Modular QoS CLI Three-Level Hierarchical Policer

Last Updated: May 2, 2012

The Modular QoS CLI (MQC) Three-Level Hierarchical Policer extends the traffic policing functionality by allowing you to configure traffic policing at three levels of policy map hierarchies; a primary level, a secondary level, and a tertiary level. Traffic policing may be configured at any or all of these levels, depending on the needs of your network. Configuring traffic policing in a three-level hierarchical structure provides a high degree of granularity for traffic policing.

Feature Specifications for the Modular QoS CLI (MQC) Three-Level Hierarchical Policer

Feature History	
Release	Modification
12.2(13)T	This feature was introduced.
Supported Platforms	
For platforms supported in Cisco IOS Release 12.2(13)T, consult Cisco Feature Navigator.	

- Finding Feature Information, page 1
- Restrictions for the Modular QoS CLI Three-Level Hierarchical Policer, page 2
- Information About the Modular QoS CLI Three-Level Hierarchical Policer, page 3
- How to Configure the Modular QoS CLI Three-Level Hierarchical Policer, page 5
- Configuration Examples for the Modular QoS CLI Three-Level Hierarchical Policer, page 10
- Additional References, page 12

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for the Modular QoS CLI Three-Level Hierarchical Policer

If traffic policing is configured at both the top level and secondary levels, note the following caveats:

- When traffic policing is configured at both the primary and secondary levels, the traffic policer at the secondary level acts only on packets sent by the policer at the top level.

However, the packet classification for the policy map at the secondary level occurs before the primary level policer has acted on the classes. When this situation occurs, the class counters for the policy map at the secondary level may not be equal to the number of packets acted upon by the second level policer.

The following output of the **show policy-map interface**command helps to illustrate this point. In this sample output two policy maps (called "primary_level," and "secondary_level," respectively) have been configured. The primary_level policy map contains a class map called "c1," and the secondary_level policy map contains a class map called "c3".

```
> > > show policy interface serial5/0.1
> > >  Serial5/0.1
> > >
> > >   Service-policy output: primary_level
> > >
> > >     Class-map: c1 (match-all)
> > >       24038 packets, 3004750 bytes
> > >       30 second offered rate 0 bps, drop rate 0 bps
> > >       Match: any
> > >       police:
> > >           cir 300000 bps, bc 9375 bytes
> > >         conformed 18105 packets, 2263125 bytes; actions:
> > >           transmit
> > >         exceeded 5933 packets, 741625 bytes; actions:              (*)
> > >           drop
> > >         conformed 0 bps, exceed 0 bps
> > >
> > >       Service-policy : secondary_level
> > >
> > >         Class-map: c3 (match-all)
> > >           24038 packets, 3004750 bytes

> > >           30 second offered rate 0 bps, drop rate 0 bps
> > >           Match: any
> > >           police:                (<= Indicates traffic policing has been configured)
> > >               cir 200000 bps, bc 3000 bytes
> > >               pir 250000 bps, be 3000 bytes
> > >             conformed 12047 packets, 1505875 bytes; actions:       (**)
> > >               set-frde-transmit
> > >             exceeded 3004 packets, 375500 bytes; actions:          (**)
> > >               set-frde-transmit
> > >             violated 3054 packets, 381750 bytes; actions:          (**)
> > >               set-frde-transmit
> > >             conformed 0 bps, exceed 0 bps, violate 0 bps
> > >
> > >         Class-map: class-default (match-any)
> > >           0 packets, 0 bytes
> > >           30 second offered rate 0 bps, drop rate 0 bps
> > >           Match: any
> > >             0 packets, 0 bytes
> > >             30 second rate 0 bps
```

Note the following about this example:

- ◦ The class counter for the class map called "c3" shows 24038 packets (italicized in the example).
  - ◦ Traffic policing has been configured in the policy map, and the traffic policing feature for class map "c3" shows a total of 18105 packets -- 12047 conformed packets, plus 3004 exceeded packets, plus 3054 violated packets (indicated by the double asterisks ("**") in the example). This total is because 5933 packets have already been dropped in class map "c1" (indicated by the "*" in the example).
  - ◦ Therefore, only 18105 packets (24038 packets minus 5933 packets) are acted upon by the traffic policing feature configured in the second_level policy map.
- In this implementation of the Modular QoS CLI (MQC) Three-Level Hierarchical Policer, traffic policing at the primary level does not guarantee fairness in sharing bandwidth among the child classes. If packets from two different classes arrive at the same rate and then go through a traffic policer, the output rates of the two classes could be different because this feature acts as an aggregate policer.

