Configuring Modular QoS Congestion Management on Cisco IOS XR Software

Congestion management controls congestion after it has occurred on a network. Congestion can be managed on Cisco IOS XR software by using packet queueing methods, and by shaping the packet flow through use of traffic regulation mechanisms.

Packet queueing methods define packet scheduling or the order in which packets are dequeued to the interface for transmission on the physical wire. Furthermore, queueing methods support minimum bandwidth guarantees and low latencies based on the order and number of times that a queue is serviced.

The following types of queueing methods and traffic regulation mechanisms are supported on the Cisco IOS XR software:

- Modified Deficit Round Robin (MDRR)
- Low-latency queueing (LLQ) with strict priority queueing (PQ)
- Traffic shaping
- Traffic policing

Feature History for Configuring Modular QoS Congestion Management on Cisco IOS XR Software

Release	Modification
Release 3.2	The Congestion Avoidance feature was introduced with the exception of the following: • Two-rate policer and two-token bucket algorithm • The **pir** and **violate-action** keywords for the **police** command • Packet-by-packet MDRR scheduling mechanism
Release 3.3.0	The **police** command was changed to the **police rate** command and the syntax changed.
Release 3.4.0	The **police rate** command enters policy map police configuration mode to configure the conform, exceed and violate actions. The following new commands were added: **conform-action**, **exceed-action** and **violate-action**. The **cos**, **qos-group**, **atm-clp**, and **transmit** actions were added to the policer.
Release 3.5.0	The **show policy-map interface** command output was updated to show when a policy is suspended on a multilink or T3 interface.

Release 3.6.0	Increased Class scale from 32 to 1000 classes per policy map. Unallocated remaining bandwidth is equally distributed among all the queueing classes that do not have remaining bandwidth configured explicitly. For shape and police percentage parameters in child policy, reference is relative to the maximum rate of the parent. For bandwidth percentage parameters in child policy, reference is relative to the minimum bandwidth of the parent class. If bandwidth is not configured in parent class, guaranteed service rate of parent class is used as reference.
Release 3.7.0	The Multi-Action Set/Policer feature was introduced.
Release 3.8.0	The Frame Relay QoS on Layer 2 VPN was introduced. The Policer Granularity feature was introduced.
Release 3.9.0	The granularity of the rates specified in the **bandwidth**, police-rate, and **shape average** commands was changed from 64 kbps to 8 kbps.

Contents

Prerequisites for Configuring QoS Congestion Management on Cisco IOS XR Software

The following 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 QoS Congestion Management on Cisco IOS XR Software

To implement QoS congestion management features in this document, you must understand the following concepts:

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 in Cisco IOS XR software 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 queueing 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 queueing 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. See the *Configuring Modular QoS Congestion Management on Cisco IOS XR Software* module for information about the traffic descriptor.

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 queueing 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.

Modified Deficit Round Robin

MDRR is a class-based composite scheduling mechanism that allows for queueing of up to eight traffic classes. It operates in the same manner as class-based weighted fair queueing (CBWFQ) and allows definition of traffic classes based on customer match criteria (such as access lists); however, MDRR does not use the weighted fair queueing algorithm.

When MDRR is configured in the queueing strategy, nonempty queues are served one after the other. Each time a queue is served, a fixed amount of data is dequeued. The algorithm then services the next queue. When a queue is served, MDDR keeps track of the number of bytes of data that were dequeued in excess of the configured value. In the next pass, when the queue is served again, less data is dequeued to compensate for the excess data that was served previously. As a result, the average amount of data dequeued per queue is close to the configured value. In addition, MDRR allows for a strict priority queue for delay-sensitive traffic.

Each queue within MDRR is defined by two variables:

- Quantum value—Average number of bytes served in each round.
- Deficit counter—Number of bytes a queue has sent in each round. The counter is initialized to the quantum value.

Packets in a queue are served as long as the deficit counter is greater than zero. Each packet served decreases the deficit counter by a value equal to its length in bytes. A queue can no longer be served after the deficit counter becomes zero or negative. In each new round, the deficit counter for each nonempty queue is incremented by its quantum value.

Note In general, the quantum size for a queue should not be smaller than the maximum transmission unit (MTU) of the interface to ensure that the scheduler always serves at least one packet from each nonempty queue.

Low-Latency Queueing with Strict Priority Queueing

The LLQ feature brings strict priority queueing (PQ) to the MDRR scheduling mechanism. PQ in strict priority mode ensures that one type of traffic is sent, possibly at the expense of all others. For PQ, a low-priority queue can be detrimentally affected, and, in the worst case, never allowed to send its packets if a limited amount of bandwidth is available or the transmission rate of critical traffic is high.

Strict PQ allows delay-sensitive data, such as voice, to be dequeued and sent before packets in other queues are dequeued.

LLQ enables the use of a single, strict priority queue within MDRR 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 access lists, protocols, IP precedence, and IP differentiated service code point (DSCP) values. Moreover, within an access list you can specify that traffic matches are allowed based on the DSCP value that is set using the first six bits of the IP type of service (ToS) byte in the IP header.

Qos-Group-Based Queuing

QoS-group based queuing is supported in the ingress and egress directions.

The local output queue selection for packets sent to the fabric on the ingress card is based on a fabric modular QoS CLI (MQC) policy. The match criteria for the policy is the MPLS Exp bits of the topmost label. For information fabric QoS, see the *Configuring Fabric Quality of Service Policies and Classes on Cisco IOS XR Software* module.

In the egress direction, the MQC policy attached to the Layer 2 ATM sub-interface is used to select and configure the segmentation and reassembly (SAR) queue. All the traffic, from different ATM virtual circuits (VCs), is directed to the default output queue in the buffer management ASIC (BMA). To differentiate between different classes of service, an MQC policy is attached to the main interface to select the output queues (including the low latency queuing (LLQ)) on the Layer 2 interfaces.

The QoS Group setting in the ingress direction is preserved across the fabric and can be used to choose the output queue in the egress direction. This functionality helps in achieving egress shaping based on the MPLS Exp values in the disposition router. Additionally, the discard class setting preserved across the fabric can be used to perform WRED on the output queues.

