Class-Based Policing

Class-based policing allows you to control the maximum rate of traffic that is transmitted or received on an interface. Class-based policing is often configured on interfaces at the edge of a network to limit traffic into or out of the network.

- Finding Feature Information, on page 1
- Information About Class-Based Policing, on page 1
- Restrictions for Class-Based Policing, on page 2
- How to Configure Class-Based Policing, on page 2
- Configuration Examples for Class-Based Policing, on page 6
- Additional References, on page 9

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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.

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.

Information About Class-Based Policing

Class-Based Policing Functionality

The Class-Based Policing feature performs the following functions:

- Limits the input transmission rate of a class of traffic based on user-defined criteria.

Class-based policing allows you to control the maximum rate of traffic transmitted or received on an interface. The Class-Based Policing feature is applied when you attach a traffic policy that contains the class-based policing configuration to an interface.

The Class-Based Policing feature works with a token bucket mechanism. There are currently two types of token bucket algorithms: a single token bucket algorithm and a two-token bucket algorithm. A single token

bucket system is used when the **violate-action** option is not specified, and a two-token bucket system is used when the **violate-action** option is specified.

Benefits of Class-Based Policing

Bandwidth Management Through Rate Limiting

Class-based policing allows you to control the maximum rate of traffic transmitted or received on an interface. Class-based policing is often configured on interfaces at the edge of a network to limit traffic into or out of the network. In most class-based policing configurations, traffic that falls within the rate parameters is transmitted, whereas traffic that exceeds the parameters is dropped or transmitted with a different priority.

Packet Marking

Packet marking allows you to partition your network into multiple priority levels or classes of service (CoS). A packet is marked and these markings can be used to identify and classify traffic for downstream devices.

- Use class-based policing to set the IP precedence or DSCP values for packets entering the network. Networking devices within your network can then use the adjusted IP precedence values to determine how the traffic should be treated.
- Use class-based policing to assign packets to a QoS group. The router uses the QoS group to determine how to prioritize packets.

Traffic can be marked without using the Class-Based Policing feature.

Restrictions for Class-Based Policing

- Class-based policing on sub-interfaces is *not* supported.
- Policing is supported for ingress policy maps only.
- Hierarchical policing (policing at both parent level and child level) is *not* supported. However, Egress two-level policer is supported provided PHB level priority policer is configured.
- Conditional marking is *not* supported.

How to Configure Class-Based Policing

Configuring a Traffic Policing Service Policy

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables higher privilege levels, such as privileged EXEC mode.

	Command or Action	Purpose
	`Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map** [**match-all** | **match-any**] *class-map-name* **Example:** `Router(config)# class-map match-any MATCH_PREC`	Specifies the name of the class map to be created and enters QoS class map configuration mode. • The class map defines the criteria to use to differentiate the traffic. For example, you can use the class map to differentiate voice traffic from data traffic, based on a series of match criteria defined using the **match** command. **Note** If the **match-all** or **match-any** keyword is not specified, traffic must match all the match criteria to be classified as part of the traffic class.
Step 4	**match ip precedence** *precedence-value* **Example:** `Router(config-cmap)# match ip precedence 0`	Enables packet matching on the basis of the IP precedence values you specify. **Note** You can enter up to four matching criteria, as number abbreviation (0 to 7) or criteria names (critical, flash, and so on), in a single match statement.
Step 5	**exit** **Example:** `Router(config-cmap)# exit`	Returns to global configuration mode.
Step 6	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map POLICE-SETTING`	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy, and enters QoS policy-map configuration mode.
Step 7	**class** {*class-name* | **class-default**} **Example:** `Router(config-pmap)# class MATCH_PREC`	Specifies the name of the class whose policy you want to create or change or specifies the default class (commonly known as the class-default class) before you configure its

	Command or Action	Purpose
		policy, and enters policy-map class configuration mode.
Step 8	**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 set-qos-transmit 1 violate-action drop`	Configures traffic policing according to burst sizes and any optional actions specified.
Step 9	**exit** **Example:** `Router(config-pmap-c)# exit`	(Optional) Exits policy-map class configuration mode.
Step 10	**exit** **Example:** `Router(config-pmap)# exit`	(Optional) Exits QoS policy-map configuration mode.
Step 11	**interface** *interface-type interface-number* **Example:** `Router(config)# interface GigabitEthernet 0/0/1`	Configures an interface type and enters interface configuration mode. • Enter the interface type and interface number.
Step 12	**service-policy** {**input** | **output**} *policy-map-name* **Example:** `Router(config-if)# service-policy input POLICE-SETTING`	Attaches a policy map to an interface. • Enter either the **input** or **output** keyword and the policy map name.
Step 13	**end** **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode and returns to privileged EXEC mode.

