QoS: Classification Configuration Guide, Cisco IOS Release 12.2SX

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

Last Updated: August 11, 2011

and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Configuring Committed Access Rate

This module describes the tasks for configuring committed access rate (CAR) and distributed CAR (DCAR).

Note In Cisco IOS Release 12.2 SR, CAR is not supported on the Cisco 7600 series router.

For complete conceptual information about these features, see the "Classification Overview"module and the "Policing and Shaping Overview" module.

For a complete description of the CAR commands in this module, see the Cisco IOS Quality of Service Solutions Command Reference. To locate documentation of other commands that appear in this module, use the command reference master index or search online.

Note CAR and DCAR can only be used with IP traffic. Non-IP traffic is not rate limited. CAR and DCAR can be configured on an interface or subinterface. However, CAR and DCAR are not supported on the Fast EtherChannel, tunnel, or PRI interfaces, nor on any interface that does not support Cisco Express Forwarding (CEF). CEF must be enabled on the interface before you configure CAR or DCAR. CAR is not supported for Internetwork Packet Exchange (IPX) packets.

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.

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.

Committed Access Rate Configuration Task List

The CAR and DCAR services limit the input or output transmission rate on an interface or subinterface based on a flexible set of criteria. CAR is often configured on interfaces at the edge of a network to limit traffic into or out of the network.

CAR can rate limit traffic based on certain matching criteria, such as incoming interface, IP precedence, or IP access list. You configure the actions that CAR will take when traffic conforms to or exceeds the rate limit.

You can set CAR rate policies that are associated with one of the following:

- All IP traffic
- IP precedence
- MAC address
- IP access list, both standard and extended. Matching to IP access lists is more processor-intensive than matching based on other criteria.

Each interface can have multiple CAR policies, corresponding to different types of traffic. For example, low priority traffic may be limited to a lower rate than high-priority traffic. With multiple rate policies, the router examines each policy in the order entered until the packet matches. If a match is not found, the default action is to send.

The rate policies can be independent; each rate policy deals with a different type of traffic. Alternatively, rate policies can be cascading; a packet can be compared to multiple different rate policies in succession. You can configure up to 100 rate policies on a subinterface.

Note Because of the linear search for the matching rate-limit statement, the CPU load increases with the number of rate policies.

Basic CAR and DCAR functionality requires that the following criteria be defined:

- Packet direction, incoming or outgoing.
- An average rate, determined by a long-term average of the transmission rate. Traffic that falls under this rate will always conform.
- A normal burst size, which determines how large traffic bursts can be before some traffic is considered to exceed the rate limit.
- An excess burst size (Be). Traffic that falls between the normal burst size and the Excess Burst size exceeds the rate limit with a probability that increases as the burst size increases. CAR propagates bursts. It does no smoothing or shaping of traffic.

Table 1 Rate-Limit Command Action Keywords

Keyword	Description
continue	Evaluates the next **rate-limit** command.
drop	Drops the packet.
set-prec-continue *new-prec*	Sets the IP Precedence and evaluates the next **rate-limit** command.

Keyword	Description
set-prec-transmit *new-prec*	Sets the IP Precedence and sends the packet.
transmit	Sends the packet.

- IP Precedence or MAC Address, page 3
- IP Access List, page 3

IP Precedence or MAC Address

Use the **access-list rate-limit** command to classify packets using either IP Precedence or MAC addresses. You can then apply CAR policies using the **rate-limit** command to individual rate-limited access lists. Packets with different IP precedences or MAC addresses are treated differently by the CAR service. See the section Example Rate Limiting in an IXP, page 7 for an example of how to configure a CAR policy using MAC addresses.

IP Access List

Use the **access-list** command to define CAR policy based on an access list. The *acl-index* argument is an access list number. Use a number from 1 to 99 to classify packets by precedence or precedence mask. Use a number from 100 to 199 to classify by MAC address.

Note If an access list is not present, the **rate-limit** command will act as if no access list is defined and all traffic will be rate limited accordingly.

When you configure DCAR on Cisco 7000 series routers with RSP7000 or Cisco 7500 series routers with a VIP2-40 or greater interface processor, you can classify packets by group, to allow you to partition your network into multiple priority levels or classes of service. This classification is achieved by setting IP precedences based on different criteria for use by other QoS features such as Weighted Random Early Detection (WRED) or weighted fair queueing (WFQ).

Configuring CAR and DCAR for All IP Traffic

SUMMARY STEPS

1. Router(config)# **interface***interface-type interface-number*
2. Router(config-if)# **rate-limit** {**input** | **output**} *bps burst-normal burst-max* **conform-action** *action* **exceed-action** *action*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **interface***interface-type interface-number*	Specifies the interface or subinterface. This command puts the router in interface configuration mode.

	Command or Action	Purpose
Step 2	Router(config-if)# **rate-limit** {**input** | **output**} *bps burst-normal burst-max* **conform-action** *action* **exceed-action** *action*	Specifies a basic CAR policy for allConfiguring CAR and DCAR for All IP Traffic, page 3ef"> Table 1 for a description of conform and exceed *action* keywords.

Configuring CAR and DCAR Policies

SUMMARY STEPS

1. Router(config-if)# **interface** *interface-type interface-number*
2. Router(config-if)# **rate-limit** {**input** | **output**} [**access-group** [**rate-limit**] *acl-index*] *bps burst-normal burst-max* **conform-action** *action* **exceed-action** *action*
3. Router(config-if) **exit**
4. Router(config)# **access-list rate-limit** *acl-index* {*precedence* | *mac-address*| **mask** *prec-mask*}
5. Do one of the following:
   - Router(config)# **access-list** *acl-index* {**deny** | **permit**} *source*[*source-wildcard*]
   - Router(config)# **access-list** *acl-index* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard*[**precedence** *precedence*][**tos** *tos*] [**log**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config-if)# **interface** *interface-type interface-number*	Specifies the interface or subinterface. This command puts the router in interface configuration mode.
Step 2	Router(config-if)# **rate-limit** {**input** | **output**} [**access-group** [**rate-limit**] *acl-index*] *bps burst-normal burst-max* **conform-action** *action* **exceed-action** *action*	Specifies the rate policy for each particular class of traffic. See Configuring CAR and DCAR Policies, page 4 for a description of the **rate-limit** command action keywords. Repeat this command for each different class of traffic.
Step 3	Router(config-if) **exit**	(Optional) Returns to global configuration mode. **Note** This change in configuration mode is needed only if you complete optional Configuring CAR and DCAR Policies, page 4 or Configuring CAR and DCAR Policies, page 4.
Step 4	Router(config)# **access-list rate-limit** *acl-index* {*precedence* | *mac-address*| **mask** *prec-mask*}	(Optional) Specifies a rate-limited access list. Repeat this command if you wish to specify a new access list.

	Command or Action	Purpose
Step 5	Do one of the following: • Router(config)# **access-list** *acl-index* {**deny** | **permit**} *source*[*source-wildcard*] • Router(config)# **access-list** *acl-index* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard*[**precedence** *precedence*][**tos** *tos*] [**log**]	(Optional) Specifies a standard or extended access list. Repeat this command to further configure the access list or specify a new access list.

Configuring a Class-Based DCAR Policy

SUMMARY STEPS

1. Router(config-if)# **interface** *interface-type interface-number*
2. Router(config-if)# **rate-limit** {**input** | **output**} [**access-group** [**rate-limit**] *acl-index*] *bps burst-normal burst-max* **conform-action** *action* **exceed-action** *action*
3. Router(config-if)# **random-detect precedence** *precedence min-threshold max-threshold mark-prob-denominator*
4. Do one of the following:
   - Router(config-if)# **access-list** *acl-index* {**deny** | **permit**} *source*[*source-wildcard*]
   - Router(config-if)# **access-list** *acl-index* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard*[**precedence** *precedence*] [**tos** *tos*] [**log**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config-if)# **interface** *interface-type interface-number*	Specifies the interface or subinterface. This command puts the router in interface configuration mode.
Step 2	Router(config-if)# **rate-limit** {**input** | **output**} [**access-group** [**rate-limit**] *acl-index*] *bps burst-normal burst-max* **conform-action** *action* **exceed-action** *action*	Specifies the rate policy for each particular class of traffic. See Configuring a Class-Based DCAR Policy, page 5 for a description of the **rate-limit** command action keywords. Repeat this command for each different class of traffic.
Step 3	Router(config-if)# **random-detect precedence** *precedence min-threshold max-threshold mark-prob-denominator*	Configures WRED and specifies parameters for packets with specific IP Precedence.

	Command or Action	Purpose
Step 4	Do one of the following: • Router(config-if)# **access-list** *acl-index* {**deny** | **permit**} *source*[*source-wildcard*] • Router(config-if)# **access-list** *acl-index* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard*[**precedence** *precedence*] [**tos** *tos*] [**log**]	(Optional) Specifies a standard or extended access list. Repeat this command to further configure the access list or specify a new access list.

Monitoring CAR and DCAR

Command	Purpose
`Router#` **show access-lists**	Displays the contents of current IP and rate-limited access lists.
`Router#` **show access-lists rate-limit** [*access-list-number*]	Displays information about rate-limited access lists.
`Router#` **show interfaces** [*interface-type interface-number*] **rate-limit**	Displays information about an interface configured for CAR.

CAR and DCAR Configuration Examples

Example Subrate IP Services

The following example illustrates how to configure a basic CAR policy that allows all IP traffic. In the example, the network operator delivers a physical T3 link to the customer, but offers a less expensive 15 Mbps subrate service. The customer pays only for the subrate bandwidth, which can be upgraded with additional access bandwidth based on demand. The CAR policy limits the traffic rate available to the customer and delivered to the network to the agreed upon rate limit, plus the ability to temporarily burst over the limit.

```
interface hssi 0/0/0
rate-limit output 15000000 2812500 5625000 conform-action transmit exceed-action drop
ip address 10.1.0.9 255.255.255.0
```

Example Input and Output Rate Limiting on an Interface

In this example, a customer is connected to an Internet service provider (ISP) by a T3 link. The ISP wants to rate limit transmissions from the customer to 15 Mbps of the 45 Mbps. In addition, the customer is allowed to send bursts of 2,812,500 bytes. All packets exceeding this limit are dropped. The following commands are configured on the High-Speed Serial Interface (HSSI) of the ISP connected to the customer:

```
interface Hssi0/0/0
 description 45Mbps to R1
 rate-limit input 15000000 2812500 2812500 conform-action transmit exceed-action drop
 ip address 200.200.14.250 255.255.255.252
 rate-limit output 15000000 2812500 2812500 conform-action transmit exceed-action drop
```

The following sample output shows how to verify the configuration and monitor CAR statistics using the **show interfaces rate-limit** command:

```
Router# show interfaces hssi 0/0/0 rate-limit
Hssi0/0/0 45Mbps to R1
 Input
  matches: all traffic
   params: 15000000 bps, 2812500 limit, 2812500 extended limit
   conformed 8 packets, 428 bytes; action: transmit
   exceeded 0 packets, 0 bytes; action: drop
   last packet: 8680ms ago, current burst: 0 bytes
   last cleared 00:03:59 ago, conformed 0 bps, exceeded 0 bps
 Output
  matches: all traffic
   params: 15000000 bps, 2812500 limit, 2812500 extended limit
   conformed 0 packets, 0 bytes; action: transmit
   exceeded 0 packets, 0 bytes; action: drop
   last packet: 8680ms ago, current burst: 0 bytes
   last cleared 00:03:59 ago, conformed 0 bps, exceeded 0 bps
```

Example Rate Limiting in an IXP

The following example uses rate limiting to control traffic in an Internet Exchange Point (IXP). Because an IXP comprises many neighbors around an FDDI ring, MAC address rate-limited access lists are used to control traffic from a particular ISP. Traffic from one ISP (at MAC address 00e0.34b0.7777) is compared to a rate limit of 80 Mbps of the 100 Mbps available on the FDDI connection. Traffic that conforms to this rate is sent. Nonconforming traffic is dropped.

```
interface Fddi2/1/0
 rate-limit input access-group rate-limit 100 80000000 15000000 30000000 conform-action
transmit exceed-action drop
 ip address 200.200.6.1 255.255.255.0
!
access-list rate-limit 100 00e0.34b0.7777
```

The following sample output shows how to verify the configuration and monitor the CAR statistics using the **show interfaces rate-limit** command:

```
Router# show interfaces fddi2/1/0 rate-limit
Fddi2/1/0
 Input
  matches: access-group rate-limit 100
   params: 800000000 bps, 15000000 limit, 30000000 extended limit
   conformed 0 packets, 0 bytes; action: transmit
   exceeded 0 packets, 0 bytes; action: drop
   last packet: 4737508ms ago, current burst: 0 bytes
   last cleared 01:05:47 ago, conformed 0 bps, exceeded 0 bps
```

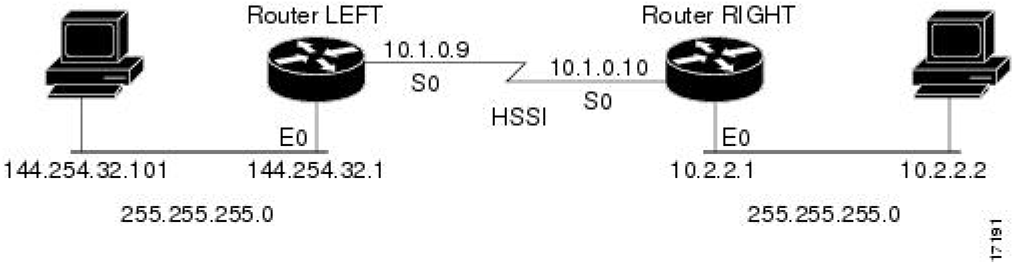
Example Rate Limiting by Access List

The following example shows how CAR can be used to limit the rate by application to ensure capacity for other traffic including mission-critical applications:

- All World Wide Web traffic is sent. However, the IP precedence for Web traffic that conforms to the first rate policy is set to 5. For nonconforming Web traffic, the IP precedence is set to 0 (best effort).
- File Transfer Protocol (FTP) traffic is sent with an IP precedence of 5 if it conforms to the second rate policy. If the FTP traffic exceeds the rate policy, it is dropped.
- Any remaining traffic is limited to 8 Mbps, with a normal burst size of 15,000 bytes and an Excess Burst size of 30,000 bytes. Traffic that conforms is sent with an IP precedence of 5. Traffic that does not conform is dropped.

