AT&T

QoS Configuration Guide, 17.2.0

November 2017

Supporting AT&T Vyatta Network Operating System

Contents

Copyright Statement

The training materials and other content provided herein for assistance in training on the Vyatta vRouter may have references to Brocade as the Vyatta vRouter was formerly a Brocade product prior to AT&T's acquisition of Vyatta. Brocade remains a separate company and is not affiliated to AT&T.

About This Guide

This guide describes the QoS architecture and how to configure QoS on AT&T products that run on the AT&T Vyatta Network Operating System (referred to as a virtual router, vRouter, or router in the guide).

QoS

Overview

This section covers the following topics of the quality of service (QoS) features on the AT&T Vyatta vRouter:

- QoS architecture *(page 9)*
- Monitoring *(page 13)*
- Configuration limits *(page 10)*
- Queue and traffic classes *(page 10)*
- Mapping *(page 11)*
- Classification *(page 12)*
- Remark *(page 13)*
- Frame overhead *(page 13)*

QoS architecture

QoS allows network administrators to identify different traffic flows and treat them according to their individual requirements, rather than simply using the default mechanism, which is to directly forward traffic to hardware. QoS provides queue prioritization for traffic that is transmitted out a physical interface.

In addition to the default queuing mechanism, the AT&T Vyatta vRouter provides a variety of QoS mechanisms for identifying and treating the various traffic flows that pass through an interface. In general, mechanisms apply to outbound traffic.

The QoS default queue priority is first in, first out (FIFO).

The general work flow for nondefault QoS mechanisms is as follows:

1. Create a QoS policy.
2. Apply the policy to an interface.

A QoS policy identifies traffic flows and specifies how each flow is to be treated. Policies allow traffic flows to be classified into a queue belonging to a traffic class. The traffic classes provide a priority queue mechanism for the flows.

If no QoS policy is set on an interface, the default behavior allows traffic to skip QoS processing and pass directly to the destination interface.

To configure QoS on the AT&T Vyatta vRouter, the commands can be found under the following configuration command nodes:

- `policy qos name` *policy-name* **`shaper bandwidth`**
- `policy qos name` *policy-name* **`shaper burst`**
- `policy qos name` *policy-name* **`shaper class`**
- `policy qos name` *policy-name* **`shaper default`**
- `policy qos name` *policy-name* **`shaper description`**
- `policy qos name` *policy-name* **`shaper frame-overhead`**
- `policy qos name` *policy-name* **`shaper period`**
- `policy qos name` *policy-name* **`shaper profile`**
- `policy qos name` *policy-name* **`shaper traffic-class`**

To define QoS policy definitions, use the following command:

```
vyatta@vyatta# set policy qos name policy1 shaper ?
Possible completions:
   bandwidth      Bandwidth limit
   burst          Burst size
```

```
+> class          Class number
   default        Qos profile for default traffic
   description    Description for this queuing policy
   frame-overhead Framing overhead
   period         Enforcement period (ms)
+> profile        QoS traffic profile
+> traffic-class  Traffic Class
```

To assign a policy to an interface, use the following command:

`set interfaces dataplane` *interface_name* **`policy`** **`qos`** policy_name

QoS is supported on all dataplane interfaces except tunnel interfaces.

Configuration limits

The following are the configuration limits of QoS:

- 4 queues per traffic class
- 4 traffic classes per QoS profile
- 255 classes per QoS policy
- 256 profiles per QoS policy

Queue and traffic classes

Queuing configuration is the QoS scheduling algorithmbased on class and interface. Four queues are providedper traffic class and up to four traffic classes are available in the class, for a total of 16 configurable queues as configured in the profile.

Traffic classes are prioritized in ascending order. The four queues within a traffic class are scheduled in weighted round robin (WRR) order. The weights can be configured to provide different bandwidth allocations for each queue within a traffic class.

Table 1: Queue-traffic class mapping

Traffic class	WRR queues
Traffic-class 3	4 WRR slots (1 through 100)
Traffic-class 2	4 WRR slots (1 through 100)
Traffic-class 1	4 WRR slots (1 through 100)
Traffic-class 0	4 WRR slots (1 through 100)

> **Note:** While 16 queues are now supported, only a maximum 4 queues can share the same traffic class. It is possible to have unused traffic classes (that is, have no queues assigned).

Each QoS policy provides the following attributes:

There are up to 256 profiles per policy.

- There is strict priority scheduling by traffic class within a QoS profile.
- You can configure a maximum of 4 WRR queues per traffic class.
- You can check the queues and their assigned traffic class by using the CLI during the validation stage.
- You can configure a traffic-class with no queues assigned.

Each queue has the following configurable parameters:

- Traffic class
    - Strict-priority assignment.

  - Must be set for each queue.
  - Priorities are ordered from 0 (highest priority) to 3 (lowest priority).
  - A maximum of four queues can be assigned to a traffic class.
  - Queues are serviced by the round robin method.
- Weight
  - The Weighted Round Robin value.
  - Determines the proportion of bandwidth a queue receives when multiple queues share the same priority.
  - Can be a number between 1 and 100. This number does not necessarily need to represent a percentage.
  - The default weight is 1.

Mapping

QoS mapping is based on priority for IPv4 or IPv6 traffic. Packets are mapped to queues based on either 802.1p priority (if present) or Differentiated Services Code Point (DSCP) for IPv4 or IPv6 traffic.

Each profile has a table mapping of all the possible Priority Code Point (PCP) and DSCP traffic to queue. For the default mapping, a packet's DSCP or PCP value is mapped into the corresponding traffic class with the range being spread evenly as shown in the following tables.

The following table shows the default DSCP value to traffic class/queue mapping.

Table 2: Default DSCP value to traffic class and queue mapping

DSCP value	Traffic Class	Queue
48-64	0	0
32-47	1	0
16-31	2	0
0-15	3	0

The following table shows the default PCP value to traffic class/queue mapping.

Table 3: Default PCP value to traffic class and queue mapping

PCP value	Traffic Class	Queue
6-7	0	0
4-5	1	0
2-3	2	0
0-1	3	0

To configure values for DSCP through the CLI, you must use numeric format, symbolic format, or a range of numbers. The numeric form must conform to the standard POSIX input method: a decimal number and a hex number preceded by 0x.

Table 4: DSCP values

Name	Decimal	Hex
default	0	0x00
af11	10	0x0A

Name	Decimal	Hex
af12	12	0x0C
af13	14	0x0E
af21	18	0x12
af22	20	0x14
af23	22	0x16
af31	26	0x1A
af32	28	0x1C
af33	30	0x1E
af41	34	0x22
af42	36	0x24
af43	38	0x26
cs1	8	0x08
cs2	16	0x10
cs3	24	0x18
cs4	32	0x20
cs5	40	0x28
cs6	48	0x30
cs7	56	0x38
ef	23	0x2E

Lists must be comma separated items or a number range separated by a minus sign (-).

PCP mapping is used only if the packet has a VLAN header and the profile that the packet is shaped by has a PCP map explicitly configured with the `set policy qos name shaper profile map pcp` command.

Classification

QoS classification uses a subset of the packet classification that is used in policy-based routing and firewall. QoS classification allows matching of packets based on the source and destination values of IP and MAC addresses as well as DSCP and PCP values.

> **Note:** You can configure rules to match IPv4 ICMP, IPv6 ICMP, IPv6 routing header, or TCP without specifying the respective protocol, provided that a protocol specific match option is present. For example TCP flags, ICMP type.

The QoS classification process assigns a packet to a class. These classes are identified by one or more match rules based on a subset of the firewall command syntax.

Classes are evaluated in numerical order. The first class that matches is used (that is, they are final). The class numbers do not have to be sequential (and the system accepts gaps in the number sequence), but the largest class number determines the size of the internal data structures. Therefore, using large numbers is discouraged. Even though classes look like firewall rules, they are not stateful. Each class is either associated with an action which can either be a QoS scheduling profile or drop.

Classifying a packet based on the TCP/IP n-tuple can be configured through the following command:

`set policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *match-name* **`protocol tcp`**

Remark

The access control list (ACL) can include rules to remark a packet by changing the DSCP or PCP values. Changes that are made during the classification process occur before the packet is evaluated for scheduling.

For example, if the QoS scheduler has a rule to set all DSCP packets to traffic class 0, then these packets are set to queue 0 in traffic class 0 (the highest-priority traffic class).

You can remark DSCP and PCP values with the `set policy qos name` *policy-name* **`shaper class`** *1* **`match`** *1* **`mark`** [**`dscp`** | **`pcp`**] *value* command.

Frame overhead

QoS can be adjusted to adapt to the constraints of the destination system. Configure frame overhead which makes allowances for additional bytes of a packet as a result of the underlying link-layer protocols. Use the `set policy qos name` *policy-name* **`shaper frame-overhead`** command to configure frame overhead.

Monitoring

To display QoS statistics and the configuration of the mapping of packets to queues, use the `show queuing` or `monitor queuing` operational command.

Per packet logging

You can configure the vRouter to log every packet that matches a network packet filter rule.

> **Note:** Per packet logging generates large amounts of output and can negatively affect the performance of the entire system. Use per packet logging only for debugging purposes.

When logging is enabled, all log messages appear in the /var/log/dataplane/vplane.log file. This file is rotated and compressed daily, and the last seven log files are automatically maintained by the system.

AT&T recommends limiting per packet logging to debugging. Per packet logging occurs in the forwarding paths and can greatly reduce the throughput of the system and dramatically increase the disk space used for the log files.

To implement per packet logging for debugging purposes, include the **log** keyword when specifying a rule. When the logging option is specified, a log message containing the parameters of the packet is generated and logged.

QoS on bonded link

QoS is supported with all bonding modes.

> **Note:** You cannot configure QoS on a bonding interface member. However, you can configure QoS on the actual bonding interface.

QoS policies

The AT&T Vyatta vRouter supports the following QoS features on outbound traffic:

RED and WRED

The QoS policy random-detect mechanism is a congestion-avoidance mechanism based on traffic class and includes Random Early Detection (RED) and Weighted Random Early Detection (WRED).

Congestion occurs when output buffers are allowed to fill such that packets must be dropped. Congestion can cause global resynchronization of TCP hosts as multiple hosts reduce their transmission rates to try to clear the congestion; this congestion can significantly affect network performance. As congestion clears, the network increases transmission rates again until the point at which congestion reoccurs. This cycle of congestion and clearing does not make the best use of the available bandwidth.

RED determines the likelihood of a packet being dropped in the outgoing queue and queues them accordingly to an interface. It reduces the chance that network congestion occurs by randomly dropping packets when the output interface begins to show signs of congestion. The packet drops act as a signal to the source to decrease its transmission rate which, in turn, helps avoid conditions of congestion and reduces the chance of global resynchronization, making better use of network bandwidth.

WRED takes RED one step further by providing a way to attach precedence to different traffic streams. Differential QoS can then be provided to different traffic streams by dropping more packets from some streams than from others.

RED is configured per queue weight, probability, and a maximum and minimum threshold queue depth. After a minimum threshold is met, QoS begins to drop packets at increasing rates until the maximum threshold is met, at which time the system drops all packets for the queue.

Exponentially Weighted Moving Average (EWMA) tracks traffic queue length based on traffic rates and the passage of time. EWMA can be assigned a filter with a weight value.

If RED is disabled, all traffic classes are handled as strict drop tail (drop packets when queue is full).

Bandwidth

Allows the bandwidth associated with a shaping node to be configured with an absolute value or a percentage of the interface bandwidth.

Round-robin

The QoS-policy round-robin mechanism is a simple scheduling algorithm. In round-robin queuing, each queue is scheduled in turn. The default behavior is to distribute the bandwidth evenly.

Weighted Round-Robin (WRR) is designed to spread the available bandwidth among the queues according to the assigned weight.

Traffic shaper

The QoS-policy shaper mechanism controls the transmission rate of outgoing traffic, particularly limiting bursts of packets and limiting bandwidth.

When a policy is configured, it can be applied to a class of a packet and a behavior can be applied to packet to direct how the packet is handled at the outgoing interface.

The QoS-policy shaper provides queuing that is based on the token bucket shaping algorithm. This algorithm allows for bursting if a bucket has tokens to spend.

The shaper algorithm limits bandwidth usage based on class and then allocates any leftover bandwidth.

`Round-robin`, on the other hand, attempts to divide all available bandwidth equally between the defined classes.

Traffic class

The QoS-policy priority-queue mechanism is a scheduling algorithm. Packets are placed in the queues based on match criteria associated with each queue. Packets are retrieved from the queues in priority order. Packets in lower priority queues will not be transmitted until those in higher priority queues have been sent. If packets continually fill higher priority queues, those waiting on lower priority queues will not be serviced until the higher priority traffic load abates.

Queueing at the traffic class level is based on strict priority. To avoid having lower priority queues deprived of bandwidth, configure a maximum bandwidth for the traffic class. After the maximum is reached, the lower priority queues will be scheduled.

Default-traffic prioritization

By default, a packet is prioritized based on the value in its PCP or DSCP field and sent to one of the queues. The packets on the highest priority queue are sent out first, followed by those on the next-highest priority queue, followed by those on the lowest priority queue. Within each queue, packets are sent through the interface based on traffic class assigned to a queue, then on a Weighted Round-Robin (WRR) handling.

If traffic arrives at a queue faster than it can be delivered (for example, because of bandwidth limitations), it is buffered within the system. If more data arrives than the system can buffer, the excess is dropped.

Data traffic is divided in this way because providing equal levels of service for all traffic is not always desirable. Some types of traffic, by their nature, should be treated differently than others. For example, voice traffic is very sensitive to delay and, if it is not processed accordingly, could be unintelligible. Data, on the other hand, is not sensitive to delay, but is sensitive to corruption.

QoS Configuration Examples

Configuration examples

This section provides the following quality of service (QoS) configuration examples for implementing QoS traffic shaping on outbound traffic:

- Configuring a QoS policy *(page 16)*
- Configuring a class profile *(page 17)*
- Configuring traffic class *(page 18)*
- Configuring RED *(page 19)*
- Configuring mapping to queues *(page 20)*
- Configuring ACLs *(page 21)*
- Configuring WRR *(page 22)*
- Configuring remarking *(page 23)*
- QoS configuration example *(page 24)*

Configuring a QoS policy

For all QoS configurations, begin by creating a QoS policy.

