CHAPTER 40

Configuring IPv6 QoS

This chapter describes how to configure IPv6 quality of service (QoS) on the Cisco ME 3400 Ethernet Access switch. Cisco IOS Release 12.2(60)EZ adds support for IPv6 QoS.

The switch supports QoS for both IPv4 and IPv6 traffic when a dual IPv4 and IPv6 SDM template is configured. For more information about dual IPv4 and IPv6 templates, see the "Dual IPv4 and IPv6 SDM Templates" section on page 7-2.

Note To enable IPv6 QoS, you must configure the dual IPv4 and IPv6 Switch Database Management (SDM) template on the switch. Select the template by entering the **sdm prefer dual-ipv4-and-ipv6** global configuration command. You must reload the switch after configuring the template.

This chapter describes how to configure QoS for IPv6 traffic only on the Cisco ME 3400 Ethernet Access switch. Chapter 34, "Configuring QoS" describes how to configure QoS for both IPv4 and IPv6 traffic.

For more information about Cisco IOS MQC commands, see the "Cisco IOS Quality of Service Solutions Command Reference" at this site:

http://www.cisco.com/en/US/docs/ios/qos/configuration/guide/12_4/qos_12_4_book.html

For related information, see these chapters:

- For information about configuring IPv6 on the switch, see Chapter 37, "Configuring IPv6 Unicast Routing."
- For information about configuring IPv6 access control lists (ACLs), see Chapter 39, "Configuring IPv6 ACLs."
- For information about SDM templates, see Chapter 7, "Configuring SDM Templates."
- For information about configuring IPv4 QoS, see Chapter 34, "Configuring QoS."

Note For complete syntax and usage information for the commands used in this chapter, see the command reference for this release or the Cisco IOS documentation referenced in the procedures.

This chapter contains these sections:

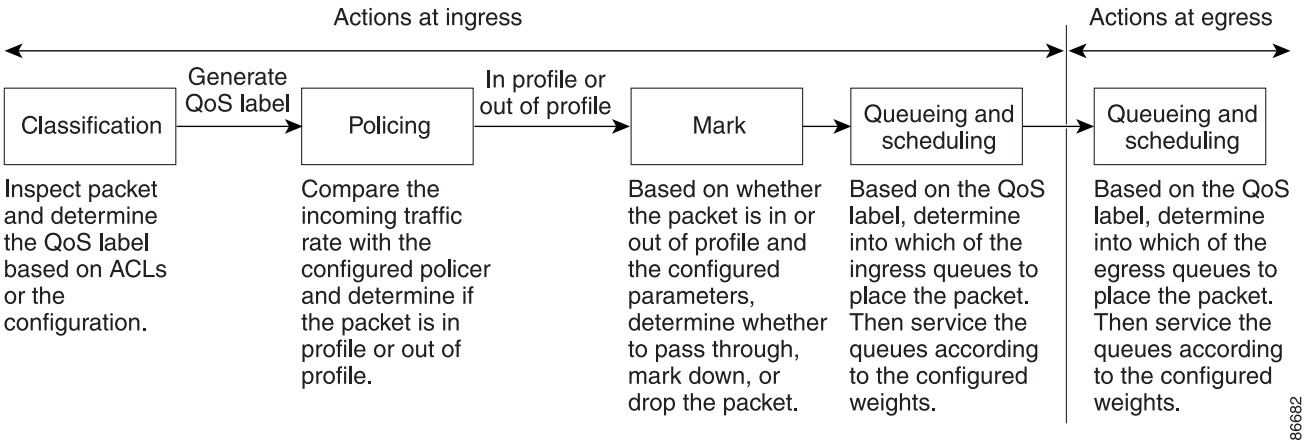
Understanding IPv6 QoS

QoS features supported for IPv6 environments include packet classification, queueing, class-based packet marking, and policing of IPv6 packets. IPv6 packets are forwarded by paths that are different from those for IPv4.

All QoS features available for IPv6 environments are managed from the modular QoS command-line interface (MQC). The MQC allows you to define traffic classes, create and configure traffic policies (policy maps), and then attach those traffic policies to interfaces.