Monitoring and Maintaining Traffic Policing

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**show policy-map** **Example:** `Router# show policy-map`	Displays all configured policy maps.
Step 3	**show policy-map** *policy-map-name* **Example:** `Router# show policy-map pmap`	Displays the user-specified policy map.
Step 4	**show policy-map interface** **Example:** `Router# show policy-map interface`	Verifies that the Class-Based Policing feature is configured on your interface. If the feature is configured on your interface. • The command output displays policing statistics.

Verifying Class-Based Traffic Policing

Use the **show policy-map interface** command to verify that the Class-Based Policing feature is configured on your interface. If the feature is configured on your interface, the **show policy-map interface** command output displays policing statistics.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show policy-map interface** **Example:** `Router# show policy-map interface`	Verifies that the Class-Based Policing feature is configured on your interface. If the feature is configured on your interface. • The command output displays policing statistics.
Step 3	**show policy-map interface** *type interface* **Example:** `Router# show policy-map interface GigabitEthernet 0/0/1`	Displays traffic statistics for policies applied to a specific interface.

	Command or Action	Purpose
Step 4	**show policy-map interface** *type interface* **service instance** *service-instance number* **Example:** `Router# show policy-map interface GigabitEthernet 0/0/1 service instance 1`	Displays the policy map information for a given service instance under an interface.
Step 5	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Example: Verifying Class-Based Traffic Policing

```
Router# show policy-map interface
  FastEthernet1/1/1
  service-policy output: x
    class-map: a (match-all)
      0 packets, 0 bytes
      5 minute rate 0 bps
      match: ip precedence 0
      police:
        1000000 bps, 10000 limit, 10000 extended limit
        conformed 0 packets, 0 bytes; action: transmit
        exceeded 0 packets, 0 bytes; action: drop
        conformed 0 bps, exceed 0 bps, violate 0 bps
```

Troubleshooting Tips

Check the interface type. Verify that class-based policing is supported on your interface.

Configuration Examples for Class-Based Policing

Example Configuring a Service Policy That Includes Traffic Policing

In the following example, class-based policing is configured with the average rate at 8000 bits per second, the normal burst size at 1000 bytes, and the excess burst size at 1000 bytes for all packets leaving the interface.

```
class-map access-match
 match access-group 1
 exit
policy-map police-setting
 class access-match
  police 8000 1000 1000 conform-action transmit exceed-action set-qos-transmit 1
violate-action drop
  exit
```

```
exit
service-policy output police-setting
```

The treatment of a series of packets leaving FastEthernet interface 1/1/1 depends on the size of the packet and the number of bytes remaining in the conform and exceed token buckets. The series of packets are policed based on the following rules:

- If the previous arrival of the packet was at T1 and the current arrival of the packet is at T, the bucket is updated with T - T1 worth of bits based on the token arrival rate. The refill tokens are placed in the conform bucket. If the tokens overflow the conform bucket, the overflow tokens are placed in the exceed bucket. The token arrival rate is calculated as follows:

(time between packets < which is equal to T - T1 > * policer rate)/8 bytes

- If the number of bytes in the conform bucket is greater than the length of the packet (for example, B), then the packet conforms and B bytes should be removed from the bucket. If the packet conforms, B bytes are removed from the conform bucket and the conform action is taken. The exceed bucket is unaffected in this scenario.
- If the number of bytes in the conform bucket is less than the length of the packet, but the number of bytes in the exceed bucket is greater than the length of the packet (for example, B), the packet exceeds and B bytes are removed from the bucket.
- If the number bytes in the exceed bucket B is fewer than 0, the packet violates the rate and the violate action is taken. The action is complete for the packet.

In this example, the initial token buckets starts full at 1000 bytes. If a 450-byte packet arrives, the packet conforms because enough bytes are available in the conform token bucket. The conform action (send) is taken by the packet, and 450 bytes are removed from the conform token bucket (leaving 550 bytes).

If the next packet arrives 0.25 seconds later, 250 bytes are added to the conform token bucket ((0.25 * 8000)/8), leaving 800 bytes in the conform token bucket. If the next packet is 900 bytes, the packet does not conform because only 800 bytes are available in the conform token bucket.