The following table shows how to configure a QoS policy.

Table 5: Configuring a QoS policy

Step	Command
Define a policy for default traffic.	`vyatta@R1# set policy qos name policy1 shaper default def`
Specify an associated profile and bandwidth for default traffic.	`vyatta@R1# set policy qos name policy1 shaper profile def bandwidth 3mbit`
Apply the policy to a dataplane interface.	`vyatta@R1# set interfaces dataplane dp0s3 policy qos policy1`
Commit the configuration.	`vyatta@R1# commit`

View the configuration.	`vyatta@R1# show policy qos name policy1`
	`name policy1 {`
	`        shaper {`
	`                default def`
	`                profile def {`
	`                        bandwidth 3mbit`
	`                }`
	`        }`
	`}`

Configuring a class profile

The profile is the description of a policy for a customer. The profile is used to describe different throughput groups. For example, Premium, Normal, Guest.

The following table shows how to configure a class profile. For all QoS configurations, begin by defining a QoS policy, described in Configuring a QoS policy *(page 16)*.

Table 6: Configuring a class profile

Step	Command
Specify the QoS policy class and protocol.	`vyatta@R1# set policy qos name policy1 shaper class 1 match 1 protocol tcp`
Specify the profile name of the QoS policy class.	`vyatta@R1# set policy qos name policy1 shaper class 1 profile prof1`
Specify the bandwidth for the defined QoS profile.	`vyatta@R1# set policy qos name policy1 shaper profile prof1 bandwidth 1mbit`
Define the mapping of DSCP traffic to a queue for the QoS policy.	`vyatta@R1#  set policy qos name policy1 shaper profile prof1 map dscp af11 to 3`
Define the mapping of traffic class to a queue for the QoS policy.	`vyatta@R1# set policy qos name policy1 shaper profile prof1 queue 3 traffic-class 1`
Commit the configuration.	`vyatta@R1# commit`

<table>
<tr><th>Step</th><th>Command</th></tr>
<tr><td>View the configuration.</td><td><pre>
vyatta@R1# show policy qos name policy1
name policy1 {
        shaper {
                class 1 {
                        match 1 {
                                protocol tcp
                        }
                        profile prof1
                }
                default def
                profile def {
                        bandwidth 3mbit
                }
                profile prof1 {
                        bandwidth 1mbit
                        map {
                                dscp af11 {
                                        to 3
                                }
                        }
                        queue 3 {
                                traffic-class
 1
                        }
                }
        }
}
</pre></td></tr>
</table>

Configuring traffic class

Traffic class configuration is applied globally per profile.

The following procedure shows how to configure traffic class. For all QoS configurations, begin by defining a QoS policy, described in Configuring a QoS policy *(page 16)*.

Table 7: Configuring traffic class

Step	Command
Create the traffic class and assign it a name and bandwidth.	`vyatta@R1# set policy qos name policy1 shaper traffic-class 1 bandwidth 300kbit`
Define the queue limit as the number of packets queued before dropping.	`vyatta@R1# set policy qos name policy1 shaper traffic-class 1 queue-limit 128`
Commit the configuration.	`vyatta@R1# commit`

<table>
<tr><th>Step</th><th>Command</th></tr>
<tr><td>View the configuration.</td><td><pre>vyatta@R1# show qos name policy1 shaper
 traffic-class
       traffic-class 1 {
               bandwidth 300kbit
               queue-limit 128
       }
}</pre></td></tr>
</table>

Configuring RED

The following procedure shows how to configure RED. For all QoS configurations, begin by defining a QoS policy, described in Configuring a QoS policy *(page 16)*.

Table 8: Configuring random early detection (RED)

<table>
<tr><th>Step</th><th>Command</th></tr>
<tr><td>Create a traffic class and assign it a number from 0 through 3. Needed only if traffic class is not already defined.</td><td><pre>vyatta@R1# set policy qos name policy1 shaper
 traffic-class 1</pre></td></tr>
<tr><td>Configure the exponentially weighted moving average (EWMA) filter weight with a number from 1 through 12.</td><td><pre>vyatta@R1# set policy qos name policy1 shaper
 traffic-class 1 random-detect filter-weight
 1</pre></td></tr>
<tr><td>Configure the maximum value for the inverse of packet marking probability with a number from 1 through 255.</td><td><pre>vyatta@R1# set policy qos name policy1
 shaper traffic-class 1 random-detect mark-
probability 2</pre></td></tr>
<tr><td>Configure the maximum threshold for the queue with the number of packets from 1 through 1023.</td><td><pre>vyatta@R1# set policy qos name policy1 shaper
 traffic-class 1 random-detect max-threshold
 100</pre></td></tr>
<tr><td>Configure the minimum threshold for the queue with of the number of packets from 1 through 1022.</td><td><pre>vyatta@R1# set policy qos name policy1 shaper
 traffic-class 1 random-detect min-threshold
 5</pre></td></tr>
<tr><td>Commit the configuration.</td><td><pre>vyatta@R1# commit</pre></td></tr>
<tr><td>View the configuration.</td><td><pre>vyatta@R1# show policy qos name policy1
 shaper traffic-class 1
traffic-class 1 {
       random-detect {
               filter-weight 1
               mark-probability 2
               max-threshold 100
               min-threshold 5
       }
}</pre></td></tr>
</table>

Configuring mapping to queues

The following procedure shows how to configure mapping to queues for DSCP traffic. For all QoS configurations, begin by defining a QoS policy, described in Configuring a QoS policy *(page 16)*.

Table 9: Configuring mapping

Step	Command
Create a mapping of DSCP traffic types 10 and 11-13 to queue 1.	`vyatta@R1# set policy qos name policy1 shaper profile profile1 map dscp 10,11-13 to 1`
Create a mapping of DSCP traffic types 5-8 to queue 3.	`vyatta@R1# set policy qos name policy1 shaper profile profile1 map dscp 5-8 to 3`
Create a mapping of traffic class 0 and queue 1.	`vyatta@R1# set policy qos name policy1 shaper profile profile1 queue 1 traffic-class 0`
Create a mapping of traffic class 1 and queue 3.	`vyatta@R1# set policy qos name policy1 shaper profile profile1 queue 3 traffic-class 1`
Assign bandwidth to traffic class 0.	`vyatta@R1# set policy qos name policy1 shaper profile profile1 traffic-class 0 bandwidth 200kbit`
Assign bandwidth to traffic class 1.	`vyatta@R1# set policy qos name policy1 shaper profile profile1 traffic-class 1 bandwidth 15`
Commit the configuration.	`vyatta@R1# commit`

<table>
<tr><th>Step</th><th>Command</th></tr>
<tr><td>View the configuration.</td><td><pre>
vyatta@R1# show policy qos name policy1
 shaper profile profile1
profile profile1 {
       map {
               dscp 5-8 {
                       to 3
               }
               dscp 10,11-13 {
                       to 1
               }
       }
       queue 1 {
               traffic-class 0
       }
       traffic-class 0 {
               bandwidth 200kbit
       }
       queue 3 {
               traffic-class 1
       }
       traffic-class 0 {
               bandwidth 15kbit
       }
}
</pre></td></tr>
</table>

Configuring ACLs

Access Control Lists (ACLs) are based on the source and destination address, port, and protocol values.

The following table shows how to configure an ACL. For all QoS configurations, begin by defining a QoS policy, described in Configuring a QoS policy *(page 16)*.

Table 10: Configuring an ACL

Step	Command
Create a class that matches the policy rule.	`vyatta@R1# set policy qos name policy1 shaper class 1`
Create a profile for class 1.	`vyatta@R1# set policy qos name policy1 shaper class 1 profile profile1`
Define the protocol type of the traffic to match.	`vyatta@R1# set policy qos name policy1 shaper class 1 match http-in protocol tcp`
Define the source port of the traffic to match.	`vyatta@R1# set policy qos name policy1 shaper class 1 match http-in source port http`
Define the destination port of the traffic to match.	`vyatta@R1# set policy qos name policy1 shaper class 1 match http-out destination port http`
Define the protocol type of the traffic to match.	`vyatta@R1# set policy qos name policy1 shaper class 1 match http-out protocol tcp`

<table>
<tr><th>Step</th><th>Command</th></tr>
<tr><td>Commit the configuration.</td><td><pre>vyatta@R1# commit</pre></td></tr>
<tr><td>View the configuration using the <code>show policy</code> command.</td><td><pre>vyatta@R1:~$ show policy qos name policy1
 shaper class
class 1 {
        match http-in {
                destination {
                        port http
                }
                protocol tcp
                source {
                        port http
                }
        }
        match http-out {
                protocol tcp
        }
        profile profile1
}</pre></td></tr>
</table>

Configuring WRR

The following procedure shows how to configure WRR. The example assigns WRR to queues 2 and 3.

For all QoS configurations, begin by defining a QoS policy, described in Configuring a QoS policy *(page 16)*.

Table 11: WRR

Step	Command
Map a DSCP value to queue 2.	`vyatta@R1# set policy qos name policy1 shaper profile profile1 map dscp af11 to 2`
Map a DSCP value to queue 3.	`vyatta@R1# set policy qos name policy1 shaper profile profile1 map dscp af21 to 3`
Assign queue 2 to a traffic class.	`vyatta@R1# set policy qos name policy1 shaper profile profile1 queue 2 traffic-class 1`
Assign queue 3 to a traffic class.	`vyatta@R1# set policy qos name policy1 shaper profile profile1 queue 3 traffic-class 1`
Define the weight value for queue 2.	`vyatta@R1# set policy qos name policy1 shaper profile profile1 queue 2 weight 100`
Define the weight value of queue 3.	`vyatta@R1# set policy qos name policy1 shaper profile profile1 queue 3 weight 200`
Commit the configuration.	`vyatta@R1# commit`

<table>
<tr><th>Step</th><th>Command</th></tr>
<tr><td>View the configuration.</td><td><pre>vyatta@R1# show policy qos name policy1
profile def {
   map {
     dscp af11 {
       to 2
     }
     dscp af21 {
       to 3
     }
   }
   queue 2 {
     traffic-class 1
     weight 100
   }
   queue 3 {
     traffic-class 1
     weight 200
   }
}</pre></td></tr>
</table>

Configuring remarking

If the QoS scheduler has a rule to set all DSCP packets to traffic class 0, then all packets are set to the lowest priority queue 3.

DSCP and PCP values can be remarked by the user through the `set policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`mark`** command.

For all QoS configurations, begin by defining a QoS policy, described in Configuring a QoS policy *(page 16)*.

The following procedure shows the remarking of DSCP packets.

Table 12: Configuring remarking

Step	Command
Create the class-matching rule and provide a description to identify it in the `show policy` command output.	`vyatta@R1# set policy qos name policy1 shaper class 1 match match1 description "dscp class 40"`
Define the criteria to match the destination port.	`vyatta@R1# set policy qos name policy1 shaper class 1 match match1 destination port bgp`
Define the criteria to match DSCP packets.	`vyatta@R1# set policy qos name policy1 shaper class 1 match match1 mark dscp 40`
Define the criteria to match a protocol.	`vyatta@R1# set policy qos name policy1 shaper class 1 match match1 protocol tcp`
Commit the configuration.	`vyatta@R1# commit`

<table>
<tr><th>Step</th><th>Command</th></tr>
<tr><td>View the configuration.</td><td>

```
vyatta@R1# show policy qos name policy1
name policy1 {
       shaper {
               class 1 {
                       match match1 {
                               description
 "dscp class 40"
                               destination {
                                       port
 bgp
                               }
                               mark {
                                       dscp
 40
                               }
                               protocol tcp
                       }
               }
       }
}
```

</td></tr>
</table>

QoS configuration example

The following example shows the configuration of four traffic classes:

```
vyatta@R1# show policy

policy {
     qos {
       name policy1 {
         shaper {
             default example-queue
             description "example"
             profile example-queue {
                 bandwidth 1Gbit
                 map {
                     dscp 24 {
                         to 3
                     }
                     dscp 25 {
                         to 2
                     }
                     dscp 40 {
                         to 1
                     }
                     dscp 46 {
                         to 0
                     }
                 }
                 queue 0 {
                     description dscp46
                     traffic-class 0
                 }
                 queue 1 {
                     description dscp40
                     traffic-class 0
                     weight 60
                 }
                 queue 2 {
                     description dscp25
                     traffic-class 1
                     weight 30
                 }
                 queue 3 {
                     description dscp24
                     traffic-class 1
                     weight 10
                 }
             }
             traffic-class 0 {
                 bandwidth 590000
                 description "Highest priority"
             }
             traffic-class 1 {
                 description "Best effort"
                 bandwidth 390000
             }
         }
       }
     }
}
```

Monitoring QoS

This section provides the following QoS monitoring examples:

- Statistics *(page 25)*
- Priority maps *(page 26)*
- Monitoring QoS statistics *(page 26)*

Statistics

The QoS scheduler keeps track of the number of packets and bytes that pass through the system.