Figure 40-1 shows the basic QoS model for IPv6 traffic.

Figure 40-1 Basic QoS Model



To configure QoS in IPv6 networks, follow the same steps that you would follow to implement QoS in IPv4 networks. At a very high level, the basic steps for implementing IPv6 QoS are as follows:

1. Define a traffic class. Use the **class-map [match-all | match-any]** class-map-name global configuration command to define a traffic class and to enter class-map configuration mode.
2. Create a traffic policy to associate the traffic class with one or more QoS features. Use the **policy-map policy-map-name** global configuration command to create a traffic policy and to enter policy-map configuration mode. A traffic policy defines the QoS features to associate with the specified traffic class.
3. Attach the traffic policy to an interface. Use the **service-policy interface** configuration command to attach the policy map to an interface for packets entering or leaving the interface. Specify whether the traffic policy characteristics are applied to incoming or outgoing packets.

Configuring IPv6 QoS

Classifying Traffic

Classifying network traffic allows you to organize packets into traffic classes or categories on the basis of whether the traffic matches specific criteria.

Restrictions

The following restrictions apply when classifying IPv6 traffic:

- Classification based on inner vlan (match vlan inner) for QinQ services is not supported for IPv6 traffic on the Cisco ME 3400 Ethernet Access switch.
- IPv4 and IPv6 classification criteria cannot be configured simultaneously in the same class-map, but they can be configured in different class-maps in the same policy.

Note These restrictions apply to IPv6 traffic only. See Chapter 34, "Configuring QoS" for guidelines and restrictions that apply to both IPv4 and IPv6 traffic.

Using IPv6 ACLS to Classify IPv6 Traffic

You can classify IPv6 traffic using IPv6 ACLs. For information about configuring IPv6 ACLs, see Chapter 39, "Configuring IPv6 ACLs."

The Cisco ME 3400 Ethernet Access switch supports matching IPv6 traffic based on source or destination address, IPv6 DSCP and layer 4 source or destination port by defining ACLs and using the **match access-group** class-map configuration command to match on the ACL.

SUMMARY STEPS

1. **configure terminal**
2. **ipv6 access-list** *access-list-name*
3. **{deny | permit} protocol** {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix/ prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* [*port-number*]] [**dscp** *value*] [**fragments**] [**log**] [**log-input**] [**routing**] [**sequence** *value*] [**time-range** *name*]

   {**deny** | **permit**} **tcp** {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6- prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* [*port-number*]] [**ack**] [**dscp** *value*] [**established**] [**fin**] [**log**] [**log-input**] [**neq** {*port* | *protocol*}] [**psh**] [**range** {*port* | *protocol*}] [**rst**] [**routing**] [**sequence** *value*] [**syn**] [**time-range** *name*] [**urg**]

   {**deny** | **permit**} **udp**
   {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* [*port-number*]] [**dscp** *value*] [**log**] [**log-input**] [**neq** {*port* | *protocol*}] [**range** {*port* | *protocol*}] [**routing**] [**sequence** *value*] [**time-range** *name*]

{**deny** | **permit**} **icmp** {*source-ipv6-prefix*/*prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix*/*prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* [*port-number*]] [*icmp-type* [*icmp-code*] | *icmp-message*] [**dscp** *value*] [**log**] [**log-input**] [**routing**] [**sequence** *value*] [**time-range** *name*]

4. **end**
5. **show ipv6 access-list**
6. **copy running-config startup-config**

DETAILED STEPS

Complete the following steps in privileged EXEC mode, to create an IPv6 ACL.

	Command	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 2	**ipv6 access-list** *access-list-name* **Example:** `Switch(config)# ipv6 access-list CISCO`	Defines an IPv6 access list using a name, and enters IPv6 access-list configuration mode.