The MQC policy on the main interface co-exists with an MQC policy on the sub-interface, however, the two policies are independent of one another, which means the match criteria of one policy cannot be dependent on the set actions of another. This is applicable only to the virtual circuit (VC) and virtual path (VP) modes, since, the main interface is used for Layer 2 in the port mode.

The following MQC actions are supported on the egress MQC policy used to select the output queue:

- MDRR using the **bandwidth** and **bandwidth percent** command
- the **shaping** and **shaping percent** command
- the **priority** command for LLQ selection
- the **random-detect** command based on the *discard class*
- the **queue-limit** command
- policing (engine-based)

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.

Traffic Shaping for Frame Relay on Layer 2 VPN

The shaping of all packets of the circuit is allowed in the disposition path. Shaping is configured in the imposition path by configuring bandwidth, queue-limit parameters, and WRED parameters.

The Layer 2 circuit acts like a subinterface for supporting QoS, which means that when a policy with queuing actions is attached to a Layer 2 circuit, it is assigned its own port and queues.

Note The matching criteria are **qos-group** and **discard-class** in the disposition path.

Traffic Shaping for ATM on Layer 2 VPN

The **shape** command under the PVC submode is applicable to the attachment circuits (AC) in the virtual circuit (VC) mode and the virtual path (VP) mode.

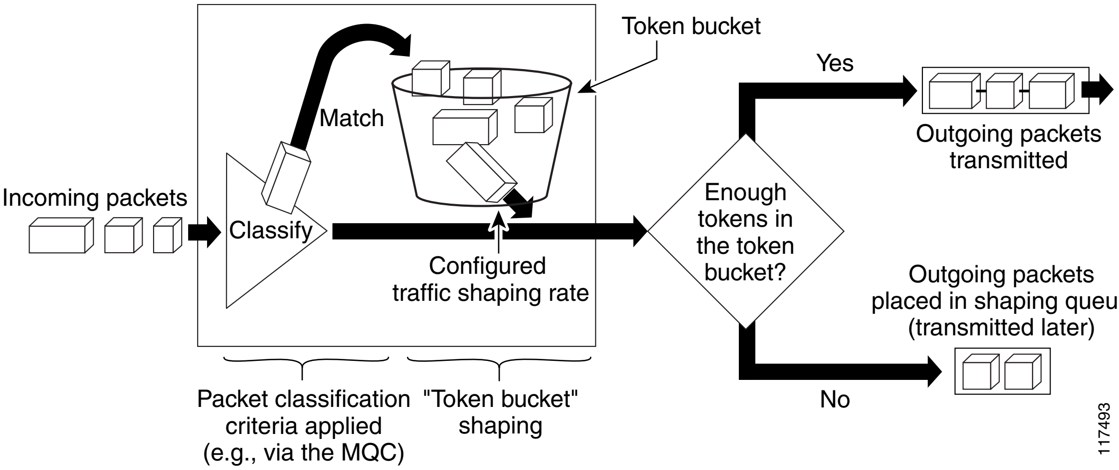
Note The default shape is UBR at line rate.

Traffic-Shaping Mechanism Regulates Traffic

When incoming packets arrive at an interface, the packets are classified using a classification technique, such as an access control list (ACL) or 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.

Figure 1 illustrates how a traffic shaping mechanism regulates traffic flow.

Figure 1 How a Traffic Shaping Mechanism Regulates Traffic


Token bucket
Match
Yes
Outgoing packets transmitted
Incoming packets
Classify
Enough tokens in the token bucket?
Configured traffic shaping rate
Outgoing packets placed in shaping queue (transmitted later)
Packet classification criteria applied (e.g., via the MQC)
"Token bucket" shaping
No
117493

Packets matching the specified criteria are placed in the token bucket. The maximum size of the token bucket is the confirm burst (Bc) size plus the Be size. The token bucket is filled at a constant rate of Bc worth of tokens at every Tc. This is the configured traffic shaping rate.

If the traffic shaping mechanism is active (that is, packets exceeding the configured traffic shaping rate already exist in a transmission queue) at every Tc, the traffic shaper checks to see if the transmission queue contains enough packets to send (that is, up to either Bc [or Bc plus Be] worth of traffic).

If the traffic shaper is not active (that is, there are no packets exceeding the configured traffic shaping rate in the transmission queue), the traffic shaper checks the number of tokens in the token bucket. One of the following occurs:

- If there are enough tokens in the token bucket, the packet is sent (transmitted).
- If there are not enough tokens in the token bucket, the packet is placed in a shaping queue for transmission at a later time.

Traffic Policing

In general, traffic policing allows you to control the maximum rate of traffic sent or received on an interface, and to partition a network into multiple priority levels or class of service (CoS).

Traffic policing manages the maximum rate of traffic through a token bucket algorithm. The token bucket algorithm can use the user-configured values to determine the maximum rate of traffic allowed on an interface at a given moment in time. The token bucket algorithm is affected by all traffic entering or leaving (depending on where the traffic policy with traffic policing is configured) and is useful in managing network bandwidth in cases in which several large packets are sent in the same traffic stream.

Traffic entering the interface with traffic policing configured is placed into one of these categories. Within these three categories, users can decide packet treatments. For instance, packets that conform can be configured to be sent, packets that exceed can be configured to be sent with a decreased priority, and packets that violate can be configured to be dropped.

Traffic policing is often configured on interfaces at the edge of a network to limit the rate of traffic entering or leaving the network. In the most common traffic policing configurations, traffic that conforms is sent and traffic that exceeds is sent with a decreased priority or is dropped. Users can change these configuration options to suit their network needs.

Note Configured values take into account the Layer 2 encapsulation applied to traffic. This applies to both ingress and egress policing. For POS/SDH transmission, the encapsulation is considered to be 4 bytes. For Ethernet, the encapsulation is 14 bytes; whereas for 802.1Q, the encapsulation is 18 bytes.

Traffic policing also provides a certain amount of bandwidth management by allowing you to set the burst size (Bc) for the committed information rate (CIR). When the peak information rate (PIR) is supported, a second token bucket is enforced and the traffic policer is then called a two-rate policer.

Note The two-rate policer and two-token bucket algorithm is supported in Cisco IOS XR software.

For Cisco IOS XR software, a single-rate, two-color policer is supported that provides one token bucket with two actions for each packet: a conform action and an exceed action.