To view the QoS statistics for all devices, use the following command:

```
vyatta@R1:~$ show queuing
```

Example:

```
vyatta@R1:~$ show queuing
Interface        Prio    Packets            Bytes  Tail-drop   RED-drop
------------------------------------------------------------------------
dp0s4               0          0                0          0          0
                    1          0                0          0          0
                    2          0                0          0          0
                    3          0                0          0          0
dp0s3               0          0                0          0          0
                    1          0                0          0          0
                    2          0                0          0          0
                    3         13             1754          0          0
```

To view the QoS statistics for a single interface, use the following command:

```
vyatta@R1:~$ show queuing interface
```

Example:

```
vyatta@R1:~$ show queuing dp0s4
Class    Prio  WRR Qlength    Packets            Bytes Tail-drop  RED-drop
---------------------------------------------------------------------------
0           0    0       0          0                0         0         0
                 1       0          0                0         0         0
                 2       0          0                0         0         0
                 3       0          0                0         0         0
            1    0       0          0                0         0         0
```

```
                 1       0          0                0        0         0
                 2       0          0                0        0         0
                 3       0          0                0        0         0
            2    0       0          0                0        0         0
                 1       0          0                0        0         0
                 2       0          0                0        0         0
                 3       0          0                0        0         0
            3    0       0          0                0        0         0
                 1       0          0                0        0         0
                 2       0          0                0        0         0
                 3       0          0                0        0         0
```

To view queuing class statistics, use the following command:

```
vyatta@R1:~$ show queuing class
Interface        Prio    Packets          Bytes  Match
--------------------------------------------------------------------------------
dp0s5            1       0                0      proto 6 to any port 179 tag 41943041  apply mark
 dscp
vyatta@vyatta:~$
```

Example:

```
vyatta@R1:~$ show queuing dp0p2p1.100
...
```

Priority maps

To view individual DSCP maps, use the following command:

```
show queuing interface map dscp
```

Example:

```
vyatta@R1:~$ show queuing dp0s4 map dscp
DSCP->TC:WRR map for default: (dscp=d1d2)
     d2 |    0    1    2    3    4    5    6    7    8    9
  d1    |
  ------+---------------------------------------------------
     0  |  3:0  3:0  3:0  3:0  3:0  3:0  3:0  3:0  3:0  3:0
     1  |  3:0  3:0  3:0  3:0  3:0  3:0  2:0  2:0  2:0  2:0
     2  |  2:0  2:0  2:0  2:0  2:0  2:0  2:0  2:0  2:0  2:0
     3  |  2:0  2:0  1:0  1:0  1:0  1:0  1:0  1:0  1:0  1:0
     4  |  1:0  1:0  1:0  1:0  1:0  1:0  1:0  1:0  0:0  0:0
     5  |  0:0  0:0  0:0  0:0  0:0  0:0  0:0  0:0  0:0  0:0
     6  |  0:0  0:0  0:0  0:0
```

To view the 802.1p priority code point map, use the following command:

```
show queuing interface map pcp
```

Example:

```
vyatta@R1:~$ show queuing dp0s4 map pcp
Class Of Service->TC:WRR map for default
  PCP |    0    1    2    3    4    5    6    7
  ----+------------------------------------------
      |  3:0  3:0  2:0  2:0  1:0  1:0  0:0  0:0
```

Monitoring QoS statistics

The `monitor queuing` command provides a dynamic view of dataplane queue statistics by showing changes to statistics over time. A positive number indicates an increase in a particular traffic statistic while a negative number indicates a decrease in a particular traffic statistic over the past one-second period. Use the `monitor queuing` command to view changes in the general flow of traffic over time.

```
vyatta@R1:~$ monitor queuing

Interface          Prio    Packets            Bytes  Tail-drop   RED-drop
--------------------------------------------------------------------------
dp0p1s2               0         -2             -196          0          0
                      1          0                0          0          0
                      2          0                0          0          0
                      3          0                0          0          0
```

Use **Ctrl-C** to cancel this operation.

QoS Commands

interfaces bonding <dpFbondx> policy qos <policy-name>

Applies a QoS policy to an Ethernet link bond group.

Syntax:
`set interfaces bonding` *dpFbondx* **`policy`** **`qos`** *policy-name*

dpFbondx
The identifier for a bond group. The identifier ranges from **`dp0bond0`** through **`dp0bond99`**.
policy-name
The name of a QoS policy.

Configuration mode

```
interfaces {
 bonding dpFbondx {
  policy {
   qos policy-name
  }
 }
}
```

Use the `set` form of this command to apply a QoS policy to an Ethernet link bond group.

interfaces bonding <dpFbondx> vif <vif-id> policy qos <policy-name>

Applies a QoS policy to a virtual Ethernet link bond group.

Syntax:
`set interfaces bonding` *dpFbondx* **`vif`** *vif-id* **`policy`** **`qos`** *policy-name*

dpFbondx
The identifier a bond group. The identifier ranges from **`dp0bond0`** through **`dp0bond99`**.
vif-id
A virtual interface ID.
policy-name
The name of a QoS policy.

Configuration mode

```
interfaces {
 bonding dpFbondx {
  vif vif-id {
   policy {
    qos policy-name
   }
  }
 }
}
```

Use the `set` form of this command to apply a QoS policy to a virtual Ethernet link bond group.

interfaces dataplane <interface-name> policy qos <policy-name>

Applies a QoS policy to an interface.

Syntax:
`set interfaces dataplane` *interface-name* **`policy qos`** *policy-name*

Syntax:
`delete interfaces dataplane` *interface-name* **`policy qos`**

Syntax:
`show interfaces dataplane` *interface-name* **`policy qos`**

`dataplane` ***interface-name***
The name of the dataplane interface.

`policy qos` ***policy-name***
The name of a QoS policy.

Configuration mode

```
interfaces {
    dataplane interface-name {
        policy {
           qos policy-name
        }
    }
}
```

Use the `set` form of this command to apply a QoS policy to an interface. QoS is supported on all dataplane interfaces except tunnel interfaces.

Use the `delete` form of this command to delete a QoS policy from an interface.

Use the `show` form of this command to display the QoS policies that are applied to an interface.

interfaces dataplane <interface-name> vif <vif-id> policy qos <policy-name>

Applies a QoS policy to a virtual interface.

Syntax:
`set interfaces dataplane` *interface-name* **`vif`** *vif-id* **`policy qos`** *policy-name*

Syntax:
`delete interfaces dataplane` *interface-name* **`vif`** *vif-id* **`policy qos`**

Syntax:
`show interfaces dataplane` *interface-name* **`vif`** *vif-id* **`policy qos`**

`dataplane` ***interface-name***
The name of the dataplane interface.

`vif` ***vif-id***
The virtual interface ID.

`policy qos` ***policy-name***
The name of a QoS policy.

Configuration mode

```
interfaces {
        dataplane interface-name {
                }
                vif vif-id {
                        policy {
                                qos policy-name
                        }
                }
        }
}
```

Use the `set` form of this command to apply a QoS policy to a virtual interface.

Use the `delete` form of this command to delete a QoS policy from a virtual interface.

Use the `show` form of this command to display the QoS policies that are applied to a virtual interface.

monitor queuing

Monitors traffic for dataplane queues.

Syntax:

```
monitor queuing
```

Operational mode

Use this command to start monitoring traffic for dataplane queues.

Use **Ctrl-C** to stop the monitoring operation.

The `monitor queuing` command displays the following information:

Output field	Description
Interface	Interface for which packets are queued.
Prio	Traffic class for which statistics are displayed.
Packets	Increase or decrease in the number of packets that are matched since the last refresh. A positive number indicates an increase. A negative number indicates a decrease.
Bytes	Increase or decrease in the number of bytes that are matched since the last refresh. A positive number indicates an increase. A negative number indicates a decrease.
Tail-drop	Increase or decrease in the number of packets dropped because the queue is full since the last refresh. A positive number indicates an increase. A negative number indicates a decrease.
RED-drop	Increase or decrease in the number of packets dropped due to random early detection (RED) since the last refresh. A positive number indicates an increase. A negative number indicates a decrease.

The following example shows how to start dataplane queue monitoring and provides a traffic snapshot. A positive number indicates an increase in a particular traffic statistic while a negative number indicates a decrease in a particular traffic statistic over the past one-second period.

```
vyatta@R1:~$ monitor queuing

Interface         Prio    Packets            Bytes  Tail-drop   RED-drop
```

```
-------------------------------------------------------------------
dp0p1s2             0        -2            -196        0         0
                    1         0               0        0         0
                    2         0               0        0         0
                    3         0               0        0         0
```

policy qos name <policy-name>

Creates a QoS policy.

Syntax:
`set policy qos name` *policy-name*

Syntax:
`delete policy qos [` **`name`** *policy-name* `]`

Syntax:
`show policy qos name`

`name` ***policy-name***
A name for the QoS policy.

Configuration mode

```
policy {
    qos {
        name policy-name {
        }
    }
}
```

Use the `set` form of this command to create a QoS policy.

Use the `delete` form of this command to delete a QoS policy.

Use the `show` form of this command to display the QoS policy configuration.

policy qos name <policy-name> shaper bandwidth <limit>

Defines the bandwidth of a QoS policy.

Syntax:
`set policy qos name` *policy-name* **`shaper bandwidth`** { *number* | *number-and-suffix* }

Syntax:
`delete policy qos name` *policy-name* **`shaper bandwidth`** [*number* | *number-and-suffix*]

Syntax:
`show policy qos name` *policy-name* **`shaper bandwidth`**

`name` ***policy-name***
The name of a QoS policy.

`bandwidth` ***limit***
The bandwidth rate as a number followed by no space and a scaling suffix representing the rate (for example, 10mbit). The following suffixes are supported:

No suffix: Kilobits per second

`gbit`: Gigabits per second

`mbit`: Megabits per second

kbit: Kilobits per second

gbps: Gigabytes per second

mbps: Megabytes per second

kbps: Kilobytes per second

x%: Percent of total bandwidth

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                bandwidth
                    bandwidth number%
                    bandwidth number
                    bandwidth number-and-suffix
            }
        }
    }
}
```

Use the `set` form of this command to define the bandwidth of a QoS policy.

Use the `delete` form of this command to delete the bandwidth of a QoS policy.

Use the `show` form of this command to display the bandwidth of a QoS policy.

policy qos name <policy-name> shaper burst <limit>

Sets the burst size limit of a QoS policy.

Syntax:
`set policy qos name` *policy-name* **shaper burst** *limit*

Syntax:
`delete policy qos name` *policy-name* **shaper burst** [*limit*]

Syntax:
`show policy qos name` *policy-name* **shaper burst**

`name` ***policy-name***
The name of a QoS policy.

`burst` ***limit***
The burst size limit in number of bytes. The number can range from 1 through 312500000.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                burst limit
            }
        }
    }
}
```

Use the `set` form of this command to set the burst size limit of a QoS policy.

Use the `delete` form of this command to delete the burst size limit of a QoS policy.

Use the `show` form of this command to display the burst size limit of a QoS policy.

policy qos name <policy-name> shaper class <class-id> description <description>

Describes a QoS policy class for ease of identification when viewing a configuration.

Syntax:
`set policy qos name` *policy-name* **`shaper class`** *class-id* **`description`** *description*

Syntax:
`delete policy qos name` *policy-name* **`shaper class`** *class-id* **`description`**

Syntax:
`show policy qos name` *policy-name* **`shaper class`** *class-id* **`description`**

`name` ***policy-name***
The name of a QoS policy.

`class` ***class-id***
The number of the QoS policy class. The number ranges from 1 through 255.

`description` ***description***
A description of the QoS policy class.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                class class-id {
                    description description
                }
            }
        }
    }
}
```

Use the `set` form of this command to create a description of a QoS policy class.

Use the `delete` form of this command to delete the description of a QoS policy class.

Use the `show` form of this command to display the description of a QoS policy class.

policy qos name <policy-name> shaper class <class-id> match <rule-name> action <action>

Defines the action to take on packets when the packets meets the match criteria.

Syntax:
`set policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`action`** { **`drop`** | **`pass`** }

Syntax:
`delete policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`action`** [**`drop`** | **`pass`**]

Syntax:
`show policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`action`**

`name` ***policy-name***
The name of a QoS policy.

`class` ***class-id***

The number of the QoS policy class. The number ranges from 1 through 255.

`match` ***rule-name***

The name of the class-matching rule—the rule that specifies the class that must be matched.

`action [ drop | pass ]`

The action to take when the rule matches (pass the packets through or drop them). The default action is pass.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                class class-id {
                    match rule-name {
                        action {drop | pass}
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to define the action to take on packets when the packets meets the match criteria.

Use the `delete` form of this command to delete the configuration that defines the action to take on packets when the packets meet the match criteria.

Use the `show` form of this command to display the configuration that defines the action to take on packets when the packets meet the match criteria.

policy qos name <policy-name> shaper class <class-id> match <match-name> application name <name>

Matches applications (for example, social media) by name.

Syntax:

`set policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *match-name* **`application name`** *name*

Syntax:

`delete policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *match-name* **`application name`** *name*

Syntax:

`show policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *match-name* **`application name`** *name*

policy-name

The name of a QoS policy.

class-id

The number of a QoS policy class. The number ranges from 1 through 255.

match-name

The name of a class-matching rule—the rule that specifies the class that must be matched.

name

The name of the application. You can configure a single application name to be matched from a list of Deep Packet Inspection (DPI) engine applications at the most granular level. For more information about DPI, refer to AT&T Vyatta Network Operating System Policy-based Routing Configuration Guide.

Configuration mode

```
policy {
        qos {
                name policy-name {
                        shaper {
                                class class-id {
                                        match match-name {
                                                application {
                                                        name application-name
                                                }
                                        }
                                }
                        }
                }
        }
}
```

Use the `set` form of this command to match applications by name.

Use the `delete` form of this command to delete applications by name.

Use the `show` form of this command to display applications by name.

The following table lists related commands that are documented elsewhere.

Related commands documented elsewhere	
show application name <name>	Displays the application name and associated application types. (Refer to AT&T Vyatta Network Operating System Policy-based Routing Configuration Guide.)
show application type <type>	Displays the application names associated with the given application type. (Refer to AT&T Vyatta Network Operating System Policy-based Routing Configuration Guide.)

policy qos name <policy-name> shaper class <class-id> match <match-name> application type <type>

Matches applications by type.

Syntax:
`set policy qos type` *policy-name* **`shaper class`** *class-id* **`match`** *match-name* **`application type`** *type*

Syntax:
`delete policy qos type` *policy-name* **`shaper class`** *class-id* **`match`** *match-name* **`application type`** *type*

Syntax:
`show policy qos type` *policy-name* **`shaper class`** *class-id* **`match`** *match-name* **`application type`** *type*

policy-name
The name of a QoS policy.

class-id
The number of a QoS policy class. The number ranges from 1 through 255.

match-name
The name of a class-matching rule—the rule that specifies the class that must be matched.

type
A type of application. The application type provides access to less granular groups of DPI classifications such as analytics, database, social networking. An application can have multiple application types. You can configure a single application type to be matched from a list of Deep Packet Inspection (DPI) engine

applications at the most granular level. For more information about DPI, refer to AT&T Vyatta Network Operating System Policy-based Routing Configuration Guide.

Configuration mode

