

Quality of Service

Have you ever participated in a long-distance phone call that involved a satellite connection? The conversation might 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 video, cannot tolerate long latency times. Quality of service (QoS) is a feature that lets you give priority to critical traffic, prevent bandwidth hogging, and manage network bottlenecks to prevent packet drops.

Note For the ASASM, we suggest performing QoS on the switch instead of the ASASM. Switches have more capability in this area.

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

- Information About QoS, page 19-1
- Licensing Requirements for QoS, page 19-5
- Guidelines and Limitations, page 19-5
- Configuring QoS, page 19-6
- Monitoring QoS, page 19-16
- Feature History for QoS, page 19-19

Information About QoS

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

This section describes the QoS features supported by the ASA and includes the following topics:

- Supported QoS Features, page 19-2
- What is a Token Bucket?, page 19-2
- Information About Policing, page 19-3
- Information About Priority Queuing, page 19-3
- Information About Traffic Shaping, page 19-4
- DSCP and DiffServ Preservation, page 19-5

Supported QoS Features

The ASA supports the following QoS features:

- Policing—To prevent individual flows from hogging the network bandwidth, you can limit the maximum bandwidth used per flow. See Information About Policing, page 19-3 for more information.
- Priority queuing—For critical traffic that cannot tolerate latency, such as Voice over IP (VoIP), you can identify traffic for Low Latency Queuing (LLQ) so that it is always transmitted ahead of other traffic. See Information About Priority Queuing, page 19-3 for more information.
- Traffic shaping—If you have a device that transmits packets at a high speed, such as a ASA with Fast Ethernet, and it is connected to a low speed device such as a cable modem, then the cable modem is a bottleneck at which packets are frequently dropped. To manage networks with differing line speeds, you can configure the ASA to transmit packets at a fixed slower rate. See Information About Traffic Shaping, page 19-4 for more information.

What is a Token Bucket?

A token bucket is used to manage a device that regulates the data in a flow. For example, the regulator might be a traffic policer or a traffic shaper. A token bucket itself has no discard or priority policy. Rather, a token bucket discards tokens and leaves to the flow the problem of managing its transmission queue if the flow overdrives the regulator.

A token bucket is a formal definition of a rate of transfer. It has three components: a burst size, an average rate, and a time interval. Although the average rate is generally represented as bits per second, any two values may be derived from the third by the relation shown as follows:

average rate = burst size / time interval

Here are some definitions of these terms:

- Average rate—Also called the committed information rate (CIR), it specifies how much data can be sent or forwarded per unit time on average.
- Burst size—Also called the Committed Burst (Bc) size, it specifies in bits or bytes per burst how much traffic can be sent within a given unit of time to not create scheduling concerns. (For traffic shaping, it specifies bits per burst; for policing, it specifies bytes per burst.)
- Time interval—Also called the measurement interval, it specifies the time quantum in seconds per burst.

In the token bucket metaphor, tokens are put into the bucket at a certain rate. The bucket itself has a specified capacity. If the bucket fills to capacity, newly arriving tokens are discarded. Each token is permission for the source to send a certain number of bits into the network. To send a packet, the regulator must remove from the bucket a number of tokens equal in representation to the packet size.

If not enough tokens are in the bucket to send a packet, the packet either waits until the bucket has enough tokens (in the case of traffic shaping) or the packet is discarded or marked down (in the case of policing). If the bucket is already full of tokens, incoming tokens overflow and are not available to future packets. Thus, at any time, the largest burst a source can send into the network is roughly proportional to the size of the bucket.

Note that the token bucket mechanism used for traffic shaping has both a token bucket and a data buffer, or queue; if it did not have a data buffer, it would be a policer. For traffic shaping, packets that arrive that cannot be sent immediately are delayed in the data buffer.

For traffic shaping, a token bucket permits burstiness but bounds it. It guarantees that the burstiness is bounded so that the flow will never send faster than the token bucket capacity, divided by the time interval, plus the established rate at which tokens are placed in the token bucket. See the following formula:

(token bucket capacity in bits / time interval in seconds) + established rate in bps = maximum flow speed in bps

This method of bounding burstiness also guarantees that the long-term transmission rate will not exceed the established rate at which tokens are placed in the bucket.

