

Congestion Management

Congestion management controls congestion after it has occurred on a network. Congestion is managed by using packet queuing methods and by shaping the packet flow through use of traffic regulation mechanisms.

This chapter provides details about congestion management techniques.

- Prerequisites for Configuring QoS Congestion Management, on page 1
- Information About Configuring Congestion Management, on page 1
- Congestion Management Overview, on page 1
- Traffic Shaping, on page 2
- How to Configure QoS Congestion Management, on page 3

Prerequisites for Configuring QoS Congestion Management

These prerequisites are required for configuring QoS congestion management on your network:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- You must be familiar with Cisco IOS XR QoS configuration tasks and concepts.

Information About Configuring Congestion Management

Congestion Management Overview

Congestion management features allow you to control congestion by determining the order in which a traffic flow (or packets) is sent out an interface based on priorities assigned to packets. Congestion management entails the creation of queues, assignment of packets to those queues based on the classification of the packet, and scheduling of the packets in a queue for transmission. The congestion management features allow you to specify creation of a different number of queues, affording greater or lesser degree of differentiation of traffic, and to specify the order in which that traffic is sent.

During periods with light traffic flow, that is, when no congestion exists, packets are sent out the interface as soon as they arrive. During periods of transmit congestion at the outgoing interface, packets arrive faster than

the interface can send them. If you use congestion management features, packets accumulating at an interface are queued until the interface is free to send them; they are then scheduled for transmission according to their assigned priority and the queuing method configured for the interface. The router determines the order of packet transmission by controlling which packets are placed in which queue and how queues are serviced with respect to each other.

In addition to queuing methods, QoS congestion management mechanisms, such as policers and shapers, are needed to ensure that a packet adheres to a contract and service. Both policing and shaping mechanisms use the traffic descriptor for a packet.

Policers and shapers usually identify traffic descriptor violations in an identical manner through the token bucket mechanism, but they differ in the way they respond to violations. A policer typically drops traffic flow; whereas, a shaper delays excess traffic flow using a buffer, or queuing mechanism, to hold the traffic for transmission at a later time.

Traffic shaping and policing can work in tandem. For example, a good traffic shaping scheme should make it easy for nodes inside the network to detect abnormal flows.

Low-Latency Queueing with Strict Priority Queueing

Strict Priority Queueing (PQ) allows delay-sensitive data, such as voice, to be dequeued and sent before packets in other queues are dequeued. Upto four priority queues are supported. The highest priority queue is referred to as PQ1; the lowest priority queue is referred to as PQ4.

Low-Latency Queueing (LLQ) enables the use of a single, strict priority queue at the class level, allowing you to direct traffic belonging to a class. To rank class traffic to the strict priority queue, you specify the named class within a policy map and then configure the **priority** command for the class. (Classes to which the **priority** command is applied are considered priority classes.) Within a policy map, you can give one or more classes priority status. When multiple classes within a single policy map are configured as priority classes, all traffic from these classes is enqueued to the same, single, strict priority queue.

Through use of the **priority** command, you can assign a strict PQ to any of the valid match criteria used to specify traffic. These methods of specifying traffic for a class include matching on protocols, IP precedence, and IP differentiated service code point (DSCP) values.

Traffic Shaping

Traffic shaping allows you to control the traffic flow exiting an interface to match its transmission to the speed of the remote target interface and ensure that the traffic conforms to policies contracted for it. Traffic adhering to a particular profile can be shaped to meet downstream requirements, thereby eliminating bottlenecks in topologies with data-rate mismatches.

To match the rate of transmission of data from the source to the target interface, you can limit the transfer of data to one of the following:

- A specific configured rate
- A derived rate based on the level of congestion

The rate of transfer depends on these three components that constitute the token bucket: burst size, mean rate, and time (measurement) interval. The mean rate is equal to the burst size divided by the interval.

When traffic shaping is enabled, the bit rate of the interface does not exceed the mean rate over any integral multiple of the interval. In other words, during every interval, a maximum of burst size can be sent. Within the interval, however, the bit rate may be faster than the mean rate at any given time.

When the peak burst size equals 0, the interface sends no more than the burst size every interval, achieving an average rate no higher than the mean rate. However, when the peak burst size is greater than 0, the interface can send as many as the burst size plus peak burst bits in a burst, if in a previous time period the maximum amount was not sent. Whenever less than the burst size is sent during an interval, the remaining number of bits, up to the peak burst size, can be used to send more than the burst size in a later interval.