```
policy {
       qos {
               type policy-name {
                       shaper {
                               class class-id {
                                       match match-name {
                                               application {
                                                       type application-type
                                               }
                                       }
                               }
                       }
               }
       }
}
```

Use the `set` form of this command to match applications by type.

Use the `delete` form of this command to delete applications by type.

Use the `show` form of this command to display applications by type.

The following table lists related commands that are documented elsewhere.

Related commands documented elsewhere	
show application name <name>	Displays the application name and associated application types. (Refer to AT&T Vyatta Network Operating System Policy-based Routing Configuration Guide.)
show application type <type>	Displays the application names associated with the given application type. (Refer to AT&T Vyatta Network Operating System Policy-based Routing Configuration Guide.)

policy qos name <policy-name> shaper class <class-id> match <rule-name> description <description>

Describes a QoS policy class for ease of identification when viewing a configuration.

Syntax:

`set policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`description`** *description*

Syntax:

`delete policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`description`**

Syntax:

`show policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`description`**

`name` ***policy-name***

The name of a QoS policy.

`class` ***class-id***

The number of the QoS policy class. The number ranges from 1 through 255.

`match` ***rule-name***

The name of the class-matching rule—the rule that specifies the class that must be matched.

`description` ***description***

A description of the QoS queuing policy to use as a reference when viewing the configuration. If the description contains multiple words, they must be enclosed within single or double quotation marks. Text that includes carriage returns is not supported inside the quotation marks. There are no other restrictions on the use of text. Creating a description for an existing QoS policy replaces any existing description.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                class class-id {
                    match {
                        description description
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to add a description of a QoS policy class.

Use the `delete` form of this command to delete the description of a QoS policy class.

Use the `show` form of this command to display the description of a QoS policy class.

policy qos name <policy-name> shaper class <class-id> match <rule-name> destination <destination>

Defines a destination address, MAC address or port for a QoS policy class rule.

Syntax:
`set policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`destination`** { **`address`** *address* | **`mac-address`** *address* | **`port`** *port* }

Syntax:
`delete policy qos name` *policy-name* **`shaper class`** *class-id* **`destination`** [**`address`** | **`mac-address`** | **`port`**]

Syntax:
`show policy qos name` *policy-name* **`shaper class`** *class-id* **`destination`**

`name` ***policy-name***

Name of a QoS policy.

`class` ***class-id***

Specifies the numeric identifier of a QoS policy class. The numeric identifier ranges from 1 through 255.

`match` ***rule-name***

Specifies the name of a class-matching rule.

`address` ***address***

Specifies an address to match. Address formats are as follows:

address-group name: An address group that is configured with a list of addresses.

ip-address: An IPv4 address.

ip-address/prefix: An IPv4 network address, where 0.0.0.0/0 matches any network.

! *ip-address*: All IP addresses except the specified IPv4 address.

! *ip-address/prefix*: All IP addresses except the specified IPv4 network address.

ipv6-address: An IPv6 address; for example, fe80::20c:29fe:fe47:f89.

ip-address/prefix: An IPv6 network address, where ::/0 matches any network; for example, fe80::20c:29fe:fe47:f88/64.

! *ipv6-address*: All IP addresses except the specified IPv6 address.

! *ip-address/prefix*: All IP addresses except the specified IPv6 network address.

When both an address and a port are specified, then a packet is considered a match only when both the address and the port match.

`mac-address` ***address***

Specifies a media access control (MAC) address to match. The address format is six 8-bit numbers, separated by colons, in hexadecimal; for example, 00:0a:59:9a:f2:ba.

`port` ***port***

Specifies a port to match. Port formats are as follows:

- *port-group name*: A port group that is configured with a list of ports.
- *port name*: A port name as shown in `/etc/services`, for example, http.
- *start-end*: A range of port numbers, for example, 1001-1005.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                class class-id {
                    match rule-name {
                        destination {
                           address address
                           mac-address address
                           port port
                        }
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to define a destination address, MAC address, or port as a match criterion for a QoS policy class rule.

Use the `delete` form of this command to delete the destination address, MAC address, or port as a match criterion for a QoS policy class rule.

Use the `show` form of this command to display the destination parameter configuration for a QoS policy class rule.

policy qos name <policy-name> shaper class <class-id> match <rule-name> disable

Disables a QoS policy class rule.

Syntax:

`set policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`disable`**

Syntax:

`delete policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`disable`**

The rule is enabled.

`name` ***policy-name***

Name of a QoS policy.

`class` ***class-id***

Specifies the numeric identifier of a QoS policy class. The numeric identifier ranges from 1 through 255.

`match` ***rule-name***

Specifies the name of a class-matching rule.

`disable`

Specifies disabling the rule.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                class class-id {
                    match rule-name {
                        disable
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to disable a rule for a QoS policy class.

Use the `delete` form of this command to re-enable a rule for a QoS policy class.

policy qos name <policy-name> shaper class <class-id> match <rule-name> dscp <value>

Defines a differentiated services code point (DSCP) value as a match criterion of a QoS policy class.

Syntax:

`set policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`dscp`** *value*

Syntax:

`delete policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`dscp`** [*value*]

Syntax:

`show policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name*

`name` ***policy-name***

The name of a QoS policy.

`class` ***class-id***

The number of the QoS policy class. The number ranges from 1 through 255.

`match` ***rule-name***

The name of the class-matching rule.

`dscp` ***value***

The DSCP value of a packet. The values can range from 0 through 63.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                class class-id {
                    match rule-name {
                        dscp value
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to define a specific DSCP value as a match criterion of a QoS policy class.

Use the `delete` form of this command to delete a specific DSCP value as a match criterion of a QoS policy class.

Use the `show` form of this command to display the match criteria of a QoS policy class.

policy qos name <policy-name> shaper class <class-id> match <rule-name> ethertype <type>

Defines an Ethernet type as a match criterion for a QoS policy class.

Syntax:
`set policy qos name` *policy-name* **`shaper`** **`class`** *class-id* **`match`** *rule-name* **`ethertype`** *type*

Syntax:
`delete policy qos name` *policy-name* **`shaper`** **`class`** *class-id* **`match`** *rule-name* **`ethertype`**

Syntax:
`show policy qos name` *policy-name* **`shaper`** **`class`** *class-id* **`match`** *rule-name* **`ethertype`**

`name` ***policy-name***

Name of a QoS policy.

`class` ***class-id***

Specifies the numeric identifier of a QoS policy class. The numeric identifier ranges from 1 through 255.

`match` ***rule-name***

Specifies the name of a class-matching rule.

`ethertype` ***type***

Specifies the Ethernet type to match on. You can specify any Ethernet name listed in `/etc/ethertypes`, for example, IPv4. The Ethernet type can be specified by using the name format, hexadecimal format, or decimal format.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                class class-id {
                    match rule-name {
                        ethertype ethertype
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to define an Ethernet type as a match criterion for a QoS policy class rule.

Use the `delete` form of this command to delete an Ethernet type that is configured as a match criterion for a QoS policy class rule.

Use the `show` form of this command to display an Ethernet type that is configured as a match criterion for a QoS policy class rule.

policy qos name <policy-name> shaper class <class-id> match <rule-name> fragment

Define fragmented packets as the match criteria of a QoS policy class.

Syntax:
`set policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`fragment`**

Syntax:
`delete policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`fragment`**

Syntax:
`show policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`fragment`**

`name` ***policy-name***
The name of a QoS policy.

`class` ***class-id***
The number of the QoS policy class. The number ranges from 1 through 255.

`match` ***rule-name***
The name of the class-matching rule—the rule that specifies the class that must be matched.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                class class-id {
                    match rule-name
                    fragment
                }
            }
        }
    }
}
```

Use the `set` form of this command to define fragmented packets as the match criteria of a class.

Use the `delete` form of this command to delete fragmented packets as the match criteria of a class.

Use the `show` form of this command to display the match criteria of a class.

policy qos name <policy-name> shaper class <class-id> match <rule-name> icmp <icmp>

Defines an IPv4 ICMP type number, code number, name, or group as a match criterion for a QoS policy class rule.

Syntax:
`set policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`icmp`** { **`type`** *number* [**`code`** *number*] | **`name`** *name* | **`group`** *group* }

Syntax:
`delete policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`icmp`** [**`type`** [*number* **`code`**] | **`name`** | **`group`**]

Syntax:

show policy qos name *policy-name* **shaper** **class** *class-id* **match** *rule-name* **icmp** [**type** [*number* **code**] | **name** | **group**]

name ***policy-name***

Name of a QoS policy.

class ***class-id***

Specifies the numeric identifier of a QoS policy class. The numeric identifier ranges from 1 through 255.

match ***rule-name***

Specifies the name of a class-matching rule.

type ***number***

Specifies the numeric identifier of an ICMP type. The numeric identifier ranges from 0 through 255.

code ***number***

Specifies the numeric identifier of an ICMP code. The numeric identifier ranges from 0 through 255.

name ***name***

Specifies the name of an ICMP type.

group ***group***

Specifies an IPv4 ICMP group.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                class class-id {
                    match rule-name {
                        icmp {
                            type number {
                                code number
                            }
                            name name
                            group group
                        }
                    }
                }
            }
        }
    }
}
```

You can specify an ICMP type code by type; for example, 128 (echo-request), or by a type and code pair; for example, type 1 and code 4 (port-unreachable). Alternatively, you can specify the ICMP type code explicitly by using the **name** *name* parameter; for example, name echo-request.

For a list of ICMP types and codes, refer to ICMP Types *(page 76)*.

Use the set form of this command to define an IPv4 ICMP type number, code number, name, or group as a match criterion for a QoS policy class rule.

Use the delete form of this command to delete an IPv4 ICMP match criterion for a QoS policy class rule.

Use the show form of this command to display the IPv4 ICMP match criterion for a QoS policy class rule.

policy qos name <policy-name> shaper class <class-id> match <rule-name> icmpv6 <icmpv6>

Defines an IPv6 ICMP type number, code number, name, or group as a match criterion for a QoS policy class rule.

Syntax:

set policy qos name *policy-name* **shaper** **class** *class-id* **match** *rule-name* **icmpv6** { **type** *number* [**code** *number*] | **name** *name* | **group** *group* }

Syntax:
delete policy qos name *policy-name* **shaper class** *class-id* **match** *rule-name* **icmpv6** [**type** [*number* **code**] | **name** | **group**]

Syntax:
show policy qos name *policy-name* **shaper class** *class-id* **match** *rule-name* **icmpv6** [**type** [*number* **code**] | **name** | **group**]

name ***policy-name***

Name of a QoS policy.

class ***class-id***

Specifies the numeric identifier of a QoS policy class. The numeric identifier ranges from 1 through 255.

match ***rule-name***

Specifies the name of a class-matching rule.

type ***number***

Specifies the numeric identifier of an ICMPv6 type. The numeric identifier ranges from 0 through 255.

code ***number***

Specifies the numeric identifier of an ICMPv6 code. The numeric identifier ranges from 0 through 255.

name ***name***

Specifies the name of an ICMPv6 type.

group ***group***

Specifies an IPv6 ICMP group.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                class class-id {
                    match rule-name {
                        icmpv6 {
                            type number {
                                code number
                            }
                            name name
                            group group
                        }
                    }
                }
            }
        }
    }
}
```

You can specify an ICMPv6 type code by type; for example, 128 (echo-request), or by a type and code pair; for example, type 1 and code 4 (port-unreachable). Alternatively, you can specify the ICMP type code explicitly by using the **name** *name* parameter; for example, `name echo-request`.

For a list of ICMPv6 types and codes, refer to ICMPv6 Types *(page 79)*.

Use the `set` form of this command to define an IPv6 ICMP type number, code number, name, or group as a match criterion for a QoS policy class rule.

Use the `delete` form of this command to delete an IPv6 ICMP match criterion for a QoS policy class rule.

Use the `show` form of this command to display the IPv6 ICMP match criterion for a QoS policy class rule.

policy qos name <policy-name> shaper class <class-id> match <rule-name> ipv6-route type <number>

Defines an IPv6 source-routing header as a match criterion for a QoS policy class rule.

Syntax:
`set policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`ipv6-route type`** *number*

Syntax:
`delete policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`ipv6-route type`**

Syntax:
`show policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`ipv6-route type`**

`name` ***policy-name***

Name of a QoS policy.

`class` ***class-id***
Specifies the numeric identifier of a QoS policy class. The numeric identifier ranges from 1 through 255.

`match` ***rule-name***
Specifies the name of a class-matching rule.

`type` ***number***
Specifies the numeric identifier of an IPv6 route type. The numeric identifier ranges from 0 through 255.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                class class-id {
                    match rule-name {
                        ipv6-route {
                            type number
                        }
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to define an IPv6 route type as a match criterion for a QoS policy class rule.

Use the `delete` form of this command to delete the IPv6 route type configured as a match criterion for a QoS policy class rule.

Use the `show` form of this command to display the IPv6 route type configured as a match criterion for a QoS policy class rule.

policy qos name <policy-name> shaper class <class-id> match <rule-name> log

Enables logging for a match rule of a QoS policy class.