Information About Policing

Policing is a way of ensuring that no traffic exceeds the maximum rate (in bits/second) that you configure, thus ensuring that no one traffic flow or class can take over the entire resource. When traffic exceeds the maximum rate, the ASA drops the excess traffic. Policing also sets the largest single burst of traffic allowed.

Information About Priority Queuing

LLQ priority queuing lets you prioritize certain traffic flows (such as latency-sensitive traffic like voice and video) ahead of other traffic.

The ASA supports two types of priority queuing:

- Standard priority queuing—Standard priority queuing uses an LLQ priority queue on an interface (see Configuring the Standard Priority Queue for an Interface, page 19-8), while all other traffic goes into the “best effort” queue. 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 called *tail drop*. To avoid having the queue fill up, you can increase the queue buffer size. You can also fine-tune the maximum number of packets allowed into the transmit queue. These options let you control the latency and robustness of the priority queuing. Packets in the LLQ queue are always transmitted before packets in the best effort queue.
- Hierarchical priority queuing—Hierarchical priority queuing is used on interfaces on which you enable a traffic shaping queue. A subset of the shaped traffic can be prioritized. The standard priority queue is not used. See the following guidelines about hierarchical priority queuing:
  - Priority packets are always queued at the head of the shape queue so they are always transmitted ahead of other non-priority queued packets.
  - Priority packets are never dropped from the shape queue unless the sustained rate of priority traffic exceeds the shape rate.
  - For IPsec-encrypted packets, you can only match traffic based on the DSCP or precedence setting.
  - IPsec-over-TCP is not supported for priority traffic classification.

Information About Traffic Shaping

Traffic shaping is used to match device and link speeds, thereby controlling packet loss, variable delay, and link saturation, which can cause jitter and delay.

Note Traffic shaping is only supported on the ASA 5505.

- Traffic shaping must be applied to all outgoing traffic on a physical interface or in the case of the ASA 5505, on a VLAN. You cannot configure traffic shaping for specific types of traffic.
- Traffic shaping is implemented when packets are ready to be transmitted on an interface, so the rate calculation is performed based on the actual size of a packet to be transmitted, including all the possible overhead such as the IPsec header and L2 header.
- The shaped traffic includes both through-the-box and from-the-box traffic.
- The shape rate calculation is based on the standard token bucket algorithm. The token bucket size is twice the Burst Size value. See What is a Token Bucket?, page 19-2.
- When bursty traffic exceeds the specified shape rate, packets are queued and transmitted later. Following are some characteristics regarding the shape queue (for information about hierarchical priority queuing, see Information About Priority Queuing, page 19-3):
  - The queue size is calculated based on the shape rate. The queue can hold the equivalent of 200-milliseconds worth of shape rate traffic, assuming a 1500-byte packet. The minimum queue size is 64.
  - When the queue limit is reached, packets are tail-dropped.
  - Certain critical keep-alive packets such as OSPF Hello packets are never dropped.
  - The time interval is derived by *time_interval* = *burst_size* / *average_rate*. The larger the time interval is, the burstier the shaped traffic might be, and the longer the link might be idle. The effect can be best understood using the following exaggerated example:

    Average Rate = 1000000

    Burst Size = 1000000

    In the above example, the time interval is 1 second, which means, 1 Mbps of traffic can be bursted out within the first 10 milliseconds of the 1-second interval on a 100 Mbps FE link and leave the remaining 990 milliseconds idle without being able to send any packets until the next time interval. So if there is delay-sensitive traffic such as voice traffic, the Burst Size should be reduced compared to the average rate so the time interval is reduced.

How QoS Features Interact

You can configure each of the QoS features alone if desired for the ASA. Often, though, you configure multiple QoS features on the ASA so you can prioritize some traffic, for example, and prevent other traffic from causing bandwidth problems.

See the following supported feature combinations per interface:

- Standard priority queuing (for specific traffic) + Policing (for the rest of the traffic).