Multiple Action Set

The Multiple Action Set feature allows you to mark packets with multiple action sets (conditional and unconditional) through a class map.

To support multiple action sets, the following combinations are supported of conform and exceed actions:

- set-qos-group, set-discard-class
- set-mpls-exp-imp, set-qos-group
- set-mpls-exp-imp, set-discard-class
- set-mpls-exp-imp, set-qos-group and set-discard-class
- set-fr-de, set-prec-tunnel
- set-fr-de, set-dscp-tunnel
- set-fr-de, set-mpls-exp-imp
- set-mpls-exp-imp, set-prec-tunnel
- set-mpls-exp-imp, set-dscp-tunnel
- set-mpls-exp-imp, set-clp

Note If partial multiple set actions are used, hierarchical policing is not supported.

Table 1 shows a summary of marking combinations that are supported on conditional ingress policer marking and unconditional ingress QoS marking.

Table 1 Marking Combinations Supported on Conditional Ingress Policer and Unconditional Ingress QoS Marking

Marking Combination	Layer 2 Interfaces	Layer 3 Interfaces
set qos-group + set discard-class	Y	N
set mpls exp imposition + set qos-group	Y	Y
set mpls exp imposition + set discard-class	Y	Y

Table 1 Marking Combinations Supported on Conditional Ingress Policer and Unconditional Ingress QoS Marking

Marking Combination	Layer 2 Interfaces	Layer 3 Interfaces
set mpls exp imposition + set precedence tunnel	N	Y
set mpls exp imposition + set dscp tunnel	N	Y
set mpls exp imposition + set qos-group + set discard-class	Y	Y
set clp + set mpls exp imposition	Y	N
set frame-relay DE + set mpls exp imposition	Y	N
set frame-relay DE + set precedence tunnel	Y	N
set frame-relay DE + set DSCP tunnel	Y	N

Packet Marking Through the IP Precedence Value, IP DSCP Value, and the MPLS Experimental Value Setting

In addition to rate-limiting, traffic policing allows you to independently mark (or classify) the packet according to whether the packet conforms or violates a specified rate. Packet marking also allows you to partition your network into multiple priority levels or CoS. Packet marking as a policer action is conditional marking.

Use the traffic policer to set the IP precedence value, IP DSCP value, or Multiprotocol Label Switching (MPLS) experimental value for packets that enter the network. Then networking devices within your network can use this setting to determine how the traffic should be treated. For example, the Weighted Random Early Detection (WRED) feature uses the IP precedence value to determine the probability that a packet is dropped.

If you want to mark traffic but do not want to use traffic policing, see the "Class-based, Unconditional Packet Marking Examples" section to learn how to perform packet classification.

Note Marking IP fields on an MPLS-enabled interface results in non-operation on that particular interface.

Traffic Policing for Frame Relay on L2VPN

Policing each circuit individually in the ingress and egress path is allowed. The policer can be either one-rate, two-color (1R2C) or two-rate, three-color (2R3C) color-blind only. The policer action for conforming or exceeding packets can be one of the following:

- **transmit**
- **drop**
- **set mpls exp imposition/topmost <exp>** (AToM only)
- **set tunnel prec/dscp** (L2TPv3 only)
- **set qos-group** *qos-group*
- **set discard-class** *discard-class*
- **set fr-de and set mpls exp** (AToM only)
- **set fr-de**
- **set tunnel prec/dscp** (L2TPv3 only)

Note For a 2R3C policer, the action for violating packets is **drop**.

Note The matching criteria is based on the incoming Frame Relay discard eligibility (DE) bit in the Frame Relay header in the imposition path.

Traffic Policing on Layer 2 ATM Interfaces

Traffic policing is supported on the Layer 2 ATM interfaces in the ingress imposition path. The OAM cells are policed along with the user cells unless the QoS policy is explicitly configured to exclude the OAM cells from being policed.

Note Policing is supported for the virtual circuit (VC), and the virtual path (VP) modes. However, policing is not supported for the port mode on the Layer 2 ATM interfaces.

Different match criteria can be used in the policy map with class-default matching all the traffic including the OAM cells.

Policing is performed on the ATM Adaptation Layer type 0 (AAL0) cells but translates to ATM Adaptation Layer type 5 (AAL5) packets as described below:

- AAL5 packet conforms, if all the cells in the packet conform to peak cell rate (PCR) and sustainable cell rate (SCR) buckets.
- AAL5 packet exceeds, if at least one cell does not conform to the SCR bucket.
- AAL5 packet violates, if at least one cell does not conform to the PCR bucket.

The following policer options are supported:

- Rate in cells per second, and percent
- Peak rate in cells per second, and percent
- Delay tolerance in microseconds
- Maximum burst size in cells

The following policer actions are supported on the Layer 2 ATM interfaces in the ingress direction:

- **transmit**
- **drop**
- **set mpls exp imposition** <*exp*> (AToM only)
- **set qos-group** <*qos-group*> (AToM and local switching)
- **set discard-class** <*discard-class*> (AToM and local switching)
- **set atm-clp** (Exceed action only, AToM and local switching)
- **drop** (Violate action)

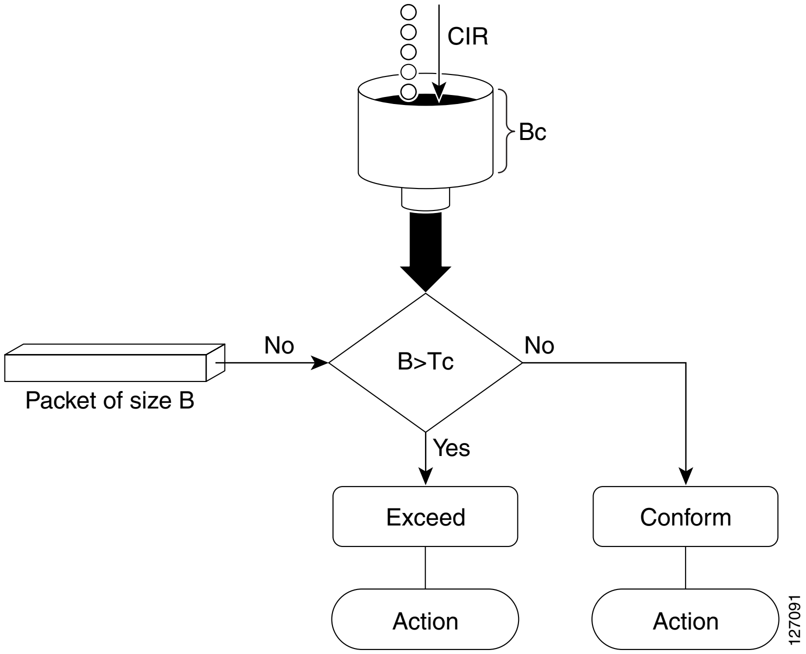
Multiple policing action is supported on the Layer 2 ATM interfaces using the following combination:

- **set mpls exp imposition** and **set atm-clp**.

