CHAPTER 13

Configuring Quality of Service

This chapter describes the QoS features built into your ML-Series card and how to map QoS scheduling at both the system and interface levels. This chapter contains the following major sections:

Understanding ML-Series QoS

The ML-Series card incorporates QoS features to provide control over access to network bandwidth resources. This control enables providers to implement priorities specified in Service Level Agreements (SLAs) and offers tools to enable traffic engineering.

The ML-Series QoS provides the ability to classify each packet in the network based on its interface of arrival, bridge group, class of service (CoS), IP precedence, and IP differentiated services code points. When classified, further QoS functions can be applied to each packet as it traverses the network.

Policing is also provided by the ML-Series card to ensure that no attached equipment submits more than a pre-defined amount of bandwidth into the network. This feature limits the bandwidth available to a customer, and provides a mechanism to support traffic engineering.

Priority marking allows Ethernet IEEE 802.1P CoS bits to be marked,as they exit the ML-Series card. This feature operates on the outer IEEE 802.1P tag when coupled with QinQ.

Per class flow queuing is provided to enable fair access to excess network bandwidth, and low latency queuing is supported for voice traffic. It allows allocation of bandwidth to support service-level agreements and ensure applications with high network resource requirements are adequately served. Buffers are allocated to queues dynamically from a shared resource pool. The allocation process incorporates the instantaneous system load as well as the allocated bandwidth to each queue to optimize buffer allocation to each queue.

The ML-Series card uses an advanced Weighted Deficit Round Robin (WDRR) scheduling process to provide fair access to excess bandwidth as well as guaranteed throughput to each class flow.

Admission control is a process that is invoked each time that service is configured on the ML-Series card to ensure that the card's available QoS resources are not overcommitted. In particular, admission control ensures that no configurations are accepted where a sum of the committed bandwidths on an interface exceed the total bandwidth of that interface.

The QoS bandwidth allocation of Multicast and Broadcast traffic is handled separately and differently than Unicast traffic. Aggregate Multicast and Broadcast traffic are given a fixed bandwidth commit of 10% on each interface, and treated as best effort for traffic exceeding 10%. Multicast and Broadcast are supported at line-rate.

Configuring QoS

Configuring a QoS policy typically requires classifying traffic into classes, configuring policies applied to those traffic classes, and attaching policies to interfaces.

This section contains this configuration information:

Classifying Traffic by Using Class Maps

You use the class-map global configuration command to isolate a specific traffic flow (or class) from all other traffic and to name it. The class map defines the criteria to use to match against a specific traffic flow to further classify the traffic of an interface. Match statements can include bridge-group, input-interface, IP precedence values, CoS, or IP DSCP values criterion. In use, the traffic class applies only to a specific interface on which it is applied (via a policy map). The traffic classification is not global, but the traffic class definition can be re-used for multiple interfaces or policy maps.

A single hidden class map always exists, named class-default, which is defined as **match-any**. This can be used to match all packets on any input or output that has an applied policy map.

Beginning in privileged EXEC mode, follow these steps to create a class map and to define the match criterion to classify traffic:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters global configuration mode.
Step 2	`Router(config)#` **`class-map`** `[{`**`match-all`**` | `**`match any`**`}]` *`class-map-name`*	Creates a class map, and enters class-map configuration mode. • Use the **match-all** keyword to perform a logical-AND of all matching statements under this class map. All match criteria in the class map must be matched. • Use the **match-any** keyword to perform a logical-OR of all matching statements under this class map. One or more match criteria must be matched. For *class-map-name*, specify the name of the class map. If neither the **match-all** nor **match-any** keyword is specified, the default is **match-all**.

	Command	Purpose

Step 3	`Router(config-cmap)#` **`match`** *`keyword`*	Defined the match keyword to classify traffic.
		The following are valid keyword choices:
		`any`
		`bridge-group`
		`cos`
		`input-interface`
		`ip dscp`
		`ip precedence`
		The `input-interface` choice is not valid when applied to the INPUT of an interface (redundant).
		There is no default match criterion.
		Multiple match criteria are supported. The command matches either ALL or ANY of the criteria, as controlled by the **match-all** and **match-any** subcommands of the class-map command.