  You cannot configure priority queuing and policing for the same set of traffic.
- Traffic shaping (for all traffic on an interface) + Hierarchical priority queuing (for a subset of traffic).

You cannot configure traffic shaping and standard priority queuing for the same interface; only hierarchical priority queuing is allowed. For example, if you configure standard priority queuing for the global policy, and then configure traffic shaping for a specific interface, the feature you configured last is rejected because the global policy overlaps the interface policy.

Typically, if you enable traffic shaping, you do not also enable policing for the same traffic, although the ASA does not restrict you from configuring this.

DSCP and DiffServ Preservation

- DSCP markings are preserved on all traffic passing through the ASA.
- The ASA does not locally mark/remark any classified traffic, but it honors the Expedited Forwarding (EF) DSCP bits of every packet to determine if it requires “priority” handling and will direct those packets to the LLQ.
- DiffServ marking is preserved on packets when they traverse the service provider backbone so that QoS can be applied in transit (QoS tunnel pre-classification).

Licensing Requirements for QoS

The following table shows the licensing requirements for this feature:

Model	License Requirement
ASAv	Standard or Premium License.
All other models	Base License.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single context mode only. Does not support multiple context mode.

Firewall Mode Guidelines

Supported in routed firewall mode only. Does not support transparent firewall mode.

IPv6 Guidelines

Does not support IPv6.

Model Guidelines

- Traffic shaping is only supported on the ASA 5505. Multi-core models (such as the ASA 5500-X) do not support shaping.
- (ASA 5512-X through ASA 5555-X) Priority queuing is not supported on the Management 0/0 interface.

- (ASASM) Only policing is supported.

Additional Guidelines and Limitations

- QoS is applied unidirectionally; only traffic that enters (or exits, depending on the QoS feature) the interface to which you apply the policy map is affected. See Feature Directionality, page 1-2 for more information.
- For traffic shaping, you can only use the **class-default** class map, which is automatically created by the ASA, and which matches all traffic.
- For priority traffic, you cannot use the **class-default** class map.
- For hierarchical priority queuing, for encrypted VPN traffic, you can only match traffic based on the DSCP or precedence setting; you cannot match a tunnel group.
- For hierarchical priority queuing, IPsec-over-TCP traffic is not supported.
- You cannot configure traffic shaping and standard priority queuing for the same interface; only hierarchical priority queuing is allowed.
- For standard priority queuing, the queue must be configured for a physical interface or, for the ASA 5505 or ASASM, a VLAN.
- For policing, to-the-box traffic is not supported.
- For policing, traffic to and from a VPN tunnel bypass interface is not supported.
- For policing, when you match a tunnel group class map, only outbound policing is supported.

Configuring QoS

This section includes the following topics:

- Determining the Queue and TX Ring Limits for a Standard Priority Queue, page 19-7
- Configuring the Standard Priority Queue for an Interface, page 19-8
- Configuring a Service Rule for Standard Priority Queuing and Policing, page 19-9
- Configuring a Service Rule for Traffic Shaping and Hierarchical Priority Queuing, page 19-13

Determining the Queue and TX Ring Limits for a Standard Priority Queue

To determine the priority queue and TX ring limits, use the worksheets below.

Table 19-1 shows how to calculate the priority queue size. 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 (called *tail drop*). To avoid having the queue fill up, you can adjust the queue buffer size according to the Configuring the Standard Priority Queue for an Interface, page 19-8.

Table 19-1 Queue Limit Worksheet

Step 1	_____ *Outbound bandwidth (Mbps or Kbps)*[1]	Mbps	x	**125**	=	_____ *# of bytes/ms*		
		Kbps	x	**.125**	=	_____ *# of bytes/ms*		
Step 2	_____ *# of bytes/ms from Step 1*	÷	_____ *Average packet size (bytes)*[2]	x	_____ *Delay (ms)*[3]	=	_____ *Queue limit (# of packets)*	

1. For example, DSL might have an uplink speed of 768 Kbps. Check with your provider.
2. Determine this value from a codec or sampling size. For example, for VoIP over VPN, you might use 160 bytes. We recommend 256 bytes if you do not know what size to use.
3. The delay depends on your application. For example, the recommended maximum delay for VoIP is 200 ms. We recommend 500 ms if you do not know what delay to use.