The figure below illustrates the configuration. Notice that two access lists are created to classify the Web and FTP traffic so that they can be handled separately by CAR.

Figure 1



Router LEFT Configuration

```
interface Hssi0/0/0
description 45Mbps to R2
rate-limit output access-group 101 20000000 3750000 7500000 conform-action set-prec-
transmit 5 exceed-action set-prec-transmit 0
rate-limit output access-group 102 10000000 1875000 3750000 conform-action
set-prec-transmit 5 exceed-action drop
rate-limit output 8000000 1500000 3000000 conform-action set-prec-transmit 5
exceed-action drop
ip address 10.1.0.9 255.255.255.0
!
access-list 101 permit tcp any any eq www
access-list 102 permit tcp any any eq ftp
```

The following sample output shows how to verify the configuration and monitor CAR statistics using the **show interfaces rate-limit** command:

```
Router# show interfaces hssi 0/0/0 rate-limit
Hssi0/0/0 45Mbps to R2
 Input
  matches: access-group 101
   params: 20000000 bps, 3750000 limit, 7500000 extended limit
   conformed 3 packets, 189 bytes; action: set-prec-transmit 5
   exceeded 0 packets, 0 bytes; action: set-prec-transmit 0
   last packet: 309100ms ago, current burst: 0 bytes
   last cleared 00:08:00 ago, conformed 0 bps, exceeded 0 bps
  matches: access-group 102
   params: 10000000 bps, 1875000 limit, 3750000 extended limit
   conformed 0 packets, 0 bytes; action: set-prec-transmit 5
   exceeded 0 packets, 0 bytes; action: drop
   last packet: 19522612ms ago, current burst: 0 bytes
   last cleared 00:07:18 ago, conformed 0 bps, exceeded 0 bps
  matches: all traffic
   params: 8000000 bps, 1500000 limit, 3000000 extended limit
   conformed 5 packets, 315 bytes; action: set-prec-transmit 5
   exceeded 0 packets, 0 bytes; action: drop
   last packet: 9632ms ago, current burst: 0 bytes
   last cleared 00:05:43 ago, conformed 0 bps, exceeded 0 bps
```

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Marking Network Traffic

Marking network traffic allows you to set or modify the attributes for traffic (that is, packets) belonging to a specific class or category. When used in conjunction with network traffic classification, marking network traffic is the foundation for enabling many quality of service (QoS) features on your network. This module contains conceptual information and the configuration tasks for marking network traffic.

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.

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.

Prerequisites for Marking Network Traffic

In order to mark network traffic, Cisco Express Forwarding must be configured on both the interface receiving the traffic and the interface sending the traffic.

Restrictions for Marking Network Traffic

Traffic marking can be configured on an interface, a subinterface, or an ATM permanent virtual circuit (PVC). Marking network traffic is not supported on the following interfaces:

- Any interface that does not support Cisco Express Forwarding
- ATM switched virtual circuit (SVC)

- Fast EtherChannel
- PRI
- Tunnel

Information About Marking Network Traffic

Purpose of Marking Network Traffic

Traffic marking is a method used to identify certain traffic types for unique handling, effectively partitioning network traffic into different categories.

After the network traffic is organized into classes by traffic classification, traffic marking allows you to mark (that is, set or change) a value (attribute) for the traffic belonging to a specific class. For instance, you may want to change the class of service (CoS) value from 2 to 1 in one class, or you may want to change the differentiated services code point (DSCP) value from 3 to 2 in another class. In this module, these values are referred to as attributes.

Attributes that can be set and modified include the following:

- Cell loss priority (CLP) bit
- CoS value of an outgoing packet
- Discard eligible (DE) bit setting in the address field of a Frame Relay frame
- Discard-class value
- DSCP value in the type of service (ToS) byte
- MPLS EXP field value in the topmost label on either an input or an output interface
- Multiprotocol Label Switching (MPLS) experimental (EXP) field on all imposed label entries
- Precedence value in the packet header
- QoS group identifier (ID)
- ToS bits in the header of an IP packet

Benefits of Marking Network Traffic

Improved Network Performance

Traffic marking allows you to fine-tune the attributes for traffic on your network. This increased granularity helps single out traffic that requires special handling, and thus, helps to achieve optimal application performance.

Traffic marking allows you to determine how traffic will be treated, based on how the attributes for the network traffic are set. It allows you to segment network traffic into multiple priority levels or classes of service based on those attributes, as follows:

- Traffic marking is often used to set the IP precedence or IP DSCP values for traffic entering a network. Networking devices within your network can then use the newly marked IP precedence values to determine how traffic should be treated. For example, voice traffic can be marked with a particular IP precedence or DSCP and low latency queuing (LLQ) can then be configured to put all packets of that mark into a priority queue. In this case, the marking was used to identify traffic for LLQ.

- Traffic marking can be used to identify traffic for any class-based QoS feature (any feature available in policy-map class configuration mode, although some restrictions exist).
- Traffic marking can be used to assign traffic to a QoS group within a router. The router can use the QoS groups to determine how to prioritize traffic for transmission. The QoS group value is usually used for one of the two following reasons:
  - To leverage a large range of traffic classes. The QoS group value has 100 different individual markings, as opposed to DSCP and Precedence, which have 64 and 8, respectively.
  - If changing the Precedence or DSCP value is undesirable.
- If a packet (for instance, in a traffic flow) needs to be marked to differentiate user-defined QoS services is leaving a router and entering a switch, the router can set the CoS value of the traffic, because the switch can process the Layer 2 CoS header marking. Alternatively, the Layer 2 CoS value of the traffic leaving a switch can be mapped to the Layer 3 IP or MPLS value.
- Weighted random early detection (WRED) uses precedence values or DSCP values to determine the probability that the traffic will be dropped. Therefore, the Precedence and DSCP can be used in conjunction with WRED.

Two Methods for Marking Traffic Attributes

There are two methods for specifying and marking traffic attributes:

- You can specify and mark the traffic attribute by using a **set** command.

With this method, you configure individual **set** commands for the traffic attribute that you want to mark.

- You can specify and mark the traffic attribute by creating a mapping table (called a "table map").

With this method, you configure the traffic attributes that you want to mark once in a table map and then the markings can be propagated throughout the network.

These methods are further described in the sections that follow.

- Method One Using a set Command, page 13
- Method Two Using a Table Map, page 14
- Traffic Marking Procedure Flowchart, page 16

Method One Using a set Command

You specify the traffic attribute you want to change with a **set**command configured in a policy map. The table below lists the available **set**commands and the corresponding attribute. The table also includes the network layer and the network protocol typically associated with the traffic attribute.

Table 2 set Commands and Corresponding Traffic Attribute, Network Layer, and Protocol

set Commands[1]	Traffic Attribute	Network Layer	Protocol
set atm-clp	CLP bit	Layer 2	ATM

[1] Cisco IOS set commands can vary by release. For more information, see the command documentation for the Cisco IOS release that you are using

set Commands[1]	Traffic Attribute	Network Layer	Protocol
set cos	Layer 2 CoS value of the outgoing traffic	Layer 2	ATM, Frame Relay
set discard-class	discard-class value	Layer 2	ATM, Frame Relay
set dscp	DSCP value in the ToS byte	Layer 3	IP
set fr-de	DE bit setting in the address field of a Frame Relay frame	Layer 2	Frame Relay
set ip tos (route-map)	ToS bits in the header of an IP packet	Layer 3	IP
set mpls experimental imposition	MPLS EXP field on all imposed label entries	Layer 3	MPLS
set mpls experimental topmost	MPLS EXP field value in the topmost label on either an input or an output interface	Layer 3	MPLS
set precedence	precedence value in the packet header	Layer 3	IP
set qos-group	QoS group ID	Layer 3	IP, MPLS

If you are using individual **set** commands, those **set** commands are specified in a policy map. The following is a sample of a policy map configured with one of the **set** commands listed in the table above.

In this sample configuration, the **set atm-clp**command has been configured in the policy map (policy1) to mark the CLP attribute.

```
policy-map policy1
 class class1
 set atm-clp
 end
```

Method Two Using a Table Map

You can create a table map that can be used to mark traffic attributes. A table map is a kind of two-way conversion chart that lists and maps one traffic attribute to another. A table map supports a many-to-one type of conversion and mapping scheme. The table map establishes a to-from relationship for the traffic attributes and defines the change to be made to the attribute. That is, an attribute is set *to* one value that is taken *from* another value. The values are based on the specific attribute being changed. For instance, the Precedence attribute can be a number from 0 to 7, while the DSCP attribute can be a number from 0 to 63.

[1] **Cisco IOS set commands can vary by release. For more information, see the command documentation for the Cisco IOS release that you are using**

The following is a sample table map configuration:

```
table-map table-map1

 map from 0 to 1

 map from 2 to 3

 exit
```

The table below lists the traffic attributes for which a to-from relationship can be established using the table map.

Table 3 Traffic Attributes for Which a To-From Relationship Can Be Established

The "To" Attribute	The "From" Attribute
Precedence	CoS
	QoS group
DSCP	CoS
	QoS group
CoS	Precedence
	DSCP
QoS group	Precedence
	DSCP
	MPLS EXP topmost
MPLS EXP topmost	QoS group
MPLS EXP imposition	Precedence
	DSCP

Once the table map is created, you configure a policy map to use the table map. In the policy map, you specify the table map name and the attributes to be mapped by using the **table** keyword and the *table-map-name* argument with one of the commands listed in the table below.

Table 4 Commands Used in Policy Maps to Map Attributes

Command Used in Policy Maps	Maps These Attributes
set cos dscp table *table-map-name*	CoS to DSCP
set cos precedence table *table-map-name*	CoS to Precedence
set dscp cos table *table-map-name*	DSCP to CoS

Command Used in Policy Maps	Maps These Attributes
set dscp qos-group table *table-map-name*	DSCP to qos-group
set mpls experimental imposition dscp table *table-map-name*	MPLS EXP imposition to DSCP
set mpls experimental imposition precedence table *table-map-name*	MPLS EXP imposition to precedence
set mpls experimental topmost qos-group table *table-map-name*	MPLS EXP topmost to QoS-group
set precedence cos table *table-map-name*	Precedence to CoS
set precedence qos-group table *table-map-name*	Precedence to QoS-group
set qos-group dscp table *table-map-name*	QoS-group to DSCP
set qos-group mpls exp topmost table *table-map-name*	QoS-group to MPLS EXP topmost
set qos-group precedence table *table-map-name*	QoS-group to Precedence

The following is an example of a policy map (policy2) configured to use the table map (table-map1) created earlier:

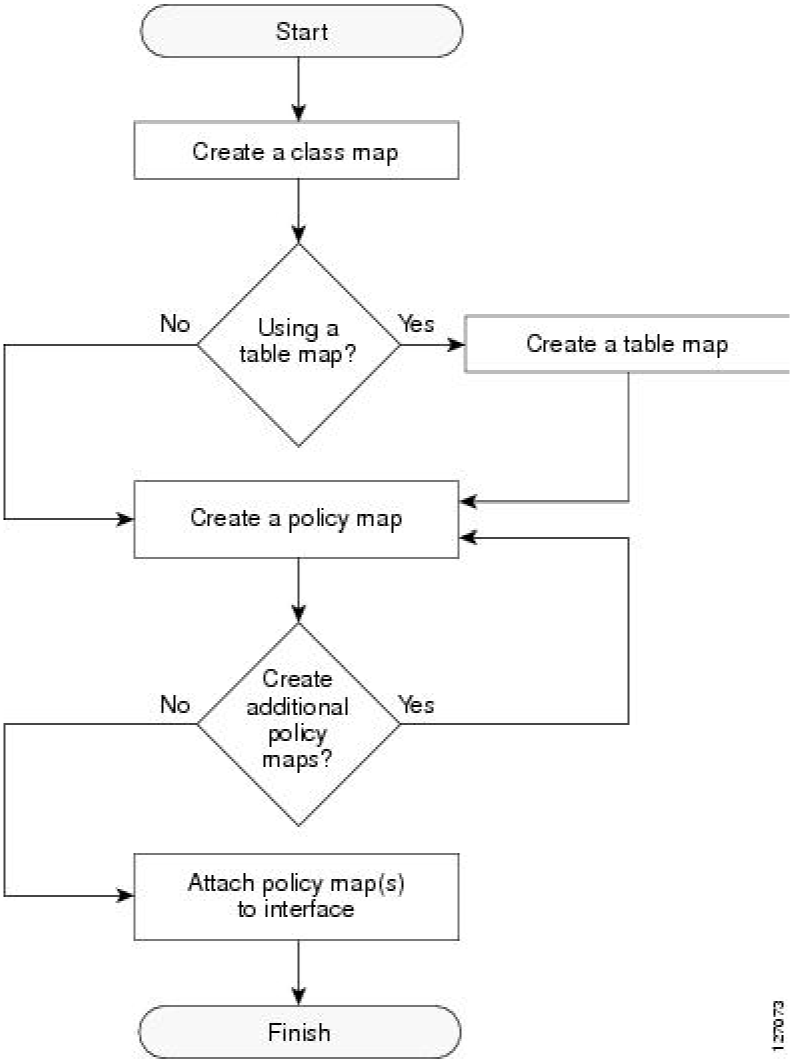
```
policy map policy2

 class class-default

 set cos dscp table table-map1

 exit
```

In this example, a mapping relationship was created between the CoS attribute and the DSCP attribute as defined in the table map.