Syntax:
`set policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`log`**

Syntax:

delete policy qos name *policy-name* **shaper** **class** *class-id* **match** *rule-name* **log**

Syntax:
show policy qos name *policy-name* **shaper** **class** *class-id* **match** *rule-name*

name ***policy-name***
The name of a QoS policy.
class ***class-id***
The number of the QoS policy class. The number ranges from 1 through 255.
match ***rule-name***
The name of the class-matching rule—the rule that specifies the class that must be matched.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                class class-id {
                    match rule-name {
                        log
                    }
                }
            }
        }
    }
}
```

Use the set form of this command to enable logging for a QoS rule. This command causes debug messages similar to the following ICMP log message to be written into /var/log/dataplane/vplane.log.

Use the delete form of this command to disable logging for a QoS rule.

Use the show form of this command to display the match criteria of a class.

policy qos name <policy-name> shaper class <class-id> match <rule-name> mark

Remarks the PCP or DSCP value of packets that match a previously defined match rule for this class.

Syntax:
set policy qos name *policy-name* **shaper** **class** *class-id* **match** *rule-name* **mark** { **dscp** *value* | **pcp** *value* }

Syntax:
delete policy qos name *policy-name* **shaper** **class** *class-id* **match** *rule-name* **mark** { **dscp** [*value*] | **pcp** [*value*] }

Syntax:
show policy qos name *policy-name* **shaper** **class** *class-id* **match** *rule-name*

name ***policy-name***
The name of a QoS policy.
class ***class-id***
The number of the QoS policy class. The number ranges from 1 through 255.
match ***rule-name***
The name of the class-matching rule—the rule that specifies the class that must be matched.
dscp ***value***
The DSCP value of a packet. the numbers can range from 0 through 63.
pcp ***value***

The PCP value that ranges from 0 through 7. PCP matches packets with headers containing the 802.1 priority code point.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                class class-id {
                    match name {
                    mark {
                      dscp value
                      pcp value }
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to define a specific PCP or DSCP value as a remark value of a QoS policy class.

Use the **`delete`** form of this command to delete specific PCP or DSCP value as a remark value of a QoS policy class.

Use the `show` form of this command to display the remark value.

policy qos name <policy-name> shaper class <class-id> match <rule-name> pcp <number>

Defines a priority code point (PCP) number as a match criterion of a QoS policy class.

Syntax:
`set policy qos name` *policy-name* **`shaper`** **`class`** *class-id* **`match`** *rule-name* **`pcp`** *number*

Syntax:
`delete policy qos name` *policy-name* **`shaper`** **`class`** *class-id* **`match`** *rule-name* **`pcp`** [*number*]

Syntax:
`show policy qos name` *policy-name* **`shaper`** **`class`** *class-id* **`match`** *rule-name*

`name` ***policy-name***
The name of a QoS policy.

`class` ***class-id***
The number of the QoS policy class. The number ranges from 1 through 255.

`match` ***rule-name***
The name of the class-matching rule—the rule that specifies the class that must be matched.

`pcp` ***number***
The PCP number that ranges from 0 through 7. PCP matches packets with headers containing the 802.1 priority code point.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                class class-id {
                    match name {
                        pcp number
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to define a specific PCP value as a match criterion of a QoS policy class.

Use the `delete` form of this command to delete PCP as a match criterion of a QoS policy class.

Use the `show` form of this command to display the match criteria of a class.

policy qos name <policy-name> shaper class <class-id> match <rule-name> police bandwidth <limit>

Defines the policing rule for bandwidth for a QoS policy class.

Syntax:
`set policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`police bandwidth`** { *rate* | *rate-and-suffix* }

Syntax:
`delete policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`police bandwidth`** [*rate* | *rate-and-suffix*]

Syntax:
`show policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`police bandwidth`**

`name` ***policy-name***
The name of a QoS policy.

`class` ***class-id***
The number of the QoS policy class. The number ranges from 1 through 255.

`match` ***rule-name***
The name of the class-matching rule—the rule that specifies the class that must be matched.

`bandwidth` ***limit***
The maximum bandwidth. You can use the following syntax to specify the rate:

- **`bandwidth`** *number* **`%`**
  You can specify the bandwidth as a percentage. Use a number from 0 through 100.
- **`bandwidth`** *number* [*suffix*]
  You can use a suffix to specify the rate. The following are the supported suffix values:
  - [*unit*]**`bit`**
    Use this suffix format to specify the rate in bits per second.
  - [*unit*]**`bps`**
    Use this suffix format to specify the rate in bytes per second.

  To specify the rate as a decimal value, you can replace the *unit* parameter with one of the following keywords:

  - **`K`** (Kilo)
  - **`M`** (Mega)
  - **`G`** (Giga)

  To specify the rate as a binary value, you can replace the *unit* parameter with one of the following keywords:

  - **`Ki`** (Kilo)
  - **`Mi`** (Mega)
  - **`Gi`** (Giga)

For example, **bandwidth 1Kbit** means 1000 bits per second, **bandwidth 3Gbps** means 3 gigabytes per second, and **101Mibit** means 105906176 bits per second.

If a suffix is not specified, the default is Kbit. For example, **bandwidth 5** means 5000 bits per second.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                class class-id {
                    match rule-name {
                        police {
                            bandwidth number
                            bandwidth number-and-suffix
                            bandwidth number%
                        }
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to define the bandwidth policing rule of a QoS policy class.

Use the `delete` form of this command to delete the bandwidth policing rule of a QoS policy class.

Use the `show` form of this command to display the bandwidth policing rule of a QoS policy class.

policy qos name <policy-name> shaper class <class-id> match <rule-name> police burst <limit>

Defines the policing rule for traffic burst size limit for a match rule of a QoS policy class.

Syntax:
`set policy qos name` *policy-name* **shaper class** *class-id* **match** *rule-name* **police burst** *limit*

Syntax:
`delete policy qos name` *policy-name* **shaper class** *class-id* **match** *rule-name* **police** [**burst** *limit*]

Syntax:
`show policy qos name` *policy-name* **shaper class** *class-id* **match** *rule-name* **police**

`name` ***policy-name***
The name of a QoS policy.

`class` ***class-id***
The number of the QoS policy class. The number ranges from 1 through 255.

`match` ***rule-name***
The name of the class-matching rule—the rule that specifies the class that must be matched.

`burst` ***limit***
The burst size limit in number of bytes. The number can range from 0 through 312500000.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                class class-id {
                    match rule-name {
                        police {
                            burst limit
                        }
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to define the burst size limit policing rule for a match rule of a QoS policy class.

Use the `delete` form of this command to delete the burst size limit policing rule for a match rule of a QoS policy class.

Use the `show` form of this command to display the burst size limit policing rule for a match rule of a QoS policy class.

policy qos name <policy-name> shaper class <class-id> match <rule-name> police ratelimit <limit>

Defines the rate limit in packets per second for a match rule of a QoS policy class.

Syntax:

`set policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`police ratelimit`** *limit*

Syntax:

`delete policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`police ratelimit`**

Syntax:

`show policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`police ratelimit`**

`name` ***policy-name***

The name of a QoS policy.

`class` ***class-id***

The number of the QoS policy class. The number ranges from 1 through 255.

`match` ***rule-name***

The name of the class-matching rule—the rule that specifies the class that must be matched.

`ratelimit` ***limit***

The number of packets that can be sent in a second.

*n***`kpps`**: Thousands of packets per second.

*n***`mpps`**: Millions packets per second.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                class class-id {
                    match rule-name {
                        police {
                            ratelimit limit
                        }
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to define the rate limit in packets per second for a match rule of a QoS policy class.

Use the `delete` form of this command to delete the rate limit in packets per second for a match rule of a QoS policy class.

Use the `show` form of this command to display the rate limit in packets per second for a match rule of a QoS policy class.

policy qos name <policy-name> shaper class <class-id> match <rule-name> police then action <action>

Defines drop action on packets for a match rule of a QoS policy class when traffic exceeds policed bandwidth.

Syntax:
`set policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`police then action drop`**

Syntax:
`delete policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`police then action`**

Syntax:
`show policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`police`**

`name` ***policy-name***
The name of a QoS policy.

`class` ***class-id***
The number of the QoS policy class. The number ranges from 1 through 255.

`match` ***rule-name***
The name of the class-matching rule.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                class class-id {
                    match rule-name {
                       police  {
                           then
                           action drop
                       }
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to define the drop configuration on packets for a QoS policy class when traffic for the match rule exceeds policed bandwidth. This command applies only when the traffic rate has been exceeded within a policing period.

Use the `delete` form of this command to delete the drop configuration on packets for a match rule of a QoS policy class when traffic for the match rule exceeds policed bandwidth.

Use the `show` form of this command to display the drop policing rule of a match rule of a QoS policy class.

policy qos name <policy-name> shaper class <class-id> match <rule-name> police then mark <type>

Defines the policing rule for DSCP or PCP marking of packets when traffic exceeds policed bandwidth for a match rule of a QoS policy class.

Syntax:
`set policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`police then mark`** { **`dscp`** *dscp-value* | **`pcp`** *pcp-value* }

Syntax:
`delete policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`police then mark`** [**`dscp`** | **`pcp`**]

Syntax:
`show policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`police then mark`**

`name` ***policy-name***
The name of a QoS policy.

`class` ***class-id***
The number of the QoS policy class. The number ranges from 1 through 255.

`match` ***rule-name***
The name of the class-matching rule.

`pcp` ***pcp-value***
The PCP value. The value ranges from 0 to 7.

`dscp` ***dscp-value***
The DSCP value. The value ranges from 0 to 63, or is one of the standard DSCP tags. See the table in Mapping *(page 11)*.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                class class-id {
                    match rule-name {
                        police {
                            then
                            mark dscp
                            mark pcp
                        }
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to define the policing rule for DSCP or PCP marking of packets when traffic exceeds policed bandwidth, for a match rule of a QoS policy class.

Use the `delete` form of this command to delete the policing rule for DSCP or PCP marking of packets when traffic exceeds policed bandwidth, for a match rule of a QoS policy class.

Use the `show` form of this command to display the policing rule for DSCP or PCP marking of packets when traffic exceeds policed bandwidth, for a match rule of a QoS policy class.

policy qos name <policy-name> shaper class <class-id> match <rule-name> protocol <protocol>

Defines a protocol type for a rule to match if the protocol is present in the packet.

Syntax:
`set policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`protocol`** { *text* | *0-255* | **`all`** }

Syntax:
`delete policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`protocol`** [*text* | *0-255* | **`all`**]

Syntax:
`show policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`protocol`**

`name` ***policy-name***
The name of a QoS policy.

`class` ***class-id***
The number of the QoS policy class. The number ranges from 1 through 255.

`match` ***rule-name***
The name of the class-matching rule—the rule that specifies the class that must be matched.

`protocol` { *text* | *0-255* | `all` }
The name of an IP protocol.

text: IP protocol name from `/etc/protocols`, for example, **`tcp`** or **`udp`**.

0-255: The IP protocol number located in the IP header.

`all`: All IP protocols

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                class class-id {
                    match rule-name {
                        protocol protocol
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to define a matching protocol of a match rule of a QoS policy class.

Use the `delete` form of this command to delete a matching protocol of a match rule of a QoS policy class.

Use the `show` form of this command to display a matching protocol of a match rule of a QoS policy class.

policy qos name <policy-name> shaper class <class-id> match <rule-name> source <source>

Defines a source address, MAC address or port for a match rule of a QoS policy class.

Syntax:
`set policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`source`** { **`address`** *address* | **`mac-address`** *address* | **`port`** *port* }

Syntax:
delete policy qos name *policy-name* **shaper class** *class-id* **source** [**address** | **mac-address** | **port**]

Syntax:
show policy qos name *policy-name* **shaper class** *class-id* **source**

policy-name

Name of a QoS policy.

class ***class-id***

Specifies the numeric identifier of a QoS policy class. The numeric identifier ranges from 1 through 255.

match ***rule-name***

Specifies the name of a class-matching rule.

address ***address***

Specifies an address to match. Address formats are as follows:

address-group name: An address group that is configured with a list of addresses.

ip-address: An IPv4 address.

ip-address/prefix: An IPv4 network address, where 0.0.0.0/0 matches any network.

! *ip-address*: All IP addresses except the specified IPv4 address.

! *ip-address/prefix*: All IP addresses except the specified IPv4 network address.

ipv6-address: An IPv6 address; for example, fe80::20c:29fe:fe47:f89.

ip-address/prefix: An IPv6 network address, where ::/0 matches any network; for example, fe80::20c:29fe:fe47:f88/64.

! *ipv6-address*: All IP addresses except the specified IPv6 address.

! *ip-address/prefix*: All IP addresses except the specified IPv6 network address.

When both an address and a port are specified, then a packet is only considered a match when both the address and the port match.

mac-address ***address***

Specifies a media access control (MAC) address to match. The address format is six 8-bit numbers, separated by colons, in hexadecimal; for example, 00:0a:59:9a:f2:ba.

port ***port***

Specifies a port to match. Port formats are as follows:

- *port-group name*: A port group that is configured with a list of ports.
- *port name*: A port name as shown in `/etc/services`, for example, http.
- *start-end*: A range of port numbers, for example, 1001-1005.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                class class-id {
                    match rule-name {
                        source {
                            address address
                            mac-address address
                            port port
                        }
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to define a matching source address, MAC address, or port for a match rule of a QoS policy class.

Use the `delete` form of this command to delete the matching source address, MAC address, or port for a match rule of a QoS policy class.

Use the `show` form of this command to display the matching source parameter configuration for a match rule of a QoS policy class.

policy qos name <policy-name> shaper class <class-id> match <rule-name> tcp flags <flags>

Defines the flags in the TCP header as match rule criteria for a QoS policy class.

Syntax:
`set policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`tcp flags`** *flags*

Syntax:
`delete policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`tcp flags`** *flags*

Syntax:
`show policy qos name` *policy-name* **`shaper class`** *class-id* **`match`** *rule-name* **`tcp flags`**

`name` ***policy-name***
The name of a QoS policy.

`class` ***class-id***
The ID number of the QoS policy class. The number ranges from 1 through 999999.

`match` ***rule-name***
The name of the class-matching rule—the rule that specifies the class that must be matched.

`flags` ***flags***
The TCP flags to apply. Allowed values: SYN ACK FIN RST URG PSH. Use commas to separate multiple values.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                class class-id {
                    match rule-name {
                        tcp flags flags
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to define the flags in the TCP header as match rule criteria for a QoS policy class.

Use the `delete` form of this command to delete the flags in the TCP header as match rule criteria for a QoS policy class.

Use the `show` form of this command to display the flags for a QoS policy.

policy qos name <policy-name> shaper class <class-id> profile <profile-name>

Associates a QoS profile that belongs to a QoS policy to a QoS policy class.

Syntax:
`set policy qos name` *policy-name* **`shaper class`** *class-id* **`profile`** *profile-name*

Syntax:
`delete policy qos name` *policy-name* **`shaper class`** *class-id* **`profile`** [*profile-name*]

Syntax:
`show policy qos name` *policy-name* **`shaper class`** *class-id* **`profile`**

`name` ***policy-name***
The name of a QoS policy.

`class` ***class-id***
The ID number of the QoS policy class. The number ranges from 1 through 255.

`profile` ***profile-name***
The name of a QoS profile.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                class class-id {
                    profile profile-name
                }
             }
        }
    }
}
```

Use the `set` form of this command to create an association between the class and a profile name of a QoS policy class.

Use the `delete` form of this command to delete an association between the class and a profile name of a QoS policy class.

Use the `show` form of this command to display an association between the class and a profile name of a QoS policy class.

policy qos name <policy-name> shaper default <default-name>

Defines a QoS traffic-queuing profile to apply to traffic that does not match any defined classes.

Syntax:
`set policy qos name` *policy-name* **`shaper default`** *default-name*

Syntax:
`delete policy qos name` *policy-name* **`shaper default`** [*default-name*]

Syntax:
`show policy qos name` *policy-name* **`shaper default`** *default-name*

`name` ***policy-name***
: The name of a QoS policy.

`default` ***default-name***
: The name of a QoS profile to apply to default traffic. This attribute is required and must be configured.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                default default-name {
                }
            }
        }
    }
}
```

Use the `set` form of this command to define the QoS profile to apply to default traffic.

Use the `delete` form of this command to delete a QoS profile for default traffic.

Use the `show` form of this command to display a QoS profile for default traffic.

policy qos name <policy-name> shaper description <description>

Describes a QoS policy.

Syntax:
`set policy qos name` *policy-name* **`shaper description`** *description*

Syntax:
`delete policy qos name` *policy-name* **`description`**

Syntax:
`show policy qos name` *policy-name* **`description`**

`name` ***policy-name***
: The name of a QoS policy.

`description` ***description***
: A description of the QoS queuing policy to use as a reference when viewing the configuration. If the description contains multiple words, they must be enclosed within single or double quotation marks. Text that includes carriage returns is not supported inside the quotation marks. There are no other restrictions on the use of text. Creating a description for an existing QoS policy replaces any existing description.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                description description {
                }
            }
        }
    }
}
```

Use the `set` form of this command to describe a QoS queuing policy.

Use the `delete` form of this command to delete the description of a QoS policy.

Use the `show` form of this command to display the description of a QoS policy.

policy qos name <policy-name> shaper frame-overhead <bytes>

Enables the frame overhead scheduler which takes into account the additional bytes added by the underlying link layer protocols.

Syntax:
`set policy qos name` *policy-name* **`shaper frame-overhead`** *bytes*

Syntax:
`delete policy qos name` *policy-name* **`shaper frame-overhead`** [*bytes*]

Syntax:
`show policy qos name` *policy-name* **`shaper frame-overhead`**

The Ethernet frame overhead of 22 bytes.

`name` ***policy-name***
The name of a QoS policy.

`frame-overhead` ***bytes***
The Ethernet frame overhead in bytes. The number range is 0 through 1000.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                frame-overhead bytes {
                }
            }
        }
    }
}
```

Use the `set` form of this command to enable the frame overhead scheduler which takes into account the additional bytes added by the underlying link layer protocols.

Use the `delete` form of this command to delete the configuration for the frame overhead scheduler.

Use the `show` form of this command to display the configuration for the frame overhead scheduler.

policy qos name <policy-name> shaper profile <profile-name>

Creates a QoS policy profile.

Syntax:
`set policy qos name` *policy-name* **`shaper profile`** *profile-name*

Syntax:
`delete policy qos name` *policy-name* **`shaper profile`** [*profile-name*]

Syntax:

show policy qos name *policy-name* **shaper profile**

name ***policy-name***
The name of a QoS policy.

profile ***profile-name***
The name of a QoS profile.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                profile profile-name {
                }
            }
        }
    }
}
```

After a profile has been created, use other QoS commands to configure attributes for bandwidth, burst, class, default, description, map, queue, and size.

Use the `set` form of this command to create a QoS profile.

Use the `delete` form of this command to delete a QoS profile.

Use the `show` form of this command to display a QoS profile.

policy qos name <policy-name> shaper profile <profile-name> bandwidth <limit>

Defines the maximum bandwidth of a QoS traffic-queuing profile.

Syntax:
set policy qos name *policy-name* **shaper profile** *profile-name* **bandwidth** { *number* | *number-and-suffix* }

Syntax:
delete policy qos name *policy-name* **shaper profile** *profile-name* **bandwidth** [*number* | *number-and-suffix*]

Syntax:
show policy qos name *policy-name* **shaper profile** *profile-name* **bandwidth**

name ***policy-name***
The name of a QoS policy.

profile ***profile-name***
The name of a QoS profile.

bandwidth ***limit***
The bandwidth rate as a number followed by no space and a scaling suffix representing the rate (for example, 10mbit). The following suffixes are supported:

No suffix: Kilobits per second.

mbit: Megabits per second.

mbps: Megabytes per second.

gbit: Gigabits per second.

kbps: Kilobytes per second.

gbps: Gigabytes per second.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                profile profile-name {
                    bandwidth number
                    bandwidth number-and-suffix
                }
            }
        }
    }
}
```

Use the `set` form of this command to define the bandwidth of a QoS profile.

Use the `delete` form of this command to delete the bandwidth of a QoS profile.

Use the `show` form of this command to display the bandwidth of a QoS profile.

policy qos name <policy-name> shaper profile <profile-name> burst <limit>

Defines the maximum burst for a QoS profile.

Syntax:

`set policy qos name` *policy-name* **`shaper profile`** *profile-name* **`burst`** *limit*

Syntax:

`delete policy qos name` *policy-name* **`shaper profile`** *profile-name* **`burst`** *limit*

Syntax:

`show policy qos name` *policy-name* **`shaper profile`** *profile-name* **`burst`**

`name` ***policy-name***

The name of a QoS policy.

`profile` ***profile-name***

The name of a QoS profile.

`burst` ***limit***

The burst size limit in number of bytes. The number can range from 0 through 312500000.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                profile profile-name {
                    burst limit
                }
            }
        }
    }
}
```

Use the `set` form of this command to define the burst size limit of a QoS policy.

Use the `delete` form of this command to delete the burst size limit of a QoS policy.

Use the `show` form of this command to display the burst size limit of a QoS policy.

policy qos name <policy-name> shaper profile <profile-name> description

Describes a QoS profile.

Syntax:
`set policy qos name` *policy-name* **`shaper profile`** *profile-name* **`description`** *description*

Syntax:
`delete policy qos name` *policy-name* **`shaper profile`** *profile-name* **`description`**

Syntax:
`show policy qos name` *policy-name* **`shaper profile`** *profile-name* **`description`**

`name` ***policy-name***
    The name of a QoS policy.

`profile` ***profile-name***
    The name of a QoS profile.

`description` ***description***
    A description of the QoS profile to use as a reference when viewing the configuration. If the description contains multiple words, they must be enclosed within single or double quotation marks. Text that includes carriage returns is not supported inside the quotation marks. There are no other restrictions on the use of text. Creating a description for an existing QoS profile replaces any existing description.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                profile profile-name {
                    description description
                }
            }
        }
    }
}
```

The text entered as the description must be kept in quotation marks. The description must be kept to a single line; this command does not support carriage returns, otherwise there are no restrictions of the use of text.

Use the `set` form of this command to create the description of a QoS profile.

Use the `delete` form of this command to delete the description of a QoS profile.

Use the `show` form of this command to display the description of a QoS profile.

policy qos name <policy-name> shaper profile <profile-name> map dscp <value> to <queue-id>

Overrides the default DSCP to queue mapping for a QoS profile.

Syntax:
`set policy qos name` *policy-name* **`shaper profile`** *profile-name* **`map dscp`** *value* **`to`** *queue-id*

Syntax:
`delete policy qos name` *policy-name* **`shaper profile`** *profile-name* **`map dscp`** *value* **`to`** [*queue-id*]

Syntax:

`show policy qos name` *policy-name* **`shaper profile`** *profile-name* **`map dscp`**

`name` ***policy-name***
The name of a QoS policy.

`profile` ***profile-name***
The name of the QoS profile.

`dscp` ***value***
Specifies the DSCP value as the match criteria. The supported values are AF11 through AF13, AF21 through AF23, AF31 through AF33, AF41 through AF43, CS1 through CS7, default, and EF. See Mapping. *(page 11)*

`to` ***queue-id***
Specifies the number of the destination queue. The queue number ranges from 0 through 15.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                profile profile-name {
                    map {
                        dscp value {
                            to queue-id
                        }
                    }
                }
          }
        }
    }
}
```

Use the `set` form of this command to override the default DSCP to queue mapping for a QoS profile.

Use the `delete` form of this command to delete the mapping of traffic with a specific DSCP value to a specific queue of a QoS profile.

Use the `show` form of this command to display mapping of traffic with a specific DSCP value to a specific queue of a QoS profile.

policy qos name <policy-name> shaper profile <profile-name> map pcp <value> to <queue-id>

Defines the mapping of priority code point (PCP) traffic to a queue for a QoS profile.

Syntax:
`set policy qos name` *policy-name* **`shaper profile`** *profile-name* **`map pcp`** *value* **`to`** *queue-id*

Syntax:
`delete policy qos name` *policy-name* **`shaper profile`** *profile-name* **`map pcp`** *value* **`to`** [*queue-id*]

Syntax:
`show policy qos name` *policy-name* **`shaper profile`** *profile-name* **`map pcp`**

`name` ***policy-name***
The name of a QoS policy.

`profile` ***profile-name***
The name of a QoS profile.

`pcp` ***value***
The PCP value that ranges from 0 through 7. PCP matches packets with headers containing the 802.1 priority code point.

`to` ***queue-id***

Specifies the number of the destination queue. The queue number ranges from 0 through 15.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                profile profile-name {
                    map {
                        pcp value {
                            to queue-id
                        }
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to map traffic with a specific PCP value to a specific queue of a QoS profile. Use PCP mapping only if the packet has a VLAN header and the profile that the packet is shaped by has the PCP map explicitly configured with at least one instance of this command.

Use the `delete` form of this command to delete the mapping of traffic with a specific PCP value to a specific queue of a QoS profile.

Use the `show` form of this command to display the mapping of traffic with a specific PCP value to a specific queue of a QoS profile.

policy qos name <policy-name> shaper profile <profile-name> period <number>

Defines the length of time that a burst is limited to for a QoS profile.

Syntax:

`set policy qos name` *policy-name* **`shaper`** **`profile`** *profile-name* **`period`** *number*

Syntax:

`delete policy qos name` *policy-name* **`shaper`** **`profile`** *profile-name* **`period`** [*number*]

Syntax:

`show policy qos name` *policy-name* **`shaper`** **`profile`** *profile-name* **`period`**

`name` ***policy-name***

The name of the QoS policy.

`profile` ***profile-name***

The name of the QoS profile.

`period` ***number***

The enforcement period in milliseconds. The numbers range from 1 through 3000.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                profile profile-name {
                    period number
                }
            }
        }
    }
}
```

Use the `set` form of this command to define the length of time that a burst is limited to for a QoS profile. For example, if maximum burst size is one MB and the period is 20 milliseconds, one MB of traffic can be sent every 20 milliseconds.

Use the `delete` form of this command to delete the enforcement interval period of the profile.

Use the `show` form of this command to display the enforcement interval period (in milliseconds) of the profile.

policy qos name <policy-name> shaper profile <profile-name> queue <queue-id>

Defines the queue ID on the output port of a packet for forwarding or scheduling, depending on how it is configured.

Syntax:

`set policy qos name` *policy-name* **`shaper`** **`profile`** *profile-name* **`queue`** *queue-id*

Syntax:

`delete policy qos name` *policy-name* **`shaper`** **`profile`** *profile-name* **`queue`** [*queue-id*]

Syntax:

`show policy qos name` *policy-name* **`shaper`** **`profile`** *profile-name* **`queue`**

`name` ***policy-name***

The name of a QoS policy.

`profile` ***profile-name***

The name of a QoS profile.

`queue` ***queue-id***

The packet queue identifier. The numbers range from 0 through 15.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                profile profile-name {
                    queue queue-id
                }
            }
        }
    }
}
```

The queue ID is associated with a QoS policy profile. The queue can restrict traffic based on bandwidth and burst.

A total of 16 queues (0 through 15) can be configured for a policy.

Use the `set` form of this command to define the queue ID on the output port of a packet for forwarding or scheduling depending on what is configured.

Use the `delete` form of this command to delete the queue ID on the output port of a packet for forwarding or scheduling depending on what is configured.

Use the `show` form of this command to display the queue ID on the output port of a packet for forwarding or scheduling, depending on what is configured.

policy qos name <policy-name> shaper profile <profile-name> queue <queue-id> description <description>

Describes a QoS queue.

Syntax:
`set policy qos name` *policy-name* **`shaper profile`** *profile-name* **`queue`** *queue-id* **`description`** *description*

Syntax:
`delete policy qos name` *policy-name* **`shaper profile`** *profile-name* **`queue`** *queue-id* **`description`**

Syntax:
`show policy qos name` *policy-name* **`shaper profile`** *profile-name* **`queue`** *queue-id* **`description`**

`name` ***policy-name***
The name of a QoS policy.

`profile` ***profile-name***
The name of a QoS profile.

`queue` ***queue-id***
The packet queue identifier. The numbers range from 0 through 15.