Regulation of Traffic with the Policing Mechanism

Figure 2 illustrates how a single-rate token bucket policer marks packets as either conforming or exceeding a CIR.

Figure 2 How a Traffic Policing Mechanism Regulates Traffic



The time interval between token updates (Tc) to the token bucket is updated at the CIR value each time a packet arrives at the traffic policer. The Tc token bucket can contain up to the Bc value. If a packet of size B is greater than the Tc token bucket, then the packet exceeds the CIR value and a configured action is performed. If a packet of size B is less than the Tc token bucket, then the packet conforms and a different configured action is performed.

Traffic Shaping Versus Traffic Policing

Although traffic shaping and traffic policing can be implemented together on the same network, there are distinct differences between them, as shown in Table 2.

Table 2 Differences Between Traffic Shaping and Traffic Policing

	Traffic Shaping	Traffic Policing
Triggering Event	• Occurs automatically at regular intervals (Tc). or Occurs whenever a packet arrives at an interface.	• Occurs whenever a packet arrives at an interface.
What it Does	• Classifies packets. • If a packet does not meet match criteria, no further action is taken. • Packets meeting match criteria are sent (if there are enough tokens in the token bucket) or Packets are placed in a queue for transmission later. • If the number of packets in the queue exceed the queue limit, the packets are dropped.	• Classifies packets. • If a packet does not meet match criteria, no further action is taken. • Packets meeting match criteria and conforming to or exceeding a specified rate, receive the configured policing action (for example, drop, send, mark, then send). • Packets are not placed in a queue for transmission later.

Policer Granularity

Table 3 shows the default policer granularity values.

Table 3 Policer Granularity Default Values

SPA Interface Processor	Policer Granularity Default Value
Cisco 12000 SIP-401	64 kbps
Cisco 12000 SIP-501	64 kbps
Cisco 12000 SIP-601	64 kbps

The Policer Granularity feature allows you to override the default policer granularity values.

The police rate you set should be a multiple of the policer granularity. For example, if the police rate is set to 72 kbps but the default policer granularity is 64 kbps, the effective police rate is 64 kbps. To get an actual police rate of 72 kbps, configure the policer granularity to 8 kbps. Because 72 is a multiple of 8, the police rate will be exactly 72 kbps.

Policer granularity values, whether default or configured, apply to the SPA Interface Processor (SIP) and to all shared port adapters (SPAs) that are installed in the SIP.

How to Configure QoS Congestion Management on Cisco IOS XR Software

This section contains instructions for the following tasks:

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. MDRR is implemented as the scheduling algorithm.

The **bandwidth remaining** command specifies a weight for the class to the MDRR. The MDRR algorithm derives the weight for each class from the bandwidth remaining value allocated to 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 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.

SUMMARY STEPS

1. **configure**
2. **policy-map** *policy-name*
3. **class** *class-name*
4. **bandwidth** {*rate* [*units*] | **percent** *value*}
5. **bandwidth remaining percent** *value*
6. **exit**
7. **class** *class-name*
8. **bandwidth** {*rate* [*units*] | **percent** *percent-value*}
9. **bandwidth remaining percent** *value*
10. **exit**

11. **exit**
12. **interface** *type interface-path-id*
13. **service-policy** {**input** | **output**} *policy-map*
14. **end**
    or
    commit
15. **show policy-map interface** *type interface-path-id* [**input** | **output**]

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`configure`**
	Example:
	`RP/0/0/CPU0:router# configure`
		Enters global configuration mode.

Step 2	**`policy-map`** *`policy-name`*
	Example:
	`RP/0/0/CPU0:router(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/RP0/CPU0:router(config-pmap)# class class1`
		Specifies the name of the class whose policy you want to create or change.

Step 4	**`bandwidth`** `{`*`rate`* `[`*`units`*`]|` **`percent`** *`value`*`}`
	Example:
	`RP/0/0/CPU0:router(config-pmap-c)# bandwidth percent 50`
		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 50 percent of the interface bandwidth.

Step 5	**`bandwidth remaining percent`** *`value`*
	Example:
	`RP/0/0/CPU0:router(config-pmap-c)# bandwidth remaining percent 20`
		Specifies how to allocate leftover bandwidth to various classes.
		• The remaining bandwidth of 40 percent is shared by class class1 and class2 (see Steps 8 and 9) in a 20:80 ratio: class class1 receives 20 percent of the 40 percent, and class class2 receives 80 percent of the 40 percent.

Step 6	**`exit`**
	Example:
	`RP/0/0/CPU0:router(config-pmap-c)# exit`
		Returns the router to policy map configuration mode.

Step 7	**`class`** *`class-name`*
	Example:
	`RP/0/0/CPU0:router(config-pmap)# class class2`
		Specifies the name of a different class whose policy you want to create or change.

	Command or Action	Purpose

Step 8	**bandwidth** {*rate* [*units*] | **percent** *value*}
	Example:
	`RP/0/0/CPU0:router(config-pmap-c)# bandwidth percent 10`
		Specifies the bandwidth allocated for a class belonging to a policy map.
		• In this example, class class2 is guaranteed 10 percent of the interface bandwidth.