Regulation of Traffic with the Shaping Mechanism

When incoming packets arrive at an interface, the packets are classified using a classification technique, such as the setting of the IP Precedence bits through the Modular QoS CLI (MQC). If the packet matches the specified classification, the traffic-shaping mechanism continues. Otherwise, no further action is taken.

How to Configure QoS Congestion Management

Configuring Guaranteed and Remaining Bandwidths

The **bandwidth** command allows you to specify the minimum guaranteed bandwidth to be allocated for a specific class of traffic.

The **bandwidth remaining** command specifies a weight for the class. If you do not configure the **bandwidth remaining** command for any class, the leftover bandwidth is allocated equally to all classes for which **bandwidth remaining** is not explicitly specified.

Guaranteed Service rate of a queue is defined as the bandwidth the queue receives when all the queues are congested. It is defined as:

Guaranteed Service Rate = minimum bandwidth + excess share of the queue

Restrictions

- The bandwidth reservation should be at the rate configured at Layer 1.
- The amount of bandwidth configured should be large enough to also accommodate Layer 2 overhead.
- A policy map can have all class bandwidths specified in kilobits per second or percentages but not a mixture of both in the same class.

The **bandwidth** command is supported only on policies configured on outgoing interfaces.

Configuring Guaranteed Bandwidth

Procedure

Step 1 **configure**

Step 2 **policy-map** *policy-name*

Example:

```
RP/0/(config)# policy-map policy1
```

Enters policy map configuration mode.

- Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.

Step 3 **class** *class-name*

Example:

```
RP/0/(config-pmap)# class class1
```

Specifies the name of the class whose policy you want to create or change.

Step 4 **bandwidth** {*rate* [*units*]| **percent** *percentage-value*}

Example:

```
RP/0/(config-pmap-c)# bandwidth percent 40
```

Enters policy map class configuration mode.

- Specifies the bandwidth allocated for a class belonging to a policy map.
- In this example, class class1 is guaranteed 40 percent of the interface bandwidth.

Step 5 **exit**

Example:

```
RP/0/(config-pmap-c)# exit
```

Returns the router to policy map configuration mode.

Step 6 **class** *class-name*

Example:

```
RP/0/(config-pmap)# class class2
```

Specifies the name of the class whose policy you want to create or change.

Step 7 **bandwidth** {*rate* [*units*]| **percent** *percentage-value*}

Example:

```
RP/0/(config-pmap-c)# bandwidth percent 40
```

Enters policy map class configuration mode.

- Specifies the bandwidth allocated for a class belonging to a policy map.
- In this example, class class2 is guaranteed 40 percent of the interface bandwidth.

Step 8 **exit**

Example:

```
RP/0/(config-pmap-c)# exit
```

Returns the router to policy map configuration mode.

Step 9 **class** *class-name*

Example:

```
RP/0/(config-pmap)# class class-default
```

Specifies the name of the class whose policy you want to create or change.

Step 10 **bandwidth** {*rate* [*units*]| **percent** *percentage-value*}

Example:

```
RP/0/(config-pmap-c)# bandwidth percent 20
```

Enters policy map class configuration mode.

- Specifies the bandwidth allocated for a class belonging to a policy map.
- In this example, class class-default is guaranteed 20 percent of the interface bandwidth.

Step 11 **exit**

Example:

```
RP/0/(config-pmap-c)# exit
```

Returns the router to policy map configuration mode.

Step 12 **exit**

Example:

```
RP/0/(config-pmap)# exit
```

Returns the router to global configuration mode.

Step 13 **interface** *type interface-path-id*

Example:

```
RP/0/(config)# interface tengige 0/2/0/0
```

Enters interface configuration mode and configures an interface.

Step 14 **service-policy output** *policy-map*

Example:

```
RP/0/ (config-if)# service-policy output policy1
```

Attaches a policy map to an output interface to be used as the service policy for that interface.

- In this example, the traffic policy evaluates all traffic leaving that interface.

Step 15 **end** or **commit**

Example:

```
RP/0/(config-if)# end
```

or

```
RP/0/(config-if)# commit
```

Saves configuration changes.

- When you issue the **end** command, the system prompts you to commit changes:

  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:

  Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.

  Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.

  Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 16 **show policy-map interface** *type interface-path-id* **output**

Example:

```
RP/0/ # show policy-map interface tengige 0/2/0/0
```

(Optional) Displays policy configuration information for all classes configured for all service policies on the specified interface.

Configuring Bandwidth Remaining

Procedure

Step 1 **configure**

Step 2 **policy-map** *policy-name*

Example:

```
RP/0/(config)# policy-map policy1
```

Enters policy map configuration mode.

- Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.

Step 3 **class** *class-name*

Example:

```
RP/0/(config-pmap)# class class1
```

Specifies the name of the class whose policy you want to create or change.

Step 4 **bandwidth remaining percent** *percentage-value*

Example:

```
RP/0/(config-pmap-c)# bandwidth remaining percent 40
```

Specifies how to allocate leftover bandwidth for class class1.

Step 5 **exit**

Example:

```
RP/0/(config-pmap-c)# exit
```

Returns the router to policy map configuration mode.

Step 6 **class** *class-name*

Example:

```
RP/0/(config-pmap)# class class2
```

Specifies the name of the class whose policy you want to create or change.

Step 7 **bandwidth remaining percent** *percentage-value*

Example:

```
RP/0/(config-pmap-c)# bandwidth remaining percent 40
```

Specifies how to allocate leftover bandwidth for class class2.

Step 8 **exit**

Example:

```
RP/0/(config-pmap-c)# exit
```

Returns the router to policy map configuration mode.

Step 9 **class** *class-name*

Example:

```
RP/0/(config-pmap)# class class-default
```

Specifies the name of the class whose policy you want to create or change.

Step 10 **bandwidth remaining percent** *percentage-value*

Example:

```
RP/0/(config-pmap-c)# bandwidth remaining percent 20
```

Specifies how to allocate leftover bandwidth for class class-default.

Step 11 **exit**

Example:

```
RP/0/(config-pmap-c)# exit
```

Returns the router to policy map configuration mode.

Step 12 **exit**

Example:

```
RP/0/(config-pmap)# exit
```

Returns the router to global configuration mode.

Step 13 **interface** *type interface-path-id*

Example:

```
RP/0/(config)# interface tengige 0/2/0/0
```

Enters interface configuration mode and configures an interface.

Step 14 **service-policy output** *policy-map*

Example:

```
RP/0/(config-if)# service-policy output policy1
```

Attaches a policy map to an output interface to be used as the service policy for that interface.

- In this example, the traffic policy evaluates all traffic leaving that interface.

Step 15 **end** or **commit**

Example:

```
RP/0/(config-if)# end
```

or

```
RP/0/(config-if)# commit
```

Saves configuration changes.

- When you issue the **end** command, the system prompts you to commit changes:

  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:

  Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.

  Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.

  Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 16 **show policy-map interface** *type interface-path-id* **output**

Example:

```
RP/0/ # show policy-map interface tengige 0/2/0/0
```

(Optional) Displays policy configuration information for all classes configured for all service policies on the specified interface.

Configuring Low-Latency Queueing with Strict Priority Queueing

The **priority** command configures LLQ with strict priority queuing (PQ) that allows delay-sensitive data such as voice to be dequeued and sent before packets in other queues are dequeued. When a class is marked as high priority using the **priority** command, you must configure a policer to limit the priority traffic. This configuration ensures that the priority traffic does not constrain all the other traffic on the line card, which protects low priority traffic from limitations. Use the **police** command to explicitly configure the policer.

Note Five levels of priorities are supported: priority level 1, priority level 2, priority level 3, priority level 4, and the priority level normal. If no priority level is configured, the default is priority level normal.

Restrictions

- Within a policy map, you can give one or more classes priority status. When multiple classes within a single policy map are configured as priority classes, all traffic from these classes is queued to the same single priority queue.
- The **shape average**,**bandwidth**, and **random-detect**commands cannot be configured in the same class with the **priority** command.

Procedure

Step 1 **configure**

Step 2 **policy-map** *policy-name*

Example:

```
RP/0/(config)# policy-map voice
```

Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy and enters the policy map configuration mode.

Step 3 **class** *class-name*

Example:

```
RP/0/(config-pmap)# class voice
```

Specifies the name of the class whose policy you want to create or change and enters the policy map class configuration mode.

Step 4 **priority**[**level** *priority_level*]

Example:

```
RP/0/(config-pmap-c)# priority level 1
```

Specifies priority to a class of traffic belonging to a policy map. If no priority level is configured, the default is priority 1.

Step 5 **exit**

Example:

```
RP/0/(config-pmap)# exit
```

Returns the router to policy map class configuration mode.

Step 6 **police rate** {[*units*] | **percent** *percentage*}} [**burst**] [**peak-burst** *peak-burst* [*burst-units*]] [**peak-rate** *value* [*units*]] | **percent** *percentage*]

Example:

```
RP/0/(config-pmap-c)# police rate 250
```

Configures traffic policing and enters policy map police configuration mode. In this example, the low-latency queue is restricted to 250 kbps to protect low-priority traffic from starvation and to release bandwidth.