Traffic Marking Procedure Flowchart

The figure below illustrates the order of the procedures for configuring traffic marking.

Figure 2



MQC and Network Traffic Marking

To configure network traffic marking, you use the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC).

The MQC is a CLI structure that allows you to complete the following tasks:

- Specify the matching criteria used to define a traffic class.
- Create a traffic policy (policy map). The traffic policy defines the QoS policy actions to be taken for each traffic class.
- Apply the policy actions specified in the policy map to an interface, subinterface, or ATM PVC by using the **service-policy** command.

Traffic Classification Compared with Traffic Marking

Traffic classification and traffic marking are closely related and can be used together. Traffic marking can be viewed as an additional action, specified in a policy map, to be taken on a traffic class.

Traffic classification allows you to organize into traffic classes on the basis of whether the traffic matches specific criteria. For example, all traffic with a CoS value of 2 is grouped into one class, and traffic with DSCP value of 3 is grouped into another class. The match criterion is user-defined.

After the traffic is organized into traffic classes, traffic marking allows you to mark (that is, set or change) an attribute for the traffic belonging to that specific class. For instance, you may want to change the CoS value from 2 to 1, or you may want to change the DSCP value from 3 to 2.

The match criteria used by traffic classification are specified by configuring a **match** command in a class map. The marking action taken by traffic marking is specified by configuring a **set** command in a policy map. These class maps and policy maps are configured using the MQC.

The table below compares the features of traffic classification and traffic marking.

Table 5 Traffic Classification Compared with Traffic Marking

	Traffic Classification	Traffic Marking
Goal	Groups network traffic into specific traffic classes on the basis of whether the traffic matches the user-defined criterion.	After the network traffic is grouped into traffic classes, modifies the attributes for the traffic in a particular traffic class.
Configuration Mechanism	Uses class maps and policy maps in the MQC.	Uses class maps and policy maps in the MQC.
CLI	In a class map, uses **match** commands (for example, **match cos**) to define the traffic matching criterion.	Uses the traffic classes and matching criterion specified by traffic classification. In addition, uses **set** commands (for example, **set cos**) in a policy map to modify the attributes for the network traffic. If a table map was created, uses the **table** keyword and *table-map-name* argument with the **set** commands (for example, **set cos precedence table** *table-map-name*) in the policy map to establish the to-from relationship for mapping attributes.

How to Mark Network Traffic

- Creating a Class Map for Marking Network Traffic, page 19
- Creating a Table Map for Marking Network Traffic, page 20
- Creating a Policy Map for Applying a QoS Feature to Network Traffic, page 22
- Attaching the Policy Map to an Interface, page 25

- Configuring QoS When Using IPsec VPNs, page 27

Creating a Class Map for Marking Network Traffic

Note The **match fr-dlci** command is included in the steps below. The **match fr-dlci**command is just an example of one of the **match** commands that can be used. See the command documentation for the Cisco IOS release that you are using for a complete list of **match** commands.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** *class-map-name* [**match-all**| **match-any**]
4. **match fr-dlci** *dlci-number*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**class-map** *class-map-name* [**match-all**| **match-any**]
	Example:
	`Router(config)# class-map class1`
		Creates a class map to be used for matching traffic to a specified class and enters class-map configuration mode.
		• Enter the class map name.

	Command or Action	Purpose
Step 4	**match fr-dlci** *dlci-number* **Example:** `Router(config-cmap)# match fr-dlci 500`	(Optional) Specifies the Frame Relay DLCI number as a match criterion in a class map. **Note** The **match fr-dlci** command classifies traffic on the basis of the Frame Relay DLCI number. The **match fr-dlci**command is just an example of one of the **match** commands that can be used. The **match** commands vary by Cisco IOS release. See the command documentation for the Cisco IOS release that you are using for a complete list of **match** commands.

Step 5	**end**
	Example:
	`Router(config-cmap)# end`
		(Optional) Returns to privileged EXEC mode.

Creating a Table Map for Marking Network Traffic

Note If you are not using a table map, skip this procedure and advance to Creating a Policy Map for Applying a QoS Feature to Network Traffic, page 22.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **table-map** *table-map-name* **map from** *from-value* **to** *to-value* [**default** *default-action-or-value*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**table-map** *table-map-name* **map from** *from-value* **to** *to-value* [**default** *default-action-or-value*] **Example:** **Example:** `Router(config)# table-map table-map1 map from 2 to 1`	Creates a table map using the specified name and enters tablemap configuration mode. • Enter the name of the table map you want to create. • Enter each value mapping on a separate line. Enter as many separate lines as needed for the values you want to map. • The **default** keyword and *default-action-or-value* argument set the default value (or action) to be used if a value is not explicitly designated.

Step 4	**end**
	Example:
	`Router(config-tablemap)#`
	`end`
		(Optional) Exits tablemap configuration mode and returns to privileged EXEC mode.

Creating a Policy Map for Applying a QoS Feature to Network Traffic

Note

- The **set atm-clp** command is supported on the following adapters only:
  - Enhanced ATM Port Adapter (PA-A3)
  - ATM Inverse Multiplexer over ATM Port Adapter with 8 T1 Ports (PA-A3-8T1IMA)
  - ATM Inverse Multiplexer over ATM Port Adapter with 8 E1 Ports (PA-A3-8E1IMA)
- Before modifying the encapsulation type from IEEE 802.1 Q to ISL, or vice versa, on a subinterface, detach the policy map from the subinterface. After changing the encapsulation type, reattach the policy map.
- A policy map containing the **set qos-group** command can only be attached as an input traffic policy. QoS group values are not usable for traffic leaving a router.
- A policy map containing the **set cos**command can only be attached as an output traffic policy.
- A policy map containing the **set atm-clp** command can be attached as an output traffic policy only. The **set atm-clp** command does not support traffic that originates from the router.

Note The **set cos**command and **set cos dscp table** *table-map-name*command are shown in the steps t The **set cos**command and **set cos dscp table** *table-map-name*command are examples the **set** co that can be used when marking traffic. Other **set** commands can be used. For a list of other **set** c see Creating a Policy Map for Applying a QoS Feature to Network Traffic, page 22 and Creati Map for Applying a QoS Feature to Network Traffic, page 22.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** {*class-name* | **class-default**}
5. **set cos** *cos-value*
6.
7. **set cos dscp table** *table-map-name*
8. Router(config-pmap-c)# set cos 2
9.
10. Router(config-pmap-c)# set cos dscp table table-map1
11. **end**
12. **show policy-map**
13.
14. **show policy-map** *policy-map* **class** *class-name*
15. Router# show policy-map
16.
17. Router# show policy-map policy1 class class1
18. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map policy1`	Specifies the name of the policy map created earlier and enters policy-map configuration mode. - Enter the policy map name.

	Command or Action	Purpose

Step 4	**class** {*class-name* | **class-default**}
	Example:
	`Router(config-pmap)# class class1`
		Specifies the name of the class whose policy you want to create and enters policy-map class configuration mode. This class is associated with the class map created earlier.
		- Enter the name of the class or enter the **class-default**keyword.

Step 5	**set cos** *cos-value* **Example:**	(Optional) Sets the CoS value in the type of service (ToS) byte. **Note** The **set cos**command is an example of one of the **set** commands that can be used when marking traffic. Other **set** commands can be used. For a list of other **set** commands, see Creating a Policy Map for Applying a QoS Feature to Network Traffic, page 22.
Step 6		or
Step 7	**set cos dscp table** *table-map-name* **Example:**	(Optional) If a table map has been created earlier, sets the CoS value based on the DSCP value (or action) defined in the table map. **Note** The **set cos dscp table** *table-map-name*command is an example of one of the commands that can be used. For a list of other commands, see Creating a Policy Map for Applying a QoS Feature to Network Traffic, page 22.
Step 8	Router(config-pmap-c)# set cos 2	
Step 9		
Step 10	Router(config-pmap-c)# set cos dscp table table-map1	

Step 11	**end**
	Example:
	`Router(config-pmap-c)# end`
		Returns to privileged EXEC mode.

Step 12	**show policy-map**	(Optional) Displays all configured policy maps.
Step 13		or
Step 14	**show policy-map** *policy-map* **class** *class-name* **Example:**	(Optional) Displays the configuration for the specified class of the specified policy map. - Enter the policy map name and the class name.
Step 15	Router# show policy-map	
Step 16		
Step 17	Router# show policy-map policy1 class class1	

	Command or Action	Purpose
Step 18	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

- What to Do Next, page 25

What to Do Next

Create and configure as many policy maps as you need for your network. To create and configure additional policy maps, repeat the steps in the Creating a Policy Map for Applying a QoS Feature to Network Traffic, page 22. Then attach the policy maps to the appropriate interface, following the instructions in the Attaching the Policy Map to an Interface, page 25.

Attaching the Policy Map to an Interface

Note Depending on the needs of your network, policy maps can be attached to an interface, a subinterface, or an ATM permanent virtual circuit (PVC).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [**name-tag**]
4. **pvc** [*name*] *vpi* / *vci* [**ilmi**|**qsaal**|**smds**| **l2transport**]
5. **exit**
6. **service-policy** {**input** | **output**} *policy-map-name*
7. **end**
8. **show policy-map interface** *type number*
9. **exit**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [**name-tag**] **Example:** `Router(config)# interface serial4/0`	Configures an interface type and enters interface configuration mode. • Enter the interface type and number.
Step 4	**pvc** [*name*] *vpi* / *vci* [**ilmi**|**qsaal**|**smds**|**l2transport**] **Example:** `Router(config-if)# pvc cisco 0/16`	(Optional) Creates or assigns a name to an ATM permanent virtual circuit (PVC), specifies the encapsulation type on an ATM PVC, and enters ATM virtual circuit configuration mode. • Enter the PVC name, the ATM network virtual path identifier, and the network virtual channel identifier. **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, advance to Attaching the Policy Map to an Interface, page 25.
Step 5	**exit** **Example:** `Router(config-atm-vc)# exit`	(Optional) Returns to interface configuration mode. **Note** This step is required only if you are attaching the policy map to an ATM PVC and you completed Attaching the Policy Map to an Interface, page 25. If you are not attaching the policy map to an ATM PVC, advance to Attaching the Policy Map to an Interface, page 25.
Step 6	**service-policy** {**input** | **output**} *policy-map-name* **Example:** `Router(config-if)# service-policy input policy1`	Attaches a policy map to an input or output interface. • Enter the policy map name. **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.

Step 7	**end**
	Example:
	`Router(config-if)# end`
		Returns to privileged EXEC mode.

	Command or Action	Purpose

Step 8	**show policy-map interface** *type number*
	Example:
	`Router#`
	`show policy-map interface serial4/0`
		(Optional) Displays traffic statistics of all classes configured for all service policies on the specified interface, subinterface, or PVC on the interface.
		When there are multiple instances of the same class in a policy-map, and this policy-map is attached to an interface,
		show policy-map interface `<interface_name>` **output class** `<class-name>`
		returns only the first instance.
		• Enter the interface type and number.

Step 9	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Configuring QoS When Using IPsec VPNs

Note This task is required only if you are using IPsec Virtual Private Networks (VPNs). Otherwise, this task is not necessary. For information about IPsec VPNs, see the "Configuring Security for VPNs with IPsec" module.

Note This task uses the **qos pre-classify** command to enable QoS preclassification for the packet. QoS preclassification is not supported for all fragmented packets. If a packet is fragmented, each fragment might received different preclassifications.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num*
4. **exit**
5. **interface** *type number* [**name-tag**]
6. **qos pre-classify**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* **Example:** `Router(config)# crypto map mymap 10`	Enters crypto map configuration mode and creates or modifies a crypto map entry. • Enter the crypto map name and sequence number.
Step 4	**exit** **Example:** `Router(config-crypto-map)# exit`	Returns to global configuration mode.
Step 5	**interface** *type number* [**name-tag**] **Example:** `Router(config)# interface serial4/0`	Configures an interface type and enters interface configuration mode. • Enter the interface type and number.
Step 6	**qos pre-classify** **Example:** `Router(config-if)# qos pre-classify`	Enables QoS preclassification.

Step 7	**end**
	Example:
	`Router(config-if)# end`
		(Optional) Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for Marking Network Traffic

Example Creating a Class Map for Marking Network Traffic

The following is an example of creating a class map to be used for marking network traffic. In this example, a class called class1 has been created. The traffic with a Frame Relay DLCI value of 500 will be put in this class.

```
Router> enable

Router# configure terminal

Router(config)# class-map class1

Router(config-cmap)# match fr-dlci 500

Router(config-cmap)# end
```

Example Table Map for Marking Network Traffic

In the following example, the **table-map** (value mapping) command has been used to create and configure a table map called table-map1. This table map will be used to establish a to-from relationship between one traffic-marking value and another.

In table-map1, a traffic-marking value of 0 will be mapped to a value of 1.

```
Router> enable
Router# configure terminal
Router(config)# table-map
 table-map1 map from 0 to 1

Router(config-tablemap)#
 end
```

Example Policy Map for Applying a QoS Feature to Network Traffic

Policy Map Configured to Use set Command

The following is an example of creating a policy map to be used for traffic marking. In this example, a policy map called policy1 has been created, and the **set dscp**command has been configured for class1.

```
Router> enable
```

```
Router# configure terminal
Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# set dscp 2
Router(config-pmap-c)# end
```

Policy Map Configured to Use a Table Map

A policy map called policy1 has been created and configured to use table-map1 for setting the precedence value. In this example, the CoS value will be set according to the DSCP value defined in table-map1 created previously.