`description` ***description***
A description of the QoS queue to use as a reference when viewing the configuration. If the description contains multiple words, they must be enclosed within single or double quotation marks. Text that includes carriage returns is not supported inside the quotation marks. There are no other restrictions on the use of text. Creating a description for an existing QoS queue replaces any existing description.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                profile profile-name {
                    queue queue-id {
                        description description
                    }
                 }
            }
        }
    }
}
```

Use the `set` form of this command to describe a QoS queue.

Use the `delete` form of this command to delete the description to a QoS queue.

Use the `show` form of this command to display the description to a QoS queue.

policy qos name <policy-name> shaper profile <profile-name> queue <queue-id> traffic-class <traffic-class>

Defines the traffic class ID of a queue for a QoS policy.

Syntax:
`set policy qos name` *policy-name* **`shaper profile`** *profile-name* **`queue`** *queue-id* **`traffic-class`** *traffic-class*

Syntax:

`delete policy qos name` *policy-name* **`shaper`** **`profile`** *profile-name* **`queue`** *queue-id* **`traffic-class`** [*traffic-class*]

Syntax:
`show policy qos name` *policy-name* **`shaper`** **`profile`** *profile-name* **`queue`** *queue-id* **`traffic-class`**

`name` ***policy-name***
The name of a QoS policy.

`profile` ***profile-name***
The name of a QoS profile.

`queue` ***queue-id***
The packet queue identifier. The numbers range from 0 through 15.

`traffic-class` ***traffic-class***
Class identification number. The numbers range from 0 through 3.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                profile profile-name {
                    queue queue-id {
                        traffic-class traffic-class
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to define the traffic class ID of a queue for a QoS policy.

Use the `delete` form of this command to delete the traffic class ID of a queue for a QoS policy.

Use the `show` form of this command to display the traffic class ID of a queue for a QoS policy.

policy qos name <policy-name> shaper profile <profile-name> queue <queue-id> weight <weight-number>

Defines the WRR weight number for a queue.

Syntax:
`set policy qos name` *policy-name* **`shaper`** **`profile`** *profile-name* **`queue`** *queue-id* **`weight`** *weight-number*

Syntax:
`delete policy qos name` *policy-name* **`shaper`** **`profile`** *profile-name* **`queue`** *queue-id* **`weight`** [*weight-number*]

Syntax:
`show policy qos name` *policy-name* **`shaper`** **`profile`** *profile-name* **`queue`** *queue-id* **`weight`**

`name` ***policy-name***
The name of a QoS policy.

`profile` ***profile-name***
The name of a QoS profile.

`queue` ***queue-id***
The packet queue identifier. The numbers range from 0 through 15.

`weight` ***weight-number***
WRR numerical number. The numbers range from 1 through 100.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                profile profile-name {
                   queue queue-id {
                      weight weight-number
                   }
               }
            }
        }
    }
}
```

Use the `set` form of this command to define the WRR weight number for a queue.

Use the `delete` form of this command to delete the WRR weight number for a queue.

Use the `show` form of this command to display the WRR weight number for a queue.

policy qos name <policy-name> shaper profile <profile-name> traffic-class <traffic-class> bandwidth <limit>

Defines the maximum bandwidth of a traffic class for a QoS profile.

Syntax:
`set policy qos name` *policy-name* **`shaper profile`** *profile-name* **`traffic-class`** *traffic-class* **`bandwidth`** { *number%* | *number* | *number-and-suffix* }

Syntax:
`delete policy qos name` *policy-name* **`shaper profile`** *profile-name* **`traffic-class`** *traffic-class* **`bandwidth`** { *number%* | *number* | *number-and-suffix* }

Syntax:
`show policy qos name` *policy-name* **`shaper profile`** *profile-name* **`traffic-class`** *traffic-class* **`bandwidth`**

`name` ***policy-name***
The name of a QoS policy.

`profile` ***profile-name***
The name of a QoS profile.

`traffic-class` ***traffic-class***
The ID of the traffic class. The number ranges from 0 through 3.

`bandwidth` ***limit***
The bandwidth rate as a percentage (1 through 100%) or a number followed by a scaling suffix representing the rate (<number><suffix>). Suffixes are either 'bit' for bits-per-second or 'bps' for bytes-per-second. These can be preceded by a decimal (K,M,G) or binary (Ki,Mi,Gi) multiplier. No suffix refers to Kbit (1000 bits per second).

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                profile profile-name {
                    traffic-class traffic-class {
                        bandwidth number
                        bandwidth number-and-suffix
                        bandwidth 100%
                    }
                }
            }
        }
    }
}
```

Use the `set` form of this command to define the bandwidth limit of a traffic class for a QoS profile.

Use the `delete` form of this command to delete the bandwidth limit of a traffic class for a QoS profile.

Use the `show` form of this command to display the bandwidth limit of a traffic class for a QoS profile.

policy qos name <policy-name> shaper profile <profile-name> traffic-class <traffic-class> description <description>

Describes a traffic class of a QoS profile.

Syntax:
`set policy qos name` *policy-name* **`shaper profile`** *profile-name* **`traffic-class`** *traffic-class* **`description`** *description*

Syntax:
`delete policy qos name` *policy-name* **`shaper profile`** *profile-name* **`traffic-class`** *traffic-class* **`description`**

Syntax:
`show policy qos name` *policy-name* **`shaper profile`** *profile-name* **`traffic-class`** *traffic-class* **`description`**

`name` ***policy-name***
The name of a QoS policy.

`profile` ***profile-name***
The name of a QoS profile.

`traffic-class` ***traffic-class***
The ID of the traffic class. The number ranges from 0 through 3.

`description` ***description***
The description of a traffic class as a reference notation when viewing the configuration. If the description contains multiple words, they must be enclosed within single or double quotation marks. Text that includes carriage returns is not supported inside the quotation marks. There are no other restrictions on the use of text.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                profile profile-name {
                    traffic-class traffic-class {
                        description description
                    }
                }
            }
        }
    }
}
```

The text entered as the description must be kept in quotation marks. The description must be kept to a single line; this command does not support carriage returns, otherwise there are no restrictions of the use of text.

Use the `set` form of this command to define the description of a traffic class of a QoS profile.

Use the `delete` form of this command to delete the description of a traffic class of a QoS profile.

Use the `show` form of this command to display the description of a traffic class of a QoS profile.

policy qos name <policy-name> shaper traffic-class <traffic-class> bandwidth <limit>

Defines the bandwidth rate of a QoS traffic class.

Syntax:
`set policy qos name` *policy-name* **`shaper traffic-class`** *traffic-class* **`bandwidth`** { *number%* | *number* | *number-and-suffix* }

Syntax:
`delete policy qos name` *policy-name* **`shaper traffic-class`** *traffic-class* **`bandwidth`** { *number%* | *number* | *number-and-suffix* }

Syntax:
`show policy qos name` *policy-name* **`shaper traffic-class`** *traffic-class* **`bandwidth`**

`name` ***policy-name***
The name of the QoS policy.

`traffic-class` ***traffic-class***
The ID of the traffic class. The number ranges from 0 through 3.

`bandwidth` ***limit***
The bandwidth rate as a percentage (1 through 100%) or a number followed by a scaling suffix representing the rate (<number><suffix>). Suffixes are either 'bit' for bits-per-second or 'bps' for bytes-per-second. These can be preceded by a decimal (K,M,G) or binary (Ki,Mi,Gi) multiplier. No suffix refers to Kbit (1000 bits per second).

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                traffic-class traffic-class {
                    bandwidth number%
                    bandwidth number
                    bandwidth number-and-suffix
                }
            }
        }
    }
}
```

Use the `set` form of this command to define the bandwidth of a QoS policy's traffic class.

Use the `delete` form of this command to delete the bandwidth of a QoS policy's traffic class.

Use the `show` form of this command to display the bandwidth of a QoS policy's traffic class.

policy qos name <policy-name> shaper traffic-class <traffic-class> description

Describes a traffic-class for ease of identification when viewing a configuration.

Syntax:

`set policy qos name` *policy-name* **`shaper`** **`traffic-class`** *traffic-class* **`description`** *description*

Syntax:
`delete policy qos name` *policy-name* **`shaper`** **`traffic-class`** *traffic-class* **`description`**

Syntax:
`show policy qos name` *policy-name* **`shaper`** **`traffic-class`** *traffic-class* **`description`**

`name` ***policy-name***
: The name of a QoS policy.

`traffic-class` ***traffic-class***
: The ID of the traffic class. The number ranges from 0 through 3.

`description` ***description***
: A description of the QoS traffic class to use as a reference when viewing the configuration. If the description contains multiple words, they must be enclosed within single or double quotation marks. Text that includes carriage returns is not supported inside the quotation marks. There are no other restrictions on the use of text. Creating a description for an existing QoS traffic class replaces any existing description.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                traffic-class traffic-class {
                    description description
                }
            }
        }
    }
}
```

Use the `set` form of this command to describe a traffic class for ease of identification when viewing a configuration.

Use the `delete` form of this command to delete the description of a traffic class.

Use the `show` form of this command to display the description of a traffic class.

policy qos name <policy-name> shaper traffic-class <traffic-class> queue-limit <number>

Defines the queue limit of a QoS traffic class.

Syntax:
`set policy qos name` *policy-name* **`shaper`** **`traffic-class`** *traffic-class* **`queue-limit`** *number*

Syntax:
`delete policy qos name` *policy-name* **`shaper`** **`traffic-class`** *traffic-class* **`queue-limit`** [*number*]

Syntax:
`show policy qos name` *policy-name* **`shaper`** **`traffic-class`** *traffic-class* **`queue-limit`**

`name` ***policy-name***
: The name of a QoS policy.

`traffic-class` ***traffic-class***
: The ID of the traffic class. The number ranges from 0 through 3.

`queue-limit` ***number***

The queue limit in number of packets. The number range from 1 through 8192 and must be a power of 2. To support queue limits, the policy must be a port-level policy.

> **Note:** Although you can enter a queue-limit number that ranges from 1 through 65535, if you enter a number that is greater than 8192, this command sets the queue limit to 8192.

Configuration mode

```
policy {
    qos {
        name policy-name {
            shaper {
                traffic-classs traffic-class {
                    queue-limit
                    number
                }
            }
        }
    }
}
```

Use the `set` form of this command to define the queue limit of a QoS traffic class.

Use the `delete` form of this command to deletes the queue limit of a QoS traffic class.

Use the `show` form of this command to display the queue limit of a QoS traffic class.

policy qos name <policy-name> shaper traffic-class <traffic-class> random-detect filter-weight <weight>

Defines the exponentially weighted moving average (EWMA) filter parameter for a QoS traffic class.

Syntax:
`set policy qos name` *policy-name* **`shaper`** **`traffic-class`** *traffic-class* **`random-detect`** **`filter-weight`** *weight*

Syntax:
`delete policy qos name` *policy-name* **`shaper`** **`traffic-class`** *traffic-class* **`random-detect`** **`filter-weight`** [*weight*]

Syntax:
`show policy qos name` *policy-name* **`shaper`** **`traffic-class`** *traffic-class* **`random-detect`** **`filter-weight`**

`name` ***policy-name***
The name of a QoS policy.

`traffic-class` ***traffic-class***
The ID of the traffic class. The number ranges from 0 through 3.

`filter-weight` ***weight***
The exponentially weighted moving average (EWMA) filter weight. The number ranges from 1 through 12.

Configuration mode

```
policy {
    qos {
        name policy-name {
            traffic-class traffic-class {
                random-detect {
                    filter-weight weight
                }
            }
        }
    }
}
```

Use the `set` form of this command to define the EWMA filter parameter for a QoS traffic class.

Use the `delete` form of this command to delete the EWMA filter parameter for a QoS traffic class.

Use the `show` form of this command to display the EWMA filter parameter for a QoS traffic class.

policy qos name <policy-name> shaper traffic-class <traffic-class> random-detect mark-probability <number>

Defines the packet marking probability (in an inverse) filter number for a QoS traffic class.

Syntax:
`set policy qos name` *policy-name* **`shaper traffic-class`** *traffic-class* **`random-detect mark-probability`** *mark-probability*

Syntax:
`delete policy qos name` *policy-name* **`shaper traffic-class`** *traffic-class* **`random-detect mark-probability`** [*mark-probability*]

Syntax:
`show policy qos name` *policy-name* **`shaper traffic-class`** *traffic-class* **`random-detect mark-probability`**

`name` ***policy-name***
The name of a QoS policy.

`traffic-class` ***traffic-class***
The ID of the traffic class. The number ranges from 0 through 3.

`mark-probability` ***mark-probability***
The maximum value for the inverse packet marking probability filter for a QoS traffic class—a rate of 1/*x* where *x* is the mark-probability number. The number ranges from 1 through 255.

Configuration mode

```
policy {
    qos {
        name policy-name {
            traffic-class traffic-class {
                random-detect {
                    mark-probability mark-probability
                }
            }
        }
    }
}
```

When the maximum queue depth is met, the system drops packets at a rate of 1/*x* where x is the mark-probability number.

Use the `set` form of this command to define the inverse of packet marking probability filter number for a QoS traffic class.

Use the `delete` form of this command to delete the inverse of packet marking probability filter number for a QoS traffic class.

Use the `show` form of this command to display the inverse of packet marking probability filter number for a QoS traffic class.

policy qos name <policy-name> shaper traffic-class <traffic-class> random-detect max-threshold <level>

Defines the maximum threshold level for a QoS traffic class.

Syntax:
set policy qos name *policy-name* **shaper traffic-class** *traffic-class* **random-detect max-threshold** *max-threshold*

Syntax:
delete policy qos name *policy-name* **shaper traffic-class** *traffic-class* **random-detect max-threshold** [*max-threshold*]

Syntax:
show policy qos name *policy-name* **shaper traffic-class** *traffic-class* **random-detect max-threshold**

name ***policy-name***
: The name of a QoS policy.

traffic-class ***traffic-class***
: The ID of the traffic class. The number ranges from 0 through 3.

max-threshold ***max-threshold***
: The maximum threshold level number. The number ranges from 1 through 1023.

Configuration mode

```
policy {
    qos {
        name policy-name {
            traffic-class traffic-class {
                random-detect {
                    max-threshold max-threshold
                }
            }
        }
    }
}
```

Use the `set` form of this command to define the maximum threshold number for a QoS traffic class.

Use the `delete` form of this command to delete the maximum threshold number for a QoS traffic class.

