Control Protocol (TCP) packets and define two policers, all packets ending with 0 in the last address bit increment the first policer, while all packets ending with 1 in the address bit increment the second policer. As another example, if you want to filter all TCP packets and define 256 policers, matching is based on the last octet of the destination address field. You achieve both cases by specifying an appropriate subnet prefix length.

Prefix-specific action is supported for the IP version 4 (IPv4) inet address family.

To configure prefix-specific actions, include the prefix-action statement and specify an action name.

To enable a prefix-specific counter, include the count statement.

To configure the destination address range specified for a prefix-specific policer or counter, include the destination-prefix-length statement.

To enable a set of prefix-specific policers, include the policer statement and specify the policer name.

To configure the source address range specified for a prefix-specific policer or counter, include the `source-prefix-length` statement.

To configure the total address range of the subnet supported, include the `subnet-prefix-length` statement. The source or destination prefix length must be larger than the subnet prefix length.

Prefix-specific action applies to a specific prefix length, and not to a specific interface. You can add an interface policer policies at the aggregate level for a specific interface. You could also use the `next term` action to configure all Hypertext Transfer Protocol (HTTP) traffic to each host to transmit at 500 Kbps and have the total HTTP traffic limited to 1 Mbps.

The maximum number of policers you can configure for one subnet is 65,536. If you configure more than 65,536 policers, you receive an error message.

Examples: Configuring Prefix-Specific Actions

Create a prefix-specific policer operating on the source address and apply it to the input interface:

```
[edit]
firewall {
  policer host-policer {
    filter-specific;
    if-exceeding {
      bandwidth-limit bps;
      burst-size-limit bytes;
    }
    then {
      discard;
    }
  }
  family inet {
    prefix-action ftp-policer-set {
      count;
      destination-prefix-length 32;
      policer host-policer;
      subnet-prefix-length 24;
    }
    filter filter-ftp {
      term term1 {
        from {
          destination-address 10.10.10/24;
          destination-port ftp;
        }
        then {
          prefix-action ftp-policer-set;
        }
      }
    }
  }
}
```

Filter all packets going to the /24 subnet, letting them pass to the prefix-specific action policers. In the policer set, the last octet of the source address field of the packet is used to index into the respective prefix-specific action policers.

```
[edit]
firewall {
  policer 1Mbps-policer {
    if-exceeding {
      bandwidth-limit 1m;
      burst-size-limit 63k;
    }
  }
  family inet {
    prefix-action per-source-policer {
      policer 1Mbps-policer;
      subnet-prefix-length 24;
      source-prefix-length 32;
    }
  }
  filter limit-all-hosts {
    term one {
      from {
        source-address {
          10.10.10.0/24;
        }
      }
      then prefix-action per-source-policer;
    }
  }
}
```

In the above case, all packets are subjected to the prefix-specific action policing. The last octet of the source address field of the packet is used to index into the corresponding policer. In other words, all packets ending with 0x(00000000) match the first policer and all packets ending in 0x(00000001) match the second policer.

Therefore, 256 policers are created and shared by all addresses. In this case, 10.1.1.1, 10.2.2.1, 10.4.5.1 ... 10.x.x.1 share the same 1Mbps-policer; 10.1.1.2, 10.2.2.2, 10.4.5.2 ... 10.x.x.2 share another 1Mbps-policer, and so on.

Subject packets belonging to the 10.10.10.0/24 subnet are subject to policing by the prefix-specific action policers. Because 128 policers defined in the policer set, the /24 subnet can be thought of as being split into two /25 subnets, both of them sharing the same prefix-specific action set. Therefore, 10.10.10.1 and 10.10.10.129 share the same 1Mbps policer, 10.10.10.2 and 10.10.10.130 share another 1Mbps policer, and so on.

```
[edit]
firewall {
  policer 1Mbps-policer {
    if-exceeding {
      bandwidth-limit 1m;
      burst-size-limit 63k;
    }
  }
}
```

```

family inet {
  prefix-action per-source-policer {
    policer 1Mbps-policer;
    subnet-prefix-length 25;
    source-prefix-length 32;
  }
}
filter limit-all-hosts {
  term one {
    from {
      source-address {
        10.10.10.0/24;
      }
    }
    then prefix-action per-source-policer;
  }
}
}

```

Define 256 policers based on the last octet of the source address field. However, you are only allowing a subset of that to pass through the match condition. As a result, only the lower half of the set is used.

```

[edit]
firewall {
  policer 1Mbps-policer {
    if-exceeding {
      bandwidth-limit 1m;
      burst-size-limit 63k;
    }
  }
  family inet {
    prefix-action per-source-policer {
      policer 1Mbps-policer;
      subnet-prefix-length 24;
      source-prefix-length 32;
    }
  }
  filter limit-all-hosts {
    term one {
      from {
        source-address {
          10.10.10.0/25;
        }
      }
      then prefix-action per-source-policer;
    }
  }
}
}