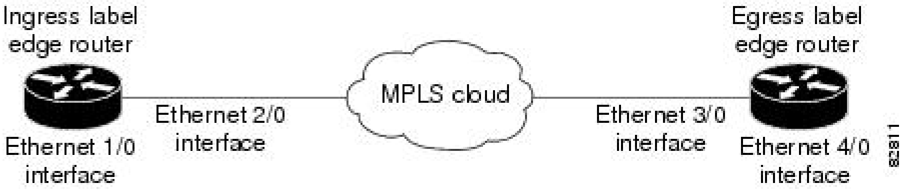
```
Router(config)# policy map policy1

Router(config-pmap)# class class-default

Router(config-pmap-c)#
 set cos dscp table table-map1

Router(config-pmap-c)#
 end
```

Note As an alternative to configuring the **set cos dscp table table-map1** command shown in the example, you could configure the command without specifying the **table** keyword and the applicable *table-map-nam*e argument (that is, you could configure the **set cos dscp**command). When the command is configured without the **table** keyword and applicable table map name, the values are copied from the specified categories. In this case, the DSCP value is copied and used to set the CoS value. When the DSCP value is copied and used for the CoS value only the *first 3 bits* (that is, the class selector bits) of the DSCP value will be used to set the CoS value. For example, if the DSCP value is EF (101110), the first 3 bits of this DSCP value will be used to set the CoS value, resulting in a CoS value of 5 (101).

Policy Map Configured to Use a Table Map for Mapping MPLS EXP Values

This section contains an example of a policy map configured to map MPLS experimental (EXP) values. The figure below illustrates the network topology for this configuration example.

Figure 3



For this configuration example, traffic arrives at the input interface (an Ethernet 1/0 interface) of the ingress label edge router (LER). The precedence value is copied and used as the MPLS EXP value of the traffic when the MPLS label is imposed. This label imposition takes place at the ingress LER.

The traffic leaves the ingress LER through the output interface (an Ethernet 2/0 interface), traverses through the network backbone into the MPLS cloud, and enters the egress LER.

At the input interface of the egress LER (an Ethernet 3/0 interface), the MPLS EXP value is copied and used as the QoS group value. At the output interface of the egress LER (an Ethernet 4/0 interface), the QoS group value is copied and used as the precedence value.

To accomplish configuration described above, three separate policy maps were required--policy1, policy2, and policy3. Each policy map is configured to convert and propagate different traffic-marking values.

The first policy map, policy1, is configured to copy the precedence value of the traffic and use it as the MPLS EXP value during label imposition.

```
Router(config)# policy-map policy1

Router(config-pmap)# class class-default

Router(config-pmap-c)#
 set mpls experimental imposition precedence

Router(config-pmap-c)#
 end
```

When the traffic leaves the LER through the output interface (the Ethernet 2/0 interface), the MPLS EXP value is copied from the precedence value during MPLS label imposition. Copying the MPLS EXP value from the precedence value ensures that the MPLS EXP value reflects the appropriate QoS treatment. The traffic now proceeds through the MPLS cloud into the egress LER.

A second policy map called policy2 has been configured to copy the MPLS EXP value in the incoming MPLS traffic to the QoS group value. The QoS group value is used for internal purposes only. The QoS group value can be used with output queueing on the output interface of the egress router. The QoS group value can also be copied and used as the precedence value, as traffic leaves the egress LER through the output interface (the Ethernet 4/0 interface).

```
Router(config)# policy-map policy2

Router(config-pmap)# class class-default

Router(config-pmap-c)#
 set qos-group mpls experimental topmost

Router(config-pmap-c)#
 end
```

A third policy map called policy3 has been configured to copy the internal QoS group value (previously based on the MPLS EXP value) to the precedence value. The QoS group value will be copied to the precedence value as the traffic leaves the egress LER through the output interface.

```
Router(config)# policy-map policy3

Router(config-pmap)# class class-default

Router(config-pmap-c)#
 set precedence qos-group

Router(config-pmap-c)#
 end
```

Configuring these policy maps as shown (and attaching them to interfaces as shown in Example Attaching the Policy Map to an Interface, page 32), causes the appropriate quality of service treatment to be preserved for the traffic as the traffic progresses along an IP network, through an MPLS cloud, and back again into an IP network.

Note This configuration could also have been accomplished by first creating a table map (used to map one value to another) and then specifying the **table** keyword and *table-map-name* argument in each of the **set** commands (for example, **set precedence qos-group table tablemap1**). In the MPLS configuration example, a table map was not created, and the **set** commands were configured without specifying the **table** keyword and *table-map-name* argument (for example, **set precedence qos-group)**. When the **set** commands are configured without specifying the **table** keyword and *table-map-name* argument, the values are copied from the specified categories. In this case, the QoS group value was copied and used to set the precedence value. When the DSCP value is copied and used for the MPLS EXP value, only the *first 3 bits* (that is, the class selector bits) of the DSCP value will be used to set the MPLS value.

Example Attaching the Policy Map to an Interface

The following is an example of attaching the policy map to the interface. In this example, the policy map called policy1 has been attached in the input direction of the Serial4/0 interface.

```
Router> enable
Router# configure terminal
Router(config)# interface serial4/0
Router(config-if)# service-policy input policy1
Router(config-if)# end
```

Example Configuring QoS When Using IPsec VPNs

The following is an example of configuring QoS when using IPsec VPNs. In this example, the **crypto map** command specifies the IPsec crypto map (mymap 10) to which the **qos pre-classify** command will be applied.

```
Router> enable
Router# configure terminal
Router(config)# crypto map mymap 10

Router(config-crypto-map)# qos pre-classify
Router(config-crypto-map)# exit
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
MQC	"Applying QoS Features Using the MQC" module
Classifying network traffic	"Classifying Network Traffic" module
IPsec and VPNs	"Configuring Security for VPNs with IPsec" module
Committed Access Rate (CAR)	"Configuring Committed Access Rate" module

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	--

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

Feature Information for Marking Network Traffic

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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.

Table 6 Feature Information for Marking Network Traffic

Feature Name	Software Releases	Feature Configuration Information
Enhanced Packet Marking	12.2(13)T	The Enhanced Packet Marking feature allows you to map and convert the marking of a packet from one value to another by using a kind of conversion chart called a table map. The table map establishes an equivalency from one value to another. For example, the table map can map and convert the class of service (CoS) value of a packet to the precedence value of the packet. This value mapping can be propagated for use on the network, as needed.

Feature Name	Software Releases	Feature Configuration Information
QoS Packet Marking	12.2(8)T	The QoS Packet Marking feature allows you to mark packets by setting the IP precedence bit or the IP differentiated services code point (DSCP) in the Type of Service (ToS) byte, and associate a local QoS group value with a packet.
Class-Based Marking	12.2(2)T	The Class-Based Packet Marking feature provides users with a user-friendly command-line interface (CLI) for efficient packet marking by which users can differentiate packets based on the designated markings.
Quality of Service for Virtual Private Networks	12.2(2)T	The QoS for VPNs feature provides a solution for making Cisco IOS QoS services operate in conjunction with tunneling and encryption on an interface. Cisco IOS software can classify packets and apply the appropriate QoS service before the data is encrypted and tunneled. The QoS for VPN feature allows users to look inside the packet so that packet marking can be done based on original port numbers and based on source and destination IP addresses. This allows the service provider to treat mission critical or multi-service traffic with higher priority across their network.
ATM Cell Loss Priority (CLP) Setting Class-Based Ethernet CoS Matching and Marking (802.1p and ISL CoS) Class-Based Marking Custom Queueing (CQ) PXF Based Frame Relay DE Bit Marking QoS Packet Marking	15.0(1)S	The ATM Cell Loss Priority (CLP) Setting, Class-Based Ethernet CoS Matching and Marking (802.1p and ISL CoS), Class-Based Marking, Custom Queueing (CQ), PXF Based Frame Relay DE Bit Marking, QoS Packet Marking and features were integrated into theCisco IOS Release 15.0(1)S release.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

QoS Tunnel Marking for GRE Tunnels

The QoS Tunnel Marking for GRE Tunnels feature introduces the capability to define and control the quality of service (QoS) for incoming customer traffic on the provider edge (PE) router in a service provider network.

Note For Cisco IOS Release 12.4(15)T2, the QoS Tunnel Marking for GRE Tunnels feature is supported only on platforms equipped with a Cisco MGX Route Processor Module (RPM-XF).

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.

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.

Prerequisites for QoS Tunnel Marking for GRE Tunnels

- You must configure Cisco Express Forwarding (CEF) on the interface before GRE tunnel marking can be used.
- You must determine the topology and interfaces that need to be configured to mark incoming traffic.

Restrictions for QoS Tunnel Marking for GRE Tunnels

- GRE tunnel marking is supported in input policy maps only and should not be configured for output policy maps.
- It is possible to configure GRE tunnel marking and the **ip tos** command at the same time. However, Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC) GRE tunnel marking has higher priority over IP ToS commands, meaning that tunnel marking always rewrites the IP header of the tunnel packet and overwrites the values set by **ip tos** commands. The priority of enforcement is as follows when these commands are used simultaneously:
- **set ip dscp tunnel** or **set ip precedence tunnel** (GRE tunnel marking)
- **ip tos reflect**
- **ip tos** *tos-value*

Note This is the designed behavior. We recommend that you configure only GRE tunnel marking and reconfigure any peers configured with the **ip tos** command to use GRE tunnel marking.

Information About QoS Tunnel Marking for GRE Tunnels

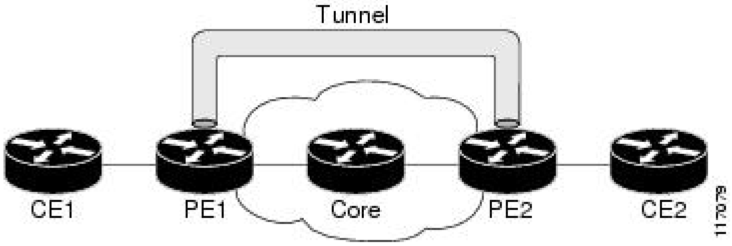
GRE Definition

Generic Routing Encapsulation (GRE) is a tunneling protocol developed by Cisco that can encapsulate a wide variety of protocol packet types inside IP tunnels, creating a virtual point-to-point link to Cisco routers at remote points over an IP internetwork.

GRE Tunnel Marking Overview

The QoS Tunnel Marking for GRE Tunnels feature allows you to define and control QoS for incoming customer traffic on the PE router in a service provider (SP) network. This feature lets you set (mark) either the IP precedence value or the differentiated services code point (DSCP) value in the header of an GRE tunneled packet. GRE tunnel marking can be implemented by using a QoS marking command, such as **set ip** {**dscp**| **precedence**} [**tunnel**], and it can also be implemented in QoS traffic policing. This feature simplifies administrative overhead previously required to control customer bandwidth by allowing you to mark the GRE tunnel header on the incoming interface on the PE routers.

The figure below shows traffic being received from the CE1 router through the incoming interface on the PE1 router on which tunnel marking occurs. The traffic is encapsulated (tunneled), and the tunnel header is marked on the PE1 router. The marked packets travel (tunnel) through the core and are decapsulated automatically on the exit interface of the PE2 router. This feature is designed to simplify classifying customer edge (CE) traffic and is configured only in the service provider network. This process is transparent to the customer sites. The CE1 and CE2 routers simply exist as a single network.



GRE Tunnel Marking and the MQC

To configure the tunnel marking for GRE tunnels, you must configure a class map and a policy map and then attach that policy map to the appropriate interface. These three tasks can be accomplished by using the MQC.

For information on using the MQC, see the "Applying QoS Features Using the MQC" module.

GRE Tunnel Marking and DSCP or IP Precedence Values

GRE tunnel marking is configured with the **set ip precedence tunnel** or **set ip dscp tunnel**command on PE routers that carry incoming traffic from customer sites. GRE tunnel marking allows you to mark the header of a GRE tunnel by setting a DSCP value from 0 to 63 or an IP precedence value from 0 to 7 to control GRE tunnel traffic bandwidth and priority.

GRE traffic can also be marked under traffic policing with the **set-dscp-tunnel-transmit**and the **set-prec-tunnel-transmit** actions (or keywords) of the **police** command. The tunnel marking value is from 0 to 63 for the **set-dscp-tunnel-transmit** actions and from 0 to 7 for the **set-prec-tunnel-transmit**command. Under traffic policing, tunnel marking can be applied with "conform" and "exceed" action statements, allowing you to automatically apply a different value for traffic that does not conform to the expected traffic rate.

After the tunnel header is marked, GRE traffic is carried through the tunnel and across the service provider network. This traffic is decapsulated on the interface of the PE router that carries the outgoing traffic to the other customer site. The configuration of GRE tunnel marking is transparent to customer sites. All internal configuration is preserved.

It is important to distinguish between the **set ip precedence** and **set ip dscp** commands and the **set ip precedence tunnel** and **set ip dscp tunnel** commands.

- The **set ip precedence** and **set ip dscp** commands are used to set the IP precedence value or DSCP value in the header of an IP packet.
- The **set ip precedence tunnel**and **set ip dscp tunnel**commands are used to set (mark) the IP precedence value or DSCP value in the tunnel header that encapsulates the GRE traffic.

Benefits of GRE Tunnel Marking

GRE tunnel marking provides a simple mechanism to control the bandwidth of customer GRE traffic. The QoS Tunnel Marking for GRE Tunnels feature is configured entirely within the service provider network and only on interfaces that carry incoming traffic on the PE routers.

- GRE Tunnel Marking and Traffic Policing, page 40

- GRE Tunnel Marking Values, page 40

GRE Tunnel Marking and Traffic Policing

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). If you use traffic policing in your network, you can also implement the GRE tunnel marking feature with the **set-dscp-tunnel-transmit**or **set-prec-tunnel-transmit** actions (or keywords) of the **police** command in policy-map class configuration mode. Under traffic policing, tunnel marking can be applied with "conform" and "exceed" action statements, allowing you to apply a different value automatically for traffic that does not conform to the expected traffic rate.

GRE Tunnel Marking Values

The range of the tunnel marking values for the **set ip dscp tunnel** and **set-dscp-tunnel-transmit**commands is from 0 to 63; and the range of values for the **set ip precedence tunnel**and **set-prec-tunnel-transmit** commands is from 0 to 7.

How to Configure Tunnel Marking for GRE Tunnels

- Configuring a Class Map, page 40
- Creating a Policy Map, page 42
- Attaching the Policy Map to an Interface or a VC, page 45
- Verifying the Configuration of Tunnel Marking for GRE Tunnels, page 47

Configuring a Class Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** [**match-all** | **match-any**] *class-map-name*
4. **match fr-de**
5. **end**

DETAILED STEPS

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

	Command or Action	Purpose
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_FRDE`
		Specifies the name of the class map to be created and enters 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.
		- Enter class map name.
		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 fr-de** **Example:** `Router(config-cmap)# match fr-de`	Enables packet matching on the basis of the specified class. You can enter one of the following three **match** commands to define the match criteria for GRE tunnel marking: - match atm clp - match cos - match fr-de **Note** This is only an example of one match criterion that you can configure with a **match** command. Other criteria include matching on the IP precedence, access group, or protocol. Enter the **match** command for the criterion that you want to specify. For more information about specifying match criteria using the MQC, see the "Applying QoS Features Using the MQC" module.

Step 5	**end**
	Example:
	`Router(config-cmap)# end`
		(Optional) Returns to privileged EXEC mode.

Creating a Policy Map

Note It is possible to configure GRE tunnel marking and the **ip tos** command at the same time. However, MQC (GRE) tunnel marking has higher priority over IP ToS commands, meaning that tunnel marking will always rewrite the IP header of the tunnel packet, overwriting the values set by **ip tos** commands. The order of enforcement is as follows when these commands are used simultaneously:

1. **set ip dscp tunnel** or **set ip precedence tunnel** (GRE tunnel marking)
2. **ip tos reflect**
3. **ip tos** tos-value

> **Note** This is the designed behavior. We recommend that you configure only GRE tunnel marking and reconfigure any peers, configured with the **ip tos** command, to use GRE tunnel marking.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** {*class-name* | **class-default**}
5. **set ip dscp tunnel** *dscp-value*
6.
7. **set ip precedence tunnel** *precedence-value*
8.
9. Do one of the following:
   - **police** *bps* [*burst-normal*] [*burst-max*] **conform-action** *action* **exceed-action** *action* [**violate-action** *action*]
10. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map TUNNEL_MARKING`	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy, and enters policy-map configuration mode. • Enter the policy map name.

Step 4	**class** {*class-name* | **class-default**}
	Example:
	`Router(config-pmap)# class MATCH_FRDE`
		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 policy. Also enters policy-map class configuration mode.
		• Enter the class name, or enter the **class-default** keyword.

Step 5	**set ip dscp tunnel** *dscp-value* **Example:** `Router(config-pmap-c)# set ip dscp tunnel 3`	Sets or marks the differentiated services code point (DSCP) value in the tunnel header of a GRE-tunneled packet on the ingress interface. The tunnel marking value is a number from 0 to 63 when configuring DSCP. • Enter the tunnel value.
Step 6		
Step 7	**set ip precedence tunnel** *precedence-value* **Example:** `Router(config-pmap-c)# set ip precedence tunnel 3`	Sets or marks the IP precedence value in the tunnel header of a GRE-tunneled packet on the ingress interface. The tunnel marking value is a number from 0 to 7 when configuring IP precedence. • Enter the tunnel value.
Step 8		

	Command or Action	Purpose

Step 9	Do one of the following:
	• **police** *bps* [*burst-normal*] [*burst-max*] **conform-action** *action* **exceed-action** *action* [**violate-action** *action*]
	Example:
	`Router(config-pmap-c)# police 8000 conform-action set-dscp-tunnel-transmit 4`
	Example:
	`exceed-action set-dscp-tunnel-transmit 0`
	Example:
	Example:
	Example:
	`Router(config-pmap-c)# police 8000 conform-action set-prec-tunnel-transmit 4 exceed-action set-prec-tunnel-transmit 0`
	Example:
		Configures traffic policing on the basis of the bits per second (bps) specified and the actions specified.
		If you use traffic policing in your network, you can implement the GRE tunnel marking feature with the **set-dscp-tunnel-transmit**or **set-prec-tunnel-transmit** keywords of the police command instead of the **set ip dscp tunnel** or the **set ip precedence tunnel** commands.
		The tunnel marking value for the traffic policing commands is from 0 to 63 when using **set-dscp-tunnel-transmit** and from 0 to 7 when using **set-prec-tunnel-transmit**.
		• Enter the bps, any optional burst sizes, and the desired conform and exceed actions.
		• Enter the **set-dscp-tunnel-transmit** or **set-prec-tunnel-transmit** commands after the **conform-action** keyword.
		Note This is an example of one QoS feature that you can configure at this step. Other QoS features include Weighted Random Early Detection (WRED), Weighted Fair Queueing (WFQ), and traffic shaping. Enter the command for the specific QoS feature that you want to configure. For more information about QoS features, see the "Quality of Service Overview" module.

Step 10	**end**
	Example:
	`Router(config-pmap-c)# end`
		(Optional) Returns to privileged EXEC mode.

Attaching the Policy Map to an Interface or a VC

Note Policy maps can be attached to main interfaces, subinterfaces, or ATM permanent virtual circuits (PVCs). Policy maps are attached to interfaces by using the **service-policy**command and specifying either the **input** or **output** keyword to indicate the direction of the interface. This feature is supported only on ingress interfaces with the **input** keyword and should not be configured on egress interfaces with the **output** keyword.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **pvc** [*name*] *vpi* / *vci* [**ilmi** | **qsaal** | **smds**]
5. Do one of the following:
   - **service-policy** {**input**| **output**} *policy-map-name*
6. Do one of the following:
   - **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface serial 0`	Configures the specified interface type and enters interface configuration mode. • Enter the interface type.

	Command or Action	Purpose
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, specifies the encapsulation type on an ATM PVC, and enters ATM virtual circuit configuration mode.

Step 5	Do one of the following:
	• **service-policy** {**input**| **output**} *policy-map-name*
	Example:
	`Router(config-if)#`
	`service-policy input policy1`
	Example:
	Example:
	Example:
	`Router(config-if-atm-vc)#`
	`service-policy input policy1`
		Specifies the name of the policy map to be attached to the *input or output*direction of the interface.
		• 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 vary according your network configuration.
		• Enter the **input** keyword followed by the policy map name.
		Note For this feature, only the incoming interface configured with the **input** keyword is supported.

Step 6	Do one of the following:
	• **end**
	Example:
	`Router(config-if)# end`
	Example:
	Example:
	`Router(config-if-atm-vc)#`
	`end`
		(Optional) Returns to privileged EXEC mode.

Verifying the Configuration of Tunnel Marking for GRE Tunnels

SUMMARY STEPS

1. **enable**
2. **show policy-map interface** *interface-name*
3. **show policy-map** *policy-map*
4. **exit**

DETAILED STEPS

	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** *interface-name*
	Example:
	`Router#`
	`show policy-map interface serial4/0`
		(Optional) 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	**show policy-map** *policy-map*
	Example:
	`Router#`
	`show policy-map policy1`
		(Optional) Displays the configuration of all classes for a specified service policy map or all classes for all existing policy maps.
		• Enter a policy map name.

Step 4	**exit** **Example:** `Router# exit`	(Optional) Returns to user EXEC mode.

- Troubleshooting Tips, page 47

Troubleshooting Tips

The commands in the Verifying the Configuration of Tunnel Marking for GRE Tunnels, page 47 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, you find that the configuration is not functioning as expected, perform these operations to troubleshoot the configuration.

- 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.

Configuration Examples for QoS Tunnel Marking for GRE Tunnels

Example Configuring Tunnel Marking for GRE Tunnels

The following is an example of a GRE tunnel marking configuration. In this example, a class map called "MATCH_FRDE" has been configured to match traffic based on the Frame Relay DE bit.