Table 19-2 shows how to calculate the TX ring limit. This limit determines the maximum number of packets allowed into the Ethernet transmit driver before the driver pushes back to the queues on the interface to let them buffer packets until the congestion clears. This setting guarantees that the hardware-based transmit ring imposes a limited amount of extra latency for a high-priority packet.

Table 19-2 TX Ring Limit Worksheet

Step 1	_____ *Outbound bandwidth (Mbps or Kbps)*[1]	Mbps	x	**125**	=	_____ *# of bytes/ms*		
		Kbps	x	**0.125**	=	_____ *# of bytes/ms*		
Step 2	_____ *# of bytes/ms from Step 1*	÷	_____ *Maximum packet size (bytes)*[2]	x	_____ *Delay (ms)*[3]	=	_____ *TX ring limit (# of packets)*	

1. For example, DSL might have an uplink speed of 768 Kbps.Check with your provider.

2. Typically, the maximum size is 1538 bytes, or 1542 bytes for tagged Ethernet. If you allow jumbo frames (if supported for your platform), then the packet size might be larger.
3. The delay depends on your application. For example, to control jitter for VoIP, you should use 20 ms.

Configuring the Standard Priority Queue for an Interface

If you enable standard priority queuing for traffic on a physical interface, then you need to also create the priority queue on each interface. Each physical interface uses two queues: one for priority traffic, and the other for all other traffic. For the other traffic, you can optionally configure policing.

Note The standard priority queue is not required for hierarchical priority queuing with traffic shaping; see Information About Priority Queuing, page 19-3 for more information.

Restrictions

- (ASASM) The ASASM does not support priority queuing.
- (ASA 5512-X through ASA 5555-X) Priority queuing is not supported on the Management 0/0 interface.

Detailed Steps

	Command	Purpose
Step 1	**priority-queue** *interface_name* **Example:** `hostname(config)# priority-queue inside`	Creatse the priority queue, where the *interface_name* argument specifies the physical interface name on which you want to enable the priority queue, or for the ASA 5505 or ASASM, the VLAN interface name.

	Command	Purpose
Step 2	**queue-limit** *number_of_packets* **Example:** `hostname(config-priority-queue)# queue-limit 260`	Changes the size of the priority queues. The default queue limit is 1024 packets. 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 (called *tail drop*). To avoid having the queue fill up, you can use the **queue-limit** command to increase the queue buffer size. The upper limit of the range of values for the **queue-limit** command is determined dynamically at run time. To view this limit, enter **queue-limit ?** on the command line. The key determinants are the memory needed to support the queues and the memory available on the device. The **queue-limit** that you specify affects both the higher priority low-latency queue and the best effort queue.
Step 3	**tx-ring-limit** *number_of_packets* **Example:** `hostname(config-priority-queue)# tx-ring-limit 3`	Specifies the depth of the priority queues. The default tx-ring-limit is 128 packets. This command sets the maximum number of low-latency or normal priority packets allowed into the Ethernet transmit driver before the driver pushes back to the queues on the interface to let them buffer packets until the congestion clears. This setting guarantees that the hardware-based transmit ring imposes a limited amount of extra latency for a high-priority packet. The upper limit of the range of values for the **tx-ring-limit** command is determined dynamically at run time. To view this limit, enter **tx-ring-limit ?** on the command line. The key determinants are the memory needed to support the queues and the memory available on the device. The **tx-ring-limit** that you specify affects both the higher priority low-latency queue and the best-effort queue.

Examples

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 260 packets, and sets the tx-ring-limit to 3:

```
hostname(config)# priority-queue outside
hostname(config-priority-queue)# queue-limit 260
hostname(config-priority-queue)# tx-ring-limit 3
```

Configuring a Service Rule for Standard Priority Queuing and Policing

You can configure standard priority queuing and policing for different class maps within the same policy map. See How QoS Features Interact, page 19-4 for information about valid QoS configurations.