Step 4	`Router(config-cmap)#` **`end`**	Returns to privileged EXEC mode.
Step 5	`Router#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

To delete an existing class map, use the **no class-map** *class-map-name* [**match-all** | **match-any**] global configuration command. To remove a match criterion, use the **no** form of the **match** command.

This example shows how to create a class map called class1 that matches incoming traffic entering interface fastethernet0:

```
Router(config)# class-map class1
Router(config-cmap)# match input-interface fastethernet0
```

This example shows how to create a class map called class2 that matches incoming traffic with IP-precedence values of 5, 6, and 7:

```
Router(config)# class-map match-any class2
Router(config-cmap)# match ip precedence 5 6 7
```

Note If a class-map contains a match rule which specifies multiple values, such as 5 6 7 in this example, then the class-map must be match-any, not the default match-all. Without the match-any an error message is printed and the class is ignored. The supported commands which allow multiple values are **match cos**, **match ip precedence** and **match ip dscp**.

This example shows how to create a class map called class3 that matches incoming traffic based on bridge group 1:

```
Router(config)# class-map class3
Router(config-cmap)# match bridge-group 1
```

Classifying, Policing, and Marking Traffic by Using Policy Maps

A policy map specifies which traffic class to act on and the actions to take. Actions can include setting a specific Layer 2 CoS value in the traffic class and specifying the traffic bandwidth limitations for each matched traffic class (policer) and the action to take when the traffic is out of profile (marking).

A policy map also has these characteristics:

- A policy map can contain multiple class statements, each with different match criteria and actions.
- A separate policy-map class can exist for each type of traffic received through an interface.
- Each packet will be matched to one and only one class map. If multiple matches are possible for a packet, it will match the first class map entry which applies.

You can attach only one policy map per interface per direction.

Beginning in global configuration mode, follow these steps to create a policy map:

	Command	Purpose
Step 1	`Router(config)#` **`policy-map`** *`policy-map-name`*	Creates a policy map by entering the policy map name, and enters policy-map configuration mode. By default, no policy maps are defined.
Step 2	`Router(config-pmap)#` **`class`** *`class-map-name`*	Selects a traffic class to act on and enters policy-map class configuration mode, which allows actions to be specific for the class. By default, no class maps are selected.
Step 3	`Router(config-pmap-c)#` **`police`** *`rate-bps burst-byte`* **`conform-action`** `[{set-cos-transmit` *`cos value`* `|` `transmit}` **`exceed-action`** `{drop |` `set-cos-transmit` *`cos value`*`}]`	Defines a policer for the currently selected class when the policy map is applied to input. Policing is supported only on ingress, not on egress. • For *rate-bps*, specify the average traffic rate in bits per second (bps). The range is 96000 to 8000000000. • For *burst-byte*, specify the normal burst size in bytes. The range is 8000 to 64000. • Conform action options are: – Set a VLAN CoS priority value and transmit – Transmit packet (default) • Exceed action options are: – Set a CoS value and transmit – Drop packet default)
Step 4	`Router(config-pmap-c)#` **`bandwidth`** `{`*`8-2000000`* `|` *`percent`*`}`	Specifies the minimum committed bandwidth for the currently selected class. When the policy-map is applied to an output, an output queue with the proper weight is created for this class. The bandwidth command is supported only on egress, not on ingress. Valid choices are: • Rate in kilobits per second (8 to 2000000) • Percent of total available bandwidth (1 to 100) If multiple classes and bandwidth actions are specified in a single policy map, they must use the same choice in specifying bandwidth (kilobits or percent).

