Applying QoS Policies

This chapter describes how to apply QoS policies, and contains the following sections:

Overview

Have you ever participated in a long-distance phone call that involved a satellite connection? The conversation may be interrupted with brief, but perceptible, gaps at odd intervals. Those gaps are the time, called the *latency*, between the arrival of packets being transmitted over the network. Some network traffic, such as voice and streaming video, cannot tolerate long latency times. *Quality of Service* (*QoS*) is a network feature that lets you give priority to these types of traffic.

As the Internet community of users upgrades their access points from modems to high-speed broadband connections like DSL and cable, the likelihood increases that at any given time, a single user might be able to absorb most, if not all, of the available bandwidth, thus starving the other users. To prevent any one user or site-to-site connection from consuming more than its fair share of bandwidth, QoS provides a policing feature that regulates the maximum bandwidth that any user can use.

QoS refers to the capability of a network to provide better service to selected network traffic over various technologies for the best overall services with limited bandwidth of the underlying technologies.

The primary goal of QoS in the security appliance is to provide rate limiting on selected network traffic for both individual flow or VPN tunnel flow to ensue that all traffic gets its fair share of limited bandwidth.A flow can be defined in a number of ways. In the security appliance, QoS can apply to a combination of source and destination IP addresses, source and destination port number, and the TOS byte of the IP header.

QoS Concepts

QoS is a traffic-management strategy that lets you allocate network resources for both mission-critical and normal data, based on the type of network traffic and the priority you assign to that traffic. In short, QoS ensures unimpeded priority traffic and provides the capability of rate-limiting (policing) default traffic.

For example, video and voice over IP (VoIP) are increasingly important for interoffice communication between geographically dispersed sites, using the infrastructure of the Internet as the transport mechanism. Firewalls are key to securing networks by controlling access, which includes inspecting VoIP protocols. QoS is the focal point to provide clear, uninterrupted voice and video communications, while still providing a basic level of service for all other traffic passing through the device.

For voice and video to traverse IP networks in a secure, reliable, and toll-quality manner, QoS must be enabled at all points of the network. Implementing QoS lets you:

- **Simplify network operations** by collapsing all data, voice, and video network traffic onto a single backbone using similar technologies.
- **Enable new network applications**, such as integrated call center applications and video-based training, that can help differentiate enterprises in their respective market spaces and increase productivity.
- **Control resource use** by controlling which traffic receives which resources. For example, you can ensure that the most important, time-critical traffic receives the network resources (available bandwidth and minimum delay) it needs, and that other applications using the link get their fair share of service without interfering with mission-critical traffic.

QoS provides maximum rate control, or policing, for tunneled traffic for each individual user tunnel and every site-to-site tunnel. In this release, there is no minimum bandwidth guarantee.

The security appliance can police individual user traffic within a LAN-to-LAN tunnel by configuring class-maps that are not associated with the tunnel, but whose traffic eventually passes through the LAN-to-LAN tunnel. The traffic before the LAN-to-LAN tunnel can then be specifically policed as it passes through the tunnel and is policed again to the aggregate rate applied to the tunnel.

The security appliance achieves QoS by allowing two types of traffic queues for each interface: a low-latency queue (LLQ) and a default queue. Only the default traffic is subject to rate limiting.

Because QoS can consume large amounts of resources, which could degrade security appliance performance, QoS is disabled by default.

Note You must consider that in an ever-changing network environment, QoS is not a one-time deployment, but an ongoing, essential part of network design.

Implementing QoS

In general, provisioning QoS policies requires the following steps:

1. Specifying traffic classes.
2. Associating actions with each traffic class to formulate policies.
3. Activating the policies.

The specification of a classification policy—that is, the definition of traffic classes—is separate from the specification of the policies that act on the results of the classification.

A *traffic class* is a set of traffic that is identifiable by its packet content. For example, TCP traffic with a port value of 23 might be classified as a Telnet traffic class.

An *action* is a specific activity taken to protect information or resources, in this case to perform QoS functions. An action is typically associated with a specific traffic class.