The exceed token bucket, which starts full at 1000 bytes (as specified by the excess burst size, is then checked for available bytes. Because enough bytes are available in the exceed token bucket, the exceed action (set the QoS transmit value of 1) is taken, and 900 bytes are taken from the exceed bucket (leaving 100 bytes in the exceed token bucket).

If the next packet arrives 0.40 seconds later, 400 bytes are added to the token buckets ((.40 * 8000)/8). Therefore, the conform token bucket now has 1000 bytes (the maximum number of tokens available in the conform bucket, and 200 bytes overflow the conform token bucket (because only 200 bytes were needed to fill the conform token bucket to capacity). These overflow bytes are placed in the exceed token bucket, giving the exceed token bucket 300 bytes.

If the arriving packet is 1000 bytes, the packet conforms because enough bytes are available in the conform token bucket. The conform action (transmit) is taken by the packet, and 1000 bytes are removed from the conform token bucket (leaving 0 bytes).

If the next packet arrives 0.20 seconds later, 200 bytes are added to the token bucket ((.20 * 8000)/8). Therefore, the conform bucket now has 200 bytes. If the arriving packet is 400 bytes, the packet does not conform because only 200 bytes are available in the conform bucket. Similarly, the packet does not exceed because only 300 bytes are available in the exceed bucket. Therefore, the packet violates and the violate action (drop) is taken.

Verifying Class-Based Traffic Policing

Use the **show policy-map interface** command to verify that the Class-Based Policing feature is configured on your interface. If the feature is configured on your interface, the **show policy-map interface** command output displays policing statistics:

```
Router# show policy-map interface
  FastEthernet1/1/1
  service-policy output: x
    class-map: a (match-all)
      0 packets, 0 bytes
      5 minute rate 0 bps
      match: ip precedence 0
      police:
        1000000 bps, 10000 limit, 10000 extended limit
        conformed 0 packets, 0 bytes; action: transmit
        exceeded 0 packets, 0 bytes; action: drop
        conformed 0 bps, exceed 0 bps, violate 0 bps
```

Use the **show policy-map interface** *type nummber* command to view the traffic statistics for policies applied to that specific interface:

```
Router# show policy-map interface gigabitethernet 0/0/1
 GigabitEthernet0/0/1

  Service-policy input: TUNNEL_MARKING

    Class-map: MATCH_PREC (match-any)
      72417 packets, 25418367 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: ip precedence 0
      QoS Set
        ip precedence tunnel 3
          Marker statistics: Disabled

    Class-map: MATCH_DSCP (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: ip dscp default (0)
      QoS Set
        ip dscp tunnel 3
          Marker statistics: Disabled

    Class-map: class-default (match-any)
      346462 packets, 28014400 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any

  Service-policy output: POLICE-SETTING

    Class-map: MATCH_PREC (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: ip precedence 0
      police:
          cir 8000 bps, bc 1000 bytes, be 1000 bytes
        conformed 0 packets, 0 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          set-qos-transmit 1
        violated 0 packets, 0 bytes; actions:
```

```
          drop
        conformed 0000 bps, exceed 0000 bps, violate 0000 bps

    Class-map: class-default (match-any)
      31 packets, 2019 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any
```

Use the **show policy-map interface service instance** command to view the traffic statistics for policy applied to the specific service instance in that specific interface:

```
Router# show policy-map interface gig0/0/1 service instance 10
 GigabitEthernet0/0/1: EFP 10

       Service-policy input: ac1

 Class-map: ac1 (match-all)
   0 packets, 0 bytes
   5 minute offered rate 0000 bps, drop rate 0000 bps
   Match: access-group 1
   police:
       cir 50000000 bps, bc 1562500 bytes
     conformed 0 packets, 0 bytes; actions:
       transmit
     exceeded 0 packets, 0 bytes; actions:
       drop
     conformed 0000 bps, exceeded 0000 bps

 Class-map: class-default (match-any)
   0 packets, 0 bytes
   5 minute offered rate 0000 bps, drop rate 0000 bps
   Match: any
```

Additional References

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*
Traffic marking	“Marking Network Traffic” module
Traffic policing	“Traffic Policing” module
Traffic policing and shaping concepts and overview information	“Policing and Shaping Overview”
Modular Quality of Service Command-Line Interface (MQC)	“Applying QoS Features Using the MQC” module

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
Class-Based Quality of Service MIB • CISCO-CLASS-BASED-QOS-MIB • CISCO-CLASS-BASED-QOS-CAPABILITY-MIB	To locate and download MIBs for selected platforms, Cisco IOS XE Software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2697	*A Single 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