	Command	Purpose
Step 5	`Router(config-pmap-c)# priority {8-2000000 | percent}`	Specifies low latency queuing for the currently selected class. When the policy-map is applied to an output, an output queue with strict priority is created for this class. Valid choices are: • Rate in kilobits per second (8 to 2000000) • Percent of total available bandwidth (1 to 100) If multiple classes and bandwidth actions are specified in a single policy map, they must use the same choice in specifying bandwidth (kilobits or percent). **Note** When using the priority action, the traffic in that class is given a 100% commit (CIR), regardless of the rate entered as the priority rate. To ensure that CIR commitments are met for the interface, a policer must be configured on the input of all interfaces that might deliver traffic to this output class, limiting the peak rate to the priority rate entered.
Step 6	`Router(config-pmap-c)# set cos {0-7}`	This command may only be used in a policy-map applied to an output. It specifies the VLAN COS priority to set for the outbound packets in the currently selected class. If QinQ is used, the top-level VLAN tag is marked. If outbound packets have no VLAN tag, the action has no effect. This action is applied to the packet after any "set cos" action done by a policer, and will therefore override the COS set by a policer action.

Note The **set-cos-transmit** action sets (marks) a VLAN COS priority associated with the packets under/over the policed rate. The classification of any output policy map will be based on the new COS value set (even if the packet were to enter and/or exit without a VLAN tag). An output policy map may override the VLAN COS priority set for the packet. When the packet is transmitted out an interface, the associated VLAN COS priority will be written into the VLAN tag. If QinQ is used, the top-level VLAN tag at the time of transmit will be marked. If the packet is transmitted without a VLAN tag, no marking will occur.

Note When the bandwidth or priority action is used on any class in a policy map, then there must be a class defined by "match any" which has a bandwidth or priority action in that policy map. This is to ensure that all traffic can be classified into a default class which has some assigned bandwidth. The hidden class "class-default" can be used as the "match any" class, and a minimum bandwidth can be assigned to it if the class is not expected to be used or no reserved bandwidth is desired for default traffic.

Note When using the bandwidth action, excess traffic (beyond the configured commit) will be allocated any available bandwidth in proportion to the relative bandwidth commitment of it's traffic class compared to other traffic classes. Excess traffic from two classes with equal commits will have equal access to available bandwidth. Excess traffic from a class with a minimum commit might receive only a minimum share of available bandwidth compared to excess bandwidth from a class with a high commit.

Note When the policing action is used with a "match any" class (policing an entire interface), and flow control send is enabled, then flow control will be used to back-off the source port to the configured police rate, rather than discarding the over-limit traffic.

Applying Policy Map to Interface

Beginning in global configuration mode, follow these steps to apply a policy map to an interface using the **service-policy** command.

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`interface-id`*	Enters interface configuration mode, and specifies the interface to apply the policy map. Valid interfaces are limited to physical Ethernet and POS interfaces. **Note** Policy maps cannot be applied to subinterfaces, port channel interfaces, or BVIs.
Step 2	`Router(config-if)#` **`service-policy`** {**`input`** *`policy-map-name`* | **`output`** *`policy-map-name`*}	Applies a policy map to the input or output of a particular interface. Only one policy map per interface per direction is supported. • Use **input** *policy-map-name* to apply the specified policy map to the input of an interface. • Use **output** *policy-map-name* to apply the specified policy map to the output of an interface.
Step 3	`Router(config-if)#` **`end`**	Returns to privileged EXEC mode.
Step 4	`Router#` **`copy running-config startup-config`**	(Optional) Saves your entries in the configuration file.

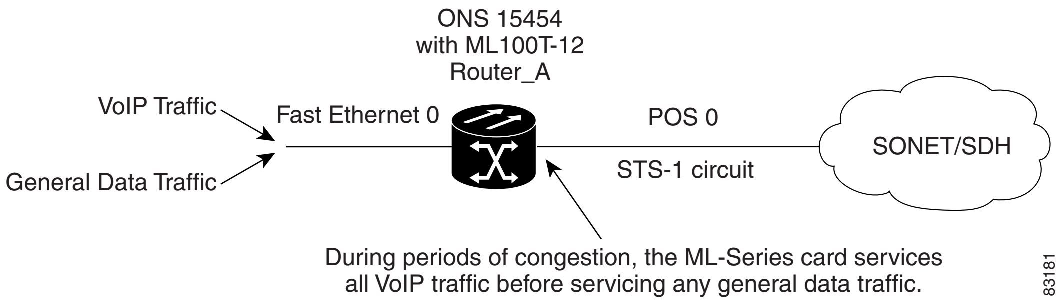
Note Applying an input policy map to a Fast Ethernet or Gigabit Ethernet interface with port based policing is required to enable flow control on that interface.