Configuring a traditional QoS policy for the security appliance consists of the following steps:

- Defining traffic classes (**class-map** command).
- Associating policies and actions with each class of traffic (**policy-map** command).
- Attaching policies to logical or physical interfaces (**service-policy** command).

Note For detailed configuration steps, see the "Configuring QoS" section on page 21-9.

The **class-map** command defines a named object representing a class of traffic, specifying the packet matching criteria that identifies packets that belong to this class. The basic form of the command is:

```
class-map class-map-name-1
    match match-criteria-1
class-map class-map-name-n
    match match-criteria-n
```

The **policy-map** command defines a named object that represents a set of policies to be applied to a set of traffic classes. An example of such a policy is policing the traffic class to some maximum rate. The basic form of the command is:

```
policy-map policy-map-name
    class class-map-name-1
        policy-1
        policy-n
    class class-map-name-n
        policy-m
        policy-m+1
```

The **service-policy** command attaches a policy-map and its associated policies to a target, named interface.

Note QoS-related policies under policy-map-name apply only to the outbound traffic, not to the inbound traffic of the named interface.

The command also indicates whether the policies apply to packets coming from or sent to the target. For example, an output policy (applied to packets exiting an interface) is applied as follows:

```
hostname(config)# service-policy policy-map-name interface outside
```

In addition, if you are differentiating between priority traffic and best-effort traffic, you must define a low-latency queue (**priority-queue** command) on each named, physical interface transmitting prioritized traffic.

The following example enables a default priority-queue with the default queue-limit and tx-ring-limit:

```
priority-queue name-interface
```

The following sections explain each of these uses in more detail.

Identifying Traffic for QoS

The **class-map** command classifies a set of traffic with which QoS actions are associated. You can use various types of match criteria to classify traffic. The **match** commands identify the traffic included in the traffic class for a class map. They include different criteria to define the traffic included in a class-map. Define a traffic class using the **class-map** global configuration command as part of configuring a security feature using Modular Policy Framework. From class-map configuration mode, you can define the traffic to include in the class using the **match** command.

After a traffic class is applied to an interface, packets received on that interface are compared to the criteria defined by the **match** statements in the class map. If the packet matches the specified criteria, it is included in the traffic class and is subjected to any actions associated with that traffic class. Packets that do not match any of the criteria in any traffic class are assigned to the default traffic class.

One such criterion is access-list. For example, in the following sequence, the **class-map** command classifies all non-tunneled TCP traffic, using an access-list named tcp_traffic:

```
hostname(config)# access-list tcp_traffic permit tcp any any
hostname(config)# class-map tcp_traffic
hostname(config-cmap)# match access-list tcp_traffic
```

When a packet is matched against a class-map, the result is either a match or a no-match.

In the following example, other, more specific match criteria are used for classifying traffic for specific, security-related tunnel groups. These specific match criteria stipulate that a match on tunnel-group (in this case, the previously-defined Tunnel-Group-1) is required as the first match characteristic to classify traffic for a specific tunnel, and it allows for an additional match line to classify the traffic (IP differential services code point, expedited forwarding).

```
hostname(config)# class-map TG1-voice
hostname(config-cmap)# match tunnel-group Tunnel-Group-1
hostname(config-cmap)# match dscp ef
```

In the following example, the **class-map** command classifies both tunneled and non-tunneled traffic according to the traffic type:

```
hostname(config)# access-list tunneled extended permit ip 10.10.34.0 255.255.255.0 20.20.10.0 255.255.255.0
hostname(config)# access-list non-tunneled extended permit tcp any any
hostname(config)# tunnel-group tunnel-grp1 type IPSec_L2L

hostname(config)# class-map browse
hostname(config-cmap)# description "This class-map matches all non-tunneled tcp traffic."
hostname(config-cmap)# match access-list non-tunneled

hostname(config-cmap)# class-map TG1-voice
hostname(config-cmap)# description "This class-map matches all dscp ef traffic for tunnel-grp 1."
hostname(config-cmap)# match dscp ef
hostname(config-cmap)# match tunnel-group tunnel-grp1

hostname(config-cmap)# class-map TG1-BestEffort
hostname(config-cmap)# description "This class-map matches all best-effort traffic for tunnel-grp1."
hostname(config-cmap)# match tunnel-group tunnel-grp1
hostname(config-cmap)# match flow ip destination-address
```