Note
For Link Aggregation Group (LAG), only the **percent** keyword is supported.

Step 7 **exit**

Example:

```
RP/0/(config-pmap-c)# exit
```

Returns the router to policy map configuration mode.

Step 8 **exit**

Example:

```
RP/0/(config-pmap)# exit
```

Returns the router to the global configuration mode.

Step 9 **interface** *type interface-path-id*

Example:

```
RP/0/(config)# interface
```

Enters interface configuration mode, and configures an interface.

Step 10 **service-policy output** *policy-map*

Example:

```
RP/0/(config-if)# service-policy output policy1
```

Attaches a policy map to an output interface to be used as the service policy for that interface. In this example, the traffic policy evaluates all traffic leaving that interface.

Step 11 **commit**

Step 12 **show policy-map interface** *type interface-path-id* **output**

Example:

```
RP/0/# show policy-map interface tengige POS 0/2/0/0HundredGigE 0/7/0/0
```

(Optional) Displays policy configuration information for all classes configured for all service policies on the specified interface.

Running configuration for priority

```
policy-map egress_BRRpriority_POLICY
class CLASS_2_egress_BRRpriorityIPV4DSCP
  priority level 4
  police rate 8 gbps
  !
!
class CLASS_3_egress_BRRpriorityIPV4DSCP
  bandwidth remaining ratio 1
!
class CLASS_1_egress_BRRpriorityIPV4DSCP
  bandwidth remaining ratio 10
!
class class-default
!
end-policy-map
!
```

How to Mitigate Control Packet Loss during Traffic Congestion at Core Interface

During a network congestion at the core interfaces, the high priority control packets may be lost randomly. To avoid this loss, we recommend employing egress queueing. This approach involves creating an egress queing policy map that prioritizes Traffic Class (TC) 6 and 7 control packets, thus ensuring uninterrupted transmission while shaping the data traffic to a maximum of 90% of the available bandwidth.

The following sample summarizes how egress queueing prioritizes the TC6 and TC7 control packets.

```
class-map match-any TC6
 match traffic-class 6
 end-class-map
!
class-map match-any TC7
 match traffic-class 7
 end-class-map
!
policy-map out_p
 class TC6
  priority level 1
 !
 class TC7
```

```
 priority level 1
!
class class-default
 shape average percent 90
!
end-policy-map
!
```

Configuring Traffic Shaping

Traffic shaping allows you to control the traffic exiting an interface to match its transmission to the speed of the remote target interface and ensure that the traffic conforms to policies contracted for it.

Procedure

Step 1 **configure**

Step 2 **policy-map** *policy-name*

Example:

```
RP/0/(config)# policy-map policy1
```

Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy and enters the policy map configuration mode.

Step 3 **class** *class-name*

Example:

```
RP/0/(config-pmap)# class class1
```

Specifies the name of the class whose policy you want to create or change and enters the policy map class configuration mode.

Step 4 **shape average** {**percent** *value* | *rate* [*units*]} [*burst-size* [*burst-units*]]

Example:

```
RP/0/(config-pmap-c)# shape average percent 50
```

Shapes traffic to the indicated bit rate according to average rate shaping in the specified units or as a percentage of the bandwidth.

Note
For Link Aggregation Group (LAG), only the **percent** keyword is supported.

Step 5 **exit**

Example:

```
RP/0/(config-pmap-c)# exit
```

Returns the router to policy map configuration mode.

Step 6 **exit**

Example:

```
RP/0/(config-pmap)# exit
```

Returns the router to global configurationmode.

Step 7 Specifies the name of the class whose policy you want to create or change.**interface** *type interface-path-id*

Example:

```
RP/0/(config)# interface tengige POS 0/2/0/0
```

Enters interface configuration mode and configures an interface.

Step 8 **service-policy output** *policy-map*

Example:

```
RP/0/(config-if)# service-policy output policy1
```

Attaches a policy map to an output interface to be used as the service policy for that interface. In this example, the traffic policy evaluates all traffic leaving that interface.

Step 9 **commit**

Step 10 **show policy-map interface** *type interface-path-id* **output**

Example:

```
RP/0/# show policy-map interface HundredGigE 0/7/0/0
```

(Optional) Displays policy configuration information for all classes configured for all service policies on the specified interface.

Running configuration for traffic shaping

```
policy-map egress_SHAPER_POLICY
class CLASS_1_egress_SHAPERIPV4PREC
  shape average 1000 mbps
!
```