To delete an existing policy map, use the **no policy-map** *policy-map-name* global configuration command. To delete an existing class map, use the **no class** *class-map-name* policy-map configuration command. To remove an existing policer, use the **no police** policy-map configuration command. To remove the policy map and interface association, use the **no service-policy** {**input** *policy-map-name* | **output** *policy-map-name*} interface configuration command.

ML-Series QoS Examples

Figure 13-1 shows an example of ML-Series QoS. The router configuration for this example is shown in Example 13-1.

Figure 13-1 ML-Series QoS Example



Example 13-1 Router A Configuration

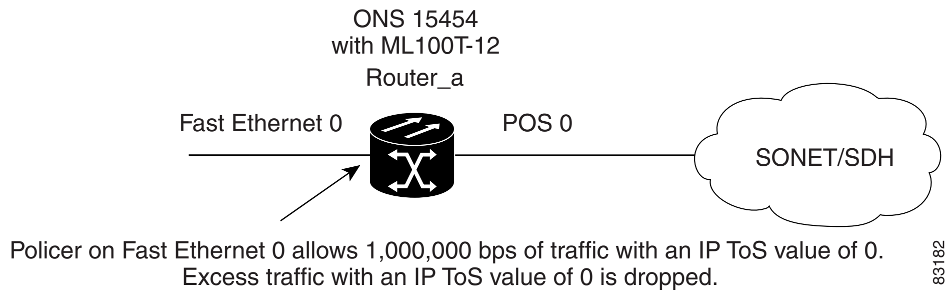
```
class-map match-all voip
  match ip precedence 5
class-map match-all default
  match any
!
!
policy-map pos0
  class default
    bandwidth   1000

  class voip
    priority 1000
!
interface FastEthernet0
 ip address 1.1.1.1 255.255.255.0
!
interface POS0
 ip address 2.1.1.1 255.255.255.0
service-policy output pos0
crc 32
 no cdp enable
 pos flag c2 1
```

ML-Series Policing Example

Figure 13-2 shows an example of ML-Series QoS.

Figure 13-2 ML-Series Policing Example



Example 13-2 shows how to configure a policer that will restrict traffic with an IP precedence of 0 to 1,000,000 bps.

Example 13-2 ML-Series Policing Sample Configuration

```
!
class-map match-all policer
  match ip precedence 0
!
policy-map police_f0
  class policer
    police 1000000 10000 conform-action transmit exceed-action drop
!
interface FastEthernet0
service-policy input police_f0
!
```

Monitoring and Verifying QoS

Table 13-1 shows Privileged EXEC commands that can be used to monitor and verify QoS status.

Table 13-1 Commands for QoS Status

Command	Purpose
`Router# show class-map name`	Displays the traffic class information of the user-specified traffic class.
`Router# show policy-map name`	Displays the user-specified policy map.
`Router# show policy-map interface interface`	Displays configurations of all input and output policies attached to an interface. Statistics displayed with this command are unsupported and show zero.

Examples of these commands are shown here:

```
Router# show class-map
 Class Map match-any class-default (id 0)
   Match any
```

```
 Class Map match-all policer (id 2)
   Match ip precedence 0

Router# show policy-map
 Policy Map police_f0
  class  policer
   police 1000000 10000 conform-action transmit exceed-action drop

Router# show policy-map interface

 FastEthernet0

  service-policy input: police_f0

    class-map: policer (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      match: ip precedence 0

    class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      match: any
        0 packets, 0 bytes
        5 minute rate 0 bps
```