Step 9	**bandwidth remaining percent** *value* **Example:** `RP/0/0/CPU0:router(config-pmap-c)# bandwidth remaining percent 80`	Specifies how to allocate leftover bandwidth to various classes. • The remaining bandwidth of 40 percent is shared by class class1 (see Steps 4 and 5) and class2 in a 20:80 ratio: class class1 receives 20 percent of the 40 percent, and class class2 receives 80 percent of the 40 percent.
Step 10	**exit** **Example:** `RP/0/0/CPU0:router(config-pmap-c)# exit`	Returns the router to policy map configuration mode.
Step 11	**exit** **Example:** `RP/0/0/CPU0:router(config-pmap)# exit`	Returns the router to global configuration mode.
Step 12	**interface** *type interface-path-id* **Example:** `RP/0/0/CPU0:router(config)# interface POS 0/2/0/0`	Enters interface configuration mode and configures an interface.
Step 13	**service-policy** {**input** | **output**} *policy-map* **Example:** `RP/0/0/CPU0:router(config-if)# service-policy output policy1`	Attaches a policy map to an input or output interface to be used as the service policy for that interface. • In this example, the traffic policy evaluates all traffic leaving that interface.

	Command or Action	Purpose

Step 14	**end**
	or
	commit
	Example:
	`RP/0/0/CPU0:router(config-if)# end`
	or
	`RP/0/0/CPU0:router(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 15	**show policy-map interface** *type interface-path-id* [**input** | **output**] **Example:** `RP/0/0/CPU0:router# show policy-map interface POS 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 low-latency queueing (LLQ), providing strict priority queueing (PQ). Strict PQ 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, we recommend that you configure a policer to limit the priority traffic. This configuration ensures that the priority traffic does not starve all of the other traffic on the line card, which protects low priority traffic from starvation. Use the **police** command to explicitly configure the policer.

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 **bandwidth**, **priority**, and **shape average** commands should not be configured together in the same class.

SUMMARY STEPS

1. **configure**
2. **policy-map** *policy-name*

3. **class** *class-name*
4. **police rate** {*rate* [*units*] | **percent** *percentage*}} [**burst** *burst-size* [*burst-units*]] [**peak-burst** *peak-burst* [*burst-units*]] [**peak-rate** *value* [*units*] | **percent** *percentage*]
5. **exceed-action** *action*
6. **priority**
7. **exit**
8. **exit**
9. **interface** *type interface-path-id*
10. **service-policy** {**input** | **output**} *policy-map*
11. **end**
    or
    commit
12. **show policy-map interface** *type interface-path-id* [**input** | **output**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**policy-map** *policy-name* **Example:** `RP/0/0/CPU0:router(config)# policy-map voice`	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/0/0/CPU0:router(config-pmap)# class voice`
		Enters policy map class configuration mode.
		• Specifies the name of the class whose policy you want to create or change.

Step 4	**police rate** {*rate* [*units*] | **percent** *percentage*}} [**burst** *burst-size* [*burst-units*]] [**peak-burst** *peak-burst* [*burst-units*]] [**peak-rate** *value* [*units*]] | **percent** *percentage*] **Example:** `RP/0/0/CPU0:router(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.
Step 5	**exceed-action** *action* **Example:** `RP/0/0/CPU0:router(config-pmap-c-police)# exceed-action drop`	Configures the action to take on packets that exceed the rate limit.

	Command or Action	Purpose

Step 6	**`priority`**
	Example:
	`RP/0/0/CPU0:router(config-pmap-c)# priority`
		Specifies priority to a class of traffic belonging to a policy map.

Step 7	**`exit`**
	Example:
	`RP/0/0/CPU0:router(config-pmap-c)# exit`
		Returns the router to policy map configuration mode.

Step 8	**`exit`**
	Example:
	`RP/0/0/CPU0:router(config-pmap)# exit`
		Returns the router to global configuration mode.

Step 9	**`interface`** *`type interface-path-id`*
	Example:
	`RP/0/0/CPU0:router(config)# interface POS 0/2/0/0`
		Enters interface configuration mode, and configures an interface.

Step 10	**`service-policy`** {**`input`** | **`output`**} *`policy-map`*
	Example:
	`RP/0/0/CPU0:router(config-if)# service-policy output policy1`
		Attaches a policy map to an input or 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	**`end`**
	or
	`commit`
	Example:
	`RP/0/0/CPU0:router(config-if)# end`
	or
	`RP/0/0/CPU0:router(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 12	**`show policy-map interface`** *`type interface-path-id`* [**`input`** | **`output`**] **Example:** `RP/0/0/CPU0:router# show policy-map interface POS 0/2/0/0`	(Optional) Displays policy configuration information for all classes configured for all service policies on the specified interface.

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.

Shaping on both incoming and outgoing interfaces is done at the Layer 3 level. Traffic shaping can be configured on Layer 2 or Layer 3 QoS packet size - it is configurable.

- On 10 Gbps IP Services Engine (Engine 5) line cards, the default QoS packet size is Layer 2. To change the QoS packet size to Layer 3, use the service-policy account nolayer2 command.
- On 2.5 Gbps IP Services Engine (Engine3) line cards, the default QoS packet size is Layer 3. QoS packet size can be changed only on outgoing interfaces of the 4-Port Gigabit Ethernet ISE line card. To change the QoS packet size, use the hw-module qos account layer2 encapsulation command.

Restrictions

The **bandwidth**, **priority**, and **shape average** commands should not be configured together in the same class.

SUMMARY STEPS

1. **configure**
2. **policy-map** *policy-name*
3. **class** *class-name*
4. **shape average** {**percent** *value* | *rate* [*units*]} [*burst-size* [*burst-units*]]
5. **exit**
6. **exit**
7. **interface** *type interface-path-id*
8. **service-policy** {**input** | **output**} *policy-map*
9. **end**
   or
   commit
10. **show policy-map interface** *type interface-path-id* [**input** | **output**]

DETAILED STEPS

	Command or Action	Purpose

Step 1	`configure`
	Example:
	`RP/0/0/CPU0:router# configure`
		Enters global configuration mode.

Step 2	`policy-map` *`policy-name`*
	Example:
	`RP/0/0/CPU0:router(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.

	Command or Action	Purpose

Step 3	**`class`** *`class-name`*
	Example:
	`RP/0/0/CPU0:router(config-pmap)# class class1`
		Enters policy map class configuration mode.
		• Specifies the name of the class whose policy you want to create or change.