```
Router> enable
Router# configure terminal
Router(config)# class-map MATCH_FRDE
Router(config-cmap)# match fr-de
Router(config-cmap)# end
```

In this part of the example configuration, a policy map called "TUNNEL_MARKING" has been created and the **set ip dscp tunnel** command has been configured in the policy map. You could use the **set ip precedence tunnel** command instead of the **set ip dscp tunnel** command if you do not use DSCP in your network.

```
Router(config)# policy-map TUNNEL_MARKING

Router(config-pmap)# class MATCH_FRDE

Router(config-pmap-c)# set ip dscp tunnel 3

Router(config-pmap-c)# end
```

Note This next part of the example configuration is not required to configure this feature if you use the **set ip dscp tunnel** or **set ip precedence tunnel** commands to enable GRE tunnel marking. This example shows how GRE tunnel marking can be enabled under traffic policing.

In this part of the example configuration, the policy map called "TUNNEL_MARKING" has been created and traffic policing has also been configured by using the **police** command and specifying the appropriate policing actions. The **set-dscp-tunnel-transmit** command can be used instead of the **set-prec-tunnel-transmit** command if you use DSCP in your network.

```
Router(config)# policy-map TUNNEL_MARKING

Router(config-pmap)# class class-default

Router(config-pmap-c)# police 8000 conform-action set-prec-tunnel-transmit 4 exceed-
```

```
action set-prec-tunnel-transmit 0
Router(config-pmap-c)# end
```

In the final part of the example configuration, the policy map is attached to serial interface 0 in the inbound (input) direction by specifying the **input** keyword of the **service-policy** command.

```
Router(config)# interface serial 0
Router(config-if)#

service-policy input TUNNEL_MARKING
Router(config-if)# end
```

Example Verifying the Tunnel Marking for GRE Tunnels Configuration

This section contains sample output from the **show policy-map interface** command and the **show policy-map** command. The output from these commands can be used to verify and monitor the feature configuration in your network.

The following is sample output from the **show policy-map interface** command. In this sample output, the character string "ip dscp tunnel 3" indicates that GRE tunnel marking has been configured to set the DSCP value in the header of a GRE-tunneled packet.

```
Router# show policy-map interface
 Serial0
Service-policy input: tunnel
    Class-map: frde (match-all)
      0 packets, 0 bytes
      30 second offered rate 0 bps, drop rate 0 bps
      Match: fr-de
      QoS Set
        ip dscp tunnel 3
          Packets marked 0
    Class-map: class-default (match-any)
      13736 packets, 1714682 bytes
      30 second offered rate 0 bps, drop rate 0 bps
      Match: any
        13736 packets, 1714682 bytes
        30 second rate 0 bps
```

The following is sample output from the **show policy-map**command. In this sample output, the character string "ip precedence tunnel 4" indicates that the GRE tunnel marking feature has been configured to set the IP precedence value in the header of an GRE-tunneled packet.

```
Router# show policy-map
Policy Map TUNNEL_MARKING
    Class MATCH_FRDE
      set ip precedence tunnel 4
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

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*
MQC	"Applying QoS Features Using the MQC" module
Tunnel marking for Layer 2 Tunnel Protocol Version 3 (L2TPv3) tunnels	"QoS: Tunnel Marking for L2TPv3 Tunnels" module
DSCP	"Overview of DiffServ for Quality of Service" module

Standards

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

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	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 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

Feature Information for QoS Tunnel Marking for GRE Tunnels

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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.

Table 7 Feature Information for QoS Tunnel Marking for GRE Tunnels

Feature Name	Releases	Feature Information
QoS Tunnel Marking for GRE Tunnels	12.4(15)T2 12.2(33)SRC 12.2(33)SB	The QoS Tunnel Marking for GRE Tunnels feature introduces the capability to define and control the QoS for incoming customer traffic on the PE router in a service provider network. **Note** For Cisco IOS Release 12.4(15)T2, the QoS Tunnel Marking for GRE Tunnels feature is supported only on platforms equipped with a Cisco MGX Route Processor Module (RPM-XF). The following commands were introduced or modified: **match atm-clp**, **match cos**, **match fr-de**, **police**, **police (two rates)**, **set ip dscp tunnel**, **set ip precedence tunnel**, **show policy-map**, **show policy-map interface**.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Classifying Network Traffic

Classifying network traffic allows you to organize traffic (that is, packets) into traffic classes or categories on the basis of whether the traffic matches specific criteria. Classifying network traffic is the foundation for enabling many quality of service (QoS) features on your network. This module contains conceptual information and the configuration tasks for classifying network traffic.

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.

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.

Prerequisites for Classifying Network Traffic

To mark network traffic, Cisco Express Forwarding (CEF) must be configured on both the interface receiving the traffic and the interface sending the traffic.

Information About Classifying Network Traffic

Purpose of Classifying Network Traffic

Classifying network traffic allows you to organize traffic (that is, packets) into traffic classes or categories on the basis of whether the traffic matches specific criteria. Classifying network traffic is the foundation for enabling other QoS features such as traffic shaping and traffic policing on your network.

The goal of network traffic classification is to group traffic based on user-defined criteria so that the resulting groups of network traffic can then be subjected to specific QoS treatments. The QoS treatments might include faster forwarding by intermediate routers and switches or reduced probability of the traffic being dropped due to lack of buffering resources.