The following example shows a way of policing a flow within a tunnel, provided the classed traffic is not specified as a tunnel, but does go *through* the tunnel. In this example, 192.168.10.10 is the address of the host machine on the private side of the remote tunnel, and the access list is named “host-over-l2l”. By creating a class-map (named “host-specific”), you can then police the “host-specific” class before the LAN-to-LAN connection polices the tunnel. In this example, the “host-specific” traffic is rate-limited before the tunnel, then the tunnel is rate-limited:

```
hostname(config)# access-list host-over-l2l extended permit ip any host 192.168.10.10
hostname(config)# class-map host-specific
hostname(config-cmap)# match access-list host-over-l2l
```

The following table summarizes the **match** command criteria available and relevant to QoS. For the full list of all match commands and their syntax, see *Cisco Security Appliance Command Reference*:

Command	Description
match access-list	Matches, by name or number, access list traffic within a class map.
match any	Identifies traffic that matches any of the criteria in the class map.
match dscp	Matches the IETF-defined DSCP value (in an IP header) in a class map. You can specify up to 64 different dscp values, defining the class as composed of packets that match any of the specified values.
match flow ip destination-address	Enables flow-based policy actions. The criteria to define flow is the destination IP address. All traffic going to a unique IP destination address is considered a flow. Policy action is applied to each flow instead of the entire class of traffic. This command always accompanies **match tunnel group**. For remote-access VPNs, this command applies to each remote-access host flow. For LAN-to-LAN VPNs, this command applies to the single aggregated VPN flow identified by the local and remote tunnel address pair.
match port	Specifies the TCP/UDP ports as the comparison criteria for packets received on that interface.
match precedence	Matches the precedence value represented by the TOS byte in the IP header. You can specify up to 8 different precedence values, defining the class as composed of packets that match any of the specified values.
match rtp	Matches traffic that uses a specific RTP port within a specified range. The allowed range is targeted at capturing applications likely to be using RTP. The packet matches the defined class only if the UDP port falls within the specified range, inclusive, and the port number is an even number.
match tunnel group	Matches every tunnel within the specified tunnel group.

In addition to the user-defined classes, a system-defined class named class-default also exists. This class-default represents all packets that do not match any of the user-defined classes, so that policies can be defined for these packets.

Defining a QoS Policy Map

The **policy-map** command configures various policies, such as security policies or QoS policies. A policy is an association of a traffic class, specified by a **class** command, and one or more actions. This section specifically deals with using the **policy-map** command to define the QoS policies for one or more classes of packets.

When you enter a **policy-map** command you enter the policy-map configuration mode, and the prompt changes to indicate this. In this mode, you can enter **class** and **description** commands. A **policy-map** command can specify multiple policies. The maximum number of policy maps is 64.

After entering the **policy-map** command, you then enter a **class** command to specify the classification of the packet traffic. The **class** command configures QoS policies for the class of traffic specified in the given class-map. A traffic class is a set of traffic that is identifiable by its packet content. For example, TCP traffic with a port value of 23 can be classified as a Telnet traffic class. The **class** commands are differentiated by their previously named and constructed class-map designations, and the associated actions follow immediately after.

The security appliance evaluates class-maps in the order in which they were entered in the policy-map configuration. It classifies a packet to the first class-map that matches the packet.

Note The order in which different types of actions in a policy-map are performed is independent of the order in which the actions appear in the command descriptions in this document.

The **priority** command provides low-latency queuing for delay-sensitive traffic, such as voice. This command selects all packets that match the associated class (TG1-voice in the previous example) and sends them to the low latency queue for priority processing.

Applying Rate Limiting

Every user's Bandwidth Limiting Traffic stream (BLT) can participate in maximum bandwidth limiting; that is, strict policing, which rate-limits the individual user's default traffic to some maximum rate. This prevents any one individual user's BLTs from overwhelming any other. LLQ traffic, however, is marked and processed downstream in a priority queue. This traffic is not rate-limited.