Command	Purpose
Step 3a {**deny** | **permit**} **protocol** {*source-ipv6-prefix*/*prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix*/ *prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* [*port-number*]] [**dscp** *value*] [**fragments**] [**log**] [**log-input**] [**routing**] [**sequence** *value*] [**time-range** *name*]	Specifies whether to deny or to permit the packet if conditions are matched. These are the conditions: • **protocol**— Enter the name or number of an Internet protocol: **ahp**, **esp**, **icmp**, **ipv6**, **pcp**, **stcp**, **tcp**, or **udp**, or an integer in the range 0 to 255 representing an IPv6 protocol number. **Note** For additional specific parameters for ICMP, TCP, and UDP, see Steps 3b through 3d. • *source-ipv6-prefix*/*prefix-length* or *destination-ipv6-prefix*/ *prefix-length*— The source or destination IPv6 network or class of networks for which to set deny or permit conditions, specified in hexadecimal and using 16-bit values between colons. • **any**—An abbreviation for the IPv6 prefix ::/0. • **host** *source-ipv6-address* or *destination-ipv6-address*—Enter the source or destination IPv6 host address for which to set deny or permit conditions, specified in hexadecimal and using 16-bit values between colons. • (Optional) *operator*—Specify an operand that compares the source or destination ports of the specified protocol. Operands are **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range**. If the operator follows the *source-ipv6-prefix*/*prefix-length* argument, it must match the source port. If the operator follows the *destination-ipv6-prefix*/*prefix-length* argument, it must match the destination port. • (Optional) *port-number*— A decimal number from 0 to 65535 or the name of a TCP or UDP port. You can use TCP port names only when filtering TCP. You can use UDP port names only when filtering UDP. • (Optional) **dscp** *value*—Match a differentiated services code point value against the traffic class value in the Traffic Class field of each IPv6 packet header. The acceptable range is from 0 to 63. • (Optional) **fragments—C**heck noninitial fragments. This keyword is visible only if the protocol is ipv6. • (Optional) **log—**Cause a logging message to be sent to the console about the packet that matches the entry. Enter **log-input** to include the input interface in the log entry. Logging is supported only for router ACLs. • (Optional) **routing**—Specify that IPv6 packets be routed. • (Optional) **sequence** *value*—Specify the sequence number for the access list statement. The acceptable range is from 1 to 4294967295. • (Optional) **time-range** *name*—Specify the time range that applies to the deny or permit statement.

	Command	Purpose
Step 3b	{**deny** | **permit**} **tcp** {*source-ipv6-prefix*/*prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix*/*prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* [*port-number*]] [**ack**] [**dscp** *value*] [**established**] [**fin**] [**log**] [**log-input**] [**neq** {*port* | *protocol*}] [**psh**] [**range** {*port* | *protocol*}] [**rst**] [**routing**] [**sequence** *value*] [**syn**] [**time-range** *name*] [**urg**] **Example:** `Switch(config-ipv6-acl)# deny tcp any any gt 5000`	(Optional) Defines a TCP access list and the access conditions. Enter **tcp** for Transmission Control Protocol. The parameters are the same as those described in Step 3a, with these additional optional parameters: • **ack**—Acknowledgment bit set. • **established**—An established connection. A match occurs if the TCP datagram has the ACK or RST bits set. • **fin**—Finished bit set; no more data from sender. • **neq** {*port* | *protocol*}—Matches only packets that are not on a given port number. • **psh**—Push function bit set. • **range** {*port* | *protocol*}—Matches only packets in the port number range. • **rst**—Reset bit set. • **syn**—Synchronize bit set. • **urg**—Urgent pointer bit set.
Step 3c	{**deny** | **permit**} **udp** {*source-ipv6-prefix*/*prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix*/*prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* [*port-number*]] [**dscp** *value*] [**log**] [**log-input**] [**neq** {*port* | *protocol*}] [**range** {*port* | *protocol*}] [**routing**] [**sequence** *value*] [**time-range** *name*]	(Optional) Defines a UDP access list and the access conditions. Enter **udp** for the User Datagram Protocol. The UDP parameters are the same as those described for TCP, except that the [*operator* [*port*]] port number or name must be a UDP port number or name, and the **established** parameter is not valid for UDP.
Step 3d	{**deny** | **permit**} **icmp** {*source-ipv6-prefix*/*prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix*/*prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* [*port-number*]] [*icmp-type* [*icmp-code*] | *icmp-message*] [**dscp** *value*] [**log**] [**log-input**] [**routing**] [**sequence** *value*] [**time-range** *name*] **Example:** `Switch(config-ipv6-acl)# permit icmp any any`	(Optional) Defines an ICMP access list and the access conditions. Enter **icmp** for Internet Control Message Protocol. The ICMP parameters are the same as those described for most IP protocols in Step 3a, with the addition of the ICMP message type and code parameters. These optional keywords have these meanings: • *icmp-type*—Enter to filter by ICMP message type, a number from 0 to 255. • *icmp-code*—Enter to filter ICMP packets that are filtered by the ICMP message code type, a number from 0 to 255. • *icmp-message*—Enter to filter ICMP packets by the ICMP message type name or the ICMP message type and code name. To see a list of ICMP message type names and code names, use the ? key or see command reference for this release.
Step 4	**end**	Returns to privileged EXEC mode.