Identifying and categorizing network traffic into traffic classes (that is, classifying packets) enables distinct handling for different types of traffic, effectively separating network traffic into different categories. This classification can be associated with a variety of match criteria such as the IP Precedence value, differentiated services code point (DSCP) value, class of service (CoS) value, source and destination MAC addresses, input interface, or protocol type. You classify network traffic by using class maps and policy maps with the Modular Quality of Service Command-Line Interface (MQC). For example, you can configure class maps and policy maps to classify network traffic on the basis of the QoS group, Frame Relay DLCI number, Layer 2 packet length, or other criteria that you specify.

Benefits of Classifying Network Traffic

Classifying network traffic allows you to see what kinds of traffic you have, organize the various kinds of network traffic into traffic classes, and treat some types of traffic differently than others. Identifying and organizing network traffic is the foundation for applying the appropriate QoS feature to that traffic, enabling you to allocate network resources to deliver optimal performance for different types of traffic. For example, high-priority network traffic or traffic matching specific criteria can be singled out for special handling, and thus, help to achieve peak application performance.

MQC and Network Traffic Classification

To configure network traffic classification, you use the Modular Quality of Service Command-Line Interface (MQC).

The MQC is a CLI structure that allows you to complete the following tasks:

- Specify the matching criteria used to define a traffic class.
- Create a traffic policy (policy map). The traffic policy defines the QoS policy actions to be taken for each traffic class.
- Apply the policy actions specified in the policy map to an interface, subinterface, or ATM permanent virtual circuit (PVC) by using the **service-policy** command.

Network Traffic Classification match Commands and Match Criteria

Network traffic classification allows you to group or categorize traffic on the basis of whether the traffic meets one or more specific criteria. For example, network traffic with a specific IP precedence can be placed into one traffic class, while traffic with a specific DSCP value can be placed into another traffic class. The network traffic within that traffic class can be given the appropriate QoS treatment, which you can configure in a policy map later.

You specify the criteria used to classify traffic with a **match** command. The table below lists the available **match** commands and the corresponding match criterion.

Table 8 match Commands and Corresponding Match Criterion

match Commands[2]	Match Criterion
match access group	Access control list (ACL) number
match any	Any match criteria
match class-map	Traffic class name
match cos	Layer 2 class of service (CoS) value
match destination-address mac	MAC address
match discard-class	Discard class value
match dscp	DSCP value
match field	Fields defined in the protocol header description files (PHDFs)
match fr-de	Frame Relay discard eligibility (DE) bit setting
match fr-dlci	Frame Relay data-link connection identifier (DLCI) number
match input-interface	Input interface name
match ip rtp	Real-Time Transport Protocol (RTP) port
match mpls experimental	Multiprotocol Label Switching (MPLS) experimental (EXP) value
match mpls experimental topmost	MPLS EXP value in the topmost label
match not	Single match criterion value to use as an unsuccessful match criterion
match packet length (class-map)	Layer 3 packet length in the IP header
match port-type	Port type
match precedence	IP precedence values
match protocol	Protocol type
match protocol (NBAR)	Protocol type known to network-based application recognition (NBAR)
match protocol citrix	Citrix protocol
match protocol fasttrack	FastTrack peer-to-peer traffic

[2] **Cisco IOS match commands can vary by release and platform. For instance, as of Cisco IOS Release 12.2(31)SB2, the match vlan (QoS) command is supported on Cisco 10000 series routers only. For more information, see the command documentation for the Cisco IOS release and platform that you are using.**

match Commands[2]	Match Criterion
match protocol gnutella	Gnutella peer-to-peer traffic
match protocol http	Hypertext Transfer Protocol
match protocol rtp	RTP traffic
match qos-group	QoS group value
match source-address mac	Source Media Access Control (MAC) address
match start	Datagram header (Layer 2) or the network header (Layer 3)
match tag (class-map)	Tag type of class map
match vlan (QoS)	Layer 2 virtual local-area network (VLAN) identification number

Traffic Classification Compared with Traffic Marking

Traffic classification and traffic marking are closely related and can be used together. Traffic marking can be viewed as an additional action, specified in a policy map, to be taken on a traffic class.

Traffic classification allows you to organize into traffic classes on the basis of whether the traffic matches specific criteria. For example, all traffic with a CoS value of 2 is grouped into one class, and traffic with DSCP value of 3 is grouped into another class. The match criterion is user-defined.

After the traffic is organized into traffic classes, traffic marking allows you to mark (that is, set or change) an attribute for the traffic belonging to that specific class. For instance, you may want to change the CoS value from 2 to 1, or you may want to change the DSCP value from 3 to 2.

The match criteria used by traffic classification are specified by configuring a **match** command in a class map. The marking action taken by traffic marking is specified by configuring a **set** command in a policy map. These class maps and policy maps are configured using the MQC.

The table below compares the features of traffic classification and traffic marking.

Table 9 Traffic Classification Compared with Traffic Marking

	Traffic Classification	Traffic Marking
Goal	Groups network traffic into specific traffic classes on the basis of whether the traffic matches the user-defined criteria.	After the network traffic is grouped into traffic classes, modifies the attributes for the traffic in a particular traffic class.
Configuration Mechanism	Uses class maps and policy maps in the MQC.	Uses class maps and policy maps in the MQC.

[2] **Cisco IOS match commands can vary by release and platform. For instance, as of Cisco IOS Release 12.2(31)SB2, the match vlan (QoS) command is supported on Cisco 10000 series routers only. For more information, see the command documentation for the Cisco IOS release and platform that you are using.**

	Traffic Classification	Traffic Marking
CLI	In a class map, uses **match** commands (for example, **match cos**) to define the traffic matching criteria.	Uses the traffic classes and matching criteria specified by traffic classification. In addition, uses **set** commands (for example, **set cos**) in a policy map to modify the attributes for the network traffic. If a table map was created, uses the **table** keyword and *table-map-name* argument with the **set** commands (for example, **set cos precedence table** *table-map-name*) in the policy map to establish the to-from relationship for mapping attributes.

How to Classify Network Traffic

- Creating a Class Map for Classifying Network Traffic, page 57
- Creating a Policy Map for Applying a QoS Feature to Network Traffic, page 58
- Attaching the Policy Map to an Interface, page 61
- Configuring QoS When Using IPsec VPNs, page 63

Creating a Class Map for Classifying Network Traffic

Note In the following task, the **match fr-dlci**command is shown in Step Creating a Class Map for Classifying Network Traffic, page 57 The **match fr-dlci**command matches traffic on the basis of the Frame Relay DLCI number. The **match fr-dlci**command is just an example of one of the **match** commands that can be used. For a list of other **match** commands, see Creating a Class Map for Classifying Network Traffic, page 57.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** *class-map-name* [**match-all**| **match-any**]
4. **match fr-dlci** *dlci-number*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**class-map** *class-map-name* [**match-all**| **match-any**]
	Example:
	`Router(config)# class-map class1`
		Creates a class map to be used for matching traffic to a specified class, and enters class-map configuration mode.
		• Enter the class map name.

Step 4	**match fr-dlci** *dlci-number* **Example:** `Router(config-cmap)# match fr-dlci 500`	(Optional) Specifies the match criteria in a class map. **Note** The **match fr-dlci** command classifies traffic on the basis of the Frame Relay DLCI number. The **match fr-dlci**command is just an example of one of the **match** commands that can be used. For a list of other **match** commands, see Creating a Class Map for Classifying Network Traffic, page 57.

Step 5	**end**
	Example:
	`Router(config-cmap)# end`
		(Optional) Returns to privileged EXEC mode.

Creating a Policy Map for Applying a QoS Feature to Network Traffic

Note In the following task, the **bandwidth** command is shown at Creating a Policy Map for Applying a QoS Feature to Network Traffic, page 58. The **bandwidth** command configures the QoS feature class-based weighted fair queuing (CBWFQ). CBWFQ is just an example of a QoS feature that can be configured. Use the appropriate command for the QoS feature you want to use.

Note Configuring bandwidth on policies that have the class-default class is supported on physical interfaces such as Gigabit Ethernet (GigE), Serial, Mobile Location Protocol (MLP), and Multilink Frame-Relay (MFR), but it is not supported on logical interfaces such as Virtual Access Interface (VAI), Subinterface, and Frame-Relay on Virtual Circuits (FR-VC).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** {*class-name* | **class-default**}
5. **bandwidth** {*bandwidth-kbps*| **remaining percent** *percentage*| **percent** *percentage*}
6. **end**
7. **show policy-map**
8.
9. **show policy-map** *policy-map* **class** *class-name*
10. Router# show policy-map
11.
12. Router# show policy-map policy1 class class1
13. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map policy1`	Specifies the name of the policy map to be created and enters policy-map configuration mode. • Enter the policy map name.

	Command or Action	Purpose

Step 4	**class** {*class-name* | **class-default**}
	Example:
	`Router(config-pmap)# class class1`
		Specifies the name of the class and enters policy-map class configuration mode. This class is associated with the class map created earlier.
		• Enter the name of the class or enter the **class-default**keyword.

Step 5	**bandwidth** {*bandwidth-kbps*| **remaining percent** *percentage*| **percent** *percentage*}
	Example:
	`Router(config-pmap-c)# bandwidth percent 50`
		(Optional) Specifies or modifies the bandwidth allocated for a class belonging to a policy map.
		• Enter the amount of bandwidth as a number of kbps, a relative percentage of bandwidth, or an absolute amount of bandwidth.
		Note The **bandwidth** command configures the QoS feature class-based weighted fair queuing (CBWFQ). CBWFQ is just an example of a QoS feature that can be configured. Use the appropriate command for the QoS feature that you want to use.

Step 6	**end**
	Example:
	`Router(config-pmap-c)# end`
		Returns to privileged EXEC mode.

Step 7	**show policy-map**	(Optional) Displays all configured policy maps.
Step 8		or
Step 9	**show policy-map** *policy-map* **class** *class-name* **Example:**	(Optional) Displays the configuration for the specified class of the specified policy map. • Enter the policy map name and the class name.
Step 10	Router# show policy-map	
Step 11		
Step 12	Router# show policy-map policy1 class class1	
Step 13	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

- What to Do Next, page 60

What to Do Next

Create and configure as many policy maps as you need for your network. To create and configure additional policy maps, repeat the steps in the Creating a Policy Map for Applying a QoS Feature to

Network Traffic task. Then attach the policy maps to the appropriate interface, following the instructions in the Attaching the Policy Map to an Interface task.

Attaching the Policy Map to an Interface

Note Depending on the needs of your network, policy maps can be attached to an interface, a subinterface, or an ATM PVC.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [**name-tag**]
4. **pvc** [*name*] *vpi* / *vci* [**ilmi**|**qsaal**|**smds**| **l2transport**]
5. **exit**
6. **service-policy** {**input** | **output**} *policy-map-name*
7. **end**
8. **show policy-map interface** *type number*
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [**name-tag**] **Example:** `Router(config)# interface serial4/0`	Configures an interface type and enters interface configuration mode. • Enter the interface type and number.

	Command or Action	Purpose
Step 4	**pvc** [*name*] *vpi* / *vci* [**ilmi**|**qsaal**|**smds**|**l2transport**] **Example:** `Router(config-if)# pvc cisco 0/16`	(Optional) Creates or assigns a name to an ATM PVC, specifies the encapsulation type on an ATM PVC, and enters ATM virtual circuit configuration mode. • Enter the PVC name, the ATM network virtual path identifier, and the network virtual channel identifier. **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, advance to Attaching the Policy Map to an Interface, page 61.
Step 5	**exit** **Example:** `Router(config-atm-vc)# exit`	(Optional) Returns to interface configuration mode. **Note** This step is required only if you are attaching the policy map to an ATM PVC and you completed Attaching the Policy Map to an Interface, page 61. If you are not attaching the policy map to an ATM PVC, advance to Attaching the Policy Map to an Interface, page 61.
Step 6	**service-policy** {**input** | **output**} *policy-map-name* **Example:** `Router(config-if)# service-policy input policy1`	Attaches a policy map to an input or output interface. • Enter the policy map name. **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.

Step 7	**end**
	Example:
	`Router(config-if)# end`
		Returns to privileged EXEC mode.

Step 8	**show policy-map interface** *type number*
	Example:
	`Router#`
	`show policy-map interface serial4/0`
		(Optional) Displays the traffic statistics of all traffic 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 type and number.

Step 9	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode.

Configuring QoS When Using IPsec VPNs

Note This task is required only if you are using IPsec Virtual Private Networks (VPNs). Otherwise, this task is not necessary. For information about IPsec VPNs, see the "Configuring Security for VPNs with IPsec" module.

Note This task uses the **qos pre-classify** command to enable QoS preclassification for the packet. QoS preclassification is not supported for all fragmented packets. If a packet is fragmented, each fragment might receive different preclassifications.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num*
4. **exit**
5. **interface** *type number* [**name-tag**]
6. **qos pre-classify**
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* **Example:** `Router(config)# crypto map mymap 10`	Enters crypto map configuration mode and creates or modifies a crypto map entry. • Enter the crypto map name and sequence number.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Router(config-crypto-map)# exit`	Returns to global configuration mode.
Step 5	**interface** *type number* [**name-tag**] **Example:** `Router(config)# interface serial4/0`	Configures an interface type and enters interface configuration mode. • Enter the interface type and number.
Step 6	**qos pre-classify** **Example:** `Router(config-if)# qos pre-classify`	Enables QoS preclassification.

Step 7	**end**
	Example:
	`Router(config-if)# end`
		(Optional) Returns to privileged EXEC mode.

Configuration Examples for Classifying Network Traffic

- Example Creating a Class Map for Classifying Network Traffic, page 65
- Example Creating a Policy Map for Applying a QoS Feature to Network Traffic, page 65
- Example Attaching the Policy Map to an Interface, page 65
- Example Configuring QoS When Using IPsec VPNs, page 66

Example Creating a Class Map for Classifying Network Traffic

The following is an example of creating a class map to be used for traffic classification. In this example, a traffic class called class1 has been created. Traffic with a Frame Relay DLCI value of 500 will be put in this traffic class.

```
Router> enable

Router# configure terminal

Router(config)# class-map class1

Router(config-cmap)# match fr-dlci 500

Router(config-cmap)# end
```

Note This example uses the **match fr-dlci**command. The **match fr-dlci**command is just an example of one of the **match** commands that can be used. For a list of other **match** commands, see Example Creating a Class Map for Classifying Network Traffic, page 65.