Policing is a way of ensuring that no traffic exceeds the maximum rate (bits/second) that you configure, thus ensuring that no one traffic flow can take over the entire resource. You use the **police** command to specify the maximum rate (that is, the rate limit for this traffic flow); this is a value in the range 8000-2000000000, specifying the maximum speed (*bits* per second) allowed.

You also specify what action, drop or transmit, to take for traffic that conforms to the limit and for traffic that exceeds the limit.

Note You can specify the drop action, but it is not functional. The action is always to transmit, except when the rate is exceeded, and even then, the action is to throttle the traffic to the maximum allowable speed.

The **police** command also configures the largest single burst of traffic allowed. A burst value in the range 1000-512000000 specifies the maximum number of instantaneous *bytes* allowed in a sustained burst before throttling to the conforming rate value.

Note Policing is applied only in the output direction.

You cannot enable both priority and policing together.

If a service policy is applied or removed from an interface that has existing VPN client/LAN-to-LAN or non-tunneled traffic already established, the QoS policy is not applied or removed from the traffic stream. To apply or remove the QoS policy for such connections, you must clear (that is, drop) the connections and re-establish them.

Note When policing is specified in the default class map, class-default, the police values of class-default are applied to the aggregated LAN-to-LAN VPN flow if there is no police command defined for tunnel-group of LAN-to-LAN VPN. In other words, the policing values of class-default are never applied to the individual flow of a LAN-to-LAN VPN that exists before encryption.

The following example builds on the configuration developed in the previous section. As in the previous example, there are two named class-maps: tcp_traffic and TG1-voice. Adding a third class-map:

```
hostname(config)# class-map TG1-best-effort
hostname(config-cmap)# match tunnel-group Tunnel-Group-1
hostname(config-cmap)# match flow ip destination-address
```

provides a basis for defining a tunneled and non-tunneled QoS policy, as follows, which creates a simple QoS policy for tunneled and non-tunneled traffic, assigning packets of the class TG1-voice to the low latency queue and setting rate limits on the tcp_traffic and TG1-best-effort traffic flows.

Note "Best effort" does not guarantee reliable packet delivery, in that it does not use a sophisticated acknowledgement system. It does, however, make a "best effort" to deliver packets to the destination.

In this example, the maximum rate for traffic of the tcp_traffic class is 56,000 bits/second and a maximum burst size of 10,500 bytes per second. For the TC1-BestEffort class, the maximum rate is 200,000 bits/second, with a maximum burst of 37,500 bytes/second. Traffic in the TC1-voice class has no policed maximum speed or burst rate because it belongs to a priority class:

```
hostname(config)# policy-map qos
hostname(config-pmap)# class tcp_traffic
hostname(config-pmap-c)# police output 56000 10500

hostname(config-pmap-c)# class TG1-voice
hostname(config-pmap-c)# priority

hostname(config-pmap-c)# class TG1-best-effort
hostname(config-pmap-c)# police output 200000 37500

hostname(config-pmap-c)# class class-default
hostname(config-pmap-c)# police output 1000000 37500
```

Note You can have up to 256 policy-maps, and up to 256 classes in a policy map. The maximum number of classes in all policy maps together is 256. For any class-map, you can have only one **match** statement associated with it, with the exception of a tunnel class. For a tunnel class, an additional ma**tch tunnel-group** statement is allowed.

Note The class **class-default** always exists. It does not need to be declared.

Activating the Service Policy

The **service-policy** command activates a **policy-map** command globally on all interfaces or on a targeted interface. An interface can be a virtual (vlan) interface or a physical interface. Only one global policy-map is allowed. If you specify the keyword **interface** and an interface name, the policy-map applies only to that interface. An interface policy-map inherits rules from the global policy-map. For rules that overlap with the global policy map, the interface policy rules will be applied. Only one interface policy-map can be applied to an interface at any one time.

In general, a **service-policy** command can be applied to any interface that can be defined by the **nameif** command.