	Command	Purpose
Step 5	**show ipv6 access-list**	Verifies the access list configuration.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Use the **no** {**deny** | **permit**} IPv6 access-list configuration commands with keywords to remove the deny or permit conditions from the specified access list.

This example configures the IPv6 access list named *CISCO*. The first deny entry in the list denies all packets that have a destination TCP port number greater than 5000. The second deny entry denies packets that have a source UDP port number less than 5000. The second deny also logs all matches to the console. The first permit entry in the list permits all ICMP packets. The second permit entry in the list permits all other traffic. The second permit entry is necessary because an implicit deny -all condition is at the end of each IPv6 access list.

```
Switch(config)# ipv6 access-list CISCO
Switch(config-ipv6-acl)# deny tcp any any gt 5000
Switch config-ipv6-acl)# deny ::/0 lt 5000 ::/0 log
Switch(config-ipv6-acl)# permit icmp any any
Switch(config-ipv6-acl)# permit any any
Switch(config-ipv6-acl)# exit
```

This example shows how to create a class map that matches incoming traffic with the IPv6 ACL named *CISCO*.

```
Switch(config)# class-map match-any IPv6-class
Switch(config-cmap)# match access-group CISCO
Switch(config-cmap)# exit
```

Using Class Maps to Classify IPv6 Traffic

Use the **class-map** global configuration command to name and to isolate a specific traffic flow (or class) from all other traffic. The Cisco ME 3400 Ethernet Access switch supports the following **match** commands in an IPv6 Qos class-map:

- **match access-group**
- **match dscp**
- **match precedence**

Except for the **match dscp** and **match precedence** commands, the functionality of all of the **match** commands is the same for both IPv4 and IPv6. The **match dscp** and **match precedence** commands match on both IPv4 and IPv6 traffic. The **match ip dscp** and **match ip precedence** commands match on IPv4 traffic only.

Note At egress there is no distinction between **match ip dscp** and **match dscp**, or between **match ip precedence** and **match precedence** in an output policy map**.** Both match statements match on both IPv4 and IPv6 traffic.

For more information about IPv4 QoS, see Chapter 34, “Configuring QoS.”

Restrictions

The following restrictions apply to the **match** command in an IPv6 QoS class-map:

- The **match dscp** and **match precedence** commands cannot be configured within the same class-map.
- The **match dscp** and **match access-group** commands cannot be configured within the same class-map.
- The **match precedence** and **match access-group** commands cannot be configured within the same class-map.
- The **match ip dscp/match ip precedence** and **match dscp/match precedence** commands cannot be configured within the same class-map.

Note You can also create class-maps during policy map creation by using the **class** policy-map configuration command.

SUMMARY STEPS

1. **configure terminal**
2. **class-map** [**match-all** | **match-any**] *class-map-name*
3. **match** {**access-group** *acl-index-or-name* | **dscp** *dscp-list* | **precedence** *ip-precedence-list*}
4. **end**
5. **show class-map**
6. **copy running-config startup-config**

DETAILED STEPS

Complete the following steps in privileged EXEC mode, to create a class map and to define the match criterion to classify traffic.