Example Creating a Policy Map for Applying a QoS Feature to Network Traffic

The following is an example of creating a policy map to be used for traffic classification. In this example, a policy map called policy1 has been created, and the **bandwidth** command has been configured for class1. The **bandwidth** command configures the QoS feature CBWFQ.

```
Router> enable
Router# configure terminal
Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# bandwidth percent 50
Router(config-pmap-c)# end
Router#
show policy-map policy1 class class1
Router# exit
```

Note This example uses the **bandwidth** command. The **bandwidth** command configures the QoS feature class-based weighted fair queuing (CBWFQ). CBWFQ is just an example of a QoS feature that can be configured. Use the appropriate command for the QoS feature that you want to use.

Example Attaching the Policy Map to an Interface

The following is an example of attaching the policy map to an interface. In this example, the policy map called policy1 has been attached in the input direction of serial interface 4/0.

```
Router> enable
```

```
Router# configure terminal
Router(config)# interface serial4/0
Router(config-if)# service-policy input policy1
Router(config-if)# end
Router#
show policy-map interface serial4/0
Router# exit
```

Example Configuring QoS When Using IPsec VPNs

The following is an example of configuring QoS when using IPsec VPNs. In this example, the **crypto map** command specifies the IPsec crypto map mymap 10, to which the **qos pre-classify** command is applied.

```
Router> enable
Router# configure terminal
Router(config)# crypto map mymap 10
Router(config-crypto-map)# exit
Router(config)# interface serial4/0
Router(config-if)# qos pre-classify
Router(config-if)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
MQC	"Applying QoS Features Using the MQC" module
Marking network traffic	"Marking Network Traffic" module
IPsec and VPNs	"Configuring Security for VPNs with IPsec" module
NBAR	"Classifying Network Traffic Using NBAR" module
CAR	"Configuring Committed Access Rate" module

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	--

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

Feature Information for Classifying Network Traffic

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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.

Table 10 ***Feature Information for Classifying Network Traffic***

Feature Name	Releases	Feature Information
Packet Classification Based on Layer 3 Packet Length	12.2(13)T	This feature provides the added capability of matching and classifying network traffic on the basis of the Layer3 length in the IP packet header. The Layer 3 length is the IP datagram plus the IP header. This new match criteria is in addition to the other match criteria, such as the IP precedence, differentiated services code point (DSCP) value, class of service (CoS), currently available.
Packet Classification Using Frame Relay DLCI Number	12.2(13)T	The Packet Classification Using the Frame Relay DLCI Number feature allows customers to match and classify traffic based on the Frame Relay data-link connection identifier (DLCI) number associated with a packet. This new match criteria is in addition to the other match criteria, such as the IP Precedence, differentiated services code point (DSCP) value, class of service (CoS), currently available.
Quality of Service for Virtual Private Networks	12.2(2)T	The QoS for VPNs feature provides a solution for making Cisco IOS QoS services operate in conjunction with tunneling and encryption on an interface. Cisco IOS software can classify packets and apply the appropriate QoS service before the data is encrypted and tunneled. The QoS for VPN feature allows users to look inside the packet so that packet classification can be done based on original port numbers and based on source and destination IP addresses. This allows the service provider to treat mission critical or multi-service traffic with higher priority across their network.

Feature Name	Releases	Feature Information
QoS: Match VLAN **Note** As of Cisco IOS Release 12.2(31)SB2, the QoS: Match VLAN feature is supported on Cisco 10000 series routers only.	12.2(31)SB2	The QoS: Match VLAN feature allows you to classify network traffic on the basis of the Layer 2 virtual local-area network (VLAN) identification number. The following commands were introduced or modified by this feature: **match vlan**(QoS), **show policy-map interface**.
Hierarchical Traffic Shaping Packet Classification Based on Layer3 Packet-Length QoS: Match VLAN	15.0(1)S	The Hierarchical Traffic Shaping, Packet Classification Based on Layer3 Packet-Length, QoS: Match VLAN features were integrated into the Cisco IOS Release 15.0(1)S release.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Flexible Packet Matching XML Configuration

The Flexible Packet Matching XML Configuration feature allows the use of eXtensible Markup Language (XML) to define traffic classes and actions (policies) to assist in blocking network attacks. The XML file used by Flexible Packet Matching (FPM) is called the traffic classification definition file (TCDF).

The TCDF gives you an alternative to the command-line interface (CLI) as a method to define traffic classification behavior. Traffic classification behavior is identical regardless of the method you use.

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.

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.

Prerequisites for the Flexible Packet Matching XML Configuration

- A protocol header definition file (PHDF) relevant to the TCDF must be loaded on the router.
- Although access to an XML editor is not required, using one might make the creation of the TCDF easier.
- You must be familiar with XML file syntax.

Restrictions for the Flexible Packet Matching XML Configuration

TCDF Image Restriction

TCDF is part of the FPM subsystem. FPM is not included in the Cisco 871 securityk9 image; therefore, TCDF parsing is not present in the Cisco 871 securityk9 image.

The Flexible Packet Matching XML Configuration has the following restrictions:

- The FPM TCDF cannot be used to mitigate an attack that requires stateful classification.
- Because FPM is stateless, it cannot keep track of port numbers being used by protocols that dynamically negotiate ports. Thus, when using the FPM TCDF, port numbers must be explicitly specified.
- FPM cannot perform IP fragmentation or TCP flow reassembly.

Information About the Flexible Packet Matching XML Configuration

Traffic Classification Definition Files for the Flexible Packet Matching XML Configuration

FPM uses a TCDF to define policies that can block attacks on the network. FPM is a packet classification feature that allows users to define one or more classes of network traffic by pairing a rich set of standard matching operators with user-defined protocol header fields. FPM users can create their own stateless packet classification criteria and define policies with multiple actions (such as drop, log, or send Internet Control Message Protocol [ICMP] unreachable) to immediately block new viruses, worms, and attacks on the network.

Before the release of the Flexible Packet Matching XML Configuration feature, FPM defined traffic classes (class maps), policies (policy maps), and service policies (attach policy maps to a class maps) through the use of CLI commands. With TCDFs, FPM can use XML as an alternative to the CLI to define classes of traffic and specify actions to apply to the traffic classes. Traffic classification behavior is the same whether you create the behavior using a TCDF or configure it using CLI commands. Once a TCDF is created, it can be loaded on any FPM-enabled device in the network.

Protocol Header Definition Files for Traffic Classification Definitions

TCDFs require that a relevant PHDF is already loaded on the device. A PHDF defines each field contained in the header of a particular protocol. Each field is described with a name, optional comment, an offset (the location of the protocol header field in relation to the start of the protocol header), and the length of the field. The total length is specified at the end of each PHDF.

The description of a traffic class in a TCDF file can contain header fields defined in a PHDF. If the PHDF is loaded on the router, the class specification to match begins with a list of the protocol headers in the packet. In the TCDF, the traffic class is associated with a policy that binds the match to an action, such as drop, log, or send ICMP unreachable.

FPM provides ready-made definitions for these standard protocols, which can be loaded onto the router with the **load protocol** command: ether.phdf, ip.phdf, tcp.phdf, and udp.phdf. You can also write your own custom PHDFs using XML if one is required for the TCDF.

Note Because PHDFs are defined via XML, they are not shown in a running configuration.

Traffic Classification Description File Format and Use

In the TCDF, you can define one or more classes of traffic and policies that describe specified actions for each class of traffic. The TCDF is an XML file that you create in a text file or with an XML editor. The file that you create must have a filename that has the .tcdf extension.

The TCDF has the following basic format. XML tags are shown in bold text for example purposes only.

```
<tdcf
>
       <class
 ...> ... </class
>
         ...
       <policy
> ... </policy
>
         ...
</tdcf
>
```

For a traffic class, you can identify a match for any field or fields against any part of the packet.

Note FPM is stateless and cannot be used to mitigate an attack that requires stateful classification, that is classify across IP fragments, across packets in a TCP stream, or peer-to-peer protocol elements.

Policies can be anything from access control, quality of service (QoS), or even routing decisions. For FPM, the associated actions (policies) might include permit, drop, log, or send ICMP unreachable.

Once loaded, the TCDF-defined classes and policies can be applied to any interface or subinterface and behave in an identical manner as the CLI-defined classes and policies. You can define policies in the TCDF and apply then to any entry point to the network to block new attacks.

Traffic Class Definitions for a Traffic Classification Definition File

A class can be any traffic stream of interest. You define a traffic stream of interest by matching a particular interface or port, a source address or destination IP address, a protocol or an application. The following sections contain information you should understand before you define the traffic class in the TCDF for FPM configuration:

- Class Element Attributes for a Traffic Classification Definition File, page 74
- Match Element for a Traffic Classification Definition File, page 75
- Operator Element Attributes for a Traffic Classification Definition File, page 75

Class Element Attributes for a Traffic Classification Definition File

The table below lists and describes the attributes that you can associate with the **class** element in a TCDF for the FPM XML configuration. The **class** element contains attributes you can use to specify the traffic class name, its description and type, where to look in the packet, what kind of match, and when the actions should apply to the traffic.

Table 11 Attributes for Use with the Class Element in a TCDF for the FPM XMLConfiguration

Attribute Name	Use	Type
name (required)	Specifies the name of the class. **Note** When you use the class element inside policy elements, you need specify the name attribute only.	String
type (required)	Specifies the type of class.	Keywords: stack or access-control
stack start	Specifies where to look in the packet. By default, the match starts at Layer 3.	Keyword: l2-start
match	Specifies the type of match to be performed on the class.	Keywords: all or any • all--All class matches must be met to perform the policy actions. • any--One or more matches within the class must be met to perform the policy actions.
undo	Directs the device to remove the class-map when set to true.	Keywords: true or false

For example, XML syntax for a stack class describing an IP, User Datagram Protocol (UDP), Simple Management Protocol (SNMP) stack might look like this:

```
<class
 name
="snmp-stack"
type
="stack">
    <match
>
```

```
       <eq

field
="ip.protocol" value="x"></eq
>
       <eq

field
="udp.dport"
value
="161"></eq
>
    </match
>
</class
>
```

Match Element for a Traffic Classification Definition File

The **match** element in the TCDF for FPM XML configuration contains **operator** elements. **Operator** elements are the following: **eq** (equal to), **neq** (not equal to), **lt** (less than), **gt** (greater than), **range** (a value in a specific range, for example, **range** 1 - 25), and **regex** (regular expression string with a maximum length of 32 characters).

In following sections, these various operators are collectively called the operator element.

Operator Element Attributes for a Traffic Classification Definition File

The table below lists and describes direct matching attributes that you can associate with the **operator**element in a TCDF for the FPM XML configuration.

Table 12 Direct Matching Attributes to Use with a Match Element in a TCDF for the FPM XML Configuration

Attribute Name	Use	Type
start	Begin the match on a predefined keyword or **Protocol.Field**, if given.	Keyword: l2-start or l3-start Otherwise, a field of a protocol as defined in the PHDF, for example, the source field in the IP protocol.
offset	Used with start attribute. Offset from the start point.	Hexadecimal or decimal number, or string constants, **Protocol.Field**, or combination of a constant and **Protocol.Field** with +, -, *, /, &, or |.
size	Used together with start and offset attributes. How much to match.	Specifies the size of the match in bytes.

Attribute Name	Use	Type
mask	Number specifying bits to be matched in protocol or field attributes. Used exclusively with field type of bitset to specify the bits of interest in a bit map.	Decimal or hexadecimal number
value	Value on which to match.	String, number, or regular expression
field	Specifies the name of the field to be compared.	Name of field as defined in the PHDF
next	Identifies the next layer of the protocol. This attribute can be used only in stack type classes.	Keyword that is the name of a protocol defined in the PHDF.
undo	Directs the device to remove the particular match operator when set to true.	Keywords: true or false

Policy Definitions for a Traffic Classification Definition File

A policy is any action that you apply to a class. You should understand the following information before defining the policy in a TCDF for the FPM XML configuration:

- Policy Element Attributes for a Traffic Classification Definition File, page 76
- Action Element for a Traffic Classification Definition File, page 77
- Traffic Classification Definition File Syntax Guidelines, page 77

Policy Element Attributes for a Traffic Classification Definition File

Policies can be anything from access control, QoS, or even routing decisions. For FPM, the associated actions or policies might include drop, log, or send ICMP unreachable. Policies describe the action to take to mitigate attacks on the network.

The table below lists and describes the attributes that you can use with the **policy** element in the TDCF for FPM XML configuration.

Table 13 Attributes for Use with the Policy Element in a TCDF for the FPM XML Configuration

Attribute Name	Use	Type
name	Name of the policy.	String
type	Specifies the type of policy map.	Keyword: access-control
undo	Directs the device to remove the policy map when set to true.	Keywords: true or false

The policy name in this example is sql-slammer, and the action defined for the policy is to drop the packet. This action is to be applied to the class that has the same name as the policy (class name= "sql-slammer").