Step 4	**`shape average`** `{`**`percent`** *`value`* `|` *`rate`* `[`*`units`*`]}` `[`*`burst-size`* `[`*`burst-units`*`]]`
	Example:
	`RP/0/0/CPU0:router(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.

Step 5	**`exit`**
	Example:
	`RP/0/0/CPU0:router(config-pmap-c)# exit`
		Returns the router to policy map configuration mode.

Step 6	**`exit`**
	Example:
	`RP/0/0/CPU0:router(config-pmap)# exit`
		Returns the router to global configuration mode.

Step 7	**`interface`** *`type interface-path-id`*
	Example:
	`RP/0/0/CPU0:router(config)# interface POS 0/2/0/0`
		Enters interface configuration mode and configures an interface.

Step 8	**`service-policy`** `{`**`input`** `|` **`output`**`}` *`policy-map`*
	`Example:`
	`RP/0/0/CPU0:router(config-if)# service-policy output policy1`
		Attaches a policy map to an input or 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	**`end`**
	or
	`commit`
	Example:
	`RP/0/0/CPU0:router(config-if)# end`
	or
	`RP/0/0/CPU0:router(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.

	Command or Action	Purpose
Step 10	`show policy-map interface` *type interface-path-id* [**input** | **output**] **Example:** `RP/0/0/CPU0:router# show policy-map interface POS 0/2/0/0`	(Optional) Displays policy configuration information for all classes configured for all service policies on the specified interface.

Configuring Traffic Policing

Traffic policing allows you to control the maximum rate of traffic sent or received on an interface.

Restrictions

set cos is not allowed as an ingress policer action.

SUMMARY STEPS

1. **configure**
2. **policy-map** *policy-name*
3. **class** *class-name*
4. **police rate** {*rate* [*units*] | **percent** *percentage*}} [**burst** *burst-size* [*burst-units*]] [**peak-burst** *peak-burst* [*burst-units*]] [**peak-rate** *value* [*units*] | **percent** *percentage*]
5. **conform-action** *action*
6. **exceed-action** *action*
7. **exit**
8. **exit**
9. **exit**
10. **interface** *type interface-path-id*
11. **service-policy** {**input** | **output**} *policy-map*
12. **end**
    or
    commit
13. **show policy-map interface** *type interface-path-id* [**input** | **output**]

Note The multi-action set/policer feature allows you to configure multiple conform and exceed actions. Hence, you can repeat the **conform-action** and **exceed-action** commands multiple times in your configuration.

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`configure`**
	Example:
	`RP/0/0/CPU0:router# configure`
		Enters global configuration mode.

Step 2	**`policy-map`** *`policy-name`*
	Example:
	`RP/0/0/CPU0:router(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/0/0/CPU0:router(config-pmap)# class class1`
		Enters policy map class configuration mode.
		• Specifies the name of the class whose policy you want to create or change.

Step 4	**`police rate`** `{`*`rate`* `[`*`units`*`]` | **`percent`** *`percentage`*`}` `[`**`burst`** *`burst-size`* `[burst-`*`units`*`]]` `[`**`peak-burst`** *`peak-burst`* `[burst-`*`units`*`]]` `[`**`peak-rate`** *`value`* `[`*`units`*`]` | **`percent`** *`percentage`*`]`
	Example:
	`RP/0/0/CPU0:router(config-pmap-c)# police rate 250000`
		Configures traffic policing and enters policy map police configuration mode. The traffic policing feature works with a token bucket algorithm.

Step 5	**`conform-action`** *`action`*
	Example:
	`RP/0/0/CPU0:router(config-pmap-c-police)# conform-action set mpls experimental topmost 3`
		Configures the action to take on packets that conform to the rate limit. The *action* argument is specified by one of the following keywords:
		• **drop**—Drops the packet.
		• **set**—Has the following keywords and arguments:
		– **atm-clp** *value*—Sets the cell loss priority (CLP) bit.
		– **cos** *value*—Sets the class of service value. Range is 0 to 7.
		– **discard-class** *value*—Sets the discard class on IP Version 4 (IPv4) or Multiprotocol Label Switching (MPLS) packets. Range is 0 to 7.
		– **dscp** [**tunnel**] *value*—Sets the differentiated services code point (DSCP) value and sends the packet.
		– **mpls experimental** {**topmost** | **imposition**} *value*—Sets the experimental (EXP) value of the Multiprotocol Label Switching (MPLS) packet topmost label or imposed label. Range is 0 to 7.
		– **precedence** [**tunnel**] *precedence*—Sets the IP precedence and sends the packet.
		• **transmit**—Transmits the packets.

	Command or Action	Purpose

Step 6	**`exceed-action`** *`action`*
	Example:
	`RP/0/0/CPU0:router(config-pmap-c-police)# exceed-action set mpls experimental topmost 4`
		Configures the action to take on packets that exceed the rate limit. The *action* argument is specified by one of the keywords specified in Step 5.

Step 7	**`exit`**
	Example:
	`RP/0/0/CPU0:router(config-pmap-c-police)# exit`
		Returns the router to policy map class configuration mode.

Step 8	**`exit`**
	Example:
	`RP/0/0/CPU0:router(config-pmap-c)# exit`
		Returns the router to policy map configuration mode.

Step 9	**`exit`**
	Example:
	`RP/0/0/CPU0:router(config-pmap)# exit`
		Returns the router to global configuration mode.

Step 10	**`interface`** *`type interface-path-id`*
	Example:
	`RP/0/0/CPU0:router(config)# interface pos 0/5/0/0`
		Enters configuration mode and configures an interface.

Step 11	**`service-policy`** `{`**`input`** `|` **`output`**`}` *`policy-map`*
	`Example:`
	`RP/0/0/CPU0:router(config-if)# service-policy output policy1`
		Attaches a policy map to an input or 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 12	**`end`**
	or
	`commit`
	Example:
	`RP/0/0/CPU0:router(config-if)# end`
	or
	`RP/0/0/CPU0:router(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.

	Command or Action	Purpose
Step 13	**show policy-map interface** *type interface-path-id* [**input** | **output**] **Example:** `RP/0/0/CPU0:router# show policy-map interface POS 0/2/0/0`	(Optional) Displays policy configuration information for all classes configured for all service policies on the specified interface.