To create a policy map, perform the following steps.

Restrictions

- You cannot use the **class-default** class map for priority traffic.
- You cannot configure traffic shaping and standard priority queuing for the same interface; only hierarchical priority queuing is allowed.
- (ASASM) The ASASM only supports policing.
- For policing, to-the-box traffic is not supported.
- For policing, traffic to and from a VPN tunnel bypass interface is not supported.
- For policing, when you match a tunnel group class map, only outbound policing is supported.

Guidelines

- For priority traffic, identify only latency-sensitive traffic.
- For policing traffic, you can choose to police all other traffic, or you can limit the traffic to certain types.

Detailed Steps

	Command	Purpose
Step 1	**class-map** *priority_map_name* **Example:** `hostname(config)# class-map priority_traffic`	For priority traffic, creates a class map to identify the traffic for which you want to perform priority queuing.
Step 2	**match** *parameter* **Example:** `hostname(config-cmap)# match access-list priority`	Specifies the traffic in the class map. See Identifying Traffic (Layer 3/4 Class Maps), page 1-12 for more information.
Step 3	**class-map** *policing_map_name* **Example:** `hostname(config)# class-map policing_traffic`	For policing traffic, creates a class map to identify the traffic for which you want to perform policing.
Step 4	**match** *parameter* **Example:** `hostname(config-cmap)# match access-list policing`	Specifies the traffic in the class map. See Identifying Traffic (Layer 3/4 Class Maps), page 1-12 for more information.
Step 5	**policy-map** *name* **Example:** `hostname(config)# policy-map QoS_policy`	Adds or edits a policy map.

	Command	Purpose
Step 6	**class** *priority_map_name* **Example:** `hostname(config-pmap)# class priority_class`	Identifies the class map you created for prioritized traffic in Step 1.
Step 7	**priority** **Example:** `hostname(config-pmap-c)# priority`	Configures priority queuing for the class.
Step 8	**class** *policing_map_name* **Example:** `hostname(config-pmap)# class policing_class`	Identifies the class map you created for policed traffic in Step 3.
Step 9	**police** {**output** | **input**} *conform-rate* [*conform-burst*] [**conform-action** [**drop** | **transmit**]] [**exceed-action** [**drop** | **transmit**]] **Example:** `hostname(config-pmap-c)# police output 56000 10500`	Configures policing for the class. See the followingoptions: • *conform-burst argument*—Specifies the maximum number of instantaneous bytes allowed in a sustained burst before throttling to the conforming rate value, between 1000 and 512000000 bytes. • **conform-action**—Sets the action to take when the rate is less than the *conform_burst* value. • *conform-rate*—Sets the rate limit for this traffic flow; between 8000 and 2000000000 bits per second.] • **drop**—Drops the packet. • **exceed-action**—Sets the action to take when the rate is between the *conform-rate* value and the *conform-burst* value. • **input**—Enables policing of traffic flowing in the input direction. • **output**—Enables policing of traffic flowing in the output direction. • **transmit**—Transmits the packet.
Step 10	**service-policy** *policymap_name* {**global** | **interface** *interface_name*} **Example:** `hostname(config)# service-policy QoS_policy interface inside`	Activates the policy map on one or more interfaces. **global** applies the policy map to all interfaces, and **interface** applies the policy to one interface. Only one global policy is allowed. You can override the global policy on an interface by applying a service policy to that interface. You can only apply one policy map to each interface.

Examples

Example 19-1 Class Map Examples for VPN Traffic

In the following example, the **class-map** command classifies all non-tunneled TCP traffic, using an ACL 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
```

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-grp1
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
192.168.10.0 255.255.255.0
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 ACL 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 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.

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

Adding a third class map 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.

Example 19-2 Priority and Policing Example

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

hostname(config)# class-map TG1-voice
hostname(config-cmap)# match tunnel-group tunnel-grp1
hostname(config-cmap)# match dscp ef

hostname(config-cmap)# class-map TG1-BestEffort
hostname(config-cmap)# match tunnel-group tunnel-grp1
hostname(config-cmap)# match flow ip destination-address

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

hostname(config-pmap-c)# service-policy qos global
```

Configuring a Service Rule for Traffic Shaping and Hierarchical Priority Queuing

You can configure traffic shaping for all traffic on an interface, and optionally hierarchical priority queuing for a subset of latency-sensitive traffic.