```
<policy

name
="sql-slammer">
    <class

name
="sql-slammer"></class
>
    <action
>drop</action
>
</policy
>
```

Action Element for a Traffic Classification Definition File

The **action** element is used to specify actions to associate with a policy. The policy with the **action** element is applied to a defined class. The **action** element can contain any of the following: permit, drop, Log, SendBackIcmp, set, RateLimit, alarm, ResetTcpConnection, and DropFlow. For example:

```
<action
>
   log
</action
>
```

Traffic Classification Definition File Syntax Guidelines

The following list describes required and optional syntax for the TCDF:

- The TCDF filename must end in the .tcdf extension, for example, sql_slammer.tcdf.
- The TCDF contains descriptions for one or more traffic classes and one or more policy actions.
- The file is encoded in the XML notation.
- The TCDF file should begin with the following version encoding:

<?xml version="1.0" encoding="UTF-8"?>

The TCDF is used to define traffic classes and the associated policies with specified actions for the purpose of blocking new viruses, worms, and attacks on the network.

The TCDF is configured in a text or XML editor. The syntax of the TCDF must comply with the XML Version 1.0 syntax and the TCDF schema. For information about Version 1.0 XML syntax, see the document at the following url:

http://www.w3.org/TR/REC-xml/

How to Create and Load Traffic Classification Definition Files

- Creating a Definition File for the FPM XML Configuration, page 78
- Loading a Definition File for the FPM XML Configuration, page 80
- Associating a Traffic Classification Definition File, page 82
- Displaying TCDF-Defined Traffic Classes and Policies, page 83

Creating a Definition File for the FPM XML Configuration

SUMMARY STEPS

1. Open a text file or an XML editor and begin the file with the XML version and encoding declaration.
2. Identify the file as a TCDF. For example:
3. Define the traffic class of interest.
4. Identify matching criteria for the defined classes of traffic. For example:
5. Define the action to apply to the defined class. For example:
6. End the traffic classification definition. For example:
7. Save the TCDF file with a filename that has a .tcdf extension, for example: slammer.tcdf.

DETAILED STEPS

Step 1 Open a text file or an XML editor and begin the file with the XML version and encoding declaration.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Step 2 Identify the file as a TCDF. For example:

Example:

```
<tcdf
>
```

Step 3 Define the traffic class of interest.
For example, a stack class describing an IP and UDP stack might be described as follows. In this example, the name of the traffic class is "ip-udp," and the class type is "stack."

Example:

```
<class

name
="ip-udp"
type
="stack"></class
>
```

In the following example, the name of the traffic class is slammer, the class type is access control, and the match criteria is all:

Example:

```
<class
 name="
slammer
"
type
```

```
="access-control"
match
="all"></class
>
```

Step 4 Identify matching criteria for the defined classes of traffic. For example:

Example:

```
    <class

name
="ip-udp"
type
="stack">
       <match
>
          <eq

field
="ip.protocol"
value
="0x11"
next
="udp"></eq
>
       </match
>
    </class
>
    <class
 name="
slammer
"
type
="access-control"
match
="all">
       <match
>
          <eq

field
="udp.dest-port"
value
="0x59A"></eq
>
          <eq

field
="ip.length"
value
="0x194"></eq
>
          <eq

start
="l3-start"
offset
="224"
size
="4"
value
="0x00401010"></eq
>
       </match
>
    </class
>
```

The traffic of interest in this TCDF matches fields defined in the PHDF files, ip.phdf and udp.phdf. The matching criteria for slammer packets is a UDP destination port number 1434 (0x59A), an IP length not to exceed 404 (0x194) bytes, and a Layer 3 position with a pattern 0x00401010 at 224 bytes from start (offset) of the IP header.

Step 5 Define the action to apply to the defined class. For example:

Example:

```
<policy

name
="fpm-udp-policy">
    <class

name
="slammer"></class
>
    <action
>Drop</action
>
</policy
>
```

The policy name in this example is fpm-udp-policy, and the action defined for the policy is to drop the packet. This action is to be applied to the class that has the name slammer.

Step 6 End the traffic classification definition. For example:

Example:

```
</tcdf
>
```

Step 7 Save the TCDF file with a filename that has a .tcdf extension, for example: slammer.tcdf.

Loading a Definition File for the FPM XML Configuration

SUMMARY STEPS

1. **enable**
2. **show protocol phdf** *protocol-name*
3. **configure terminal**
4. **load protocol** *location:filename*
5. **load classification** *location* **:** *filename*
6. **end**
7. **show class-map** [type {**stack** | **access-control**}] [*class-map-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show protocol phdf** *protocol-name* **Example:** `Router# show protocol phdf ip`	Displays protocol information from a specific PHDF. • Use this command to verify that a PHDF file relevant to the TCDF is loaded on the device.
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**load protocol** *location:filename* **Example:** `Router(config)# load protocol localdisk1:ip.phdf`	(Optional) Loads a PHDF onto a router. • The specified location must be local to the router. **Note** If the required PHDF is already loaded on the router (see Step 2), skip this step and proceed to Step 5).
Step 5	**load classification** *location* **:** *filename* **Example:** `Router(config)# load classification localdisk1:slammer.tcdf`	Loads a TCDF onto a router. • The specified location must be local to the router.

Step 6	**end**
	Example:
	`Router(config)# end`
		Exits to privileged EXEC mode.

Step 7	**show class-map** [type {**stack** | **access-control**}] [*class-map-name*] **Example:** `Router# show class-map sql-slammer`	(Optional) Displays a class map and its matching criteria. • Use this command to verify that a class defined in the TCDF file is available on the device. • The *class-map-name* argument is the name of a class in the TCDF.

Examples

The following is sample output from a **show class-map** command that displays the traffic classes defined in the TCDF after it is loaded on the router:

```
Router# show class-map
.
.
.
class-map type stack match-all ip-udp
   match field IP protocol eq 0x11 next UDP
class-map type access-control match-all slammer
   match field UDP dest-port eq 0x59A
   match field IP length eq 0x194
   match start l3-start offset 224 size 4 eq 0x4011010
.
.
.
```

- What to Do Next, page 82

What to Do Next

After you have defined the TCDF, you must apply that policy to an interface as shown in the following task "Associating a Traffic Classification Definition File, page 82."

Associating a Traffic Classification Definition File

Perform this task to associate the defination file with an interface or subinterface.

The TCDP and FPM must be configured on the device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot* / *port*
4. **service-policy type access-control**] {**input** | **output**} *policy-map-name*
5. **end**
6. **show policy-map interface type access-control**] *interface-name slot/port*[**input** | **output**]

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot* / *port* **Example:** `Router(config)# interface gigabitEthernet 0/1`	Configures an interface type and enters interface configuration mode.
Step 4	**service-policy type access-control**] {**input** | **output**} *policy-map-name* **Example:** `Router(config-if)# service-policy type access-control input sql-slammer`	Specifies the type and the name of the traffic policy to be attached to the input or output direction of an interface. • The *policy-map-name* argument is the name of a policy in the TCDF.

Step 5	**end**
	Example:
	`Router(config-if)# end`
		Exits to privileged EXEC mode.

Step 6	**show policy-map interface type access-control**] *interface-name slot/port*[**input** | **output**] **Example:** `Router# show policy-map interface gigabitEthernet 0/1`	(Optional) Displays the packet statistics of all classes that are configured for all service policies either on the specified interface or subinterface. • Use this command to verify that policy defined in TCDF is associated with the named interface.

Displaying TCDF-Defined Traffic Classes and Policies

SUMMARY STEPS

1. **enable**
2. **show class-map [type { stack** | **access-control**}**]** [*class-map-name*]
3. **show class-map type stack** [*class-map name*]
4. **show class-map type access-control** [*class-map-name*]
5. **show policy-map** [*policy-map*]
6. **exit**

DETAILED STEPS

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if prompted. For example:

Example:

```
Router> enable
Router#
```

Step 2 **show class-map [type { stack** | **access-control**}**]** [*class-map-name*]

Use this command to verify that a class defined in the TCDF file is available on the device. For example:

Example:

```
Router# show class-map
.
.
.
class-map type stack match-all ip-udp
   match field IP protocol eq 0x11 next UDP
class-map type access-control match-all slammer
   match field UDP dest-port eq 0x59A
   match field IP length eq 0x194
   match start l3-start offset 224 size 4 eq 0x4011010
.
.
.
```

Step 3 **show class-map type stack** [*class-map name*]

Use this command to display the stack type defined for the class of traffic in the TCDF file. For example:

Example:

```
Router# show class-map type stack ip-udp
class-map type stack match-all ip-udp
   match field IP protocol eq 0x11 next UDP
```

Step 4 **show class-map type access-control** [*class-map-name*]

Use this command to display the access type defined for the class in the TCDF file. For example:

Example:

```
Router# show class-map type access-control slammer
class-map type access-control match-all slammer
   match field UDP dest-port eq 0x59A
   match field IP length eq 0x194
   match start l3-start offset 224 size 4 eq 0x4011010
```

Step 5 **show policy-map** [*policy-map*]

Use this command to display the contents of a policy map defined in the TCDF. For example:

Example:

```
Router# show policy-map fpm-udp-policy
policy-map type access-control fpm-udp-policy
```

```
class slammer
  drop
```

Step 6 **exit**

Use this command to exit to user EXEC mode. For example:

Example:

```
Router# exit
Router>
```

Configuration Examples for Creating and Loading Traffic Classification Definition Files

Note The TCDF files are created in a text file or with an XML editor. In the following examples, XML tags are shown in bold text and field names in italic text. The values for the attributes are entered in quotation marks ("value").

- Example Traffic Classification Definition File for Slammer Packets, page 85
- Example Traffic Classification Definition File for MyDoom Packets, page 87

Example Traffic Classification Definition File for Slammer Packets

The following example shows how to create and load a TCDF for slammer packets (UDP 1434) for the FPM configuration. The match criteria defined within the **class** element is for slammer packets with an IP length not to exceed 404 (0x194) bytes, UDP destination port 1434 (0x59A), and pattern 0x00401010 at 224 bytes from start of IP header. This example also shows how to define the policy "sql-slammer" with the action to drop slammer packets.

```
<?xml version="1.0" encoding="UTF-8"?
>
<tcdf
>
    <class

name
="ip-udp"
type
="stack">
       <match
>
          <eq

field
="ip.protocol"
value
="0x11"
next
="udp"></eq
>
```

```
       </match
>
    </class
>
    <class
 name="
slammer
"
type
="access-control"
match
="all">
       <match
>
          <eq

field
="udp.dest-port"
value
="0x59A"></eq
>
          <eq

field
="ip.length"
value
="0x194"></eq
>
          <eq

start
="l3-start"
offset
="224"
size
="4"
value
="0x00401010"></eq
>
       </match
>
    </class
>
    <policy
 type="access-control"
name
="fpm-udp-policy">
       <class

name
="slammer"></class
>
       <action
>Drop</action
>
    </policy
>
</tcdf
>
```

The following example shows how to load the TCDF file onto the device and apply the policy defined in the file to the interface Gigabit Ethernet 0/1:

```
configure terminal
load classification localdisk1:sql-slammer.tcdf
policy-map type access-control my-policy-1
class ip-udp
service-policy fpm-udp-policy
interface gigabitEthernet 0/1
 service-policy type access-control input my-policy-1
 end
```

Example Traffic Classification Definition File for MyDoom Packets

The following example shows how to create and load a TCDF for MyDoom packets in a text file or XML editor for the FPM XML configuration. The match criteria for the MyDoom packets are as follows:

- 90 > IP length > 44
- pattern 0x47455420 at 40 bytes from start of IP header

or

- IP length > 44
- pattern 0x47455420 at 40 bytes from start of IP header

```
<tcdf
>
    <class

name
="md-stack"
type
="stack">
        <match
>
            <eq

field
="ip.protocol"
value
="6"
next
="tcp"></eq
>
        </match
>
    </class
>
    <class

type
="access-control"
name
="mydoom1">
        <match
>
            <gt

field
="ip.length"
value
="44"/>
            <lt

field
="ip.length"
value
="90"/>
            <eq

start
="ip.version"
offset
="tcp.headerlength*4+20"
size
="4"

value
="0x47455420"/>
        </match
```

```
>
    </class
>
    <class

type
="access-control"
name
="mydoom2">
        <match
>
            <gt
 field="ip.length" value="44"/>
            <eq
 start="ip.version" offset="tcp.headerlength*4+58" size="4"
                value="0x6d3a3830"/>
            <eq
 start="ip.version" offset="tcp.headerlength*4+20" size="4"
                 value="0x47455420"/>
        </match
>
    </class
>
    <policy

name
="fpm-md-stack-policy">
       <class

name
="mydoom1"></class
>
       <action
>drop</action
>
    </policy
>
    <policy

name
="fpm-md-stack-policy">
       <class

name
="mydoom2"></class
>
       <action
>drop</action
>
    </policy
>
</tcdf
>
```

The following example shows how to load the TCDF file onto the device and apply the policy defined in the file to the interface Ethernet 0/1:

```
configure terminal
load classification localdisk1:sql-slammer.tcdf
policy-map type access-control my-policy-2
class md-stack
service-policy fpm-md-stack-policy
interface Ethernet 0/1
 service-policy type access-control input my-policy-2
 end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Additional configuration information for class maps and policy maps	"Applying QoS Features Using the MQC" module
Information about and configuration tasks for FPM	"Flexible Packet Matching" module

Standards

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

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	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 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

Feature Information for Flexible Packet Matching XML Configuration

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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.

Table 14 Feature Information for Flexible Packet Matching XML Configuration

Feature Name	Releases	Feature Information
Flexible Packet Matching XML Configuration	12.4(6)T	The Flexible Packet Matching XML Configuration feature provides an Extensible Markup Language (XML)-based configuration file for Flexible Packet Matching (FPM) that can be used to define traffic classes and actions (policies) to assist in the blocking of attacks on a network. The XML file used by FPM is called the traffic classification definition file (TCDF). The TCDF gives you an alternative to the command-line interface (CLI) as a method to define traffic classification behavior. Traffic classification behavior is identical regardless of the method you use. This feature was introduced in Cisco IOS Release 12.4(6)T. The following command was introduced by this feature: **load classification**.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.