In other words, it is possible that the primary-level policer could drop packets in one class in favor of the other class. This situation would happen because the primary-level policer had enough tokens when the packets for one class arrived, but there were not enough tokens left for the other class. This pattern could continue indefinitely, based on the arrival pattern of the packets.

Information About the Modular QoS CLI Three-Level Hierarchical Policer

- Modular Quality of Service Command-Line Interface, page 3
- Packet Flow in the Modular QoS CLI Three-Level Hierarchical Policer, page 4
- Other Traffic Policing-Related Features, page 4

Modular Quality of Service Command-Line Interface

The MQC is a command-line interface (CLI) structure that allows you to create traffic policies and attach these policies to interfaces.

In the MQC, the **class-map** command is used to define a traffic class (which is then associated with a traffic policy). The purpose of a traffic class is to classify traffic.

The Modular quality of service (QoS) CLI structure consists of the following three processes:

- Defining a traffic class with the **class-map** command.
- Creating a traffic policy by associating the traffic class with one or more QoS features (using the **policy-map** command).
- Attaching the traffic policy to the interface with the **service-policy** command.

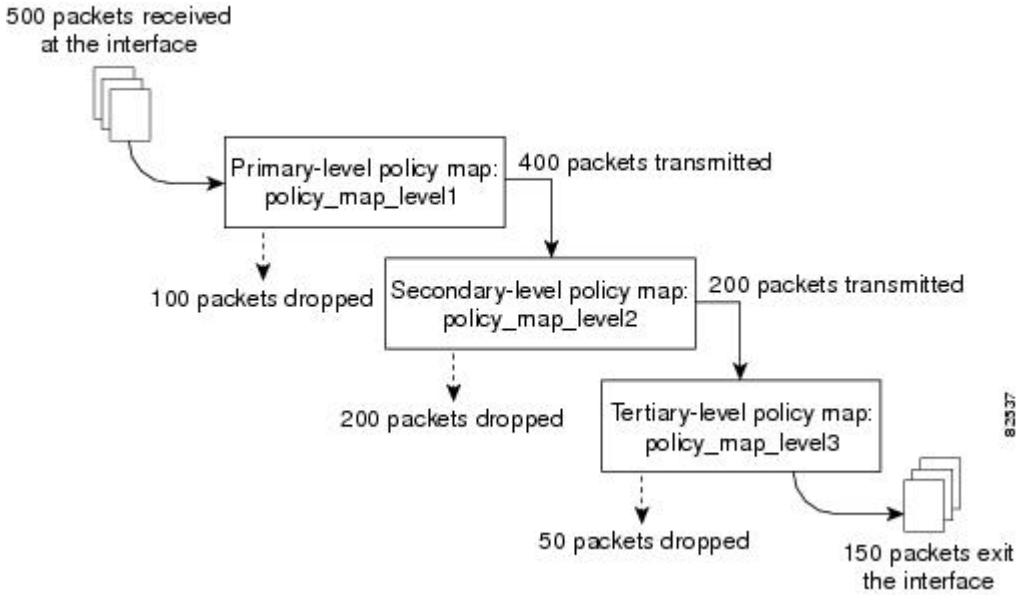
A traffic class contains three major elements: a name, a series of **match** commands, and, if more than one **match** command exists in the traffic class, an instruction on how to evaluate these **match** commands. The traffic class is named in the **class-map** command line; that is, if you enter the **class-map cisco**command while configuring the traffic class in the CLI, the traffic class would be named "cisco".

The **match** commands are used to specify various criteria for classifying packets. Packets are checked to determine whether they match the criteria specified in the **match** commands. If a packet matches the specified criteria, that packet is considered a member of the class and is forwarded according to the QoS specifications set in the traffic policy. Packets that fail to meet any of the matching criteria are classified as members of the default traffic class.