This section includes the following topics:

- (Optional) Configuring the Hierarchical Priority Queuing Policy, page 19-13
- Configuring the Service Rule, page 19-14

(Optional) Configuring the Hierarchical Priority Queuing Policy

You can optionally configure priority queuing for a subset of latency-sensitive traffic.

Guidelines

- One side-effect of priority queuing is packet re-ordering. For IPsec packets, out-of-order packets that are not within the anti-replay window generate warning syslog messages. These warnings are false alarms in the case of priority queuing. You can configure the IPsec anti-replay window size to avoid possible false alarms. See the **crypto ipsec security-association replay** command in the command reference.

- For hierarchical priority queuing, you do not need to create a priority queue on an interface.

Restrictions

- For hierarchical priority queuing, for encrypted VPN traffic, you can only match traffic based on the DSCP or precedence setting; you cannot match a tunnel group.
- For hierarchical priority queuing, IPsec-over-TCP traffic is not supported.

Detailed Steps

	Command	Purpose
Step 1	**class-map** *priority_map_name* **Example:** `hostname(config)# class-map priority_traffic`	For hierarchical priority queuing, creates a class map to identify the traffic for which you want to perform priority queuing.
Step 2	**match** *parameter* **Example:** `hostname(config-cmap)# match access-list priority`	Specifies the traffic in the class map. See Identifying Traffic (Layer 3/4 Class Maps), page 1-12 for more information. For encrypted VPN traffic, you can only match traffic based on the DSCP or precedence setting; you cannot match a tunnel group.
Step 3	**policy-map** *priority_map_name* **Example:** `hostname(config)# policy-map priority-sub-policy`	Creates a policy map.
Step 4	**class** *priority_map_name* **Example:** `hostname(config-pmap)# class priority-sub-map`	Specifies the class map you created in Step 1.
Step 5	**priority** **Example:** `hostname(config-pmap-c)# priority`	Applies the priority queuing action to a class map. **Note** This policy has not yet been activated. You must activate it as part of the shaping policy. See Configuring the Service Rule, page 19-14.

Configuring the Service Rule

To configure traffic shaping and optional hiearchical priority queuing, perform the following steps.

Restrictions

- Traffic shaping is only supported on the ASA 5505. Multi-core models (such as the ASA 5500-X) do not support shaping.
- For traffic shaping, you can only use the **class-default** class map, which is automatically created by the ASA, and which matches all traffic.

- You cannot configure traffic shaping and standard priority queuing for the same interface; only hierarchical priority queuing is allowed. See How QoS Features Interact, page 19-4 for information about valid QoS configurations.
- You cannot configure traffic shaping in the global policy.

Detailed Steps

	Command	Purpose
Step 1	**policy-map** *name* **Example:** `hostname(config)# policy-map shape_policy`	Adds or edits a policy map. This policy map must be different from the hierarchical priority-queuing map.
Step 2	**class class-default** **Example:** `hostname(config-pmap)# class class-default`	Identifies all traffic for traffic shaping; you can only use the **class-default** class map, which is defined as **match any**, because the ASA requires all traffic to be matched for traffic shaping.
Step 3	**shape average** *rate* [*burst_size*] **Example:** `hostname(config-pmap-c)# shape average 70000 4000`	Enables traffic shaping, where the **average** *rate* argument sets the average rate of traffic in bits per second over a given fixed time period, between 64000 and 154400000. Specify a value that is a multiple of 8000. See Information About Traffic Shaping, page 19-4 for more information about how the time period is calculated. The *burst_size* argument sets the average burst size in bits that can be transmitted over a given fixed time period, between 2048 and 154400000. Specify a value that is a multiple of 128. If you do not specify the *burst_size*, the default value is equivalent to 4-milliseconds of traffic at the specified average rate. For example, if the average rate is 1000000 bits per second, 4 ms worth = 1000000 * 4/1000 = 4000.
Step 4	(Optional) **service-policy** *priority_policy_map_name* **Example:** `hostname(config-pmap-c)# service-policy priority-sub-policy`	Configures hierarchical priority queuing, where the *priority_policy_map_name* is the policy map you created for prioritized traffic in the (Optional) Configuring the Hierarchical Priority Queuing Policy, page 19-13.
Step 5	**service-policy** *policymap_name* **interface** *interface_name* **Example:** `hostname(config)# service-policy shape-policy interface inside`	Activates the shaping policy map on an interface.

Examples

The following example enables traffic shaping on the outside interface, and limits traffic to 2 Mbps; priority queuing is enabled for VoIP traffic that is tagged with DSCP EF and AF13 and for IKE traffic:

```
hostname(config)# access-list ike permit udp any any eq 500
hostname(config)# class-map ike
```

```
hostname(config-cmap)# match access-list ike

hostname(config-cmap)# class-map voice_traffic
hostname(config-cmap)# match dscp EF AF13

hostname(config-cmap)# policy-map qos_class_policy
hostname(config-pmap)# class voice_traffic
hostname(config-pmap-c)# priority
hostname(config-pmap-c)# class ike
hostname(config-pmap-c)# priority

hostname(config-pmap-c)# policy-map qos_outside_policy
hostname(config-pmap)# class class-default
hostname(config-pmap-c)# shape average 2000000 16000
hostname(config-pmap-c)# service-policy qos_class_policy

hostname(config-pmap-c)# service-policy qos_outside_policy interface outside
```

Monitoring QoS

This section includes the following topics:

- Viewing QoS Police Statistics, page 19-16
- Viewing QoS Standard Priority Statistics, page 19-17
- Viewing QoS Shaping Statistics, page 19-17
- Viewing QoS Standard Priority Queue Statistics, page 19-18

Viewing QoS Police Statistics

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

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

The following is sample output for the **show service-policy police** command:

```
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 Standard Priority Statistics

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

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

The following is sample output for the **show service-policy priority** command:

```
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 Shaping Statistics

To view statistics for service policies implementing the **shape** command, use the **show service-policy** command with the **shape** keyword:

```
hostname# show service-policy shape
```

The following is sample output for the **show service-policy shape** command:

```
hostname# show service-policy shape
Interface outside
  Service-policy: shape
    Class-map: class-default

      Queueing
      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0

      shape (average) cir 2000000, bc 8000, be 8000
```

The following is sample output of the **show service policy shape** command, which includes service policies that include the **shape** command and the **service-policy** command that calls the hierarchical priority policy and the related statistics:

```
hostname# show service-policy shape

Interface outside:
  Service-policy: shape
    Class-map: class-default

      Queueing
      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0

      shape (average) cir 2000000, bc 16000, be 16000
```

```
Service-policy: voip
  Class-map: voip

    Queueing
    queue limit 64 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (pkts output/bytes output) 0/0
  Class-map: class-default

    queue limit 64 packets
    (queue depth/total drops/no-buffer drops) 0/0/0
    (pkts output/bytes output) 0/0
```

Viewing QoS Standard 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.

Feature History for QoS

Table 19-3 lists each feature change and the platform release in which it was implemented.

Table 19-3 Feature History for QoS

Feature Name	Platform Releases	Feature Information
Priority queuing and policing	7.0(1)	We introduced QoS priority queuing and policing. We introduced the following commands: **priority-queue**, **queue-limit**, **tx-ring-limit**, **priority, police, show priority-queue statistics, show service-policy police, show service-policy priority, show running-config priority-queue, clear configure priority-queue** .
Shaping and hierarchical priority queuing	7.2(4)/8.0(4)	We introduced QoS shaping and hierarchical priority queuing. We introduced the following commands: **shape**, **show service-policy shape**.
Ten Gigabit Ethernet support for a standard priority queue on the ASA 5585-X	8.2(3)/8.4(1)	We added support for a standard priority queue on Ten Gigabit Ethernet interfaces for the ASA 5585-X.