Configuring Policer Granularity

Use the Policer Granularity feature to override the default policer granularity value so that the police rate you specify is a multiple of the policer granularity.

Restrictions

The Policer Granularity feature has the following limitations:

- Supported on Cisco 12000 SIP-401, Cisco 12000 SIP-501, and Cisco 12000 SIP-601.
- Policer granularity values apply to the SIP and to all SPAs that are installed on the SIP.
- If there are policies configured on the SIP, the SIP must be reloaded for configured policer granularity changes to take effect. (If there are no policies configured on the SIP, a reload is not required.)
- Effective police rate is a multiple of the policer granularity.

SUMMARY STEPS

1. **configure**
2. **hw-module qos pol-gran** *granularity* **location** *location*
3. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**hw-module qos pol-gran** *granularity* **location** *location* **Example:** `RP/0/0/CPU0:router(config)# hw-module qos pol-gran 8 location 0/4/CPU0`	Overrides the default policer granularity for the SIP and sets it to 8 kbps. The effective police rate will be a multiple of 8. The range of granularity values is 8 kbps to 64 kbps.

	Command or Action	Purpose

Step 3	**`end`**
	or
	`commit`
	
	Example:
	`RP/0/0/CPU0:router(config-pmap-c-police)# end`
	or
	`RP/0/0/CPU0:router(config-pmap-c-police)# 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.

Configuration Examples for Configuring QoS Congestion Management on Cisco IOS XR Software

This section provides the following configuration examples:

- Traffic Shaping for an Input Interface: Example, page 91
- Traffic Policing for a Bundled Interface: Example, page 92
- Traffic Policing for an IPSec Interface: Example, page 93
- Policer Granularity: Example, page 93
- ATM QoS on Layer 2 VPN: Examples, page 93
- Multiple Action Set: Examples, page 95

Traffic Shaping for an Input Interface: Example

The following example shows how to configure a policy map on an input interface:

```
policy-map p2
 class voip
  shape average percent 20
 !
 class class-default
 !
 end-policy-map
!
```

The following example shows the display output for the previous policy map configuration:

```
RP/0/1/CPU0:router#show policy-map interface pos 0/9/0/0
Thu Mar 25 20:37:41.743 UTC

POS0/9/0/0 input: p2

Class voip
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :                   0/0                    0
    Transmitted         :                   0/0                    0
    Total Dropped       :                   0/0                    0
  Queueing statistics
    Queue ID                             : 0
    High watermark  (Unknown)            : 0
    Inst-queue-len  (packets)            : 0
    Avg-queue-len   (packets)            : 0
    Taildropped(packets/bytes)           : 0/0
Class class-default
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :                   0/0                    0
    Transmitted         :                   0/0                    0
    Total Dropped       :                   0/0                    0
```

Traffic Policing for a Bundled Interface: Example

The following example shows how to configure a policy map for a bundled interface. Note that for bundled interfaces, policing can be configured only as a percentage and not a specific rate per second:

```
policy-map p2
 class voip
  police rate percent 20
  !
 !
 class class-default
 !
 end-policy-map
!
```

The following example shows the display output for the successful policy map configuration in which policing was configuredas a percentage:

```
RP/0/8/CPU0:E5P-PE1#show policy-map interface bundle-pos 1
Thu Mar 25 21:29:40.107 PST

Bundle-POS1 input: p2

Class voip
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :                   0/0                    0
    Transmitted         :                   0/0                    0
    Total Dropped       :                   0/0                    0
  Policing statistics                (packets/bytes)     (rate - kbps)
    Policed(conform)    :                   0/0                    0
    Policed(exceed)     :                   0/0                    0
    Policed(violate)    :                   0/0                    0
    Policed and dropped :                   0/0
Class class-default
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :                   0/0                    0
    Transmitted         :                   0/0                    0
    Total Dropped       :                   0/0                    0
```

Traffic Policing for an IPSec Interface: Example

The following example shows how to configure traffic policing on an IPSec interface:

```
policy-map policy2
  class dscp4
    police rate 64000 conform transmit exceed drop
    priority
  class dscp1
    police rate 128000 conform transmit exceed drop

interface service-ipsec 2
   service-policy input pre-decrypt policy2
```

Refer to *Implementing IPSec Network Security on Cisco IOS XR Software* in the *Cisco IOS XR System Security Configuration Guide for the Cisco XR 12000 Series Router*for more information on configuring IPSec.

Policer Granularity: Example

The police rate you set should be a multiple of the policer granularity. For example, if the police rate is set to 72 kbps but the default policer granularity is 64 kbps, the effective police rate is 64 kbps. To get an actual police rate of 72 kbps, configure the policer granularity to 8 kbps. Because 72 is a multiple of 8, the police rate will be exactly 72 kbps.

This example shows how to set the policer granularity to 8 kbps:

```
hw-module qos pol-gran 8 location 0/1/CPU0
```

Use the **show qos pol-gran location** command to verify the policer granularity. The Admin value is the value configured using the **hw-module qos pol-gran location** command. The Oper value is the default policer granularity for this SIP.

```
show qos pol-gran location 0/1/CPU0

QoS Policer Granularity Rate
Admin Value:     8 kbps
Oper Value:      64 kbps
```

ATM QoS on Layer 2 VPN: Examples

The following examples shows how to configure ATM QoS on Layer 2 VPNs:

- Attaching a Service Policy to the Attachment Circuits (AC), page 93
- Configuring Policing Based on Service Type, page 94
- Configuring Dual Queue Limit, page 95

Attaching a Service Policy to the Attachment Circuits (AC)

The **service-policy** command under the PVC sub-mode is applicable to the AC in the virtual circuit (VC) mode. This command is also available for the AC in the virtual path (VP) mode and under the main interface for port mode. For the port mode, the service policy is attached in the **l2transport** sub-mode consistent with the behavior for the VC and VP modes. In non-port mode, the service policy is attached to the main interface under the interface submode.