Using the policy-map example in the previous section, the following **service-policy** command activates the policy-map "qos," defined in the previous section, for traffic on the outside interface:

```
hostname(config)# service-policy qos interface outside
```

Applying Low Latency Queueing

The security appliance allows two classes of traffic: low latency queuing (LLQ) for higher priority, latency-sensitive traffic (such as voice and video) and best effort, the default, for all other traffic. These two queues are built into the system. The security appliance recognizes QoS priority traffic and enforces appropriate QoS policies.

Because queues are not of infinite size, they can fill and overflow. When a queue is full, any additional packets cannot get into the queue and are dropped. This is *tail drop*. To avoid having the queue fill up, you can use the **queue-limit** command to increase the queue buffer size.

You can configure the low latency (priority) queue to fine-tune the maximum number of packets allowed into the transmit queue (using the **tx-ring-limit** command) and to size the depth of the priority queue (using the **queue-limit** command). This lets you control the latency and robustness of the priority queuing.

Note The upper limit of the range of values for the **queue-limit** and **tx-ring-limit** commands is determined dynamically at run time. To view this limit, enter **help** or **?** on the command line. The key determinants are the memory needed to support the queues and the memory available on the device. The range of queue-limit values is 0 through 2048 packets. The range of tx-ring-limit values is 3 through 128 packets on the PIX platform and 3 to 256 packets on the ASA platform.

Configuring Priority Queuing

You identify high priority traffic by using the **priority** command in Class mode. This command instructs the security appliance to mark as high priority the traffic selected by the class map.

For priority queuing to occur, you must create a priority queue for named, physical interfaces that transmit high priority traffic. To enable a priority queue on an interface, use the **priority-queue** command in global configuration mode. You can apply one **priority-queue** command to each physical interface defined by the **nameif** command. All other traffic is delivered on a best-effort basis.

In general, you can apply a **priority-queue** command to any physical interface that can be defined by the **nameif** command. You cannot apply a **priority-queue** command to a VLAN interface. The **priority-queue** command enters priority-queue mode, as shown by the prompt, which lets you configure the maximum number of packets allowed in the transmit queue and the size of the priority queue.

Note You cannot enable both priority queuing and policing together. In other words, only packets with normal priority can be policed; packets with high priority are not policed.

Sizing the Priority Queue

The size that you specify for the priority queue affects both the low latency queue and the best-effort queue. The **queue-limit** command specifies a maximum number of packets that can be queued to a priority queue before it drops data. This limit must be in the range of 0 through 2048 packets.

Reducing Queue Latency

The **tx-ring-limit** command lets you configure the maximum number of packets (that is, the depth) allowed to be queued in the Ethernet transmit driver ring at any given time. This allows for fine-tuning the transmit queue to reduce latency and offer better performance through the transmit driver. This limit must be in the range 3 through 128 packets on the PIX platform, with a limit of 256 packets on the ASA platform.

The default queue-limit is the number of average, 256-byte packets that the specified interface can transmit in a 500-ms interval, with an upper limit of 2048 packets. A packet that stays more than 500 ms in a network node might trigger a timeout in the end-to-end application. Such a packet can be discarded in each network node.

The default tx-ring-limit is the number of maximum 1550-byte packets that the specified interface can transmit in a 10-ms interval. This guarantees that the hardware-based transmit ring imposes no more than 10-ms of extra latency for a high-priority packet.

The following example establishes a priority queue on interface "outside" (the GigabitEthernet0/1 interface), with the default queue-limit and tx-ring-limit.

```
hostname(config)# priority-queue outside
```

The following example establishes a priority queue on the interface "outside" (the GigabitEthernet0/1 interface), sets the queue-limit to 2048 packets, and sets the tx-ring-limit to 256:

```
hostname(config)# priority-queue outside
hostname(config-priority-queue)# queue-limit 2048
hostname(config-priority-queue)# tx-ring-limit 256
```

Note When priority queuing is enabled, the security appliance empties all packets in higher priority queues before transmitting packets in lower priority queues.

Configuring QoS

The following procedure provides steps for configuring a traffic class, a policy map, and a service policy that implement QoS policing (rate limiting) or priority queuing. In addition, for priority queuing, it includes steps for enabling priority queues on interfaces.

The number of traffic classes, policy maps, and service policies needed to implement QoS varies depending upon the requirements of your network. Analyze your network and determine how many traffic classes, policy maps, and service policies needed on the security appliance you are configuring, and then use this procedure as applicable to your QoS deployment.

To configure QoS policing and priority queuing, perform the following steps:

Step 1 Determine which traffic you want to police or mark for priority queuing. For a detailed discussion of identifying QoS traffic, see the "Identifying Traffic for QoS" section on page 21-4.

Step 2 Create a class map or modify an existing class map to identify traffic that you want to police or to identify as priority traffic. Use the **class-map** command to do so, as follows:

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

where *class_map_name* is the name of the traffic class. When you enter the **class-map** command, the CLI enters class map configuration mode.

Step 3 Identify the traffic you determined in Step 1. To do so, use a **match** command. For a detailed discussion of identifying QoS traffic, see the "Identifying Traffic for QoS" section on page 21-4.

If you need to identify two or more non-contiguous ports, create an access list with the **access-list extended** command, add an ACE to match each port, and then use the **match access-list** command. The following commands show how to use an access list to identify multiple TCP ports with an access list:

```
hostname(config)# access-list acl-name any any tcp eq port_number_1
hostname(config)# access-list acl-name any any tcp eq port_number_2
hostname(config)# class-map class_map_name
hostname(config-cmap)# match access-list acl-name
```

If you need to identify a single port, use the **match port** command, as follows:

```
hostname(config-cmap)# match port {tcp | udp} port_number
```

where *port_number* is the destination port of traffic that you want to configure the security appliance to police or mark for priority queuing.

If you need to identify a range of contiguous ports, use **match port** command with the **range** keyword, as follows:

```
hostname(config-cmap)# match port {tcp | udp} range begin_port_number end_port_number
```

where *begin_port_number* is the lowest port in the range of ports and *end_port_number* is the highest port.

Step 4 Create a policy map or modify an existing policy map that you want to use to apply policing or priority queuing to the traffic identified in Step 2. For more information about QoS policy maps, see the "Defining a QoS Policy Map" section on page 21-5.

Use the **policy-map** command, as follows:

```
hostname(config-cmap)# policy-map policy_map_name
hostname(config-pmap)#
```

where *policy_map_name* is the name of the policy map. The CLI enters the policy map configuration mode and the prompt changes accordingly.

Step 5 Specify the class map, created in Step 2, that identifies the traffic to be policed or marked for priority queuing. Use the **class** command to do so, as follows:

```
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

where *class_map_name* is the name of the class map you created in Step 2. The CLI enters the policy map class configuration mode and the prompt changes accordingly.

Step 6 Configure the action for the class. You can either mark the traffic class as priority traffic or specify rate limiting for the traffic class. Do one of the following:

- If you want the traffic selected by the class map to be marked as priority traffic, enter the **priority** command.

```
hostname(config-pmap-c)# priority
```

Note Priority queuing does not occur automatically to traffic marked as priority. To enable priority queuing, you must complete Step 8 also, which enables the priority queues.

For details about priority queuing, see the "Applying Low Latency Queueing" section on page 21-8 and the **priority** command page in the *Cisco Security Appliance Command Reference*.

- If you want the security appliance to police the traffic selected by the class map, enter the **police** command.

```
hostname(config-pmap-c)# police [output] conform-rate [conform-burst] [conform-action
[drop | transmit] [exceed-action {drop | transmit}]]
```

For details about the use of the **police** command, see the "Applying Rate Limiting" section on page 21-6 and the **police** command page in the *Cisco Security Appliance Command Reference*.

Step 7 Use the **service-policy** command to apply the policy map globally or to a specific interface, as follows:

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface interface_ID]
hostname(config)#
```

where *policy_map_name* is the policy map you configured in Step 4. If you want to apply the policy map to traffic on all the interfaces, use the **global** option. If you want to apply the policy map to traffic on a specific interface, use the **interface** *interface_ID* option, where *interface_ID* is the name assigned to the interface with the **nameif** command.

The security appliance begins policing traffic and marking traffic for priority queuing, as specified.

Step 8 If in Step 6 you entered the **priority** command, you must enable priority queues on interfaces before the security appliance performs priority queuing.

For each interface on which you want the security appliance to perform priority queuing, perform the following steps:

a. Enter the **priority-queue** command:

```
hostname(config)# priority-queue interface
hostname(config-priority-queue)#
```

where *interface* is the name assigned to the physical interface whose priority queue you want to enable. VLAN interfaces do not support priority queuing. The CLI enters the Priority-queue configuration mode and the prompt changes accordingly

b. (Optional) If you want to specify a *non-default* maximum number of priority packets that can be queued, enter the **queue-limit** command, as follows:

```
hostname(config-priority-queue)# queue-limit number-of-packets
```

The default queue size is 2048 packets.

c. (Optional) If you want specify a *non-default* maximum number of packets allowed into the transmit queue, enter the **tx-ring-limit** command, as follows:

```
hostname(config-priority-queue)# tx-ring-limit number-of-packets
```

The default transmit queue size is 128 packets.

On the interfaces you enabled priority queuing, the security appliance begins performing priority queuing.

The following example creates class maps for high priority (voice) and best effort traffic for a previously configured tunnel group, named "tunnel-grp1". The "qos" policy map includes the **police** command for the best effort and the default traffic classes and the **priority** command for the voice class. The service policy is then applied to the outside interface and the priority queue for the outside interface is enabled.

Example 1 Configuring QoS Policing and Priority Queuing

```
hostname(config)# class-map TG1-voice
hostname(config-cmap)# description "This class-map matches all dscp ef traffic for tunnel-grp 1"
hostname(config-cmap)# match dscp ef
hostname(config-cmap)# match tunnel-group tunnel-grp1

hostname(config-cmap)# class-map TG1-BestEffort
hostname(config-cmap)# description "This class-map matches all best-effort traffic for tunnel-grp1"
hostname(config-cmap)# match tunnel-group tunnel-grp1
hostname(config-cmap)# match flow ip destination-address

hostname(config-cmap)# policy-map qos
hostname(config-pmap)# class TG1-voice
hostname(config-pmap-c)# priority
hostname(config-pmap-c)# class TG1-best-effort
hostname(config-pmap-c)# police output 200000 37500
hostname(config-pmap-c)# class class-default
hostname(config-pmap-c)# police output 1000000 37500

hostname(config-pmap-c)# service-policy qos interface outside
hostname(config)# priority-queue outside
hostname(config-priority-queue)# queue-limit 2048
hostname(config-priority-queue)# tx-ring-limit 256
```

Viewing QoS Configuration

This section contains the following topics:

- Viewing QoS Policy Map Configuration, page 21-13
- Viewing the Priority-Queue Configuration for an Interface, page 21-13

Viewing QoS Service Policy Configuration

To view all current service policies, including those that implement QoS policy maps, use the **show service-policy** command in privileged EXEC mode. You can limit the output to policies that include the **police** or **priority** commands by using the **police** or **priority** keywords.

Note This is the same command you use to view priority and police statistics.

The following example shows the output of **show service-policy** with the **police** keyword:

```
hostname# show service-policy police

Global policy:
    Service-policy: global_fw_policy

Interface outside:
    Service-policy: qos
        Class-map: browse
            police Interface outside:
                cir 56000 bps, bc 10500 bytes
                conformed 10065 packets, 12621510 bytes; actions: transmit
                exceeded 499 packets, 625146 bytes; actions: drop
                conformed 5600 bps, exceed 5016 bps
```

```
        Class-map: cmap2
            police Interface outside:
                cir 200000 bps, bc 37500 bytes
                conformed 17179 packets, 20614800 bytes; actions: transmit
                exceeded 617 packets, 770718 bytes; actions: drop
                conformed 198785 bps, exceed 2303 bps
```

The following example shows the output of **show service-policy** with the **priority** keyword:

```
hostname# show service-policy priority
Global policy:
    Service-policy: global_fw_policy
Interface outside:
    Service-policy: qos
        Class-map: TG1-voice
            Priority:
                Interface outside: aggregate drop 0, aggregate transmit 9383
```

Viewing QoS Policy Map Configuration

To view all policy maps, including those that include the **police** and **priority** commands, use the following command in privileged EXEC mode:

```
hostname# show running-config policy-map
```

For the foregoing examples, the output of this command would look something like the following:

```
hostname# show running-config policy-map
!
policy-map test
 class class-default
policy-map inbound_policy
 class ftp-port
  inspect ftp strict inbound_ftp
policy-map qos
 class browse
  police 56000 10500
 class TG1-voice
  priority
 class TG1-BestEffort
  police 200000 37500
```

Viewing the Priority-Queue Configuration for an Interface

To display the priority-queue configuration for an interface, enter the show running-config priority-queue command in global configuration mode. The following example shows the priority-queue configuration for the interface named "test":

```
hostname(config)# show running-config priority-queue test
priority-queue test
  queue-limit   2048
  tx-ring-limit 256
hostname(config)#
```

Viewing QoS Statistics

This section contains the following topics:

Viewing QoS Police Statistics

To view the QoS statistics for traffic policing, use the **show service-policy** command with the **police** keyword, in privileged EXEC mode:

```
hostname# show service-policy police
```

Note This is the same command you use to view configuration of policies that include the **police** keyword.

For example, the following command displays service policies that include the **police** command and the related statistics; for example:

```
hostname# show service-policy police

Global policy:
    Service-policy: global_fw_policy

Interface outside:
    Service-policy: qos
        Class-map: browse
            police Interface outside:
                cir 56000 bps, bc 10500 bytes
                conformed 10065 packets, 12621510 bytes; actions: transmit
                exceeded 499 packets, 625146 bytes; actions: drop
                conformed 5600 bps, exceed 5016 bps
        Class-map: cmap2
            police Interface outside:
                cir 200000 bps, bc 37500 bytes
                conformed 17179 packets, 20614800 bytes; actions: transmit
                exceeded 617 packets, 770718 bytes; actions: drop
                conformed 198785 bps, exceed 2303 bps
```

Viewing QoS Priority Statistics

To view statistics for service policies implementing the **priority** command, use the **show service-policy** command with the **priority** keyword, in privileged EXEC mode:

```
hostname# show service-policy priority
```

Note This is the same command you use to view configuration of policies that include the **priority** keyword.

For example, the following command displays service policies that include the **priority** command and the related statistics; for example:

```
hostname# show service-policy priority
Global policy:
    Service-policy: global_fw_policy
Interface outside:
    Service-policy: qos
        Class-map: TG1-voice
            Priority:
                Interface outside: aggregate drop 0, aggregate transmit 9383
```

Note "Aggregate drop" denotes the aggregated drop in this interface; "aggregate transmit" denotes the aggregated number of transmitted packets in this interface.

Viewing QoS Priority Queue Statistics

To display the priority-queue statistics for an interface, use the **show priority-queue statistics** command in privileged EXEC mode. The results show the statistics for both the best-effort (BE) queue and the low-latency queue (LLQ). The following example shows the use of the **show priority-queue statistics** command for the interface named test, and the command output:

```
hostname# show priority-queue statistics test

Priority-Queue Statistics interface test

Queue Type         = BE
Packets Dropped    = 0
Packets Transmit   = 0
Packets Enqueued   = 0
Current Q Length   = 0
Max Q Length       = 0

Queue Type         = LLQ
Packets Dropped    = 0
Packets Transmit   = 0
Packets Enqueued   = 0
Current Q Length   = 0
Max Q Length       = 0
hostname#
```

In this statistical report, the meaning of the line items is as follows:

- "Packets Dropped" denotes the overall number of packets that have been dropped in this queue.
- "Packets Transmit" denotes the overall number of packets that have been transmitted in this queue.
- "Packets Enqueued" denotes the overall number of packets that have been queued in this queue.
- "Current Q Length" denotes the current depth of this queue.
- "Max Q Length" denotes the maximum depth that ever occurred in this queue.