```

Accept packets from 10.10.10/24 and 10.11/16 subnets and subject them to policing by the same set of prefix-specific action policers. The policers are shared by packets across both subnets. There is a one-to-one correspondence between the 10.10.10/24 subnet. For 10.11/16, there is a many-to-one correspondence, as explained in the previous examples. Each of the 10.11.0/24, 10.11.1/24, 10.11.2/24 ... 10.11.255/24 subnets share the same prefix-specific action set.

Thus, 10.10.10.1, 10.11.1.1, 10.11.2.1 ... 10.11.x.1 share the same 1Mbps-policer; 10.10.10.2, 10.11.1.2, 10.11.2.2 ... 10.11.x.2 share another 1Mbps-policer, and so on.

```
[edit]
firewall {
  policer 1Mbps-policer {
    if-exceeding {
      bandwidth-limit 1m;
      burst-size-limit 63k;
    }
  }
  family inet {
    prefix-action per-source-policer {
      policer 1Mbps-policer;
      subnet-prefix-length 24;
      source-prefix-length 32;
    }
  }
  filter limit-all-hosts {
    term one {
      from {
        source-address {
          10.10.10/24;
          10.11/16;
        }
      }
      then prefix-action per-source-policer;
    }
  }
}
```

Examples: Classifying Traffic

Classify expedited forwarding traffic:

```
[edit]
firewall {
  policer ef-policer {
    if-exceeding {
      bandwidth-limit 300k;
      burst-size-limit 50k;
    }
    then {
      discard;
    }
  }
}
```

```

    term ef-multifield {
      then {
        loss-priority low;
        forwarding-class expedited-forwarding;
        policer ef-policer;
      }
    }
  }
}

```

Classify assured forwarding traffic:

```

firewall {
  policer af-policer {
    if-exceeding {
      bandwidth-limit 300k;
      burst-size-limit 500k;
    }
    then {
      loss-priority high;
    }
  }
  term af-multifield {
    then {
      loss-priority low;
      forwarding-class assured-forwarding;
      policer af-policer;
    }
  }
}
}

```

Configuring an Interface Set

In addition to including policers in firewall filters, you can configure an interface set that is not part of a firewall filter configuration. An interface set groups a number of interfaces into one interface set name.

To configure an interface set, include the `interface-set` statement at the [edit firewall] hierarchy level:

```

[edit firewall]
interface-set interface-set-name {
  [ interface-names ];
}

```

You must specify more than one interface name to configure an interface set. This interface set can be used for firewall filter matching.

Applying an Interface Policer

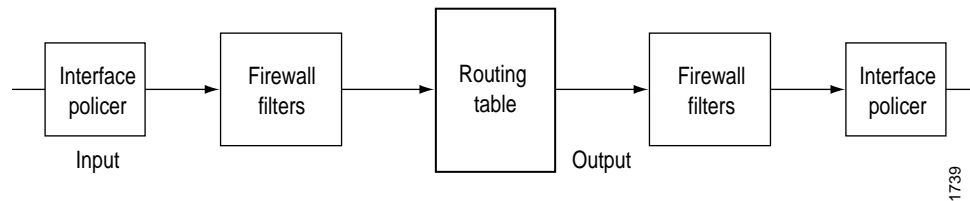
In addition to including policers in firewall filters, you can apply an interface policer that is not part of a firewall filter configuration. An interface policer can be applied to each family on an interface.

To apply an interface policer, include the policer statement 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]
  policer {
    input policer-name;
    output policer-name;
  }
```

You must first configure the policer at the [edit firewall] hierarchy level before you can apply it to an interface. Both input and output policers are allowed, and can be used in conjunction with existing firewall filters. Input interface policers are evaluated before any input firewall filters. Likewise, output interface policers are evaluated after any output firewall filters (see Figure 12 on page 214).

Figure 12: Incoming and Outgoing Interface Policers



To display a policer on a particular interface, issue the `show interfaces policers` command at the command-line interface (CLI).

Example: Applying an Interface Policer

Apply a policer on circuit cross-connect (CCC) interfaces:

```

[edit interfaces]
so-0/0/0 {
  encapsulation ppp-ccc;
  unit 0 {
    family ccc {
      policer {
        input dragnet;
      }
    }
  }
}

```

Configuring a Load-Balance Group

In addition to including policers in firewall filters, you can configure a load-balance group that is not part of a firewall filter configuration. A load-balance group contains interfaces that all use the same next-hop group characteristic to load-balance the traffic.

To configure a load-balance group, include the `load-balance-group` statement at the `[edit firewall]` hierarchy level:

```

[edit firewall]
load-balance-group group-name {
  next-hop-group [ group-names ];
}

```

Next-hop groups allow you to include multiple interfaces used to forward duplicate packets used in port mirroring. For more information about next-hop groups, see “Configuring a Next-Hop Group” on page 244.

Examples: Configuring Policing

The following example shows a complete filter configuration containing a policer. It limits all FTP traffic from a given source to certain rate limits. Traffic exceeding the limits is discarded, and the remaining traffic is accepted and counted.