Packet Flow in the Modular QoS CLI Three-Level Hierarchical Policer

The figure below illustrates the flow of packets among policy maps configured for traffic policing at each level in the hierarchy.

Figure 1 Packet Flow Among Policy Maps



In the figure above, three policy maps are configured: policy_map_level1 (the primary-level policy map), policy_map_level2 (the secondary-level policy map), and policy_map_level3 (the tertiary-level policy map). Traffic policing is configured in each policy map, and each policy map is attached to a service policy and to an interface.

In this simplified illustration, 500 packets arrive at the interface at which the policy map called "policy_map_level1" is attached. Because of the way traffic policing is configured in this policy map, 100 packets are dropped and 400 packets are transmitted.

The traffic policer at the secondary-level policy map (policy_map_level2) then evaluates the packets and treats them as determined by the way traffic policing is configured at this level. Of the 400 packets received, 200 are dropped and 200 are transmitted.

The traffic policer at the tertiary-level policy map (policy_map_level3), in turn, evaluates the 200 packets it has now received and applies the appropriate treatment as determined by the way the traffic policing is configured at this level.

Other Traffic Policing-Related Features

The Cisco IOS traffic policing software features allow you to control the maximum rate of traffic sent or received on an interface. Traffic policing is often configured on interfaces at the edge of a network to limit traffic into or out of the network. Traffic that falls within the rate parameters is sent, whereas traffic that exceeds or violates the parameters is dropped or sent with a different priority.

The Cisco IOS software currently includes the following traffic policing features:

- Traffic Policing (a single-rate policer)

- Two-Rate Policer
- Policer Enhancements -- Multiple Actions
- Percentage-Based Policing and Shaping

Previously, these features could be configured at two levels of a policy map hierarchy; the top level and one secondary level. With the Modular QoS CLI (MQC) Three-Level Hierarchical Policer, these traffic policing-related features can configured in three levels of a policy map hierarchy.

The tasks for configuring each of these traffic policing-related features is essentially the same. That is, you use the MQC to create a policy map. Then you use the **police** command to configure traffic policing for a specific class within that policy map. The policy map is then attached to an interface.

Traffic policing can be configured to specify multiple marking actions for the traffic being policed, or to use a percentage of available bandwidth when policing traffic.

How to Configure the Modular QoS CLI Three-Level Hierarchical Policer

- Configuring Traffic Policing, page 5
- Attaching the Policy Map to an Interface, page 6
- Verifying the Configuration, page 8

Configuring Traffic Policing

Traffic policing can be configured at any level of the policy map hierarchy, that is, at the primary level, secondary level, or the tertiary level.

Before configuring traffic policing, you must use the MQC to create a policy map.

SUMMARY STEPS

1. **enable**
2. **configure** {**terminal** | **memory** | **network**}
3. **policy-map** *policy-name*
4. **class-map** *class-map-name*
5. **police** *bps burst-normal burst-max* **conform-action** *action* **exceed-action** *action* **violate-action** *action*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure** {**terminal** | **memory** | **network**} **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-name* **Example:** `Router(config)# policy-map policy1`	Specifies the name of the policy map created earlier and enters policy-map configuration mode. • See the Configuring Traffic Policing, page 5. • Enter policy map name.
Step 4	**class-map** *class-map-name* **Example:** `Router(config-pmap)# class-map class1`	Specifies the name of the class map created when the policy map was created earlier and enters policy-map class configuration mode. • See the Configuring Traffic Policing, page 5. • Enter the class map name.
Step 5	**police** *bps burst-normal burst-max* **conform-action** *action* **exceed-action** *action* **violate-action** *action* **Example:** `Router(config-pmap-c)# police 8000 1000 1000 conform-action transmit exceed-action drop violate-action drop`	Configures traffic policing according to burst sizes and any optional actions specified.
Step 6	**exit** **Example:** `Router(config-pmap-c)# exit`	(Optional) Exits the policy-map class configuration mode.

Attaching the Policy Map to an Interface

After the policy map has been created and traffic policing has been configured, the policy map must be attached to an interface. Policy maps can be attached to either the input or output direction of the interface.

Depending on the needs of your network, you may need to attach the policy map to a subinterface, an ATM permanent virtual circuit (PVC), a Frame Relay data-link connection identifier (DLCI), or other type of interface.

SUMMARY STEPS

1. **enable**
2. **configure** {**terminal** | **memory** | **network**}
3. **interface** *type number*
4. **pvc** [*name*] *vpi* / *vci* [**ilmi** | **qsaal** | **smds**]
5. **service-policy** {**input**| **output**} *policy-map-name*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure** {**terminal**	**memory**	**network**} **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**interface** *type number*
	
	Example:
	
	Example:
	`Router(config-if)#`
	`interface s4/0`
		Configures an interface (or subinterface) type and enters interface configuration mode.
		• Enter the interface type number.

Step 4	**pvc** [*name*] *vpi* / *vci* [**ilmi**	**qsaal**	**smds**] **Example:** `Router(config-if)# pvc cisco 0/16 ilmi`	(Optional) Creates or assigns a name to an ATM PVC and specifies the encapsulation type on an ATM PVC. Enters ATM virtual circuit (VC) configuration mode (config-if-atm-vc). **Note** This step is required only if you are attaching the policy map to an ATM PVC. If you are not attaching the policy map to an ATM PVC, skip this step and proceed with Attaching the Policy Map to an Interface, page 6.

	Command or Action	Purpose

Step 5	**service-policy** {**input**| **output**} *policy-map-name*
	
	Example:
	`Router(config-if)#`
	`service-policy input policy1`
	
	Example:
		Specifies the name of the policy map to be attached to the input *or* output direction of the interface.
		Note Policy maps can be configured on ingress or egress routers. They can also be attached in the input or output direction of an interface. The direction (input or output) and the router (ingress or egress) to which the policy map should be attached varies according your network configuration. When using the **service-policy** command to attach the policy map to an interface, be sure to choose the router and the interface direction that are appropriate for your network configuration.
		• Enter the policy map name.

Step 6	**exit** **Example:** `Router(config-if)# exit`	(Optional) Exits interface configuration mode.

• What to Do Next, page 8

What to Do Next

If you want to configure traffic policing at another level in the policy map hierarchy, repeat the steps in the Configuring Traffic Policing, page 5 section and the Attaching the Policy Map to an Interface, page 6 section.

Verifying the Configuration

This task allows you to verify that you created the configuration you intended and that the feature is functioning correctly.

SUMMARY STEPS

1. **enable**
2. Do one of the following:
   - **show policy-map**
   -
   - **show policy-map interface** *interface-name*
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.

Step 2	Do one of the following:
	• **show policy-map**
	•
	• **show policy-map interface** *interface-name*
	
	Example:
	`Router# show policy-map`
	
	Example:
	
	Example:
	`Router#`
	`show policy-map interface s4/0`
		Displays all configured policy maps.
		or
		Displays the packet statistics of all classes that are configured for all service policies either on the specified interface or subinterface or on a specific PVC on the interface.
		• Enter the interface name.

Step 3	**exit** **Example:** `Router(config-if)# exit`	(Optional) Exits interface configuration mode.

• Troubleshooting Tips, page 9

Troubleshooting Tips

The commands in the Verifying the Configuration, page 8 section allow you to verify that you achieved the intended configuration and that the feature is functioning correctly. If after using the **show** commands listed above, the configuration is not correct or the feature is not functioning as expected, do the following:

If the configuration is not the one you intended, complete the following procedures:

- Use the **show running-config** command and analyze the output of the command.
- If the policy map does not appear in the output of the **show running-config** command, enable the **logging console** command.
- Attach the policy map to the interface again.

If the packets are not being matched correctly (for example, the packet counters are not incrementing correctly), complete the following procedures:

- Use the **show policy-map**command and analyze the output of the command.
- Use the **show running-config** command and analyze the output of the command.
- Run the **show policy-map interface** command and analyze the output of the command. Review the the following:
  - If a policy map applies queueing, and the packets are matching the correct class, but you see unexpected results, compare the number of packets to the number of packets matched.
  - If the interface is congested, and you are only seeing a small number of packets matched, check the tuning of the tx ring, and evaluate whether the queueing is happening on the tx ring. To do this, use the **show controllers** command, and look at the value of the tx count in the show output of the command.

Configuration Examples for the Modular QoS CLI Three-Level Hierarchical Policer

- Example Configuring the Modular QoS CLI Three-Level Hierarchical Policer, page 10

Example Configuring the Modular QoS CLI Three-Level Hierarchical Policer

In the following example, the Modular QoS CLI (MQC) Three-Level Hierarchical Policer has been configured for three classes within three separate policy maps. The three classes, called "c1," "c2," and "c3," respectively, have been configured using the match criteria specified as follows:

```
class-map c1
   match any
class-map c2
   match ip precedence 1 2 3
class-map c3
   match ip precedence 2
```

Next, the classes are configured in three separate policy maps, called "p_all" (the primary-level policy map), "pmatch_123" (the secondary-level policy map), and "pmatch_2" (the tertiary-level policy map), as shown below.

```
policy p_all
   class c1
      police 100000
      service-policy pmatch_123
policy pmatch_123
  class c2
     police 20000
     service-policy pmatch_2
policy pmatch_2
  class c3
     police 8000
```

The primary goal of this configuration is to limit all traffic to 100 kbps. Within this, the secondary goal is make sure that packets with precedence values of 1, 2, or 3 do not exceed 20 kbps and that packets with precedence value of 2 never exceed 8 kbps.

To verify that the classes have been configured correctly and to confirm the results of the traffic policing configuration in the policy maps, the **show policy-map** command and the **show policy-map interface**command can be used, as shown in the following sections.

The following sample output of the **show policy-map**command verifies the configuration of the classes in the policy maps:

```
Router# show policy map
  Policy Map p_all
    Class c1
     police cir 100000 bc 3000
       conform-action transmit
       exceed-action drop
      service-policy pmatch_123
  Policy Map pmatch_123
    Class c2
     police cir 20000 bc 1500
       conform-action transmit
       exceed-action drop
      service-policy pmatch_2
  Policy Map pmatch_2
    Class c3
     police cir 8000 bc 1500
       conform-action transmit
       exceed-action drop
```

The following sample output of the **show policy-map interface** command confirms the results of this configuration on the attached interface:

```
Router# show policy-map interface Ethernet3/1
 Ethernet3/1
  Service-policy output:p_all
    Class-map:c1 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match:any
      police:
          cir 100000 bps, bc 3000 bytes
        conformed 0 packets, 0 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          drop
        conformed 0 bps, exceed 0 bps,
      Service-policy :pmatch_123
        Class-map:c2 (match-all)
          0 packets, 0 bytes
          5 minute offered rate 0 bps, drop rate 0 bps
          Match:ip precedence 1  2  3
          police:
              cir 20000 bps, bc 1500 bytes
            conformed 0 packets, 0 bytes; actions:
              transmit
            exceeded 0 packets, 0 bytes; actions:
              drop
            conformed 0 bps, exceed 0 bps,
          Service-policy :pmatch_2
            Class-map:c3 (match-all)
              0 packets, 0 bytes
              5 minute offered rate 0 bps, drop rate 0 bps
              Match:ip precedence 2
              police:
                  cir 8000 bps, bc 1500 bytes
                conformed 0 packets, 0 bytes; actions:
                  transmit
                exceeded 0 packets, 0 bytes; actions:
                  drop
                conformed 0 bps, exceed 0 bps,
            Class-map:class-default (match-any)
              0 packets, 0 bytes
              5 minute offered rate 0 bps, drop rate 0 bps
```

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

Additional References

The following sections provide additional references related to the Modular QoS CLI (MQC) Three-Level Hierarchical Policer:

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
Additional information about configuring traffic policing	"Policing and Shaping Overview" module
Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC)	"Applying QoS Features Using the MQC" module
Two-rate traffic policing	"Two-Rate Policer" module
Traffic policing using multiple policer actions	"Policer Enhancements--Multiple Actions" module
Percentage-based traffic policing and shaping	"Percentage-Based Policing and Shaping" module
Frame Relay configurations	"Configuring Frame Relay" module
Frame Relay commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Wide-Area Networking Command Reference*

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
• CISCO-CLASS-BASED-QOS-CAPABILITY-MIB • CISCO-CLASS-BASED-QOS-MIB	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC 2697	*A Single Rate Three Color Marker*
RFC 2698	*A Two Rate Three Color Marker*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html