Use the `show` form of this command to display the maximum threshold number for a QoS traffic class.

policy qos name <policy-name> shaper traffic-class <traffic-class> random-detect min-threshold <level>

Defines the minimum threshold level for a QoS traffic class.

Syntax:
set policy qos name *policy-name* **shaper traffic-class** *traffic-class* **random-detect min-threshold** *min-threshold*

Syntax:
delete policy qos name *policy-name* **shaper traffic-class** *traffic-class* **random-detect min-threshold** [*min-threshold*]

Syntax:

`show policy qos name` *policy-name* **`shaper traffic-class`** *traffic-class* **`random-detect min-threshold`**

`name` ***policy-name***
: The name of a QoS policy.

`traffic-class` ***traffic-class***
: The ID of the traffic class. The number ranges from 0 through 3.

`min-threshold` ***min-threshold***
: The minimum threshold level number. The number ranges from 1 through 1022.

Configuration mode

```
policy {
    qos {
        name policy-name {
            traffic-class traffic-class {
                random-detect {
                    min-threshold min-threshold
                }
            }
        }
    }
}
```

Use the `set` form of this command to define the minimum threshold level for a QoS traffic class.

Use the `delete` form of this command to delete the minimum threshold level for a QoS traffic class.

Use the `show` form of this command to display the minimum threshold level for a QoS traffic class.

show queuing <dataplane-interface>

Displays outgoing packet actions.

Syntax:

`show queuing` [*dataplane-interface*]

dataplane-interface
: The type of dataplane interface whose QoS policies you want to display.

Operational mode

Use this command to display outgoing packet actions.

The following example shows all outgoing QoS policies.

```
vyatta@vyatta:~$ show queuing
Interface   Queue    Packets            Bytes    Dropped
-----------------------------------------------------------
dp0p4p2         0          0                0          0
                1          0                0          0
                2 2516476820       2805480368 1732333195
                3          0                0          0
vyatta@vyatta:~$
```

The following example shows specific QoS policies.

```
vyatta@vyatta:~$ show queuing dp0p1p1
dp0p4p2 Queueing:
Class       Queue    Packets            Bytes    Dropped
0               0          0                0          0
```

```
                1          0                0          0
                2          0                0          0
                3          0                0          0
1               0          0                0          0
                1          0                0          0
                2 2683633772       2058105936 2275510035
                3          0                0          0
vyatta@vyatta:~$
```

The following example shows queuing class information. In the output, the Prio column refers to priority traffic class.

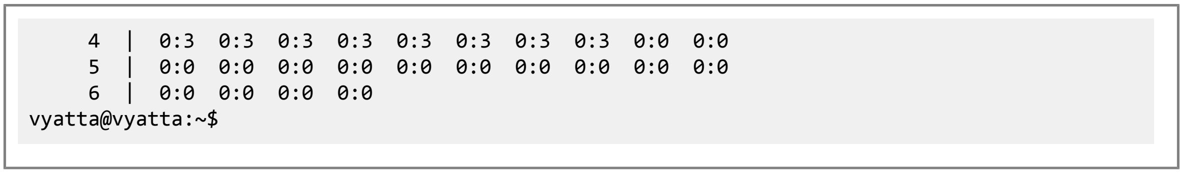
```
vyatta@vyatta:~$ show queuing class
Interface   Prio    Packets   Bytes  Match
------------------------------------------------------------
dp0s3       1       780       98312  proto 6 tag 1
vyatta@vyatta:~$
```

The following example shows PCP information.

```
vyatta@vyatta:~$ show queuing dp0s3 map pcp
Class Of Service->TC:WRR map for default

  PCP |    0    1    2    3    4    5    6    7
  ----+-----------------------------------------
      |  3:0  3:0  2:0  2:0  1:0  1:0  0:0  0:0
Class Of Service->TC:WRR map for class 1

  PCP |    0    1    2    3    4    5    6    7
  ----+-----------------------------------------
      |  3:0  3:0  2:0  2:0  1:0  1:0  0:0  0:0
vyatta@vyatta:~$
```

The following example shows DSCP information.

```
vyatta@vyatta:~$ show queuing dp0s3 map dscp
 DSCP->TC:WRR map for default: (dscp=d1d2)
  d2    |    0    1    2    3    4    5    6    7    8    9
  d1    |
  ------+---------------------------------------------------
     0  |  3:0  3:0  3:0  3:0  3:0  3:0  3:0  3:0  3:0  3:0
     1  |  3:0  3:0  3:0  3:0  3:0  3:0  2:0  2:0  2:0  2:0
     2  |  2:0  2:0  2:0  2:0  2:0  2:0  2:0  2:0  2:0  2:0
     3  |  2:0  2:0  1:0  1:0  1:0  1:0  1:0  1:0  1:0  1:0
     4  |  1:0  1:0  1:0  1:0  1:0  1:0  1:0  1:0  0:0  0:0
     5  |  0:0  0:0  0:0  0:0  0:0  0:0  0:0  0:0  0:0  0:0
     6  |  0:0  0:0  0:0  0:0
DSCP->TC:WRR map for class 1: (dscp=d1d2)

  d2    |    0    1    2    3    4    5    6    7    8    9
  d1    |
  ------+---------------------------------------------------
     0  |  3:0  3:0  3:0  3:0  3:0  3:0  3:0  3:0  3:0  3:0
     1  |  0:2  3:0  3:0  3:0  3:0  3:0  2:0  2:0  2:0  2:0
     2  |  2:0  2:0  2:0  2:0  2:0  2:0  2:0  2:0  2:0  2:0
     3  |  2:0  2:0  0:3  0:3  0:3  0:3  0:3  0:3  0:3  0:3
```



```
     4  |  0:3  0:3  0:3  0:3  0:3  0:3  0:3  0:3  0:0  0:0
     5  |  0:0  0:0  0:0  0:0  0:0  0:0  0:0  0:0  0:0  0:0
     6  |  0:0  0:0  0:0  0:0
vyatta@vyatta:~$
```

ICMP Types

This appendix lists the Internet Control Messaging Protocol (ICMP) types defined by the Internet Assigned Numbers Authority (IANA).

The IANA has developed a standard that maps a set of integers onto ICMP types. The following table lists the ICMP types and codes defined by the IANA and maps them to the literal strings that are available in the vRouter system.

Table 13: ICMP types

ICMP Type	Code	Literal	Description
0 - Echo reply	0	echo-reply	Echo reply (pong)
3 - Destination unreachable		destination- unreachable	Destination is unreachable
	0	network-unreachable	Destination network is unreachable
	1	host-unreachable	Destination host is unreachable
	2	protocol-unreachable	Destination protocol is unreachable
	3	port-unreachable	Destination port is unreachable
	4	fragmentation-needed	Fragmentation is required
	5	source-route-failed	Source route has failed
	6	network-unknown	Destination network is unknown
	7	host-unknown	Destination host is unknown
	9	network-prohibited	Network is administratively prohibited
	10	host-prohibited	Host is administratively is prohibited
	11	ToS-network-unreachable	Network is unreachable for ToS
	12	ToS-host-unreachable	Host is unreachable for ToS

ICMP Type	Code	Literal	Description
	13	communication-prohibited	Communication is administratively prohibited
	14	host-precedence-violation	Requested precedence is not permitted.
	15	precedence-cutoff	Precedence is lower than the required minimum.
4 - Source quench	0	source-quench	Source is quenched (congestion control)
5 - Redirect message		redirect	Redirected message
	0	network-redirect	Datagram is redirected for the network
	1	host-redirect	Datagram is redirected for the host
	2	ToS-network-redirect	Datagram is redirected for the ToS and network
	3	ToS-host-redirect	Datagram is redirected for the ToS and host
8 - Echo request	0	echo-request	Echo request (ping)
9 - Router advertisement	0	router-advertisement	Router advertisement
10 - Router solicitation	0	router-solicitation	Router solicitation
11 - Time exceeded		time-exceeded	Time to live (TTL) has exceeded
	0	ttl-zero-during-transit	TTL has expired in transit
	1	ttl-zero-during-reassembly	Fragment reassembly time has exceeded
12 - Parameter problem: Bad IP header		parameter-problem	Bad IP header
	0	ip-header-bad	Pointer that indicates an error
	1	required-option-missing	Missing required option
13 - Timestamp	0	timestamp-request	Request for a timestamp
14 - Timestamp reply	0	timestamp-reply	Reply to a request for a timestamp

ICMP Type	Code	Literal	Description
15 - Information request	0		Information request
16 - Information reply	0		Information reply
17 - Address mask request	0	address-mask-request	Address mask request
18 - Address mask reply	0	address-mask-reply	Address mask reply

ICMPv6 Types

This appendix lists the ICMPv6 types defined by the Internet Assigned Numbers Authority (IANA).

The Internet Assigned Numbers Authority (IANA) has developed a standard that maps a set of integers onto ICMPv6 types. The following table lists the ICMPv6 types and codes defined by the IANA and maps them to the strings literal strings available in the AT&T Vyatta vRouter system.

Table 14: ICMPv6 types

ICMPv6 Type	Code	Literal	Description
1 - Destination unreachable		destination- unreachable	
	0	no-route	No route to destination
	1	communication-prohibited	Communication with destination administratively prohibited
	2		Beyond scope of source address
	3	address-unreachable	Address unreachable
	4	port-unreachable	Port unreachable
	5		Source address failed ingress/egress policy
	6		Reject route to destination
2 - Packet too big	0	packet-too-big	
3 - Time exceeded		time-exceeded	
	0	ttl-zero-during-transit	Hop limit exceeded in transit
	1	ttl-zero-during-reassembly	Fragment reassembly time exceeded
4 - Parameter problem		parameter-problem	
	0	bad-header	Erroneous header field encountered
	1	unknown-header-type	Unrecognized Next Header type encountered

ICMPv6 Type	Code	Literal	Description
	2	unknown-option	Unrecognized IPv6 option encountered
128 - Echo request	0	echo-request (ping)	Echo request
129 - Echo reply	0	echo-reply (pong)	Echo reply
133 - Router solicitation	0	router-solicitation	Router solicitation
134 - Router advertisement	0	router-advertisement	Router advertisement
135 - Neighbor solicitation	0	neighbor-solicitation (neighbour-solicitation)	Neighbor solicitation
136 - Neighbor advertisement	0	neighbor-advertisement (neighbour-advertisement)	Neighbor advertisement

List of Acronyms

Acronym	Description
ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
AH	Authentication Header
AMI	Amazon Machine Image
API	Application Programming Interface
AS	autonomous system
ARP	Address Resolution Protocol
AWS	Amazon Web Services
BGP	Border Gateway Protocol
BIOS	Basic Input Output System
BPDU	Bridge Protocol Data Unit
CA	certificate authority
CCMP	AES in counter mode with CBC-MAC
CHAP	Challenge Handshake Authentication Protocol
CLI	command-line interface
DDNS	dynamic DNS
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuration Protocol version 6
DLCI	data-link connection identifier
DMI	desktop management interface
DMVPN	dynamic multipoint VPN
DMZ	demilitarized zone
DN	distinguished name
DNS	Domain Name System
DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
eBGP	external BGP
EBS	Amazon Elastic Block Storage
EC2	Amazon Elastic Compute Cloud
EGP	Exterior Gateway Protocol
ECMP	equal-cost multipath
ESP	Encapsulating Security Payload
FIB	Forwarding Information Base
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
HDLC	High-Level Data Link Control
I/O	Input/Output
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers

Acronym	Description
IGMP	Internet Group Management Protocol
IGP	Interior Gateway Protocol
IPS	Intrusion Protection System
IKE	Internet Key Exchange
IP	Internet Protocol
IPOA	IP over ATM
IPsec	IP Security
IPv4	IP Version 4
IPv6	IP Version 6
ISAKMP	Internet Security Association and Key Management Protocol
ISM	Internet Standard Multicast
ISP	Internet Service Provider
KVM	Kernel-Based Virtual Machine
L2TP	Layer 2 Tunneling Protocol
LACP	Link Aggregation Control Protocol
LAN	local area network
LDAP	Lightweight Directory Access Protocol
LLDP	Link Layer Discovery Protocol
MAC	medium access control
mGRE	multipoint GRE
MIB	Management Information Base
MLD	Multicast Listener Discovery
MLPPP	multilink PPP
MRRU	maximum received reconstructed unit
MTU	maximum transmission unit
NAT	Network Address Translation
NBMA	Non-Broadcast Multi-Access
ND	Neighbor Discovery
NHRP	Next Hop Resolution Protocol
NIC	network interface card
NTP	Network Time Protocol
OSPF	Open Shortest Path First
OSPFv2	OSPF Version 2
OSPFv3	OSPF Version 3
PAM	Pluggable Authentication Module
PAP	Password Authentication Protocol
PAT	Port Address Translation
PCI	peripheral component interconnect
PIM	Protocol Independent Multicast
PIM-DM	PIM Dense Mode
PIM-SM	PIM Sparse Mode
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPPoA	PPP over ATM

Acronym	Description
PPPoE	PPP over Ethernet
PPTP	Point-to-Point Tunneling Protocol
PTMU	Path Maximum Transfer Unit
PVC	permanent virtual circuit
QoS	quality of service
RADIUS	Remote Authentication Dial-In User Service
RHEL	Red Hat Enterprise Linux
RIB	Routing Information Base
RIP	Routing Information Protocol
RIPng	RIP next generation
RP	Rendezvous Point
RPF	Reverse Path Forwarding
RSA	Rivest, Shamir, and Adleman
Rx	receive
S3	Amazon Simple Storage Service
SLAAC	Stateless Address Auto-Configuration
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SONET	Synchronous Optical Network
SPT	Shortest Path Tree
SSH	Secure Shell
SSID	Service Set Identifier
SSM	Source-Specific Multicast
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus
TBF	Token Bucket Filter
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
ToS	Type of Service
TSS	TCP Maximum Segment Size
Tx	transmit
UDP	User Datagram Protocol
VHD	virtual hard disk
vif	virtual interface
VLAN	virtual LAN
VPC	Amazon virtual private cloud
VPN	virtual private network
VRRP	Virtual Router Redundancy Protocol
WAN	wide area network
WAP	wireless access point
WPA	Wired Protected Access