```

[edit]
firewall {
  policer policer-1 {
    if-exceeding {
      bandwidth-limit 400k;
      burst-size-limit 100k;
    }
    then {
      discard;
    }
  }
}

```

```

term tcp-ftp {
  from {
    source-address 10.2.3/24;
    protocol tcp;
    destination-port ftp;
  }
  then {
    policer policer-1;
    accept;
    count count-ftp;
  }
}
}

```

The following example shows a complete filter configuration containing two policers, and includes the next term action. Policer policer-1 limits all traffic from a given source to certain rate limits, then sets the forwarding class. Policer policer-2 limits all traffic to a second set of rate limits. Traffic exceeding the limits is discarded; the remaining traffic is accepted.

```

[edit]
firewall {
  policer policer-1 {
    if-exceeding {
      bandwidth-limit 10m;
      burst-size-limit 100k;
    }
    then {
      forwarding-class 0;
    }
  }
  policer policer-2 {
    if-exceeding {
      bandwidth-limit 100m;
      burst-size-limit 100k;
    }
    then {
      discard;
    }
  }
  filter f {
    term term-1 {
      then {
        policer policer-1;
        next term;
      }
    }
    term term-2 {
      then {
        policer policer-2;
        accept;
      }
    }
  }
}
}

```

The following example limits all FTP traffic from a given source to certain rate limits, but defines the policer outside the filter, thereby creating a template that can be referenced by more than one filter or more than one term within a filter. Traffic exceeding the limits is discarded, and the remaining traffic is accepted and counted.

```
[edit]
firewall {
  policer policer-1 {
    if-exceeding {
      bandwidth-limit 400k;
      burst-size-limit 100k;
    }
    then {
      discard;
    }
  }
  filter limit-ftp {
    term tcp-ftp {
      from {
        source-address 10.2.3/24;
        protocol tcp;
        destination-port ftp;
      }
      then {
        policer policer-1;
        accept;
        count count-ftp;
      }
    }
  }
}
```

The following example shows a filter intended to thwart denial-of-service (DoS) SYN attacks:

```
[edit]
firewall {
  policer syn-recvd {
    if-exceeding {
      bandwidth-limit 40k;
      burst-size-limit 15000;
    }
    then discard;
  }
  term allow-syn {
    from {
      source-address {
        192.168.12.50/32; # trusted addresses
      }
    }
    then {
      log;
      accept;
    }
  }
}
```

```

term limit-syn {
  from {
    protocol tcp;
    tcp-initial;
  }
  then {
    count limit-syn;
    policer syn-recvd;
    accept;
  }
}
term default {
  then accept;
}
}

[edit] # apply filter to lo0 to control traffic to the Routing Engine
interfaces {
  lo0 {
    unit 0 {
      family inet {
        filter {
          input syn-attack;
        }
      }
      address 172.16.4.53/32;
    }
  }
}

```

The following example uses one filter to do the following:

Stop all User Datagram Protocol (UDP) and Internet Control Message Protocol (ICMP) traffic destined to these addresses (in term a).

Send ICMP through the policer (in term b).

Accept ICMP traffic within contract and all other traffic (in term c).



NOTE: It is important to keep the terms in order; once a packet has a match within the firewall filter, it is not examined in subsequent terms. For example, if you configured the filter to send ICMP traffic through the policer before discarding ICMP and UDP traffic to those addresses, it would not work.

```

[edit firewall]
policer policer-1 {
  if-exceeding {
    bandwidth-limit 200k;
    burst-size-limit 3k;
  }
  then {
    loss-priority 1;
    forwarding-class 1;
  }
}

```

```
term a {
  from {
    destination-address {
      10.126.50.2/23;
      10.130.12.1/23;
      10.82.16.0/24 except;
      10.82.0.3/18;
    }
    protocol [icmp udp];
  }
  then {
    count packets-dropped;
    discard;
  }
}
term b {
  from {
    protocol icmp;
  }
  then policer policer-1;
}
term c {
  then accept;
}
```