	Command	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 2	**class-map** [**match-all** | **match-any**] *class-map-name* **Example:** `Switch(config)# class-map match-all VOIP-ALL`	Creates a class map, and enters class-map configuration mode. By default, no class maps are defined. • (Optional) **match-all**—Perform a logical-AND of all matching statements under this class map. All match criteria in the class map must be matched. • (Optional) **match-any**—Perform a logical-OR of all matching statements under this class map. One or more match criteria must be matched. • *class-map-name*—Specify the name of the class map. If no matching statements are specified, the default is **match-all**. **Note** A **match-all** class map cannot have more than one classification criterion (match statement).
Step 3	**match** {**access-group** *acl-index-or-name* | **dscp** *dscp-list* | **precedence** *ip-precedence-list*} **Example:** `Switch(config-cmap)# match dscp cs5`	Defines the match criterion to classify traffic. By default, no match criterion is defined. Only one match type per class map is supported, and only one ACL per class map is supported. • **access-group** *acl-index-or-name*—Specify the number or name of an ACL. Matching access groups is supported only in input policy maps. • **dscp** *dscp-list* —Matches on both IPv4 and IPv6. Enter a list of up to eight DSCP values to match against incoming packets. Separate each value with a space. You can enter multiple *dscp-list* lines to match more than eight DSCP values. The numerical range is 0 to 63. You can also configure DSCP values in other forms. See the "Classification Based on IP DSCP" section on page 34-8. • **precedence** *ip-precedence-list*—Enter a list of up to four IPv6 precedence values to match against incoming packets. Separate each value with a space. You can enter multiple *ip-precedence-list* lines to match more than four precedence values. The range is 0 to 7.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show class-map**	Verifies your entries.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Use the **no** form of the appropriate command to delete an existing class map or remove a match criterion.

This example shows how to create a class map called *VOIP-ALL*, which matches incoming traffic with a DSCP value of *cs5*.

```
Switch(config)# class-map match-all VOIP-ALL
Switch(config-cmap)# match dscp cs5
Switch(config-cmap)# exit
```

Using Table Maps to Classify IPv6 Traffic

You can use table maps to manage a large number of traffic flows with a single command. Table maps are used only in input policy maps. The Cisco ME 3400 Ethernet Access switch supports the following table maps for IPv6 traffic:

- DSCP to CoS, precedence, or DSCP
- Precedence to CoS, DSCP, or precedence
- CoS to DSCP, precedence, or CoS

For more informationon configuring the above table maps,see the "Configuring Table Maps" section on page 34-40 in Chapter 34, "Configuring QoS.".

Using Default Class to Classify IPv6 Traffic

All ingress and egress IPv6 traffic that is not classified by any of the user-defined classes will fall into class-default.

Displaying IPv6 QoS

The commands listed in Table 40-1 apply to both IPv4 and IPv6 traffic when a dual-ipv4-and-ipv6 template is configured. For explanation about available keywords, see the command reference for this release.

Table 40-1 Commands for Displaying Standard QoS Information

Command	Purpose
show class-map [*class-map-name*]	Displays QoS class-map information for all class maps or the specified class map.
show policer aggregate [*aggregate-policer-name*]	Displays information about all aggregate policers or the specified aggregate policer.
show policy-map [*policy-map-name* | **interface** [*interface-id*] [**input** | **output**] [**class** *class-name*]]	Displays QoS policy map information for the specified policy map name, interface, input or output policy maps, or policy-map class.
show cpu traffic qos	Displays the QoS marking values for CPU-generated traffic.
show running-config	Displays the configured class maps, policy maps, table maps, and aggregate policers.
show table-map [*table-map-name*]	Displays information for all configured table maps or the specified table map.

On the Cisco ME3400 Ethernet Access switch, the functionality of all of the QoS **show** commands is the same for both IPv4 and IPv6.

For more information about displaying QoS information, see the "Displaying QoS Information" section on page 34-75.