VC mode:

```
Router(config)#interface ATM0/1/0/0.2 l2transport
Router(config-subif)#pvc 10/2
Router(config-atm-vc)#service-policy output atm_policy_o
```

VP mode:

```
Router(config)#interface ATM0/1/0/0.3 l2transport
Router(config-subif)#pvp 30
Router(config-atm-vc)#service-policy input atm_policy_i
```

Port mode:

```
Router(config)#interface ATM0/1/0/0
Router(config-subif)# l2transport
Router(config-l2-transport)#service-policy input atm_policy_i
```

Main interface (non-port mode):

```
Router(config)#interface ATM0/1/0/0
    Router(config-subif)# service-policy output atm_policy_o
```

Configuring Policing Based on Service Type

The following example shows how to configure policing based on service type:

CBR or UBR:

CBR.1 (real-time traffic) and UBR (best effort, non-real time traffic) require the peak cell rate (PCR) and delay tolerance parameters to be specified for policing. The main difference between the configurations for UBR.1 and UBR.2 traffic is that for UBR.2 traffic, the exceed action includes the **set-clp-transmit** option to tag the non-conforming cells.

```
policy-map CBR1
    class class-default
          police rate <pcr> cellsps delay-tolerance <cdvt> us
               conform-action <>
               exceed-action <>
```

The police rate is expressed in percentage.

VBR.1

VBR.1 is real-time and non-real time traffic, and requires the peak cell rate (PCR), sustainable cell rate (SCR) and delay tolerance parameters to be specified for policing. In addition, the **atm-mbs** parameter can be specified to define the burst allowed on the SCR bucket.

```
policy-map VBR1
    class class-default
          police rate <scr> cellsps atm-mbs <mbs> cells peak-rate <pcr> cellsps
delay-tolerance <cdvt> us
               conform-action <>
               exceed-action <>
```

The police rate is expressed in percentage.

VBR.2 or VBR.3

VBR.2 and VBR.3 are real-time and non-real time traffic, and require the peak cell rate (PCR), sustainable cell rate (SCR) and delay tolerance parameters to be specified for policing. In addition, the **atm-mbs** parameter can be specified to define the burst allowed on the SCR bucket.

The main difference between VBR.1 and VBR.2 or VBR.3 is that the SCR bucket is for CLP0 cells only. A hierarchical policy is defined to support this configuration:

```
policy-map child
    class atm_clp0
          police rate <scr> cellsps atm-mbs <mbs> cells
               conform-action <>
               exceed-action <>
policy-map VBR2
         class class-default
          police rate <pcr> cellsps delay-tolerance <cdvt> us
               conform-action <>
               exceed-action <>
             service-policy child
```

The police rates is expressed in percentage. The child policy can contain other set actions as well.

Exclude OAM cells

OAM cells can be excluded from being policed by configuring the classification criteria.

```
policy-map child
    class atm-oam
        set <>
    class class-default
          police rate <scr> cellsps atm-mbs <mbs> cells
               conform-action <>
               exceed-action <>

policy-map VBR2
    class class-default
          police rate <pcr> cellsps delay-tolerance <cdvt> us
               conform-action <>
               exceed-action <>
      service-policy child
```

Configuring Dual Queue Limit

Dual queue limit configuration is supported on the egress Layer 2 ATM interfaces to differentiate between CLP0 and CLP1 cells.

```
policy-map q-limit
    class class-default
        queue-limit atm clp <queue-size> {[ms|us|cells]}
        queue-limit <queue-size> {[ms|us|cells]
```

Multiple Action Set: Examples

The following examples show how to configure multiple action sets for both conditional and unconditional markings in both the ingress and egress directions:

- Conditional Policer Markings in the Ingress Direction: Example, page 96
- Unconditional Quality-of-Service Markings in the Ingress Direction: Examples, page 96
- Conditional Policer Markings in the Egress Direction: Example, page 96
- Unconditional Quality-of-Service Markings in the Egress Direction: Example, page 97

Conditional Policer Markings in the Ingress Direction: Example

The following example shows how to configure conditional policer markings in the ingress direction:

```
 policy-map p1
  class c1
   police rate percent 30 peak-rate percent 50
    conform-action set qos-group 2
    conform-action set discard-class 3
    conform-action set mpls experimental imposition 3
    exceed-action set precedence tunnel 4
    exceed-action set mpls experimental imposition 4
  !
 !
  class class-default
 !
 end-policy-map
!
```

If policy map p1 is applied as an ingress policy, the following action sets are applied:

- By using the **conform-action** command, IP packets are marked with a discard class value of 3 and the MPLS experimental value for the imposition label is set to 3.
- By using the **exceed-action** command, IP packets are marked with the precedence value of 4 and the MPLS experimental value for the imposition label is set to 4.

Unconditional Quality-of-Service Markings in the Ingress Direction: Examples

The following example shows how to configure unconditional QoS markings in the ingress direction.

Note A maximum of three set actions are allowed.

```
configure
 policy-map p4
  class c1
   set discard-class 2
   set qos-group 4
   set mpls experimental imposition 3
   !
  class class-default
  !
 end-policy-map
!
```

Conditional Policer Markings in the Egress Direction: Example

The following example shows how to configure conditional policer markings in the egress direction:

```
configure
 policy-map p3
  class c1
  police rate percent 30 peak-rate percent 50
   conform-action set precedence 2
   exceed-action set dscp 4
   !
  !
  class class-default
  !
  end-policy-map
!
```

Note Only one policer conform or exceed set is allowed in the egress direction.

Unconditional Quality-of-Service Markings in the Egress Direction: Example

The following example shows how to configure the unconditional QoS markings in the egress direction:

```
configure
 policy-map p6
  class c1
   set precedence 5
   !
  class class-default
!
 end-policy-map
!
```

If policy map p6 is applied as an egress policy, IP packets are marked with the precedence value of 5 from the set precedence command.

Additional References

The following sections provide references related to implementing QoS congestion management.

Related Documents

Related Topic	Document Title
Initial system bootup and configuration	*Cisco IOS XR Getting Started Guide for the Cisco XR 12000 Series Router*
Master command reference	*Cisco XR 12000 Series Router Master Command Listing*
QoS commands	*Cisco IOS XR Modular Quality of Service Command Reference for the Cisco XR 12000 Series Router*
User groups and task IDs	"Configuring AAA Services on Cisco IOS XR Software" module of *Cisco IOS XR System Security Configuration Guide*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport