CHAPTER 35

Configuring Quality of Service

Note QoS functionality on the Catalyst 4900M and the Supervisor Engine 6-E are equivalent.

This chapter describes how to configure quality of service (QoS) by using automatic QoS (auto-QoS) commands or by using standard QoS commands on a Catalyst 4500 series switch. It describes how to specify QoS configuration on different types of interfaces (access, Layer 2 trunk, Layer 3 routed, Etherchannel) as well as VLANs. It also describes how to specify different QoS configurations on different VLANs on a given interface (per-port per-VLAN QoS). This chapter describes QoS support on Supervisor Engines II-Plus to V-10GE and on Supervisor Engine 6-E.

The QoS configuration model supported on Supervisor Engines II-Plus to V-10G is called the *switch qos* model. However, the Supervisor Engine 6-E supports a different QoS configuration model known as *MQC* (Modular QoS CLI). Please refer to the appropriate configuration section for the supervisor engine on which QoS will be configured. For more information about MQC, see the "Modular Quality of Service Command-Line Interface" section of the *Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.3*.

This chapter consists of these sections:

Note For complete syntax and usage information for the switch commands used in this chapter, first look at the *Cisco Catalyst 4500 Series Switch Command Reference* and related publications at this location:

http://www.cisco.com/en/US/products//hw/switches/ps4324/index.html

If the command is not found inthe *Catalyst 4500 Command Reference*, it will be found in the larger Cisco IOS library. Refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/en/US/products/ps6350/index.html

Overview of QoS on Catalyst 4500 Series Switch

Typically, networks operate on a *best-effort* delivery basis, which means that all traffic has equal priority and an equal chance of being delivered in a timely manner. When congestion occurs, all traffic has an equal chance of being dropped.

QoS selects network traffic (both unicast and multicast), prioritizes it according to its relative importance, and uses congestion avoidance to provide priority-indexed treatment; QoS can also limit the bandwidth used by network traffic. QoS can make network performance more predictable and bandwidth utilization more effective.

This section contains the following subsections:

Prioritization

QoS implementation is based on the DiffServ architecture. This architecture specifies that each packet is classified upon entry into the network. The classification is carried in the IP packet header, using 6 bits from the deprecated IP type of service (TOS) field to carry the classification (*class*) information. Classification can also be carried in the Layer 2 frame. These special bits in the Layer 2 frame or a Layer 3 packet are described here and shown in Figure 35-1:

- Prioritization values in Layer 2 frames:

  Layer 2 Inter-Switch Link (ISL) frame headers have a 1-byte User field that carries an IEEE 802.1p class of service (CoS) value in the three least-significant bits. On interfaces configured as Layer 2 ISL trunks, all traffic is in ISL frames.

  Layer 2 802.1Q frame headers have a 2-byte Tag Control Information field that carries the CoS value in the three most-significant bits, which are called the User Priority bits. On interfaces configured as Layer 2 802.1Q trunks, all traffic is in 802.1Q frames except for traffic in the native VLAN.

  Other frame types cannot carry Layer 2 CoS values.

  Layer 2 CoS values range from 0 for low priority to 7 for high priority.

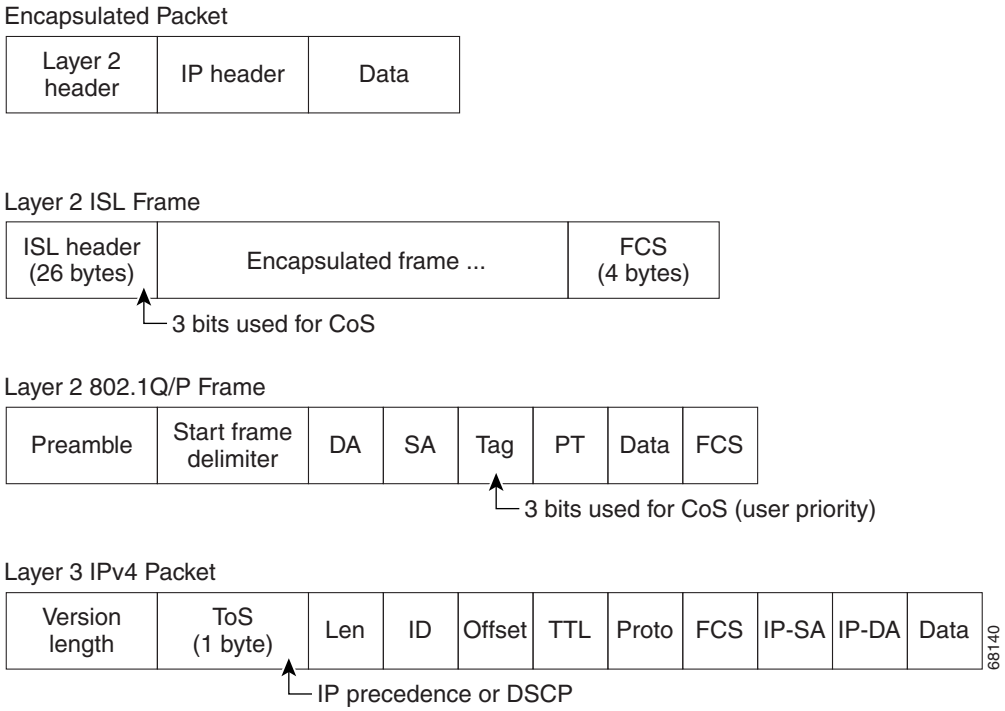
- Prioritization bits in Layer 3 packets:

  Layer 3 IP packets can carry either an IP precedence value or a Differentiated Services Code Point (DSCP) value. QoS supports the use of either value because DSCP values are backward-compatible with IP precedence values.

  IP precedence values range from 0 to 7.

DSCP values range from 0 to 63.

Figure 35-1 QoS Classification Layers in Frames and Packets



All switches and routers across the Internet rely on the class information to provide the same forwarding treatment to packets with the same class information and different treatment to packets with different class information. The class information in the packet can be assigned by end hosts or by switches or routers along the way, based on a configured policy, detailed examination of the packet, or both. Detailed examination of the packet is expected to happen closer to the edge of the network so that the core switches and routers are not overloaded.

Switches and routers along the path can use the class information to limit the amount of resources allocated per traffic class. The behavior of an individual device when handling traffic in the DiffServ architecture is called per-hop behavior. If all devices along a path provide a consistent per-hop behavior, you can construct an end-to-end QoS solution.

Implementing QoS in your network can be a simple or complex task and depends on the QoS features offered by your internetworking devices, the traffic types and patterns in your network, and the granularity of control you need over incoming and outgoing traffic.

QoS Terminology

The following terms are used when discussing QoS features:

- *Packets* carry traffic at Layer 3.
- *Frames* carry traffic at Layer 2. Layer 2 frames carry Layer 3 packets.
- *Labels* are prioritization values carried in Layer 3 packets and Layer 2 frames:
  - Layer 2 class of service (CoS) values, which range between zero for low priority and seven for high priority:

Layer 2 Inter-Switch Link (ISL) frame headers have a 1-byte User field that carries an IEEE 802.1p CoS value in the three least significant bits.

Layer 2 802.1Q frame headers have a 2-byte Tag Control Information field that carries the CoS value in the three most significant bits, which are called the User Priority bits.

Other frame types cannot carry Layer 2 CoS values.

Note On interfaces configured as Layer 2 ISL trunks, all traffic is in ISL frames. On interfaces configured as Layer 2 802.1Q trunks, all traffic is in 802.1Q frames except for traffic in the native VLAN.

- Layer 3 IP precedence values—The IP version 4 specification defines the three most significant bits of the 1-byte ToS field as IP precedence. IP precedence values range between zero for low priority and seven for high priority.
- Layer 3 differentiated services code point (DSCP) values—The Internet Engineering Task Force (IETF) has defined the six most significant bits of the 1-byte IP ToS field as the DSCP. The per-hop behavior represented by a particular DSCP value is configurable. DSCP values range between 0 and 63. See the "Configuring DSCP Maps" section on page 35-58.

Note Layer 3 IP packets can carry either an IP precedence value or a DSCP value. QoS supports the use of either value, since DSCP values are backwards compatible with IP precedence values. See Table 35-1.

Table 35-1 IP Precedence and DSCP Values

3-bit IP Precedence	6 MSb[1] of ToS						6-bit DSCP		3-bit IP Precedence	6 MSb[1] of ToS						6-bit DSCP
	8	7	6	5	4	3				8	7	6	5	4	3	
0	0	0	0	0	0	0	0		4	1	0	0	0	0	0	32
	0	0	0	0	0	1	1			1	0	0	0	0	1	33
	0	0	0	0	1	0	2			1	0	0	0	1	0	34
	0	0	0	0	1	1	3			1	0	0	0	1	1	35
	0	0	0	1	0	0	4			1	0	0	1	0	0	36
	0	0	0	1	0	1	5			1	0	0	1	0	1	37
	0	0	0	1	1	0	6			1	0	0	1	1	0	38
	0	0	0	1	1	1	7			1	0	0	1	1	1	39
1	0	0	1	0	0	0	8		5	1	0	1	0	0	0	40
	0	0	1	0	0	1	9			1	0	1	0	0	1	41
	0	0	1	0	1	0	10			1	0	1	0	1	0	42
	0	0	1	0	1	1	11			1	0	1	0	1	1	43
	0	0	1	1	0	0	12			1	0	1	1	0	0	44
	0	0	1	1	0	1	13			1	0	1	1	0	1	45
	0	0	1	1	1	0	14			1	0	1	1	1	0	46
	0	0	1	1	1	1	15			1	0	1	1	1	1	47

Table 35-1 IP Precedence and DSCP Values (continued)

3-bit IP Precedence	6 MSb[1] of ToS						6-bit DSCP		3-bit IP Precedence	6 MSb[1] of ToS						6-bit DSCP
	8	7	6	5	4	3				8	7	6	5	4	3	
2	0	1	0	0	0	0	16		6	1	1	0	0	0	0	48
	0	1	0	0	0	1	17			1	1	0	0	0	1	49
	0	1	0	0	1	0	18			1	1	0	0	1	0	50
	0	1	0	0	1	1	19			1	1	0	0	1	1	51
	0	1	0	1	0	0	20			1	1	0	1	0	0	52
	0	1	0	1	0	1	21			1	1	0	1	0	1	53
	0	1	0	1	1	0	22			1	1	0	1	1	0	54
	0	1	0	1	1	1	23			1	1	0	1	1	1	55
3	0	1	1	0	0	0	24		7	1	1	1	0	0	0	56
	0	1	1	0	0	1	25			1	1	1	0	0	1	57
	0	1	1	0	1	0	26			1	1	1	0	1	0	58
	0	1	1	0	1	1	27			1	1	1	0	1	1	59
	0	1	1	1	0	0	28			1	1	1	1	0	0	60
	0	1	1	1	0	1	29			1	1	1	1	0	1	61
	0	1	1	1	1	0	30			1	1	1	1	1	0	62
	0	1	1	1	1	1	31			1	1	1	1	1	1	63

1. MSb = most significant bit

- *Classification* is the selection of traffic to be marked.
- *Marking*, according to RFC 2475, is the process of setting a Layer 3 DSCP value in a packet; in this publication, the definition of marking is extended to include setting Layer 2 CoS values.
- *Scheduling* is the assignment of Layer 2 frames to a queue. QoS assigns frames to a queue based on internal DSCP values as shown in Internal DSCP Values, page 35-13.
- *Policing* is limiting bandwidth used by a flow of traffic. Policing can mark or drop traffic.

Basic QoS Model

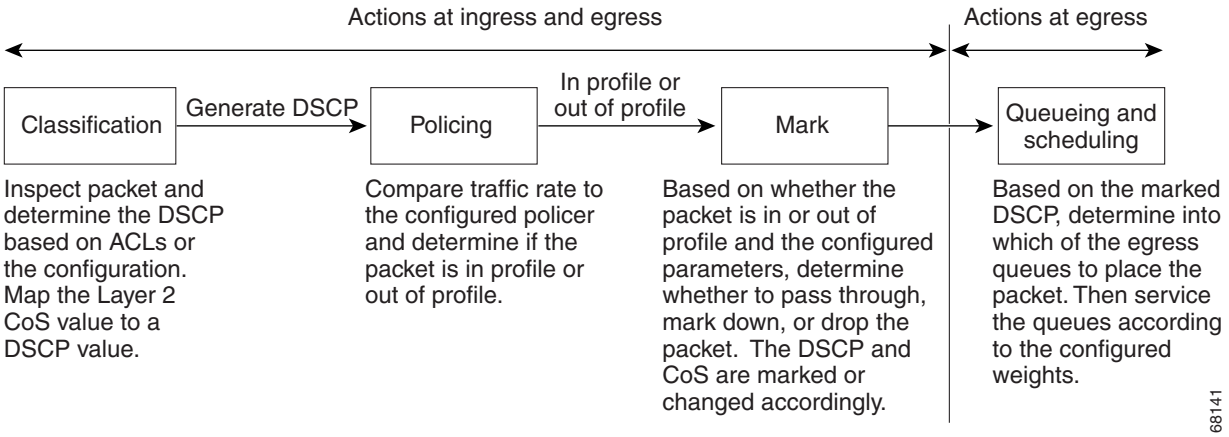
Figure 35-2 shows the basic QoS model (also referred to as Switch QoS model; it is not MQC compliant). Actions at the ingress and egress interfaces include classifying traffic, policing, and marking:

- Classifying distinguishes one kind of traffic from another. The process generates an internal DSCP for a packet, which identifies all the future QoS actions to be performed on this packet. For more information, see the "Classification" section on page 35-6.
- Policing determines whether a packet is in or out of profile by comparing the traffic rate to the configured policer, which limits the bandwidth consumed by a flow of traffic. The result of this determination is passed to the marker. For more information, see the "Policing and Marking" section on page 35-10.
- Marking evaluates the policer configuration information regarding the action to be taken when a packet is out of profile and decides what to do with the packet (pass through a packet without modification, mark down the DSCP value in the packet, or drop the packet). For more information, see the "Policing and Marking" section on page 35-10.

Actions at the egress interface include queueing and scheduling:

- Queueing evaluates the internal DSCP and determines which of the four egress queues in which to place the packet.
- Scheduling services the four egress (transmit) queues based on the sharing and shaping configuration of the egress (transmit) port. Sharing and shaping configurations are described in the "Queueing and Scheduling" section on page 35-14.

Figure 35-2 Basic QoS Model



Classification

Classification is the process of distinguishing one kind of traffic from another by examining the fields in the packet. Classification is enabled only if QoS is globally enabled on the switch. By default, QoS is globally disabled, so no classification occurs.

You specify which fields in the frame or packet that you want to use to classify incoming traffic.

Classification options are shown in Figure 35-3.

For non-IP traffic, you have the following classification options:

- Use the port default. If the packet is a non-IP packet, assign the default port DSCP value to the incoming packet.
- Trust the CoS value in the incoming frame (configure the port to trust CoS). Then use the configurable CoS-to-DSCP map to generate the internal DSCP value. Layer 2 ISL frame headers carry the CoS value in the three least-significant bits of the 1-byte User field. Layer 2 802.1Q frame headers carry the CoS value in the three most-significant bits of the Tag Control Information field. CoS values range from 0 for low priority to 7 for high priority. If the frame does not contain a CoS value, assign the default port CoS to the incoming frame.

  The trust DSCP configuration is meaningless for non-IP traffic. If you configure a port with trust DSCP and non-IP traffic is received, the switch assigns the default port DSCP.

Note On a Catalyst 4948-10GE, Supervisor V-10GE, and Supervisor V, when you send non-IP traffic(such as IPX) from port 1 with .1Q tag and a Pri=x value, for all x values, the transmit CoS at the output interface varies as srcMac changes.

"trust dscp" in a policy-map only works for IP packets. If a non-IP packet is matched by class with the “trust dscp" action, it could be transmitted with a random CoS value rather than treating the "trust dscp" as no-OP as assumed otherwise.

Workaround: Make the classification criteria in you class-maps more granular so that "trust dscp" is applied to a class that won't match non-IP packet. Separate the class-map matching both IP and non-IP packets into a set of class-maps so that "trust dscp" is applied to a class that matches only IPv4 traffic. For class-maps match non-IPv4 traffic, "trust cos" can be used.

For IP traffic, you have the following classification options:

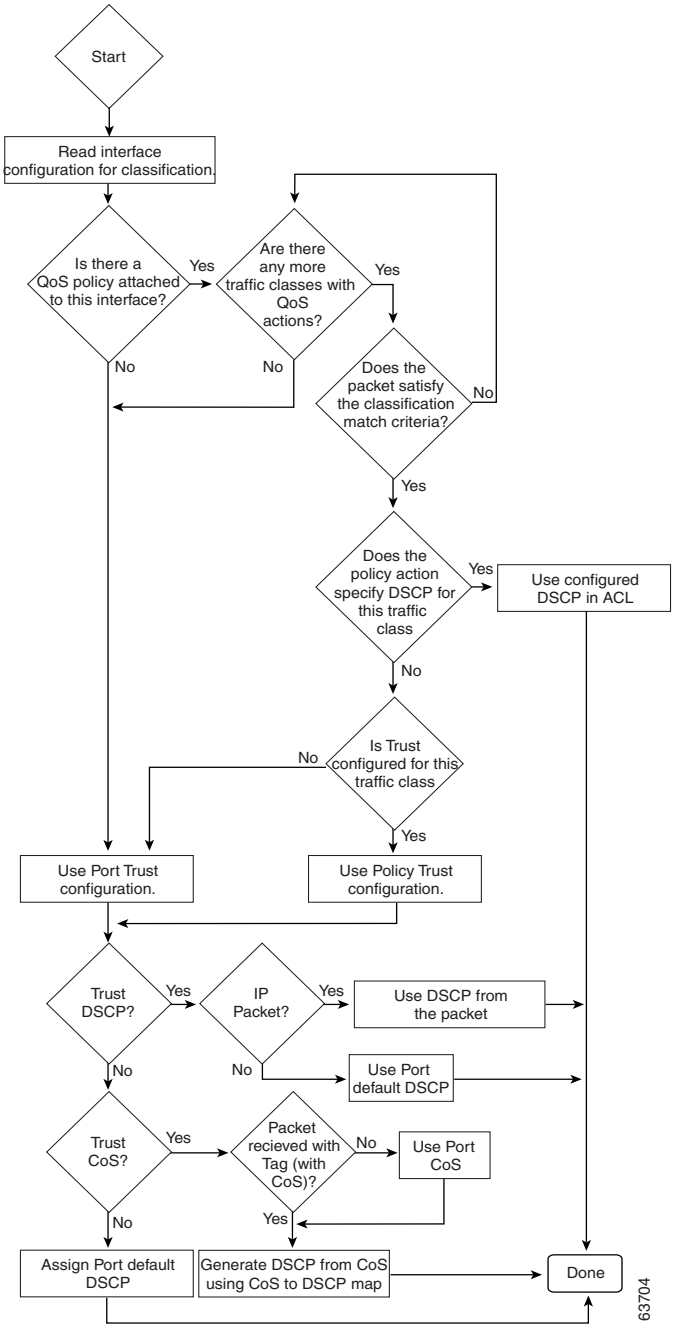
- Trust the IP DSCP in the incoming packet (configure the port to trust DSCP), and assign the same DSCP to the packet for internal use. The IETF defines the six most-significant bits of the 1-byte Type of Service (ToS) field as the DSCP. The priority represented by a particular DSCP value is configurable. DSCP values range from 0 to 63.
- Trust the CoS value (if present) in the incoming packet, and generate the DSCP by using the CoS-to-DSCP map.
- Perform the classification based on a configured IP standard or extended ACL, which examines various fields in the IP header. If no ACL is configured, the packet is assigned the default DSCP based on the trust state of the ingress port; otherwise, the policy map specifies the DSCP to assign to the incoming frame.

Note It is not possible to classify traffic based on the markings performed by an input QoS policy. In the Catalyst 4500 platform, the input and output QoS lookup happen in parallel, and therefore, input marked DSCP value cannot be used to classify traffic in the output QoS policy.

Note It is not possible to classify traffic based on *internal DSCP*. The *internal DSCP* is purely an internal classification mechanism used for all packets to determine transmit queue and transmit CoS values only.

For information on the maps described in this section, see the “Mapping Tables” section on page 35-14. For configuration information on port trust states, see the “Configuring the Trust State of Interfaces” section on page 35-53.

Figure 35-3 Classification Flowchart



Classification Based on QoS ACLs

A packet can be classified for QoS using multiple match criteria, and the classification can specify whether the packet should match all of the specified match criteria or at least one of the match criteria. To define a QoS classifier, you can provide the match criteria using the *match* statements in a class map. In the 'match' statements, you can specify the fields in the packet to match on, or you can use IP standard or IP extended ACLs. For more information, see the "Classification Based on Class Maps and Policy Maps" section on page 35-9.

If the class map is configured to match all the match criteria, then a packet must satisfy all the match statements in the class map before the QoS action is taken. The QoS action for the packet is not taken if the packet does not match even one match criterion in the class map.

If the class map is configured to match at least one match criterion, then a packet must satisfy at least one of the match statements in the class map before the QoS action is taken. The QoS action for the packet is not taken if the packet does not match any match criteria in the class map.

Note When you use the IP standard and IP extended ACLs, the permit and deny ACEs in the ACL have a slightly different meaning in the QoS context.

- If a packet encounters (and satisfies) an ACE with a "permit," then the packet "matches" the match criterion in the QoS classification.
- If a packet encounters (and satisfies) an ACE with a "deny," then the packet "does not match" the match criterion in the QoS classification.
- If no match with a permit action is encountered and all the ACEs have been examined, then the packet "does not match" the criterion in the QoS classification.

Note When creating an access list, remember that, by default, the end of the access list contains an implicit deny statement for everything if it did not find a match before reaching the end.

After a traffic class has been defined with the class map, you can create a policy that defines the QoS actions for a traffic class. A policy might contain multiple classes with actions specified for each one of them. A policy might include commands to classify the class as a particular aggregate (for example, assign a DSCP) or rate limit the class. This policy is then attached to a particular port on which it becomes effective.

You implement IP ACLs to classify IP traffic by using the **access-list** global configuration command. For configuration information, see the "Configuring a QoS Policy" section on page 35-33.

Classification Based on Class Maps and Policy Maps

A class map is a mechanism that you use to isolate and name a specific traffic flow (or class) from all other traffic. The class map defines the criterion used to match against a specific traffic flow to further classify it; the criteria can include matching the access group defined by the ACL or matching a specific list of DSCP, IP precedence, or L2 CoS values. If you have more than one type of traffic that you want to classify, you can create another class map and use a different name. After a packet is matched against the class-map criteria, you can specify the QoS actions via a policy map.

A policy map specifies the QoS actions for the traffic classes. Actions can include trusting the CoS or DSCP values in the traffic class; setting a specific DSCP or IP precedence value in the traffic class; or specifying the traffic bandwidth limitations and the action to take when the traffic is out of profile. Before a policy map can be effective, you must attach it to an interface.

You create a class map by using the **class-map** global configuration command. When you enter the **class-map** command, the switch enters the class-map configuration mode. In this mode, you define the match criteria for the traffic by using the **match** class-map configuration command.

You create and name a policy map by using the **policy-map** global configuration command. When you enter this command, the switch enters the policy-map configuration mode. In this mode, you specify the actions to take on a specific traffic class by using the **trust** or **set** policy-map configuration and policy-map class configuration commands. To make the policy map effective, you attach it to an interface by using the **service-policy** interface configuration command.

The policy map can also contain commands that define the policer, (the bandwidth limitations of the traffic) and the action to take if the limits are exceeded. For more information, see the "Policing and Marking" section on page 35-10.

A policy map also has these characteristics:

- A policy map can contain up to 255 class statements.
- You can have different classes within a policy map.
- A policy-map trust state supersedes an interface trust state.

For configuration information, see the "Configuring a QoS Policy" section on page 35-33.

Policing and Marking

After a packet is classified and has an internal DSCP value assigned to it, the policing and marking process can begin as shown in Figure 35-4.

Policing involves creating a policer that specifies the bandwidth limits for the traffic. Packets that exceed the limits are *out of profile* or *nonconforming*. Each policer specifies the action to take for packets that are in or out of profile. These actions, carried out by the marker, include passing through the packet without modification, dropping the packet, or marking down the packet with a new DSCP value that is obtained from the configurable policed-DSCP map. For information on the policed-DSCP map, see the "Mapping Tables" section on page 35-14.

You can create these types of policers:

- Individual

  QoS applies the bandwidth limits specified in the policer separately to each matched traffic class for each port/VLAN to which the policy map is attached to. You configure this type of policer within a policy map by using the **police** command under policy-map class configuration mode.

- Aggregate

  QoS applies the bandwidth limits specified in an aggregate policer cumulatively to all matched traffic flows. You configure this type of policer by specifying the aggregate policer name within a policy map by using the **police aggregate** policy-map configuration command. You specify the bandwidth limits of the policer by using the **qos aggregate-policer** global configuration command. In this way, the aggregate policer is shared by multiple classes of traffic within a policy map.

- Flow or Microflow

  With flow-based policing, all the identified flows are policed to the specified rate individually. Because the flows are dynamic, key distinguishing fields must be configured in class maps. Two flow-matching options are provided: *source ip based* (each flow with unique source IP address is treated as a new flow) and *destination ip based* (each flow with unique destination IP address is treated as new flow). For information on flow-based policer configuration, see "Configuring User Based Rate Limiting" on page 43.

When configuring policing and policers, keep these items in mind:

- For IP packets, only the length of the IP payload (the total length field in the IP header) is used by the policer for policing computation. The Layer 2 header and trailer length are not taken into account. For example, for a 64-byte Ethernet II IP packet, only 46 bytes are taken into account for policing (64 bytes - 14 byte Ethernet Header - 4 bytes Ethernet CRC).

  For non-IP packets, the Layer 2 length as specified in the Layer 2 Header is used by the policer for policing computation. To specify additional Layer 2 encapsulation length when policing IP packets, use the **qos account layer2 encapsulation** command.

- By default, no policers are configured.
- Only the average rate and committed burst parameters are configurable.
- Policing for individual and aggregate policers can occur in ingress and egress interfaces.
  - With the Supervisor Engine V-10GE (WS-X4516-10GE), 8192 policers are supported on ingress and on egress.
  - With all other supervisor engines, 1024 policers are supported on ingress and on egress.
  - The accuracy of the policer configured is +/- 1.5 per cent.

Note Four policers in ingress and egress direction are reserved.

- Policers can be of individual or aggregate type. On the Supervisor Engine V-10GE, flow based policers are supported.
- Policing for flow policers can occur on ingress Layer 3 interfaces only.
  - 512 unique flow policers can be configured on the Supervisor Engine V-10GE.

Note Because one flow policer is reserved by software, 511 unique flow policers can be defined.

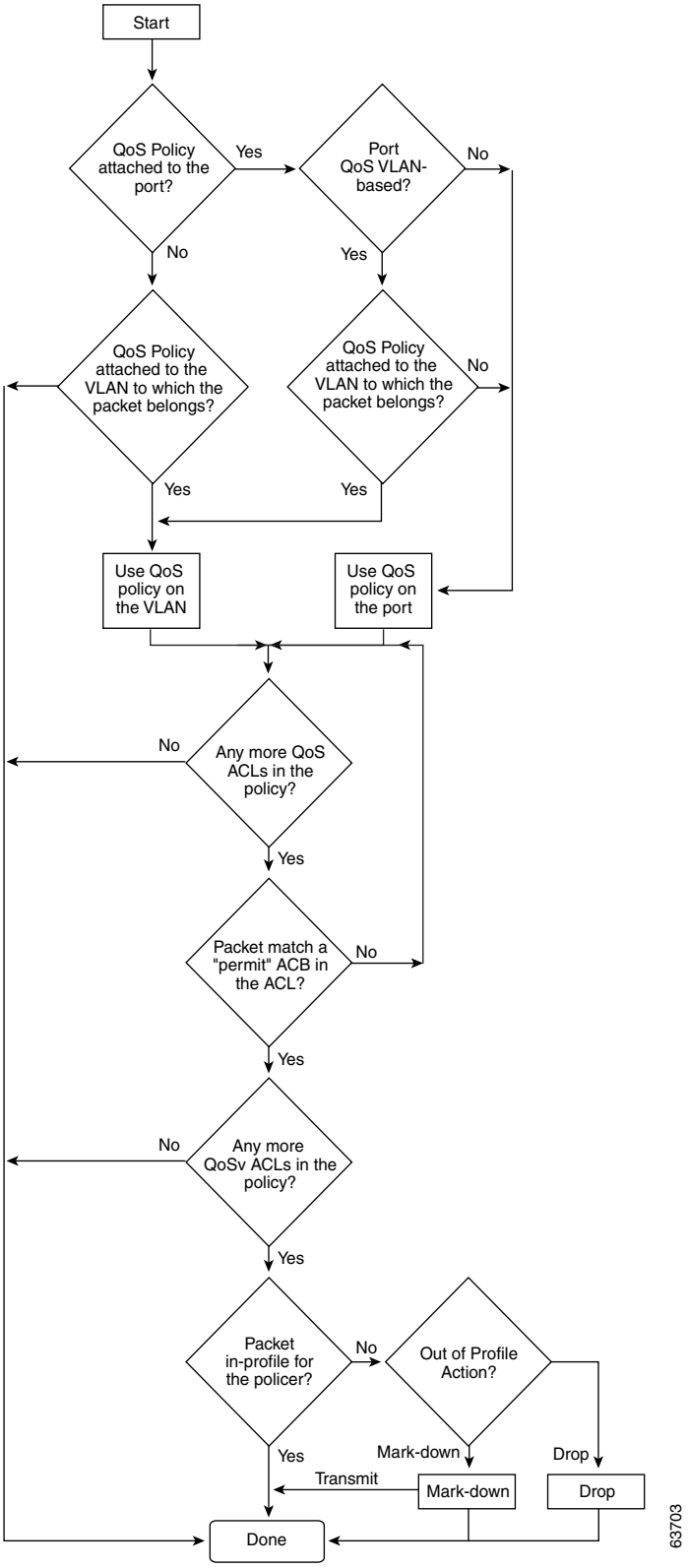
  - Greater than 100,000 flows can be microflow policed.

Note Microflow currently supports two flow matching options (source IP address based and destination IP address based). When microflow policing is used together with Netflow Statistics Collection, full flow statistics for the flows matching the source IP address or destination IP address are not available. For information on configuring Netflow Statistics, refer to "Enabling NetFlow Statistics Collection" section on page 49-7.

- On an interface configured for QoS, all traffic received or sent through the interface is classified, policed, and marked according to the policy map attached to the interface. However, if the interface is configured to use VLAN-based QoS (using the **qos vlan-based** command), the traffic received or sent through the interface is classified, policed, and marked according to the policy map attached to the VLAN (configured on the VLAN interface) to which the packet belongs. If there is no policy map attached to the VLAN to which the packet belongs, the policy map attached to the interface is used.

After you configure the policy map and policing actions, attach the policy to an ingress or egress interface by using the **service-policy** interface configuration command. For configuration information, see the "Configuring a QoS Policy" section on page 35-33 and the "Creating Named Aggregate Policers" section on page 35-31.

Figure 35-4 Policing and Marking Flowchart



Internal DSCP Values

The following sections describe the internal DSCP values:

Internal DSCP Sources

During processing, QoS represents the priority of all traffic (including non-IP traffic) with an internal DSCP value. QoS derives the internal DSCP value from the following:

- For trust-CoS traffic, from received or ingress interface Layer 2 CoS values
- For trust-DSCP traffic, from received or ingress interface DSCP values
- For untrusted traffic, from ingress interface DSCP value

The trust state of traffic is the trust state of the ingress interface unless set otherwise by a policy action for this traffic class.

QoS uses configurable mapping tables to derive the internal 6-bit DSCP value from CoS, which are 3-bit values (see the"Configuring DSCP Maps" section on page 35-58).

Egress ToS and CoS Sources

For egress IP traffic, QoS creates a ToS byte from the internal DSCP value and sends it to the egress interface to be written into IP packets. For **trust-dscp** and **untrusted** IP traffic, the ToS byte includes the original 2 least-significant bits from the received ToS byte.

Note The internal ToS value can mimic an IP precedence value (see Table 35-1 on page 35-4).

For all egress traffic, QoS uses a configurable mapping table to derive a CoS value from the internal ToS value associated with traffic (see the "Configuring the DSCP-to-CoS Map" section on page 35-60). QoS sends the CoS value to be written into ISL and 802.1Q frames.

For traffic received on an ingress interface configured to *trust CoS* using the **qos trust cos** command, the transmit CoS is always the incoming packet CoS (or the ingress interface default CoS if the packet is received untagged).

When the interface trust state is not configured to *trust dscp* using the **qos trust dscp** command, the security and QoS ACL classification always use the interface DSCP and not the incoming packet DSCP.

Mapping Tables

During QoS processing, the switch represents the priority of all traffic (including non-IP traffic) with an internal DSCP value:

- During classification, QoS uses configurable mapping tables to derive the internal DSCP (a 6-bit value) from received CoS. These maps include the CoS-to-DSCP map.
- During policing, QoS can assign another DSCP value to an IP or non-IP packet (if the packet is out of profile and the policer specifies a marked down DSCP value). This configurable map is called the policed-DSCP map.
- Before the traffic reaches the scheduling stage, QoS uses the internal DSCP to select one of the four egress queues for output processing. The DSCP-to-egress queue mapping can be configured using the **qos map dscp to tx-queue** command.

The CoS-to-DSCP and DSCP-to-CoS map have default values that might or might not be appropriate for your network.

For configuration information, see the "Configuring DSCP Maps" section on page 35-58.

Queueing and Scheduling

Each physical port has four transmit queues (egress queues). Each packet that needs to be transmitted is enqueued to one of the transmit queues. The transmit queues are then serviced based on the transmit queue scheduling algorithm.

Once the final transmit DSCP is computed (including any markdown of DSCP), the transmit DSCP to transmit queue mapping configuration determines the transmit queue. The packet is placed in the transmit queue of the transmit port, determined from the transmit DSCP. Use the **qos map dscp to tx-queue** command to configure the transmit DSCP to transmit queue mapping. The transmit DSCP is the internal DSCP value if the packet is a non-IP packet as determined by the QoS policies and trust configuration on the ingress and egress ports.

For configuration information, see the "Configuring Transmit Queues" section on page 35-55.

Active Queue Management

Active queue management (AQM) is the pro-active approach of informing you about congestion before a buffer overflow occurs. AQM is done using Dynamic buffer limiting (DBL). DBL tracks the queue length for each traffic flow in the switch. When the queue length of a flow exceeds its limit, DBL drop packets or set the Explicit Congestion Notification (ECN) bits in the packet headers.

DBL classifies flows in two categories, adaptive and aggressive. Adaptive flows reduce the rate of packet transmission once it receives congestion notification. Aggressive flows do not take any corrective action in response to congestion notification. For every active flow the switch maintains two parameters, "buffersUsed" and "credits". All flows start with "max-credits", a global parameter. When a flow with credits less than "aggressive-credits" (another global parameter) it is considered an aggressive flow and is given a small buffer limit called "aggressiveBufferLimit".

Queue length is measured by the number of packets. The number of packets in the queue determines the amount of buffer space that a flow is given. When a flow has a high queue length the computed value is lowered. This allows new incoming flows to receive buffer space in the queue. This allows all flows to get a proportional share of packets through the queue.

Because 4 transmit queues exist per interface and DBL is a per-queue mechanism, DSCP values can make DBL application more complex.

The following table provides the default DSCP-to-transmit queue mapping:

DSCP	txQueue
0-15	1
16-31	2
32-48	3
49-63	4

For example, if you are sending two streams, one with a DSCP of 16 and other with a value of 0, they will transmit from different queues. Even though an aggressive flow in txQueue 2 (packets with DSCP of 16) can saturate the link, packets with a DSCP of 0 will not be blocked by the aggressive flow, as they will transmit from txQueue 1. Thus, even without DBL, packets whose DSCP value places them in txQueue 1, 3, or 4 will not be dropped due to the aggressive flow.

Sharing Link Bandwidth Among Transmit Queues

The four transmit queues for a transmit port share the available link bandwidth of that transmit port. You can set the link bandwidth to be shared differently among the transmit queues using **bandwidth** command in interface transmit queue configuration mode. With this command, you assign the minimum guaranteed bandwidth for each transmit queue.

By default, all queues are scheduled in a round robin manner.

For systems using Supervisor Engine II-Plus, Supervisor Engine II-Plus TS, Supervisor Engine III, and Supervisor Engine IV, bandwidth can be configured on these ports only:

- Uplink ports on supervisor engines
- Ports on the WS-X4306-GB GBIC module
- Ports on the WS-X4506-GB-T CSFP module
- The 2 1000BASE-X ports on the WS-X4232-GB-RJ module
- The first 2 ports on the WS-X4418-GB module
- The two 1000BASE-X ports on the WS-X4412-2GB-TX module

For systems using Supervisor Engine V, bandwidth can be configured on all ports (10/100 Fast Ethernet, 10/100/1000BASE-T, and 1000BASE-X).

Strict Priority / Low Latency Queueing

You can configure transmit queue 3 on each port with higher priority using the **priority high** tx-queue configuration command in the interface configuration mode. When transmit queue 3 is configured with higher priority, packets in transmit queue 3 are scheduled ahead of packets in other queues.

When transmit queue 3 is configured at a higher priority, the packets are scheduled for transmission before the other transmit queues only if it has not met the allocated bandwidth sharing configuration. Any traffic that exceeds the configured shape rate is queued and transmitted at the configured rate. If the burst of traffic, exceeds the size of the queue, packets are dropped to maintain transmission at the configured shape rate.

Traffic Shaping

Traffic Shaping provides the ability to control the rate of outgoing traffic in order to make sure that the traffic conforms to the maximum rate of transmission contracted for it. Traffic that meets certain profile can be shaped to meet the downstream traffic rate requirements to handle any data rate mismatches.

Each transmit queue can be configured to transmit a maximum rate using the **shape** command. The configuration allows you to specify the maximum rate of traffic. Any traffic that exceeds the configured shape rate is queued and transmitted at the configured rate. If the burst of traffic exceeds the size of the queue, packets are dropped to maintain transmission at the configured shape rate.

Packet Modification

A packet is classified, policed, and queued to provide QoS. Packet modifications can occur during this process:

- For IP packets, classification involves assigning a DSCP to the packet. However, the packet is not modified at this stage; only an indication of the assigned DSCP is carried along. The reason for this is that QoS classification and ACL lookup occur in parallel, and it is possible that the ACL specifies that the packet should be denied and logged. In this situation, the packet is forwarded with its original DSCP to the CPU, where it is again processed through ACL software.
- For non-IP packets, classification involves assigning an internal DSCP to the packet, but because there is no DSCP in the non-IP packet, no overwrite occurs. Instead, the internal DSCP is used both for queueing and scheduling decisions and for writing the CoS priority value in the tag if the packet is being transmitted on either an ISL or 802.1Q trunk port.
- During policing, IP and non-IP packets can have another DSCP assigned to them (if they are out of profile and the policer specifies a markdown DSCP). Once again, the DSCP in the packet is not modified, but an indication of the marked-down value is carried along. For IP packets, the packet modification occurs at a later stage.

Note If you are running Supervisor Engines II-Plus, II-Plus-10GE, IV, V, V-10GE, or using a ME Catalyst 4900, Catalyst 4900, or Catalyst 4900-10GE series switch, you must enter the **qos rewrite ip dscp** command to enable update of packets and DSCP/ToS fields based on the switch's QoS policies.

Per Port Per VLAN QoS

Per-port per-VLAN QoS (PVQoS) offers differentiated quality-of-services to individual VLANs on a trunk port. It enables service providers to rate limit individual VLAN-based services on each trunk port to a business or a residence. In an enterprise Voice-over-IP environment, it can be used to rate limit voice VLAN even if an attacker impersonates an IP phone. A per-port per-VLAN service policy can be separately applied to either ingress or egress traffic.

QoS and Software Processed Packets

The Catalyst 4500 platform does not apply the QoS marking or policing configuration for any packets that are forwarded or generated by the Cisco IOS software. This means that any input or output QoS policy configured on the port or VLAN is not applied to packets if the Cisco IOS is forwarding or generating packets.

However, Cisco IOS marks all the generated control packets appropriately and uses the internal IP DSCP to determine the transmit queue on the output transmission interface. For IP packets, the internal IP DSCP is the IP DSCP field in the IP packet. For non-IP packets, Cisco IOS assigns a packet priority internally and maps it to an internal IP DSCP value.

Cisco IOS assigns an IP precedence of 6 to routing protocol packets on the control plane. As noted in RFC 791, "The Internetwork Control designation is intended for use by gateway control originators only." Specifically, Cisco IOS marks the following IP-based control packets: Open Shortest Path First (OSPF), Routing Information Protocol (RIP), Enhanced Interior Gateway Routing Protocol (EIGRP) hellos, and keepalives. Telnet packets to and from the router also receive an IP precedence value of 6. The assigned value remains with the packets when the output interface transmits them into the network.

For Layer 2 control protocols, the software assigns an internal IP DSCP. Typically, Layer 2 control protocol packets are assigned an internal DSCP value of 48 (corresponding to an IP precedence value of 6).

The internal IP DSCP is used to determine the transmit queue to which the packet is enqueued on the transmission interface. See “Configuring Transmit Queues” on page 55 for details on how to configure the DSCP to transmit queues.

The internal IP DSCP is also used to determine the transmit CoS marking if the packet is transmitted with a IEEE 802.1q or ISL tag on a trunk interface. See “Configuring the DSCP-to-CoS Map” on page 60 for details on how to configure the DSCP to CoS mapping.

Configuring Auto-QoS on Supervisor Engines II-Plus, II+10GE, IV, V, V-10GE, 4924, 4948, and 4948-10GE

Note LAN Base image does not support auto-QoS.

You can use the auto-QoS feature to simplify the deployment of existing QoS features. Auto-QoS makes assumptions about the network design, and as a result, the switch can prioritize different traffic flows and appropriately use the egress queues instead of using the default QoS behavior. (The default is that QoS is disabled. The switch then offers best-effort service to each packet, regardless of the packet content or size, and sends it from a single queue.)

When you enable auto-QoS, it automatically classifies traffic based on ingress packet label. The switch uses the resulting classification to choose the appropriate egress queue.

You use auto-QoS commands to identify ports connected to Cisco IP phones and to identify ports that receive trusted voice over IP (VoIP) traffic through an uplink. Auto-QoS then performs these functions:

- Detects the presence or absence of IP phones
- Configures QoS classification
- Configures egress queues

These sections describe how to configure auto-QoS on your switch:

Generated Auto-QoS Configuration

By default, auto-QoS is disabled on all interfaces.

When you enable the auto-QoS feature on the first interface, these automatic actions occur:

- QoS is globally enabled (**qos** global configuration command).
- DBL is enabled globally (**qos dbl** global configuration command)
- When you enter the **auto qos voip trust** interface configuration command, the ingress classification on the specified interface is set to trust the CoS label received in the packet if the specified interface is configured as Layer 2 (and is set to trust DSCP if the interface is configured as Layer 3). (See Table 35-2.)
- When you enter the **auto qos voip cisco-phone** interface configuration command, the trusted boundary feature is enabled. It uses the Cisco Discovery Protocol (CDP) to detect the presence or absence of a Cisco IP phone. When a Cisco IP phone is detected, the ingress classification on the interface is set to trust the CoS label received in the packet, if the interface is configured as Layer 2. (The classification is set to trust DSCP if the interface is configured as Layer 3.) When a Cisco IP phone is absent, the ingress classification is set to not trust the CoS label in the packet.

Note On a given port, the Cisco IP phone discovery information is not maintained on the standby supervisor engine. When the standby engine becomes active, it rediscovers the Cisco IP phone thru CDP. So, for a short time, the port will not be in the trust state after the SSO switchover.

For information about the trusted boundary feature, see the "Configuring a Trusted Boundary to Ensure Port Security" section on page 35-26.

When you enable auto-QoS by using the **auto qos voip cisco-phone** or the **auto qos voip trust** interface configuration commands, the switch automatically generates a QoS configuration based on the traffic type and ingress packet label and applies the commands listed in Table 35-2 to the interface.

Table 35-2 Generated Auto-QoS Configuration

Description	Automatically Generated Command

The switch automatically enables standard QoS and DBL configures the CoS-to-DSCP map (maps CoS values in incoming packets to a DSCP value).	`Switch(config)# qos`
	`Switch(config)# qos map cos 3 to 24`
	`Switch(config)# qos dbl`
	`Switch(config)# qos map cos 5 to 46`

The switch automatically configures the DSCP-to-Tx-queue mapping.	`Switch(config)# qos map dscp 24 25 26 27 b28 29 30 31 to tx-queue 4` `Switch(config)# qos map dscp 32 33 34 35 36 37 38 39 to tx-queue 4`

The switch automatically sets the ingress classification on the interface to trust the CoS/DSCP value received in the packet.	`Switch(config-if)# qos trust cos`
	`or`
	`Switch(config-if)# qos trust dscp`

The switch automatically creates a QoS service policy, enables DBL on the policy, and attaches it to the interface.	`Switch(config)# policy-map autoqos-voip-policy`
	`Switch(config-pmap)# class class-default`
	`Switch(config-pmap-c)# dbl`

Table 35-2 Generated Auto-QoS Configuration (continued)

Description	Automatically Generated Command
If you entered the **auto qos voip cisco-phone** command, the switch automatically enables the trusted boundary feature, which uses the CDP to detect the presence or absence of a Cisco IP phone.	`Switch(config-if)# qos trust device cisco-phone`

The switch assigns a higher priority for queue 3. Limit for shaping on queue 3 is selected so that it is 33 percent of the link speed. Configure shaping as 33 percent on those ports where sharing is supported.
This procedure ensures that the higher-priority queue does not starve other queues.
	`Switch(config-if)# tx-queue 3`
	`Switch(config-if-tx-queue)# priority high`
	`Switch(config-if-tx-queue)# shape percent 33`
	`Switch(config-if-tx-queue)# bandwidth percent 33`

Effects of Auto-QoS on the Configuration

When auto-QoS is enabled, the **auto qos voip** interface configuration command and the generated configuration are added to the running configuration.

Configuration Guidelines

Before configuring auto-QoS, you should be aware of this information:

- In this release, auto-QoS configures the switch only for VoIP with Cisco IP phones.
- To take advantage of the auto-QoS defaults, do not configure any standard-QoS commands before entering the auto-QoS commands. If necessary, you can fine-tune the QoS configuration, but we recommend that you do so only after the auto-QoS configuration is completed.
- You can enable auto-QoS on static, dynamic-access, voice VLAN access, and trunk ports.
- By default, the CDP is enabled on all interfaces. For auto-QoS to function properly, do not disable the CDP.
- To enable **auto qos voip trust** on Layer 3 interfaces, change the port to Layer 3, then apply auto-QoS to make it trust DSCP.

Enabling Auto-QoS for VoIP

To enable auto-QoS for VoIP within a QoS domain, perform this task:

	Command	Purpose
Step 1	`Switch# debug auto qos`	(Optional) Enables debugging for auto-QoS. When debugging is enabled, the switch displays the QoS commands that are automatically generated and applied when auto-QoS is enabled or disabled.
Step 2	`Switch# configure terminal`	Enters global configuration mode.

	Command	Purpose
Step 3	`Switch(config)# interface interface-id`	Enters interface configuration mode, and specify the interface that is connected to a Cisco IP phone or the uplink interface that is connected to another switch or router in the interior of the network.
Step 4	`Switch(config-if)# auto qos voip {cisco-phone | trust}`	Enables auto-QoS. The keywords have these meanings: • **cisco-phone**—If the interface is connected to a Cisco IP phone, the CoS labels of incoming packets are trusted only when the telephone is detected. • **trust**—The uplink interface is connected to a trusted switch or router, and the VoIP traffic classification in the ingress packet is trusted.
Step 5	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 6	`Switch# show auto qos interface interface-id`	Verifies your entries. This command displays the auto-QoS configuration that was initially applied; it does not display any user changes to the configuration that might be in effect.

To disable auto-QoS on an interface, use the **no auto qos voip** interface configuration command. When you enter this command, the switch changes the auto-QoS settings to the standard-QoS default settings for that interface. It does not change any global configuration performed by auto-QoS. Global configuration remains the same.

This example shows how to enable auto-QoS and to trust the CoS labels in incoming packets when the device connected to Fast Ethernet interface 1/1 is detected as a Cisco IP phone:

```
Switch(config)# interface fastethernet1/1
Switch(config-if)# auto qos voip cisco-phone
```

This example shows how to enable auto-QoS and to trust the CoS/DSCP labels in incoming packets when the switch or router connected to Gigabit Ethernet interface 1/1 is a trusted device:

```
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# auto qos voip trust
```

This example shows how to display the QoS commands that are automatically generated when auto-QoS is enabled:

```
Switch# debug auto qos
AutoQoS debugging is on
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# auto qos voip cisco-phone
```

Displaying Auto-QoS Information

To display the initial auto-QoS configuration, use the **show auto qos** [**interface** [*interface-id*]] privileged EXEC command. To display any user changes to that configuration, use the **show running-config** privileged EXEC command. You can compare the **show auto qos** and the **show running-config** command output to identify the user-defined QoS settings.

To display information about the QoS configuration that might be affected by auto-QoS, use one of these commands:

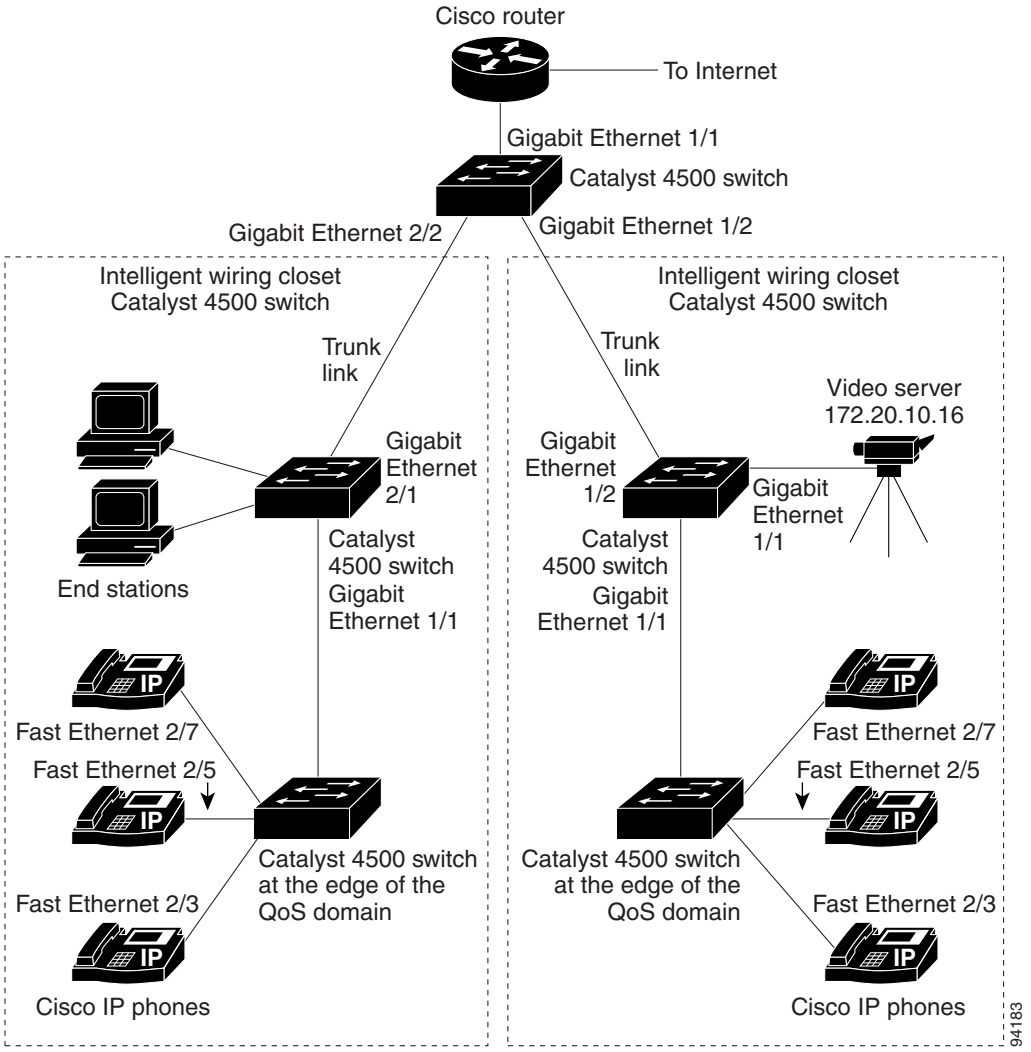
- **show qos**
- **show qos map**
- **show qos interface** [*interface-id*]

For more information about these commands, refer to the command reference for this release.

Auto-QoS Configuration Example

This section describes how you could implement auto-QoS in a network, as shown in Figure 35-5.

Figure 35-5 Auto-QoS Configuration Example Network



The intelligent wiring closets in Figure 35-5 are composed of Catalyst 4500 switches. The object of this example is to prioritize the VoIP traffic over all other traffic. To do so, enable auto-QoS on the switches at the edge of the QoS domains in the wiring closets.

Note You should not configure any standard QoS commands before entering the auto-QoS commands. You can fine-tune the QoS configuration, but we recommend that you do so only after the auto-QoS configuration is completed.

To configure the switch at the edge of the QoS domain to prioritize the VoIP traffic over all other traffic, perform this task:

	Command	Purpose
Step 1	`Switch# debug auto qos`	Enables debugging for auto-QoS. When debugging is enabled, the switch displays the QoS configuration that is automatically generated when auto-QoS is enabled.
Step 2	`Switch# configure terminal`	Enters global configuration mode.
Step 3	`Switch(config)# cdp enable`	Enables CDP globally. By default, CDP is enabled.
Step 4	`Switch(config)# interface fastethernet2/3`	Enters interface configuration mode.
Step 5	`Switch(config-if)# auto qos voip cisco-phone`	Enables auto-QoS on the interface, and specifies that the interface is connected to a Cisco IP phone. The CoS labels of incoming packets are trusted only when the IP phone is detected.
Step 6	`Switch(config)# interface fastethernet2/5`	Enters interface configuration mode.
Step 7	`Switch(config)# auto qos voip cisco-phone`	Enables auto-QoS on the interface, and specifies that the interface is connected to a Cisco IP phone.
Step 8	`Switch(config)# interface fastethernet2/7`	Enters interface configuration mode.
Step 9	`Switch(config)# auto qos voip cisco-phone`	Enables auto-QoS on the interface, and specifies that the interface is connected to a Cisco IP phone.
Step 10	`Switch(config)# interface gigabit1/1`	Enters interface configuration mode.
Step 11	`Switch(config)# auto qos voip trust`	Enables auto-QoS on the interface, and specifies that the interface is connected to a trusted router or switch.
Step 12	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 13	`Switch# show auto qos`	Verifies your entries. This command displays the auto-QoS configuration that is initially applied; it does not display any user changes to the configuration that might be in effect. For information about the QoS configuration that might be affected by auto-QoS, see the "Displaying Auto-QoS Information" section on page 35-20.

	Command	Purpose
Step 14	`Switch# show auto qos interface interface-id`	Verifies your entries. This command displays the auto-QoS configuration that was initially applied; it does not display any user changes to the configuration that might be in effect.
Step 15	`Switch# copy running-config startup-config`	Saves the **auto qos voip** interface configuration commands and the generated auto-QoS configuration in the configuration file.

Configuring QoS on Supervisor Engines II-Plus, II+10GE, IV, V, V-10GE, 4924, 4948, and 4948-10GE

Before configuring QoS, you must have a thorough understanding of these items:

- The types of applications used and the traffic patterns on your network.
- Traffic characteristics and needs of your network. Is the traffic bursty? Do you need to reserve bandwidth for voice and video streams?
- Bandwidth requirements and speed of the network.
- Location of congestion points in the network.

These sections describe how to configure QoS on the Catalyst 4000 family switch:

Default QoS Configuration

Table 35-3 shows the QoS default configuration.

Table 35-3 QoS Default Configuration

Feature	Default Value
Global QoS configuration	Disabled
Interface QoS configuration (port based)	Enabled when QoS is globally enabled
Interface CoS value	0
Interface DSCP value	0
CoS to DSCP map (DSCP set from CoS values)	CoS 0 = DSCP 0 CoS 1 = DSCP 8 CoS 2 = DSCP 16 CoS 3 = DSCP 24 CoS 4 = DSCP 32 CoS 5 = DSCP 40 CoS 6 = DSCP 48 CoS 7 = DSCP 56
DSCP to CoS map (CoS set from DSCP values)	DSCP 0–7 = CoS 0 DSCP 8–15 = CoS 1 DSCP 16–23 = CoS 2 DSCP 24–31 = CoS 3 DSCP 32–39 = CoS 4 DSCP 40–47 = CoS 5 DSCP 48–55 = CoS 6 DSCP 56–63 = CoS 7
Marked-down DSCP from DSCP map (Policed-DSCP)	Marked-down DSCP value equals original DSCP value (no markdown)
Policers	None
Policy maps	None
Transmit queue sharing	1/4 of the link bandwidth
Transmit queue size	1/4 of the transmit queue entries for the port. The transmit queue size of a port depends on the type of port, ranging from 240 packets per transmit queue to 1920 packets per transmit queue.
Transmit queue shaping	None
DCSP-to-Transmit queue map	DSCP 0–15 Queue 1 DSCP 16–31 Queue 2 DSCP 32–47 Queue 3 DSCP 48–63 Queue 4
High priority transmit queue	Disabled
With QoS disabled	
Interface trust state	Trust DSCP
With QoS enabled	With QoS enabled and all other QoS parameters at default values, QoS sets IP DSCP to zero and Layer 2 CoS to zero in all traffic transmitted.
Interface trust state	Untrusted

Configuration Guidelines

Before beginning the QoS configuration, you should be aware of this information:

- If you have EtherChannel ports configured on your switch, you must configure QoS classification and policing on the EtherChannel. The transmit queue configuration must be configured on the individual physical ports that comprise the EtherChannel.
- If the ip fragments match the source and destination configured in the ACL used to classify the traffic for quality of service, but do not match the layer 4 port numbers in the ACL, they are still matched with the ACL and may get prioritized. If the desired behavior is to give best effort service to ip fragments, following two ACEs should be added to the ACL used to classify the traffic.

```
access-list xxx deny udp any any fragments
access-list xxx deny tcp any any fragments
```

- It is not possible to match IP options against configured IP extended ACLs to enforce QoS. These packets are sent to the CPU and processed by software. IP options are denoted by fields in the IP header.
- Control traffic (such as spanning-tree BPDUs and routing update packets) received by the switch are subject to all ingress QoS processing.
- You cannot use **set** commands in policy maps if ip routing is disabled (enabled by default).
- On a dot1q tunnel port, only Layer 2 match criteria can be applied to tagged packets. However, all match criteria can be applied for untagged packets.
- On a trunk port, only Layer 2 match criteria can be applied to packets with multiple 802.1q tags.

Note QoS processes both unicast and multicast traffic.

Enabling QoS Globally

To enable QoS globally, perform this task:

	Command	Purpose
Step 1	`Switch# conf terminal`	Enter configuration mode.
Step 2	`Switch(config)# qos`	Enables QoS on the switch. Use the **no qos** command to globally disable QoS.
Step 3	`Switch(config)# end`	Exits configuration mode.
Step 4	`Switch# show qos`	Verifies the configuration.

This example shows how to enable QoS globally and verify the configuration:

```
Switch# config terminal
Switch(config)# qos
Switch(config)# end
Switch#
Switch# show qos
  QoS is enabled globally

Switch#
```

Enabling IP DSCP Rewrite

You must enter the **qos rewrite ip dscp** command to enable update of packets cos, DSCP/ToS fields based on the switch's QoS policies.

To enable IP DSCP rewrite, perform this task:

	Command	Purpose
Step 1	`Switch# conf terminal`	Enter configuration mode.
Step 2	`Switch(config)# [no] qos rewrite ip dscp`	Enables IP DSCP rewrite on the switch. Use the **no** command to disable IP DSCP rewrite
Step 3	`Switch(config)# end`	Exits configuration mode.
Step 4	`Switch# show qos`	Verifies the configuration.

This example shows how to enable IP DSCP rewrite and to verify the configuration:

```
Switch# conf terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# qos rewrite ip dscp
Switch(config)# end
Switch# show qos
QoS is enabled globally
IP header DSCP rewrite is enabled

Switch#
```

Note The **qos rewrite ip dscp** command is not supported on the Supervisor Engine 6-E and Catalyst 4900M chassis.

If you disable IP DSCP rewrite and enable QoS globally, the following events occur:

- The ToS byte on the IP packet is not modified.
- Marked and marked-down DSCP values are used for queueing.
- The internally derived DSCP (as per the trust configuration on the interface or VLAN policy) is used for transmit queue and Layer 2 CoS determination. The DSCP is not rewritten on the IP packet header.

If you disable QoS, the DSCP of the incoming packet are preserved and are not rewritten.

Configuring a Trusted Boundary to Ensure Port Security

In a typical network, you connect a Cisco IP phone to a switch port as discussed in Chapter 36, "Configuring Voice Interfaces." Traffic sent from the telephone to the switch is typically marked with a tag that uses the 802.1Q header. The header contains the VLAN information and the class of service (CoS) 3-bit field, which determines the priority of the packet. For most Cisco IP phone configurations, the traffic sent from the telephone to the switch is trusted to ensure that voice traffic is properly prioritized over other types of traffic in the network. By using the **qos trust cos** interface configuration command, you can configure the switch port to which the telephone is connected to trust the CoS labels of all traffic received on that port.

Note Starting with Cisco IOS Release 12.2(31)SG, Supervisor Engine V-10GE enables you to classify traffic based on packet's IP DSCP value irrespective of the port trust state. Because of this, even when a Cisco IP phone is not detected, data traffic can be classified based on IP DSCP values. Output queue selection is not impacted by this new behavior. It is still based on the incoming port trust configuration. For information on configuring transmit queues, refer to the "Configuring Transmit Queues" section on page 35-55".

In some situations, you also might connect a PC or workstation to the IP phone. In this case, you can use the **switchport priority extend cos** interface configuration command to configure the telephone through the switch CLI to override the priority of the traffic received from the PC. With this command, you can prevent a PC from taking advantage of a high-priority data queue.

However, if a user bypasses the telephone and connects the PC directly to the switch, the CoS labels generated by the PC are trusted by the switch (because of the trusted CoS setting) and can allow misuse of high-priority queues. The trusted boundary feature solves this problem by using the CDP to detect the presence of a Cisco IP phone (such as the Cisco IP Phone 7910, 7935, 7940, and 7960) on a switch port.

Note If CDP is not running on the switch globally or on the port in question, trusted boundary does not work.

When you configure trusted boundary on a port, trust is disabled. Then, when a phone is plugged in and detected, trust is enabled. (It may take a few minutes to detect the phone.) Now, when a phone is unplugged (and not detected), the trusted boundary feature disables the trusted setting on the switch port and prevents misuse of a high-priority queue.

Note On a given port, the Cisco IP phone discovery information is not maintained on the standby supervisor engine. When the standby engine becomes active, it rediscovers the Cisco IP phone thru CDP. So, for a short time, the port will not be in the trust state after the SSO switchover.

To enable trusted boundary on a port, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# interface interface-id`	Enters interface configuration mode, and specifies the interface connected to the IP phone. Valid interfaces include physical interfaces.
Step 3	`Switch(config)# qos trust [cos | dscp]`	Configures the interface to trust the CoS value in received traffic. By default, the port is not trusted.
Step 4	`Switch(config)# qos trust device cisco-phone`	Specifies that the Cisco IP phone is a trusted device. You cannot enable both trusted boundary and auto-QoS (**auto qos voip** interface configuration command) at the same time; they are mutually exclusive.
Step 5	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 6	`Switch# show qos interface interface-id`	Verifies your entries.
Step 7	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

To disable the trusted boundary feature, use the **no qos trust device cisco-phone** interface configuration command.

Enabling Dynamic Buffer Limiting

Note Supervisor Engine 6-E supports DBL through the MQC CLI, but not with the older *qos dbl* CLI.

Dynamic Buffer Limiting (DBL) provides active queue management on Cat4500 platforms. (Refer to "Active Queue Management" section on page 35-14 for details.)

Through "selective" DBL, you can select the flows that would be subjected (or would not be subjected) to the DBL algorithm. You ca n enable DBL globally, on specific IP DSCP values, or on specific CoS values.

The following tasks are discussed:

- Enabling DBL Globally, page 35-28
- Selectively Enable DBL, page 35-29

Enabling DBL Globally

To enable DBL globally on the switch, perform this task:

	Command	Purpose
Step 1	`Switch(config)# qos dbl`	Enables DBL on the switch. Use the **no qos dbl** command to disable AQM.
Step 2	`Switch(config)# end`	Exits configuration mode.
Step 3	`Switch# show qos dbl`	Verifies the configuration.

This example shows how to enable DBL globally and verify the configuration:

```
Switch# configure terminal
Switch(config)# qos dbl
Global DBL enabled
Switch(config)# end
Switch# show qos dbl
    QOS is enabled globally
    DBL is enabled globally on DSCP values:
          0-63
    DBL flow includes vlan
    DBL flow includes layer4-ports
    DBL does not use ecn to indicate congestion DBL exceed-action probability: 15% DBL max
    credits: 15 DBL aggressive credit limit: 10 DBL aggressive buffer limit: 2 packets
Switch#
```

You can enable DBL on the egress interface direction by applying a service-policy:

```
Switch# conf terminal
Switch(config)# policy-map dbl
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# dbl
Switch(config-pmap-c)# end
Switch#
```

```
00:08:12: %SYS-5-CONFIG_I: Configured from console by console
Switch# conf terminal
Switch(config)# int gig 1/2
Switch(config-if)# service-policy output dbl
Switch(config-if)# end
Switch#
```

Selectively Enable DBL

Note Selective DBL is not supported on Supervisor Engine 6-E.

DSCP values enable you to selectively apply DBL for IP Packets only (single or untagged). (Refer to the "Enable DBL on Specific IP DSCP Values" section on page 35-29.) To selectively apply DBL for non-IP packets or double-tagged packets (like Q-in-Q), you must use COS values as in the following section. (Refer to the "Enable DBL on Specific CoS Values" section on page 35-30.)

You can do the following:

- Enable DBL on Specific IP DSCP Values, page 35-29
- Enable DBL on Specific CoS Values, page 35-30

Enable DBL on Specific IP DSCP Values

DBL action is performed on transmit queues (4 per interface). You govern the mapping from IP DSCP to transmit queues with the **qos map dscp dscp-values to tx-queue queue-id** command. (Refer to "Configuring Transmit Queues" section on page 35-55 for details on how to do this.)

To enable DBL on specific IP DSCP values, perform this task:

	Command	Purpose
Step 1	`Switch(config)# [no] qos dbl dscp-based <value, value_range>`	Enables DBL on specific IP DSCP values.
Step 2	`Switch(config)# end`	Exits configuration mode.
Step 3	`Switch# show qos dbl`	Verifies the configuration.

This example shows how to selectively enable DBL on the DSCP values 1 through 10:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# qos dbl dscp-based 1-10
Switch(config)# end
Switch# show qos dbl
    QOS is enabled globally
    DBL is enabled globally on DSCP values:
          1-10
    DBL flow includes vlan
    DBL flow includes layer4-ports
    DBL does not use ecn to indicate congestion DBL exceed-action probability: 15%
    DBL max credits: 15
    DBL aggressive credit limit: 10
    DBL aggressive buffer limit: 2 packets
Switch#
```

This example shows how to selectively disable DBL on DSCP values 1 through 10 and to verify the configuration:

```
Switch# configure terminal
Switch(config)# no qos dbl dscp-based 1-5, 7
Switch(config)# end
Switch# show qos dbl
    QOS is enabled globally
    DBL is enabled globally on DSCP values:
          6,8-10
    DBL flow includes vlan
    DBL flow includes layer4-ports
    DBL does not use ecn to indicate congestion DBL exceed-action probability: 15% DBL max
    credits: 15 DBL aggressive credit limit: 10 DBL aggressive buffer limit: 2 packets
Switch#
```

Although you apply DBL based on class attributes other than DSCP, you still need to attach a policy-map to an egress interface ("Configuring Policy-Map Class Actions" section on page 35-37).

Provided the value has been set according to your network policies, you must configure "trust DSCP" on the ingress interface of the aggressive flow that DBL will throttle:

```
Interface <ingress>
  qos trust dscp
```

Enable DBL on Specific CoS Values

You might need to use COS values to selectively applying DBL if you intend to use non-IP packets or double-tagged packets (for example, Q-in-Q).

For single-tagged IP packets, use the following approach. Specify the global **qos dbl** dscp-based command as shown in the "Enable DBL on Specific IP DSCP Values" section on page 35-29).

```
Interface <ingress>
  switchport mode trunk
  qos trust cos
```

For non-IP packets or double-tagged packets, use the following method:

	Command	Purpose
Step 1	`Switch(config)# qos dbl`	Enables DBL globally.
Step 2	`Switch(config)# end`	Exits configuration mode.
Step 3	`Switch(config)# class-map cos`	Defines a traffic class.
Step 4	`Switch(config-cmap)# match cos x y`	Specifies CoS values used as match criteria.
Step 5	`Switch(config-cmap)# exit`	Returns to global configuration mode.
Step 6	`Switch(config)# policy-map cos`	Creates a policy map with a user-specified name.
Step 7	`Switch(config-pmap)# class cos`	Specifies the class map to be used by the policy map.
Step 8	`Switch(config-pmap-c)# dbl`	Enables DBL on the policy.
Step 9	`Switch(config-pmap-c)# end`	Returns to EXEC mode.
Step 10	`Switch# show policy-map cos`	Verifies configuration.
Step 11	`Switch# configure terminal`	Enters global configuration mode.
Step 12	`Switch(config)# interface gigabitEthernet 1/20`	Applies the configuration to an interface.

	Command	Purpose
Step 13	`Switch(config-if)# service-policy output cos`	Attaches the policy map to the interface.
Step 14	`Switch# show policy-map interface`	Verifies the configuration.

Note For more details on using CoS Mutation, refer to the "Configuring CoS Mutation" section on page 35-41.

To selectively enable DBL on CoS values 2 and 3:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# qos dbl
Switch(config)# end
Switch# configure terminal
Switch(config)# class-map cos
Switch(config-cmap)# match cos 2 3
Switch(config-cmap)# exit
Switch(config)# policy-map cos
Switch(config-pmap)# class cos
Switch(config-pmap-c)# dbl
Switch(config-pmap-c)# end
Switch# show policy-map cos
    Policy Map cos
      Class cos
        dbl
Switch# configure terminal
Switch(config)# interface gigabitEthernet 1/20
Switch(config-if)# service-policy output cos
Switch# show policy-map interface
 GigabitEthernet1/20

  Service-policy output: cos

    Class-map: cos (match-all)
      0 packets
      Match: cos  2  3
      dbl

    Class-map: class-default (match-any)
      0 packets
      Match: any
        0 packets
```

Creating Named Aggregate Policers

To create a named aggregate policer, perform this task:

Command	Purpose
`Switch(config)# qos aggregate-policer` *policer_name rate burst* `[[conform-action {transmit | drop}] [exceed-action {transmit | drop | policed-dscp-transmit}]]`	Creates a named aggregate policer.

An aggregate policer can be applied to one or more interfaces. However, if you apply the same policer to the input direction on one interface and to the output direction on a different interface, then you have created the equivalent of two different aggregate policers in the switching engine. Each policer has the same policing parameters, with one policing the ingress traffic on one interface and the other policing the egress traffic on another interface. If an aggregate policer is applied to multiple interfaces in the same direction, then only one instance of the policer is created in the switching engine.

Similarly, an aggregate policer can be applied to a port or to a VLAN. If you apply the same aggregate policer to a port and to a VLAN, then you have created the equivalent of two different aggregate policers in the switching engine. Each policer has the same policing parameters, with one policing the traffic on the configured port and the other policing the traffic on the configured VLAN. If an aggregate policer is applied to only ports or only VLANs, then only one instance of the policer is created in the switching engine.

In effect, if you apply a single aggregate policer to ports and VLANs in different directions, then you have created the equivalent of four aggregate policers; one for all ports sharing the policer in input direction, one for all ports sharing the policer in output direction, one for all VLANs sharing the policer in input direction and one for all VLANs sharing the policer in output direction.

When creating a named aggregate policer, note the following:

- The valid range of values for the *rate* parameter is as follows:
  - Minimum—32 kilobits per second
  - Maximum—32 gigabits per second

  See the "Configuration Guidelines" section on page 35-25.
- Rates can be entered in bits-per-second, or you can use the following abbreviations:
  - k to denote 1000 bps
  - m to denote 1000000 bps
  - g to denote 1000000000 bps

  Note You can also use a decimal point. For example, a rate of 1,100,000 bps can be entered as 1.1m.
- The valid range of values for the *burst* parameter is as follows:
  - Minimum—1 kilobyte
  - Maximum—512 megabytes
- Bursts can be entered in bytes, or you can use the following abbreviation:
  - k to denote 1000 bytes
  - m to denote 1000000 bytes
  - g to denote 1000000000 bytes

  Note You can also use a decimal point. For example, a burst of 1,100,000 bytes can be entered as 1.1m.
- Optionally, you can specify a conform action for matched in-profile traffic as follows:
  - The default conform action is **transmit**.
  - Enter the **drop** keyword to drop all matched traffic.

Note When you configure **drop** as the conform action, QoS configures **drop** as the exceed action.

- Optionally, for traffic that exceeds the CIR, you can specify an exceed action as follows:
  - The default exceed action is **drop**.
  - Enter the **policed-dscp-transmit** keyword to cause all matched out-of-profile traffic to be marked down as specified in the markdown map.
  - For no policing, enter the **transmit** keyword to transmit all matched out-of-profile traffic.
- You can enter the **no qos aggregate-policer** *policer_name* command to delete a named aggregate policer.

This example shows how to create a named aggregate policer with a 10 Mbps rate limit and a 1-MB burst size that transmits conforming traffic and marks down out-of-profile traffic.

```
Switch# config terminal
Switch(config)# qos aggregate-policer aggr-1 10000000 1000000 conform-action transmit
exceed-action policed-dscp-transmit
Switch(config)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show qos aggregate-policer aggr-1
Policer aggr-1
  Rate(bps):10000000 Normal-Burst(bytes):1000000
  conform-action:transmit exceed-action:policed-dscp-transmit
  Policymaps using this policer:
Switch#
```

Configuring a QoS Policy

The following subsections describe QoS policy configuration:

- Overview of QoS Policy Configuration, page 35-34
- Configuring a Class Map (Optional), page 35-34
- Configuring a Policy Map, page 35-36
- Attaching a Policy Map to an Interface, page 35-41

Note QoS policies process both unicast and multicast traffic.

Overview of QoS Policy Configuration

Configuring a QoS policy requires you to configure traffic classes and the policies that will be applied to those traffic classes, and to attach the policies to interfaces using these commands:

- **access-list** (optional for IP traffic—you can filter IP traffic with **class-map** commands):
  - QoS supports these access list types:

Protocol	Numbered Access Lists?	Extended Access Lists?	Named Access Lists?
IP	Yes: 1 to 99 1300 to 1999	Yes: 100 to 199 2000 to 2699	Yes

  - See Chapter 43, "Configuring Network Security with ACLs," for information about ACLs on the Catalyst 4500 series switches.
- **class-map** (optional)—Enter the **class-map** command to define one or more traffic classes by specifying the criteria by which traffic is classified. (See the "Configuring a Class Map (Optional)" section on page 35-34.)
- **policy-map**—Enter the **policy-map** command to define the following for each class of traffic:
  - Internal DSCP source
  - Aggregate or individual policing and marking
- **service-policy**—Enter the **service-policy** command to attach a policy map to an interface.

Configuring a Class Map (Optional)

The following subsections describe class map configuration:

Enter the **class-map** configuration command to define a traffic class and the match criteria that will be used to identify traffic as belonging to that class. Match statements can include criteria such as an ACL, an IP precedence value, or a DSCP value. The match criteria are defined with one match statement entered within the class-map configuration mode.

Creating a Class Map

To create a class map, perform this task:

Command	Purpose
`Switch(config)# [no] class-map [match-all | match-any] class_name`	Creates a named class map. Use the **no** keyword to delete a class map.

Configuring Filtering in a Class Map

To configure filtering in a class map, perform one of these tasks:

Command	Purpose
`Switch(config-cmap)# [no] match access-group {acl_index | name acl_name}`	(Optional) Specifies the name of the ACL used to filter traffic. Use the **no** keyword to remove the statement from a class map. **Note** Access lists are not documented in this publication. See the reference under **access-list** in the "Configuring a QoS Policy" section on page 35-33.
`Switch (config-cmap)# [no] match ip precedence ipp_value1 [ipp_value2 [ipp_valueN]]`	(Optional—for IP traffic only) Specifies up to eight IP precedence values used as match criteria. Use the **no** keyword to remove the statement from a class map.
`Switch (config-cmap)# [no] match ip dscp dscp_value1 [dscp_value2 [dscp_valueN]]`	(Optional—for IP traffic only) Specifies up to eight DSCP values used as match criteria. Use the **no** keyword to remove the statement from a class map.
`Switch (config-cmap)# [no] match cos value1 [value2] [value3} [value4}`	(Optional—for non-IPV4 traffic only) Specifies up to eight CoS values used as match criteria. Use the **no** keyword to remove the statement from a class map. For information on non-IPV4 traffic, see "Configuration Guidelines" section on page 35-19.
`Switch (config-cmap)# [no] match any`	(Optional) Matches any IP traffic or non-IP traffic.
`Switch (config-cmap)# match flow ip {source-address | destination-address`	(Optional) Treats each flow with a unique IP source address or destination address as a new flow.

Note Any Input or Output policy that uses a class map with the **match ip precedence** or **match ip dscp** class-map commands, requires that you configure the port on which the packet is received to **trust dscp**. If not, the IP packet DSCP/IP-precedence is not used for matching the traffic; instead, the receiving port's default DSCP is used. Starting with Cisco IOS Release 12.2(31)SG, the Supervisor Engine V-10GE enables you to classify traffic based on packet's IP DSCP value irrespective of port trust state.

Note With Cisco IOS Release 12.2(31), the Catalyst 4500 series switch supports Match CoS.

Note The interfaces on the Catalyst 4000 family switch do not support the **match classmap**, **match destination-address**, **match input-interface**, **match mpls**, **match not**, **match protocol**, **match qos-group**, and **match source-address** keywords.

Verifying Class Map Configuration

To verify class-map configuration, perform this task:

	Command	Purpose
Step 1	`Switch (config-cmap)# end`	Exits configuration mode.
Step 2	`Switch# show class-map class_name`	Verifies the configuration.

This example shows how to create a class map named *ipp5* and how to configure filtering to match traffic with IP precedence 5:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# class-map ipp5
Switch(config-cmap)# match ip precedence 5
Switch(config-cmap)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show class-map ipp5
 Class Map match-all ipp5 (id 1)
   Match ip precedence 5

Switch#
```

This example shows how to configure match CoS for non-IPV4 traffic and how to configure filtering to match traffic with CoS value of 5:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# class-map maptwo
Switch(config-cmap)# match cos 5
Switch(config-cmap)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show class-map maptwo
 Class Map match-all maptwo (id 1)
   Match cos 5

Switch#
```

Configuring a Policy Map

You can attach only one policy map to an interface. Policy maps can contain one or more policy-map classes, each with different match criteria and policers.

Configure a separate policy-map class in the policy map for each type of traffic that an interface receives. Put all commands for each type of traffic in the same policy-map class. QoS does not attempt to apply commands from more than one policy-map class to matched traffic.

The following sections describe policy-map configuration:

- Creating a Policy Map, page 35-37
- Configuring Policy-Map Class Actions, page 35-37

Creating a Policy Map

To create a policy map, perform this task:

Command	Purpose
`Switch(config)# [no] policy-map policy_name`	Creates a policy map with a user-specified name. Use the **no** keyword to delete the policy map.

Configuring Policy-Map Class Actions

These sections describe policy-map class action configuration:

- Configuring the Policy-Map Marking State, page 35-37
- Configuring the Policy-Map Class Trust State, page 35-37
- Configuring the Policy Map Class DBL State, page 35-38
- Configuring Policy-Map Class Policing, page 35-38
- Using a Named Aggregate Policer, page 35-38
- Configuring a Per-Interface Policer, page 35-38

Configuring the Policy-Map Marking State

To configure the policy map to mark the IP precedence or dscp of a packet, perform this task:

Command	Purpose
`Switch(config-pmap-c)# [no] set ip [precedence prec_value | dscp dscp_value]`	Configures the policy-map marking state, which decides the internal DSCP of the packet for subsequent processing. Use the **no** keyword to clear a configured value and return to the default.

Configuring the Policy-Map Class Trust State

To configure the policy-map class trust state, perform this task:

Command	Purpose
`Switch(config-pmap-c)# [no] trust {cos | dscp]`	Configures the policy-map class trust state, which selects the value that QoS uses as the source of the internal DSCP value (see the "Internal DSCP Values" section on page 35-13). Use the **no** keyword to clear a configured value and return to the default.

When configuring the policy-map class trust state, note the following:

- You can enter the **no trust** command to use the trust state configured on the ingress interface (this is the default).

- With the **cos** keyword, QoS sets the internal DSCP value from received or interface CoS.
- With the **dscp** keyword, QoS uses received DSCP.

Configuring the Policy Map Class DBL State

To configure the policy map class DBL state, perform this task:

Command	Purpose
`Switch(config-pmap-c)# [no] dbl`	Configures the policy-map class DBL state, which tracks the queue length of traffic flows (see the "Active Queue Management" section on page 35-14). Use the **no** keyword to clear an DBL value and return to the default.

When configuring the policy-map class DBL state, note the following:

- Any class that uses a named aggregate policer must have the same DBL configuration to work.

Configuring Policy-Map Class Policing

These sections describe configuration of policy-map class policing:

- Using a Named Aggregate Policer, page 35-38
- Configuring a Per-Interface Policer, page 35-38

Using a Named Aggregate Policer

To use a named aggregate policer (see the "Creating Named Aggregate Policers" section on page 35-31), perform this task:

Command	Purpose
`Switch(config-pmap-c)# [no] police aggregate` *aggregate_name*	Uses a previously defined aggregate policer. Use the **no** keyword to delete the policer from the policy map class.

Configuring a Per-Interface Policer

To configure a per-interface policer (see the "Policing and Marking" section on page 35-10), perform this task:

Command	Purpose
`Switch(config-pmap-c)# [no] police` *rate burst* `[[conform-action {transmit | drop}] [exceed-action {transmit | drop | policed-dscp-transmit}]]`	Configures a per-interface policer. Use the **no** keyword to delete a policer from the policy map class.

When configuring a per-interface policer, note the following:

- The valid range of values for the *rate* parameter is as follows:
  - Minimum—32 kilobits per second, entered as 32000
  - Maximum—32 gigabits per second, entered as 32000000000

Note See the "Configuration Guidelines" section on page 35-25.

- Rates can be entered in bits-per-second, or you can use the following abbreviations:
  - k to denote 1000 bps
  - m to denote 1000000 bps
  - g to denote 1000000000 bps

Note You can also use a decimal point. For example, a rate of 1,100,000 bps can be entered as 1.1m.

- The valid range of values for the *burst* parameter is as follows:
  - Minimum—1 kilobyte
  - Maximum—512 megabytes
- Bursts can be entered in bytes, or you can use the following abbreviation:
  - k to denote 1000 bytes
  - m to denote 1000000 bytes
  - g to denote 1000000000 bytes

Note You can also use a decimal point. For example, a burst of 1,100,000 bytes can be entered as 1.1m.

- Optionally, you can specify a conform action for matched in-profile traffic as follows:
  - The default conform action is **transmit**.
  - You can enter the **drop** keyword to drop all matched traffic.
- Optionally, for traffic that exceeds the CIR, you can enter the **policed-dscp-transmit** keyword to cause all matched out-of-profile traffic to be marked down as specified in the markdown map. See "Configuring the Policed-DSCP Map" section on page 35-59.
  - For no policing, you can enter the **transmit** keyword to transmit all matched out-of-profile traffic.

This example shows how to create a policy map named *ipp5-policy* that uses the class map named *ipp5*. The class map *ipp5* is configured to rewrite the packet precedence to 6 and to aggregate police the traffic that matches IP precedence value of 5:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# policy-map ipp5-policy
Switch(config-pmap)# class ipp5
Switch(config-pmap-c)# set ip precedence 6
Switch(config-pmap-c)# dbl
```

```
Switch(config-pmap-c)# police 2000000000 2000000 conform-action transmit exceed-action
policed-dscp-transmit
Switch(config-pmap-c)# end
```

This example shows how to create a policy map named cs2-policy that uses class map named cs2. The class map cos5 is configured to match on CoS 5 and to aggregate policing the traffic:

```
Switch(config)# class-map cs2
Switch(config-cmap)# match cos 5
Switch(config-cmap)# exit

Switch(config)# policy-map cs2-policy
Switch(config-pmap)# class cs2
police 2000000000 2000000 conform-action transmit exceed-action policed-dscp-transmit

Switch(config)# int g5/1
Switch(config-if)# service-policy input cs2-policy
Switch(config-if)# end

Switch# sh class-map cs2
 Class Map match-all cs2 (id 2)
   Match cos  5

Switch# sh policy-map cs2-policy
  Policy Map cs2-policy
    Class cs2
      police 2000000000 bps 2000000 byte conform-action transmit exceed-action
policed-dscp-transmit Switch#
```

Verifying Policy-Map Configuration

To verify policy-map configuration, perform this task:

	Command	Purpose
Step 1	`Switch(config-pmap-c)# end`	Exits policy-map class configuration mode. **Note** Enter additional **class** commands to create additional classes in the policy map.
Step 2	`Switch# show policy-map` *policy_name*	Verifies the configuration.

This example shows how to verify the configuration:

```
Switch# show policy-map ipp5-policy
show policy ipp5-policy
 Policy Map ipp5-policy
  class  ipp5
   set ip precedence 6
   dbl
police 2000000000 2000000 conform-action transmit exceed-action
policed-dscp-transmit
Switch#
```

Attaching a Policy Map to an Interface

To attach a policy map to an interface, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface {vlan vlan_ID | {fastethernet | gigabitethernet} slot/interface | Port-channel number}`	Selects the interface to configure.
Step 2	`Switch(config-if)# [no] service-policy input policy_map_name`	Attaches a policy map to the input direction of the interface. Use the **no** keyword to detach a policy map from an interface.
Step 3	`Switch(config-if)# end`	Exits configuration mode.
Step 4	`Switch# show policy-map interface {vlan vlan_ID | {fastethernet | gigabitethernet} slot/interface}`	Verifies the configuration.

Note You cannot enable marking commands on an interface until IP routing is enabled globally. If IP routing is disabled globally and you try to configure the service policy on an interface, the configuration is accepted but it does not take effect. You are prompted with the message: "`Set command will not take effect since CEF is disabled. Please enable IP routing and CEF globally`." To enable IP routing globally, issue the **ip routing** and **ip cef global** configuration commands. After you do this, the marking commands take effect.

This example shows how to attach the policy map named *pmap1* to Fast Ethernet interface 5/36 and to verify the configuration:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface fastethernet 5/36
Switch(config-if)# service-policy input pmap1
Switch(config-if)# end
Switch# show policy-map interface fastethernet 5/36
FastEthernet6/1

   service-policy input:p1

     class-map:c1 (match-any)
       238474 packets
       match:access-group 100
        38437 packets
       police:aggr-1
        Conform:383934 bytes Exceed:949888 bytes

    class-map:class-default (match-any)
      0 packets
      match:any
        0 packets
Switch#
```

Configuring CoS Mutation

Note Supervisor Engine 6-E does *not* support this feature.

Service providers providing Layer 2 VPNs carry double tagged or Q in Q traffic with the outer tag representing service provider's VLAN and the inner tag representing the customer's VLAN. Differentiated levels of service can be provided inside the SP network based on the CoS present in the outer tag.

By using CoS Mutation on a dot1q tunnel port, the CoS value on the outer tag of dot1q tunneled packets entering the provider core network can be derived from the CoS of the customer VLAN tag. This allows providers to preserve customer QoS semantics through their network.

CoS mutation is achieved through explicit user configuration to match on specific incoming CoS values and specifying the internal DSCP that is associated for matched packets. This internal DSCP gets converted to CoS through DSCP-CoS mapping during exit from the switch and is the CoS value that gets marked on the outer VLAN tag.

During the process, the CoS in inner tag is preserved and is carried across in the service provider's network.

The following example shows how a policy-map preserves customer VLAN IDs and CoS values throughout the network:

```
Class Map match-any c0
   Match cos  0

 Class Map match-any c1
   Match cos  1

 Class Map match-any c2
   Match cos  2

 Class Map match-any c3
   Match cos  3

 Class Map match-any c4
   Match cos  4

 Class Map match-any c5
   Match cos  5

 Class Map match-any c6
   Match cos  6

 Class Map match-any c7
   Match cos  7

Policy Map cos_mutation
    Class c0
      set dscp default

    Class c1
      set dscp cs1

    Class c2
      set dscp cs2

    Class c3
      set dscp cs3

    Class c4
      set dscp cs4

    Class c5
      set dscp cs5
```

```
    Class c6
      set dscp cs6

    Class c7
      set dscp cs7


interface GigabitEthernet5/1
 switchport access vlan 100

 switchport mode dot1q-tunnel
 service-policy input cos_mutation
```

Configuring User Based Rate Limiting

User Based Rate Limiting (UBRL) adopts microflow policing capability to dynamically learn traffic flows and rate limit each unique flow to an individual rate. UBRL is available on Supervisor Engine V-10GE with the built-in NetFlow support. UBRL can be applied to ingress traffic on routed interfaces with source or destination flow masks. It can support up to 85,000 individual flows and 511 rates. UBRL is typically used in environments where a per-user, granular rate-limiting mechanism is required; for example, the per-user outbound traffic rate could differ from the per-user inbound traffic rate.

Note By default, UBRL polices only routed IP traffic. You can use the **ip flow ingress layer2-switched** global command to police switched IP traffic. However, UBRL configuration must remain on a Layer 3 interface. With the UBRL configurations and the **ip flow ingress layer2-switched** global command, you will also be able to police intra-vlan flows. (See the "Configuring Switched/Bridged IP Flows" section on page 49-8). You do not need to enter the **ip flow ingress** command.

A flow is defined as a five-tuple (IP source address, IP destination address, IP head protocol field, Layer 4 source, and destination ports). Flow-based policers enable you to police traffic on a per flow basis. Because flows are dynamic, they require distinguishing values in the class map.

When you specify the **match flow** command with the **source-address** keyword, each flow with a unique source address is treated as a new flow. When you specify the **match flow command** with the **destination-address** keyword, each flow with a unique destination address is treated as a new flow. If the class map used by the policy map has any flow options configured, it is treated as a flow-based policy map. When you specify the **match flow** command with the **ip destination-address ip protocol L4 source-address L4 destination-address** keyword, each flow with unique IP source, destination, protocol, and Layer 4 source and destination address is treated as a new flow.

Note Microflow is only supported on Supervisor Engine V-10GE.

To configure the flow-based class maps and policy maps, perform this task:

	Command	Purpose
Step 1	`Switch(config)# class-map match-all class_name`	Creates a named class map.
Step 2	`Switch(config-cmap)# match flow ip {source-address | ip destination-address ip protocol L4 source-address L4 destination-address | destination-address}`	Specifies the key fields of the flow.

	Command	Purpose
Step 3	`Switch(config-cmap)# end`	Exits class-map configuration mode.
Step 4	`Switch# show class-map class-name`	Verifies the configuration.

Examples

Example 1

This example shows how to create a flow-based class map associated with a source address:

```
Switch(config)# class-map match-all c1
Switch(config-cmap)# match flow ip {source-address [ip destination_address ip protocol L4
source-address L4 destination address]}
Switch(config-cmap)# end
Switch#
Switch# show class-map c1
Class Map match-all c1 (id 2)
  Match flow  ip source-address
```

Example 2

This example shows how to create a flow-based class map associated with a destination address:

```
Switch(config)# class-map match-all c1
Switch(config-cmap)# match flow ip destination-address
Switch(config-cmap)# end
Switch#

Switch# show class-map c1
 Class Map match-all c1 (id 2)
   Match flow  ip destination-address
```

Example 3

Assume there are two active flows on the Fast Ethernet interface 6/1 with source addresses 192.168.10.20 and 192.168.10.21. The following example shows how to maintain each flow to 1 Mbps with an allowed burst value of 9000 bytes:

```
Switch# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# class-map c1
Switch(config-cmap)# match flow ip source-address
Switch(config-cmap)# exit
Switch(config)# policy-map p1
Switch(config-pmap)# class c1
Switch(config-pmap-c)# police 1000000 9000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface fa6/1
Switch(config-if)# service-policy input p1
Switch(config-if)# end
Switch# write memory

Switch# show policy-map interface
FastEthernet6/1

 Service-policy input: p1
```

```
  Class-map: c1 (match-all)
    15432182 packets
    Match: flow  ip source-address
    police: Per-interface
      Conform: 64995654 bytes Exceed: 2376965424 bytes

  Class-map: class-default (match-any)
    0 packets
    Match: any
      0 packets
```

Example 4

Assume there are two active flows on the Fast Ethernet interface 6/1 with destination addresses of 192.168.20.20 and 192.168.20.21. The following example shows how to maintain each flow to 1 Mbps with an allowed burst value of 9000 bytes:

```
Switch# conf terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# class-map c1
Switch(config-cmap)# match flow ip destination-address
Switch(config-cmap)# exit
Switch(config)# policy-map p1
Switch(config-pmap)# class c1
Switch(config-pmap-c)# police 1000000 9000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface fa6/1
Switch(config-if)# service-policy input p1
Switch(config-if)# end
Switch# write memory

Switch# show policy-map interface
 FastEthernet6/1

  Service-policy input: p1

    Class-map: c1 (match-all)
      2965072 packets
      Match: flow  ip destination-address
      police: Per-interface
        Conform: 6105636 bytes Exceed: 476652528 bytes

    Class-map: class-default (match-any)
      0 packets
      Match: any
        0 packets
```

Example 5

Assume that there are two active flows on FastEthernet interface 6/1:

```
SrcIp          DstIp          IpProt SrcL4Port DstL4Port
-------------------------------------------------------
192.168.10.10  192.168.20.20  20     6789       81
192.168.10.10  192.168.20.20  20     6789       21
```

With the following configuration, each flow is policed to 1000000 bps with an allowed 9000 burst value.

Note If you use the **match flow ip source-address|destination-address** command, these two flows are consolidated into one flow because they have the same source and destination address.

```
Switch# conf terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# class-map c1
Switch(config-cmap)# match flow ip source-address ip destination-address ip protocol l4
source-port l4 destination-port
Switch(config-cmap)# exit
Switch(config)# policy-map p1
Switch(config-pmap)# class c1
Switch(config-pmap-c)# police 1000000 9000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface fastEthernet 6/1
Switch(config-if)# service-policy input p1
Switch(config-if)# end
Switch# write memory
Switch# show policy-map interface
FastEthernet6/1

class-map c1
   match flow  ip source-address ip destination-address ip protocol l4 source-port l4
destination-port
!
 policy-map p1
    class c1
       police 1000000 bps 9000 byte conform-action transmit exceed-action drop
!
interface FastEthernet 6/1
  service-policy input p1

Switch# show class-map c1
 Class Map match-all c1 (id 2)
   Match flow  ip source-address ip destination-address ip protocol l4 source-port l4
destination-port

Switch# show policy-map p1
  Policy Map p1
    Class c1
      police 1000000 bps 9000 byte conform-action transmit exceed-action drop

Switch# show policy-map interface
 FastEthernet6/1

  Service-policy input: p1

    Class-map: c1 (match-all)
      15432182 packets
      Match: flow  ip source-address ip destination-address ip protocol l4 source-port l4
destination-port
      police: Per-interface
        Conform: 64995654 bytes Exceed: 2376965424 bytes


    Class-map: class-default (match-any)
      0 packets
      Match: any
        0 packets
```

Configuring Hierarchical Policers

Note Hierarchical policers are only supported on Supervisor Engine V-10GE.

You can tie flow policers with the existing policers to create dual policing rates on an interface. For example, using dual policing, you can limit all incoming traffic rates on a given interface to 50 Mbps and can limit the rate of each flow that is part of this traffic to 2 Mbps.

You can configure hierarchical policers with the **service-policy** policy-map config command. A policy map is termed *flow based* if the class map it uses matches any of the flow-based match criteria (such as **match flow ip source-address**). Each child policy map inherits all the match access-group commands of the parent.

Note You can configure only *flow based* policy maps as child policy maps. A parent policy map cannot be a flow-based policy map. Both the child policy map and parent policy map must have **match-all** in their class-map configuration.

To configure a flow based policy map as a child of an individual or aggregate policer, perform this task:

	Command	Purpose
Step 1	`Switch(config)# policy-map` *policy_name*	Specifies the individual or aggregate policy-map name.
Step 2	`Switch(config-pmap)# class` *class_name*	Specifies the class-map name of this policy map.
Step 3	`Switch(config-flow-cache)# service-policy` *service_policy_name*	Specifies the name of the flow-based policy map.

Note In a hierarchal policer configuration with parent as aggregate policer and child as microflow policer, child microflow policer matched packets report only the packets that are in the profile (that is, match the policing rate). Packets that exceed the policing rate are not reported in the class-map packet match statistics.

This example shows how to create a hierarchical policy map. A policy map with the name *aggregate-policy* has a class map with the name *aggregate-class*. A flow-based policy map with the name *flow-policy* is attached to this policy map as a child policy map.

```
Switch# config terminal
Switch(config)# policy-map aggregate-policy
Switch(config-pmap)# class aggregate-class
Switch(config-pmap-c)# service-policy flow-policy
Switch(config-pmap-c)# end
Switch#
```

In the following example, traffic in the IP address range of 101.237.0.0 to 101.237.255.255 is policed to 50 Mbps. Flows ranging from 101.237.10.0 to 101.237.10.255 are individually policed to a rate of 2 Mbps. This traffic goes through two policers: the aggregate policer and the other flow-based policer.

The following example shows the configuration for this scenario:

```
class-map match-all flow-class
 match flow ip source-address
 match access-group 20
!
class-map match-all aggregate-class
 match access-group 10
!
policy-map flow-policy
 class flow-class
   police 2000000 bps 10000 byte conform-action transmit exceed-action drop
!
policy-map aggregate-policy
 class aggregate-class
   police 50000000 bps 40000 byte conform-action transmit exceed-action drop
  service-policy flow-policy
!
access-list 10 permit 101.237.0.0 0.0.255.255
access-list 20 permit 0.0.10.0 255.255.0.255
```

The following example shows how to verify the configuration:

```
Switch# show policy-map flow-policy
 Policy Map flow-policy
   Class flow-class
     police 2000000 bps 10000 byte conform-action transmit exceed-action drop
Switch# show policy-map aggregate-policy
 Policy Map aggregate-policy
   Class aggregate-class
     police 50000000 bps 40000 byte conform-action transmit exceed-action drop
     service-policy flow-policy

Switch# show policy-map interface
FastEthernet6/1
 Service-policy input: aggregate-policy

   Class-map: aggregate-class (match-all)
     132537 packets
     Match: access-group 10
     police: Per-interface
       Conform: 3627000 bytes Exceed: 0 bytes

     Service-policy : flow-policy

       Class-map: flow-class (match-all)
         8867 packets
         Match: access-group 20
         Match: flow  ip source-address
         police: Per-interface
       Conform: 1649262 bytes Exceed: 59601096 bytes

       Class-map: class-default (match-any)
         0 packets
         Match: any            0 packets

   Class-map: class-default (match-any)
     5 packets
     Match: any        5 packets
```

Enabling Per-Port Per-VLAN QoS

The per-port per-VLAN QoS feature enables you to specify different QoS configurations on different VLANs on a given interface. Typically, you use this feature on trunk or voice VLANs (Cisco IP Phone) ports, as they belong to multiple VLANs.

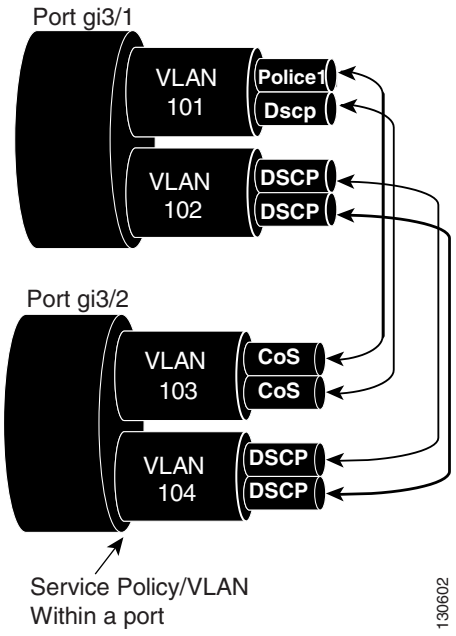
To configure per-port per-VLAN QoS, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface {fastethernet | gigabitethernet | tengigabitethernet} slot/interface | Port-channel number`	Selects the interface to configure.
Step 2	`Switch(config-if)# vlan-range vlan_range`	Specifies the VLANs involved.
Step 3	`Switch(config-if-vlan-range)# service-policy {input | output} policy-map`	Specifies the policy-map and direction.
Step 4	`Switch(config-if-vlan-range)# exit`	Exits class-map configuration mode.
Step 5	`Switch(config-if)# end`	Exits configuration interface mode.
Step 6	`Switch# show policy-map interface interface_name`	Verifies the configuration.

Example 1

Figure 35-6 displays a sample topology for configuring PVQoS. The trunk port gi3/1 is comprised of multiple VLANs (101 and 102). Within a port, you can create your own service policy per VLAN. This policy, performed in hardware, might consist of ingress and egress Policing, trusting DSCP, or giving precedence to voice packet over data.

Figure 35-6 Per-Port Per-VLAN Topology



The following configuration file shows how to perform ingress and egress policing per VLAN using the policy-map P31_QOS applied to port Gigabit Ethernet 3/1:

```
ip access-list 101 permit ip host 1.2.2.2 any
ip access-list 103 permit ip any any
Class-map match-all RT

match ip access-group 101
Class-map Match all PD

match ip access-group 103
Policy-map P31_QoS

Class RT


Police 200m 16k conform transmit exceed drop

Class PD


Police 100m 16k conform transmit exceed drop

Interface Gigabit 3/1
Switchport
Switchport trunk encapsulation dot1q
Switchport trunk allowed vlan 101-102
    Vlan range 101
        Service-policy input P31_QoS
        Service-policy output P31_QoS
    Vlan range 102
        Service-policy input P32_QoS
        Service-policy output P32_QoS
```

Example 2

Let us assume that interface Gigabit Ethernet 6/1 is a trunk port and belongs to VLANs 20, 300-301, and 400. The following example shows how to apply policy-map p1 for traffic in VLANs 20 and 400 and policy map p2 to traffic in VLANs 300 through 301:

```
Switch# configure terminal
Switch(config)# interface gigabitethernet 6/1
Switch(config-if)# vlan-range 20,400
Switch(config-if-vlan-range)# service-policy input p1
Switch(config-if-vlan-range)# exit
Switch(config-if)# vlan-range 300-301
Switch(config-if-vlan-range)# service-policy output p2
Switch(config-if-vlan-range)# end
Switch#
```

Example 3

The following command shows how to display policy-map statistics on VLAN 20 configured on Gigabit Ethernet interface 6/1:

```
Switch# show policy-map interface gigabitethernet 6/1 vlan 20
 GigabitEthernet6/1 vlan 20

  Service-policy input: p1
```

```
    Class-map: class-default (match-any)
      0 packets
      Match: any
        0 packets
      police: Per-interface
        Conform: 0 bytes Exceed: 0 bytes
```

Example 4

The following command shows how to display policy-map statistics on all VLANs configured on Gigabit Ethernet interface 6/1:

```
Switch# show policy-map interface gigabitethernet 6/1
 GigabitEthernet6/1 vlan 20

  Service-policy input: p1

    Class-map: class-default (match-any)
      0 packets
      Match: any
        0 packets
      police: Per-interface
        Conform: 0 bytes Exceed: 0 bytes

 GigabitEthernet6/1 vlan 300

  Service-policy output: p2

    Class-map: class-default (match-any)
      0 packets
      Match: any
        0 packets
      police: Per-interface
        Conform: 0 bytes Exceed: 0 bytes

 GigabitEthernet6/1 vlan 301

  Service-policy output: p2

    Class-map: class-default (match-any)
      0 packets
      Match: any
        0 packets
      police: Per-interface
        Conform: 0 bytes Exceed: 0 bytes

 GigabitEthernet6/1 vlan 400

  Service-policy input: p1

    Class-map: class-default (match-any)
      0 packets
      Match: any
        0 packets
      police: Per-interface
        Conform: 0 bytes Exceed: 0 bytes
```

Enabling or Disabling QoS on an Interface

The **qos** interface command reenables any previously configured QoS features. The **qos** interface command does not affect the interface queueing configuration.

To enable or disable QoS features for traffic from an interface, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface {vlan vlan_ID | {fastethernet | gigabitethernet} slot/interface | Port-channel number}`	Selects the interface to configure.
Step 2	`Switch(config-if)# [no] qos`	Enables QoS on the interface. Use the **no** keyword to disable QoS on an interface.
Step 3	`Switch(config-if)# end`	Exits configuration mode.
Step 4	`Switch# show qos interface`	Verifies the configuration.

This example shows how to disable QoS on interface VLAN 5:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface vlan 5
Switch(config-if)# no qos
Switch(config-if)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show qos | begin QoS is disabled
  QoS is disabled on the following interfaces:
    Vl5
<...Output Truncated...>
Switch#
```

Configuring VLAN-Based QoS on Layer 2 Interfaces

By default, QoS uses policy maps attached to physical interfaces. For Layer 2 interfaces, you can configure QoS to use policy maps attached to a VLAN. (See the "Attaching a Policy Map to an Interface" section on page 35-41.)

To configure VLAN-based QoS on a Layer 2 interface, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface {fastethernet | gigabitethernet} slot/interface | Port-channel number`	Selects the interface to configure.
Step 2	`Switch(config-if)# [no] qos vlan-based`	Configures VLAN-based QoS on a Layer 2 interface. Use the **no** keyword to disable VLAN-based QoS on an interface.
Step 3	`Switch(config-if)# end`	Exits configuration mode.
Step 4	`Switch# show qos`	Verifies the configuration.

Note If no input QoS policy is attached to a Layer 2 interface, then the input QoS policy attached to the VLAN (on which the packet is received), if any, is used even if the port is not configured as VLAN-based. If you do not want this default, attach a placeholder input QoS policy to the Layer 2 interface. Similarly,

if no output QoS policy is attached to a Layer 2 interface, then the output QoS policy attached to the VLAN (on which the packet is transmitted), if any, is used even if the port is not configured as VLAN-based. If you do not want this default, attach a placeholder output QoS policy to the layer 2 interface.

This example shows how to configure VLAN-based QoS on Fast Ethernet interface 5/42:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface fastethernet 5/42
Switch(config-if)# qos vlan-based
Switch(config-if)# end
```

This example shows how to verify the configuration:

```
Switch# show qos | begin QoS is vlan-based
QoS is vlan-based on the following interfaces:
    Fa5/42
Switch#
```

Note When a layer 2 interface is configured with VLAN-based QoS, and if a packet is received on the port for a VLAN on which there is no QoS policy, then the QoS policy attached to the port, if any is used. This applies for both Input and Output QoS policies.

Configuring the Trust State of Interfaces

This command configures the trust state of interfaces. By default, all interfaces are untrusted.

To configure the trust state of an interface, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface {vlan vlan_ID | {fastethernet | gigabitethernet} slot/interface | Port-channel number}`	Selects the interface to configure.
Step 2	`Switch(config-if)# [no] qos trust [dscp | cos]`	Configures the trust state of an interface. Use the **no** keyword to clear a configured value and return to the default.
Step 3	`Switch(config-if)# end`	Exits configuration mode.
Step 4	`Switch# show qos`	Verifies the configuration.

When configuring the trust state of an interface, note the following:

- You can use the **no qos trust** command to set the interface state to untrusted.
- For traffic received on an ingress interface configured to *trust CoS* using the **qos trust cos** command, the transmit CoS is always the incoming packet CoS (or the ingress interface default CoS if the packet is received untagged).
- When the interface trust state is not configured to *trust dscp* using the **qos trust dscp** command, the security and QoS ACL classification always use the interface DSCP and not the incoming packet DSCP.

- Starting with Cisco IOS Release 12.2(31)SG, the Supervisor Engine V-10GE enables you to classify a packet based on the packet's IP DSCP value irrespective of the port trust state. Packet transmit queuing isn't impacted by this behavior. For information on transmit queues, refer to the "Configuring Transmit Queues" section on page 35-55".

This example shows how to configure Gigabit Ethernet interface 1/1 with the **trust cos** keywords:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface gigabitethernet 1/1
Switch(config-if)# qos trust cos
Switch(config-if)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show qos interface gigabitethernet 1/1 | include trust
  Trust state: trust COS
Switch#
```

Configuring the CoS Value for an Interface

QoS assigns the CoS value specified with this command to untagged frames from ingress interfaces configured as trusted and to all frames from ingress interfaces configured as untrusted.

To configure the CoS value for an ingress interface, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface {fastethernet | gigabitethernet} slot/interface | Port-channel number}`	Selects the interface to configure.
Step 2	`Switch(config-if)# [no] qos cos default_cos`	Configures the ingress interface CoS value. Use the **no** keyword to clear a configured value and return to the default.
Step 3	`Switch(config-if)# end`	Exits configuration mode.
Step 4	`Switch# show qos interface {fastethernet | gigabitethernet} slot/interface`	Verifies the configuration.

This example shows how to configure the CoS 5 as the default on Fast Ethernet interface 5/24:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface fastethernet 5/24
Switch(config-if)# qos cos 5
Switch(config-if)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show qos interface fastethernet 5/24 | include Default COS
  Default COS is 5
Switch#
```

Configuring DSCP Values for an Interface

QoS assigns the DSCP value specified with this command to non IPv4 frames received on interfaces configured to trust DSCP and to all frames received on interfaces configured as untrusted.

To configure the DSCP value for an ingress interface, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface {fastethernet | gigabitethernet} slot/interface | Port-channel number`	Selects the interface to configure.
Step 2	`Switch(config-if)# [no] qos dscp default_dscp`	Configures the ingress interface DSCP value. Use the **no** keyword to clear a configured value and return to the default.
Step 3	`Switch(config-if)# end`	Exits configuration mode.
Step 4	`Switch# show qos interface {fastethernet | gigabitethernet} slot/interface`	Verifies the configuration.

This example shows how to configure the DSCP 5 as the default on Fast Ethernet interface 5/24:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface fastethernet 5/24
Switch(config-if)# qos dscp 5
Switch(config-if)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show qos interface fastethernet 6/1
QoS is enabled globally
Port QoS is enabled
  Port Trust State:CoS
  Default DSCP:0 Default CoS:0

  Tx-Queue    Bandwidth     ShapeRate     Priority    QueueSize
                (bps)         (bps)                   (packets)
     1        31250000      disabled      N/A         240
     2        31250000      disabled      N/A         240
     3        31250000      disabled      normal      240
     4        31250000      disabled      N/A         240
Switch#
```

Configuring Transmit Queues

The following sections describe how to configure transmit queues:

- Mapping DSCP Values to Specific Transmit Queues, page 35-56
- Allocating Bandwidth Among Transmit Queues, page 35-57

- Configuring Traffic Shaping of Transmit Queues, page 35-57
- Configuring a High Priority Transmit Queue, page 35-58

Depending on the complexity of your network and your QoS solution, you might need to perform all of the procedures in the following sections. However, you will first need to answer the following questions:

- Which packets are assigned (by DSCP value) to each queue?
- What is the size of a transmit queue relative to other queues for a given port?
- How much of the available bandwidth is allotted to each queue?
- What is the maximum rate and burst of traffic that can be transmitted out of each transmit queue?

Independent of the QoS state of an interface, a switch ensures that all the transmit queues are enabled. Because the DSCP values are trusted by default, a switch uses the appropriate transmit queues based on the DSCP to map them. This queue selection is based on the internal DSCP to transmit queue mapping table.

Mapping DSCP Values to Specific Transmit Queues

To map the DSCP values to a transmit queue, perform this task:

	Command	Purpose
Step 1	`Switch(config)# [no] qos map dscp dscp-values to tx-queue queue-id`	Maps the DSCP values to the transit queue. *dscp-list* can contain up 8 DSCP values. The *queue-id* can range from 1 to 4. Use the **no qos map dscp to tx-queue** command to clear the DSCP values from the transit queue.
Step 2	`Switch(config)# end`	Exits configuration mode.
Step 3	`Switch# show qos maps dscp tx-queues`	Verifies the configuration.

This example shows how to map DSCP values to transit queue 2.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# qos map dscp 50 to tx-queue 2
Switch(config)# end
Switch#
```

This example shows how to verify the configuration.

```
Switch# show qos maps dscp tx-queue
DSCP-TxQueue Mapping Table (dscp = d1d2)
d1 :d2  0  1  2  3  4  5  6  7  8  9
-----------------------------------
0 :   02 02 02 01 01 01 01 01 01 01
1 :   01 01 01 01 01 01 02 02 02 02
2 :   02 02 02 02 02 02 02 02 02 02
3 :   02 02 03 03 03 03 03 03 03 03
4 :   03 03 03 03 03 03 03 03 04 04
5 :   04 04 04 04 04 04 04 04 04 04
6 :   04 04 04 04
Switch#
```

Allocating Bandwidth Among Transmit Queues

To configure the transmit queue bandwidth, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface gigabitethernet slot/interface`	Selects the interface to configure.
Step 2	`Switch(config-if)# tx-queue queue_id`	Selects the transmit queue to configure.
Step 3	`Switch(config-if-tx-queue)# [no] [bandwidth rate | percent percent]`	Sets the bandwidth rate for the transmit queue. Use the **no** keyword to reset the transmit queue bandwidth ratios to the default values.
Step 4	`Switch(config-if-tx-queue)# end`	Exits configuration mode.
Step 5	`Switch# show qos interface`	Verifies the configuration.

The bandwidth rate varies with the interface.

For systems using Supervisor Engine II-Plus, Supervisor Engine II-Plus TS, Supervisor Engine III, and Supervisor Engine IV, bandwidth can be configured on these ports only:

- Uplink ports on supervisor engines
- Ports on the WS-X4306-GB GBIC module
- Ports on the WS-X4506-GB-T CSFP module
- The 2 1000BASE-X ports on the WS-X4232-GB-RJ module
- The first 2 ports on the WS-X4418-GB module
- The two 1000BASE-X ports on the WS-X4412-2GB-TX module

For systems using Supervisor Engine V, bandwidth can be configured on all ports (10/100 Fast Ethernet, 10/100/1000BASE-T, and 1000BASE-X).

This example shows how to configure the bandwidth of 1 Mbps on transmit queue 2.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface gigabitethernet 1/1
Switch(config-if)# tx-queue 2
Switch(config-if-tx-queue)#bandwidth 1000000
Switch(config-if-tx-queue)# end
Switch#
```

Configuring Traffic Shaping of Transmit Queues

To guarantee that packets transmitted from a transmit queue do not exceed a specified maximum rate, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface {fastethernet | gigabitethernet} slot/interface`	Selects the interface to configure.
Step 2	`Switch(config-if)# tx-queue queue_id`	Selects the transmit queue to configure.

	Command	Purpose
Step 3	`Switch(config-if-tx-queue)# [no] [shape rate | percent percent]`	Sets the transmit rate for the transmit queue. Use the **no** keyword to clear the transmit queue maximum rate.
Step 4	`Switch(config-if-tx-queue)# end`	Exits configuration mode.
Step 5	`Switch# show qos interface`	Verifies the configuration.

This example shows how to configure the shape rate to 1 Mbps on transmit queue 2.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface gigabitethernet 1/1
Switch(config-if-tx-queue)# tx-queue 2
Switch(config-if-tx-queue)# shape 1000000
Switch(config-if-tx-queue)# end
Switch#
```

Configuring a High Priority Transmit Queue

To configure transmit queue 3 at a higher priority, perform this task:

	Command	Purpose
Step 1	`Switch(config)# interface {fastethernet | gigabitethernet} slot/interface`	Selects the interface to configure.
Step 2	`Switch(config-if)# tx-queue 3`	Selects transmit queue 3 to configure.
Step 3	`Switch(config-if)# [no] priority high`	Sets the transmit queue to high priority. Use the **no** keyword to clear the transmit queue priority.
Step 4	`Switch(config-if)# end`	Exits configuration mode.
Step 5	`Switch# show qos interface`	Verifies the configuration.

This example shows how to configure transmit queue 3 to high priority.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface gigabitethernet 1/1
Switch(config-if-tx-queue)# tx-queue 3
Switch(config-if-tx-queue)# priority high
Switch(config-if)# end
Switch#
```

Configuring DSCP Maps

The following sections describes how to configure the DSCP maps. It contains this configuration information:

- Configuring the CoS-to-DSCP Map, page 35-59
- Configuring the Policed-DSCP Map, page 35-59
- Configuring the DSCP-to-CoS Map, page 35-60

All the maps are globally defined and are applied to all ports.

Configuring the CoS-to-DSCP Map

You use the CoS-to-DSCP map to map CoS values in incoming packets to a DSCP value that QoS uses internally to represent the priority of the traffic.

Table 35-4 shows the default CoS-to-DSCP map.

Table 35-4 Default CoS-to-DSCP Map

CoS value	0	1	2	3	4	5	6	7
DSCP value	0	8	16	24	32	40	48	56

If these values are not appropriate for your network, you need to modify them.

To modify the CoS-to-DSCP map, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# qos map cos` *cos1 ... cos8* `to dscp` *dscp*	Modifies the CoS-to-DSCP map. For *cos1...cos8*, you can enter up to 8 CoS; valid values range from 0 to 7. Separate each CoS value with a space. The *dscp* range is 0 to 63.
Step 3	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 4	`Switch# show qos maps cos-dscp`	Verifies your entries.
Step 5	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

This example shows how to configure the ingress CoS-to-DSCP mapping for CoS 0:

```
Switch# configure terminal
Switch(config)# qos map cos 0 to dscp 20
Switch(config)# end
Switch# show qos maps cos dscp

CoS-DSCP Mapping Table:
CoS:  0   1  2  3  4  5  6  7
--------------------------------
DSCP: 20  8 16 24 32 40 48 56
Switch(config)#
```

Note To return to the default map, use the **no qos cos to dscp** global configuration command.

This example shows how to clear the entire CoS-to-DSCP mapping table:

```
Switch(config)# no qos map cos to dscp
Switch(config)#
```

Configuring the Policed-DSCP Map

You use the policed-DSCP map to mark down a DSCP value to a new value as the result of a policing and marking action.

The default policed-DSCP map is a null map, which maps an incoming DSCP value to the same DSCP value.

To modify the CoS-to-DSCP map, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# qos map dscp policed` *dscp-list* `to dscp` *mark-down-dscp*	Modifies the policed-DSCP map. • For *dscp-list*, enter up to 8 DSCP values separated by spaces. Then enter the **to** keyword. • For *mark-down-dscp*, enter the corresponding policed (marked down) DSCP value.
Step 3	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 4	`Switch# show qos maps dscp policed`	Verifies your entries.
Step 5	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

To return to the default map, use the **no qos dscp policed** global configuration command.

This example shows how to map DSCP 50 to 57 to a marked-down DSCP value of 0:

```
Switch# configure terminal
Switch(config)# qos map dscp policed 50 51 52 53 54 55 56 57 to dscp 0
Switch(config)# end
Switch# show qos maps dscp policed
Policed-dscp map:
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ----------------------------------------
      0 :    00 01 02 03 04 05 06 07 08 09
      1 :    10 11 12 13 14 15 16 17 18 19
      2 :    20 21 22 23 24 25 26 27 28 29
      3 :    30 31 32 33 34 35 36 37 38 39
      4 :    40 41 42 43 44 45 46 47 48 49
      5 :    00 00 00 00 00 00 00 00 58 59
      6 :    60 61 62 63
```

Note In the previous policed-DSCP map, the marked-down DSCP values are shown in the body of the matrix. The d1 column specifies the most-significant digit of the original DSCP; the d2 row specifies the least-significant digit of the original DSCP. The intersection of the d1 and d2 values provides the marked-down value. For example, an original DSCP value of 53 corresponds to a marked-down DSCP value of 0.

Configuring the DSCP-to-CoS Map

You use the DSCP-to-CoS map to generate a CoS value.

Table 35-5 shows the default DSCP-to-CoS map.

Table 35-5 Default DSCP-to-CoS Map

DSCP value	0–7	8–15	16–23	24–31	32–39	40–47	48–55	56–63
CoS value	0	1	2	3	4	5	6	7

If the values above are not appropriate for your network, you need to modify them.

To modify the DSCP-to-CoS map, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# [no] qos map dscp` *dscp-list* `to cos` *cos*	Modifies the DSCP-to-CoS map. • For *dscp-list*, enter up to 8 DSCP values separated by spaces. Then enter the **to** keyword. • For *cos*, enter only one CoS value to which the DSCP values correspond. The DSCP range is 0 to 63; the CoS range is 0 to 7. To return to the default map, use the **no qos dscp to cos** global configuration command.
Step 3	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 4	`Switch# show qos maps dscp to cos`	Verifies your entries.
Step 5	`Switch# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

This example shows how to map DSCP values 0, 8, 16, 24, 32, 40, 48, and 50 to CoS value 0 and to display the map:

```
Switch# configure terminal
Switch(config)# qos map dscp 0 8 16 24 32 40 48 50 to cos 0
Switch(config)# end
Switch# show qos maps dscp cos
Dscp-cos map:
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ---------------------------------------
      0 :    00 00 00 00 00 00 00 00 00 01
      1 :    01 01 01 01 01 01 00 02 02 02
      2 :    02 02 02 02 00 03 03 03 03 03
      3 :    03 03 00 04 04 04 04 04 04 04
      4 :    00 05 05 05 05 05 05 05 00 06
      5 :    00 06 06 06 06 06 07 07 07 07
      6 :    07 07 07 07
```

Note In the previous DSCP-to-CoS map, the CoS values are shown in the body of the matrix. The d1 column specifies the most-significant digit of the DSCP; the d2 row specifies the least-significant digit of the DSCP. The intersection of the d1 and d2 values provides the CoS value. For example, in the DSCP-to-CoS map, a DSCP value of 08 corresponds to a CoS value of 0.

Enabling Layer 2 Control Packet QoS

Note Layer 2 Control Packet QoS is *not* supported on Supervisor Engine 6-E.

This feature solves the problem of high CPU utilization due to the ingress of a large amount of control packets. It does this by invalidating the QoS static entries corresponding to the protocol that you want to control (installed in the QoS CAM).

With this solution, hardware applies the actions corresponding to any user defined service policies that match the Layer 2 control traffic. You can deploy this mode of control through the CLI because the default mode will be the existing one.

You should configure policies to match on the required Layer 2 packets and police them to the desired level. Layer 2 control packets are essentially identified by a destination MAC address. When this feature is enabled on that packet type, if MACLs matching the desired control packets and the corresponding class-maps matching these MACLs are not already present, they will be auto-generated.

You are only required to use these class-maps in the policy-maps you create to police the control packets. You can then apply the policy-map on a per port, per vlan, or per-port-per-vlan just like any other policy-maps.

To enable Layer 2 Control Packet QoS, perform the following task:

	Command	Purpose
Step 1	`Switch# interface t`	Enters configuration mode.
Step 2	`Switch(config)# qos control-packets [bpdu-range | cdp-vtp | sstp | lldp]`	Enables Layer 2 Control Policing. You can specify the packet type that you will enable the feature on. The default is to select all packet types.
Step 3	`Switch(config)# end`	Exits configuration mode.
Step 4	`Switch# show run | inc qos control-packets`	Verifies the configuration.

The following table lists the types of packets impacted by this feature.

Table 35-6 Packet Type and Actionable Address Range

Type of packet the feature is enabled on	Range of address it acts on
BPDU-range	0180.C200.0000 BPDU 0180.C200.0002 OAM, LACP 0180.C200.0003 Eapol
CDP-VTP	0100.0CCC.CCCC
SSTP	0100.0CCC.CCCD
LLDP	0180.C200.000E

The following example illustrates how to configure Layer 2Control Packet QoS on cdp-vtp packets:

```
Switch# config terminal
Switch(config)# qos control-packets cdp-vtp
Switch(config)# end
Switch# show run | inc qos control-packets
qos control-packets cdp-vtp
```

Note If you enter **qos control-packet** without specifying any control packet types, the feature is enabled for all of them.

When the feature is enabled on all the packet types, the **qos control-packets** string appears in the output of the **show running** command:

```
Switch# show running | inc qos control-packets
qos control-packets
```

Now, if you disable the feature for sstp packets, you'll observe the following output:

```
Switch# show running | inc qos control-packets
qos control-packets bpdu-range
qos control-packets cdp-vtp
```

You can observe the status of the MACL and of the user-configured policies that will likely capture and drop/police the desired control packets with the **show running** and common relative commands.

To disable the feature, issue the **no qos control-packets [bpdu-range | cdp-vtp | sstp]** command. For example, to disable the feature on CDP-VTP packets, issue the **no qos control-packets cdp-vtp** command.

Note When you un-configure this feature for a specified protocol-type, the user- configured policies dealing with that protocol-type are immediately rendered ineffective. To save TCAM resources, you can remove the policies as well as the MACLs and class-maps (auto generated or user defined).

Note TCAM resources are not consumed when the interface is in a down state.

The following table displays the auto-generated MACLs and class-maps that are created when the feature is enabled on the corresponding packet type.

Table 35-7 Packet Types and Auto-Generated MACL/Class-Maps

Packet Type	Auto-Generated MACL/Class-Map
BPDU-range	mac access-list extended system-control-packet-bpdu-range permit any 0180.c200.0000 0000.0000.000c class-map match-any system-control-packet-bpdu-range match access-group name system-control-packet-bpdu-range
SSTP	mac access-list extended system-control-packet-sstp permit any host 0100.0ccc.cccd class-map match-any system-control-packet-sstp match access-group name system-control-packet-sstp
CDP-VTP	mac access-list extended system-control-packet-cdp-vtp permit any host 0100.0ccc.cccc class-map match-any system-control-packet-cdp-vtp match access-group name system-control-packet-cdp-vtp

The following example illustrates how to apply a MACL and policer configuration to BPDU-range packets:

- Enable the feature on the bpdu-range:

```
qos control-packets bpdu-range
```

- Create the corresponding MACL / class-maps (happens automatically):

```
mac access-list extended system-control-packet-bpdu-range
    permit any 0180.C200.0000 0000.0000.000C

class-map match-any system-control-packet-bpdu-range
    match access-group name system-control-packet-bpdu-range
```

- Create the policy-map and attach it to the desired interface / VLAN:

```
policy-map police-bpdu
  class system-control-packet-bpdu-range
    police 32000 bps 1000 byte conform-action transmit exceed-action drop

interface GigabitEthernet 1/1
      switchport trunk encapsulation dot1q
      switchport mode trunk
      vlan-range 100
          service-policy input police_bpdu
```

You should not modify the class-maps and MACLs that are generated by the system. Doing so might lead to unexpected behavior when the switch reloads or when the running configuration is updated from a file.

If you need to refine or modify these system generated class-maps or MACLs, you should create user-defined class-maps and MACLs. You can then use the newly created user-defined MACLs / class-maps to accomplish the desired policing.

Note The only restriction is that user-defined class-map names must begin with the prefix **system-control-packet-**. If the class-map does not begin with **system-control-packet-**, the configured QoS action may not be taken on certain supervisor engines.

Note There are no restrictions on the names you can use for user-defined MACLs.

For example, here are valid user defined class-maps names to police the control packets:

```
system-control-packet-bpdu1
system-control-packet-control-packet
system-control-packet-bla
```

Creating the user-defined MACLs / class-maps (as the example above) might be useful, for example, if you plan to define different class-maps for EAPOL, OAM, or BPDUs packets because the auto generated class-map system-control-packet-bpdu-range will match all of them:

```
mac access-list extended system-control-packet-bpdu
   permit any 0180.c200.0000
class-map match-any system-control-packet-bpdu
   match access-group name system-control-packet-bpdu

mac access-list extended system-control-packet-eapol
   permit any 0180.c200.0003
class-map match-any system-control-packet-eapol
   match access-group name system-control-packet-eapol
```

```
mac access-list extended system-control-packet-oam
   permit any 0180.c200.0002
class-map match-any system-control-packet-oam
   match access-group name system-control-packet-oam
```

Subsequently, you could use these class-maps to define different policers for each, instead of applying a common policer to the system-control-packet-bpdu-range.

Usage Guidelines

When this feature is enabled, you need to ensure that the existing policies applied to ports and VLANs are such that the Layer 2 control packets that you want to control are not inadvertently subjected to undesired QoS actions, and that the functionality of this feature is not impacted by other policies configured on the switch.

Before enabling QoS on the above mentioned control packets, you must examine and edit your new and existing policies to ensure that the classifiers in the policy-map matching the selected control packets are defined and configured in the correct sequence. To prevent undesirable results from actions of another classifier that may appear later in the same policy-map, you should place the classifiers matching control packets at the beginning of the policy map.

For actions associated with the class class-default, the behavior will depend on the type of supervisor engine:

- Supervisor Engine V-10GE with the built-in NetFlow support

  The actions associated with the class-default will never be applied on unmatched control packets, and a default permit action will be applied in case no control-packet class-maps are catching the control-packets before. Only the actions associated to policers that use the class-maps beginning with system-control-packet- will be applied on control packets.

- All other supervisor engines

  Actions associated with class-default are applied on unmatched control packets.

Note On Supervisor Engine V-10GE with the built-in NetFlow support, no micro-flow stats will be available for these types of packets.

Note When the feature is enabled on BPDU-range, you can police EAPOL packets only after that initial 802.1X authentication phase has completed.

Note When port security is enabled on a port that is in forwarding spanning tree state, Layer 2 control packets cannot be policed on that port.

Feature Interaction

After applying user-configured policies on each single flow, you might configure a CoPP policy *on top of* Layer 2 Control Packet QoS to rate limit the aggregate flow coming to the CPU. If so, CoPP essentially provides another level of protection for CPU by further rate-limiting on the egress side the packets already filtered in ingress on a per port per VLAN basis by the user defined policies. CoPP becomes the second level of defense while user-defined policies applied on ports and VLANs become the first level of defense.

For example, if you configure a policy-map matching and policing the BPDU-range traffic coming from interface Gigabit 1/1, VLAN 1 as follows:

```
policy-map police_bpdu_1
      class system-control-packet-bpdu-range
  police 32000 bps 1000 byte conform-action transmit exceed-action drop

   interface GigabitEthernet1/1
      switchport trunk encapsulation dot1q
      switchport mode trunk
      vlan-range 1
      service-policy input police_bpdu_1
```

and configure a second one on interface Gigabit 1/2 VLAN 2, matching and policing BPUD-range packets as follows:

```
policy-map police_bpdu_2
      class system-control-packet-bpdu-range
  police 34000 bps 1000 byte conform-action transmit exceed-action drop

   interface GigabitEthernet1/2
      switchport trunk encapsulation dot1q
      switchport mode trunk
      vlan-range 2
      service-policy input police_bpdu_2
```

the CoPP would be configured as follows:

```
policy-map system-cpp-policy
      class system-cpp-bpdu-range
  police 50000 bps 1000 byte conform-action transmit exceed-action drop
```

Note the following:

- On interface 1/1, VLAN 1, the BPDU-range packets will be policed accordingly to police_bpdu_1 at the rate of 32000 bit per second.
- On interface 1/2, VLAN 2, the BPDU-range packets will be policed accordingly to police_bpdu_2 at the rate of 34000 bit per second.
- The aggregate flow will be policed from CoPP at the CPU port at the rate of 50000 bit per second.

It is also possible to use named-aggregate policers applied to a group of ports or of VLANs to reduce the consumption of policer resources.

Note When port security is enabled on a port that is in forwarding spanning tree state, Layer 2 control packets cannot be policed on that port.

Configuring Auto-QoS on Supervisor Engine 6-E

Note LAN Base image does not support auto-QoS.

Note Auto-QoS cannot be applied to EtherChannel interfaces or VLANs.

Unlike auto-QoS on Supervisor Engines II-Plus to V-10GE, auto-QoS on Supervisor Engine 6-E employs the MQC model. This means that instead of using certain global configurations (like qos and qos dbl), auto-QoS applied to any interface on a switch with Supervisor Engine 6-E configures several global class-maps and policy-maps.

The class-maps are as follows:

```
class-map match-all AutoQos-VoIP-Control-Dscp26
  match  dscp af31
class-map match-all AutoQos-VoIP-Control-Dscp24
  match  dscp cs3
class-map match-all AutoQos-VoIP-Bearer-Cos
  match cos  5
class-map match-all AutoQos-VoIP-Control-QosGroup24
  match qos-group 24
class-map match-all AutoQos-VoIP-Control-QosGroup26
  match qos-group 26
class-map match-all AutoQos-VoIP-Bearer-QosGroup
  match qos-group 46
class-map match-all AutoQos-VoIP-Bearer-Dscp
  match  dscp ef
class-map match-all AutoQos-VoIP-Control-Cos
  match cos  3
```

The class maps are intended to identify control and data (bearer) voice traffic for either an Layer 2 or Layer 3 interface.

The policy maps are as follows:

```
policy-map AutoQos-VoIP-Input-Dscp-Policy
  class AutoQos-VoIP-Bearer-Dscp
   set qos-group 46
  class AutoQos-VoIP-Control-Dscp26
   set qos-group 26
  class AutoQos-VoIP-Control-Dscp24
   set qos-group 24
policy-map AutoQos-VoIP-Input-Cos-Policy
  class AutoQos-VoIP-Bearer-Cos
   set qos-group 46
  class AutoQos-VoIP-Control-Cos
   set qos-group 24
policy-map AutoQos-VoIP-Output-Policy
  class AutoQos-VoIP-Bearer-QosGroup
   set dscp ef
   set cos 5
    priority
   police cir percent 33
  class AutoQos-VoIP-Control-QosGroup26
   set dscp af31
   set cos 3
    bandwidth remaining percent 5
  class AutoQos-VoIP-Control-QosGroup24
   set dscp cs3
   set cos 3
    bandwidth remaining percent 5
  class class-default
    dbl
```

The three policy maps are defined as follows:

- policy-map AutoQos-VoIP-Input-Dscp-Policy

  This policy map is applied as an input service policy on an Layer 3 interface (such as an uplink connection to a neighboring switch) when auto-QoS is configured on the port.

- policy-map AutoQos-VoIP-Input-Cos-Policy

  This policy map is applied as an input service policy on an Layer 2 interface that could be either an uplink connection or a port hooked to a Cisco IP Phone.

- policy-map AutoQos-VoIP-Output-Policy

  This policy map is applied as an output policy for any port on which auto-QoS is configured, establishing policy governing egress traffic on the port based on whether it is voice data or control traffic.

The purpose of the input policy maps is to identify voice data or control traffic and mark it as such as it traverses the switch. The output policy map matches the packets on the marking occurring on ingress and then applies egress parameters such as bandwidth, policing and/or priority queuing.

The invocation of auto-QoS on a switch employing Supervisor Engine 6-E uses the same config commands used on Supervisor Engines II-Plus to V-10GE.

For switch-to-switch connections, the **[no] auto qos voice trust** command is used to apply an input and output service policy on the interface:

```
service-policy input AutoQos-VoIP-Input-Cos-Policy
```

OR

```
service-policy input AutoQos-VoIP-Input-Dscp-Policy
```

AND

```
service-policy output AutoQos-VoIP-Output-Policy
```

The selection of the input policy depends on whether the port is Layer 2 or Layer 3. For Layer 2, the policy trusts the Cos setting in the received packets. For Layer 3 ports, it relies on the DSCP value contained in the packets.

For phone connected ports, the **[no] auto qos voice cisco-phone** command is used to apply the following service policy to the port:

```
qos trust device cisco-phone

service-policy input AutoQos-VoIP-Input-Cos-Policy
```

AND

```
service-policy output AutoQos-VoIP-Output-Policy
```

It establishes a trusted boundary that recognizes Cisco IP Phones and trusts the Cos setting of the packets from the phone. If a Cisco IP Phone is not detected, the Cos field is ignored and the packets are not classified as voice traffic. Upon detecting a Cisco phone, the ingress packets are marked based on the Cos value in the packets. This marking is used on egress for proper traffic classification and handling.

Configuring QoS on Supervisor Engine 6-E

Note QoS functionality on the Catalyst 4900M and the Supervisor Engine 6-E are equivalent.

Note HQoS is not supported on Supervisor Engine 6-E.

Topics include:

MQC-based QoS Configuration

Starting with Cisco IOS Release 12.2(40)SG, a Catalyst 4500 Series Switch using Supervisor Engine 6-E employs the MQC model of QoS. To apply QoS, you use the Modular QoS Command-Line Interface (MQC), which 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, VLAN, or port and VLAN.

For more information about the MQC, see the "Modular Quality of Service Command-Line Interface" section of the *Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.3*.

Note The MQC model does not support the *trust* feature, which is available in the *switch qos* model on Supervisor Engines II-Plus through V-10GE. In the MQC model supported on the Supervisor Engine 6-E, the incoming traffic is considered trusted by default. Only when the *trusted boundary* feature is enabled on an interface can the port enter untrusted mode. In this mode, the switch marks the DSCP value of an IP packet and the CoS value of the VLAN tag on the Ethernet frame as "0".

Overview

The Supervisor Engine 6-E supports the best effort and DiffServ types of QoS deployment (RFCs 2597, 2598, 2474, 2475 define the DiffServ standards). A high level Supervisor Engine 6-E QoS model is as follows:

Step 1 The incoming packet is classified (based on different packet fields, receive port and/or VLAN) to belong to a traffic class.

Step 2 Depending on the traffic class, the packet is rate-limited/policed and its priority is optionally *marked* (typically at the edge of the network) so that lower priority packets are dropped or marked with lower priority in the packet fields (DSCP and CoS).

Step 3 After the packet has been marked, it is *looked up* for forwarding. This action obtains the transmit port and VLAN to transmit the packet.

Step 4 The packet is classified in the output direction based on the transmit port and/or VLAN. The classification takes into account any marking of the packet by input QoS.

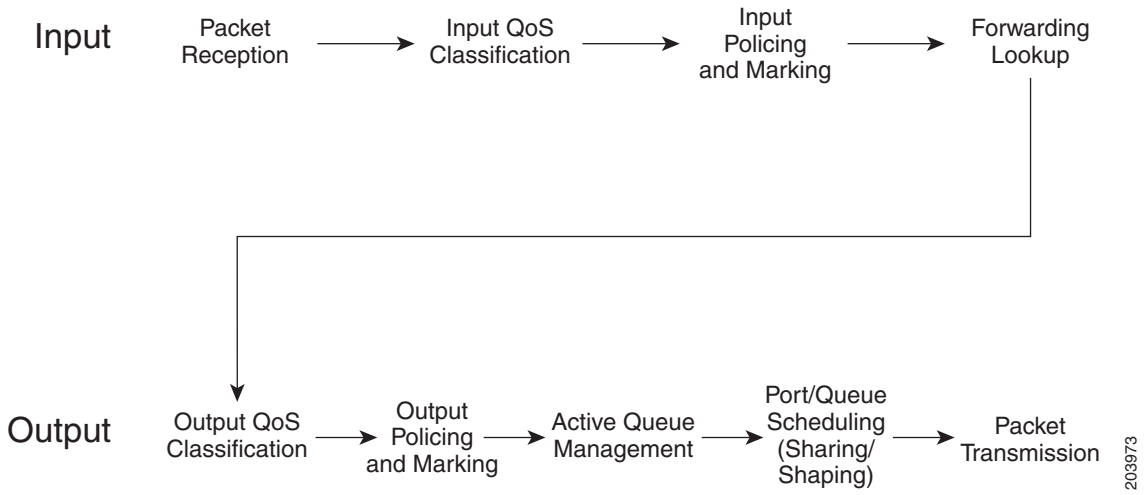
Step 5 Depending on the output classification, the packet is policed, its priority is optionally *(re-)marked*, and the transmit queue for the packet is determined depending on the traffic class.

Step 6 The transmit queue state is dynamically monitored via the AQM (Active Queue Management) algorithm and drop threshold configuration to determine whether the packet should be dropped or enqueued for transmission.

Step 7 If eligible for transmission, the packet is enqueued to a transmit queue. The transmit queue is selected based on output QoS classification criteria. The selected queue provides the desired behavior in terms of latency and bandwidth.

Figure 35-7 illustrates a high level model of Supervisor Engine 6-E.

Figure 35-7 QoS Packet Processing



Platform-supported Classification Criteria and QoS Features

The following table provides a summary of various classification criteria and actions supported on the Supervisor Engine 6-E. For details, refer to the *Catalyst 4500 Series Switch Command Reference*.

Supported classification actions	Descriptions
match access-group	Configures the match criteria for a class map on the basis of the specified ACL.
match any	Configures the match criteria for a class map to be successful match criteria for all packets.
match cos	Matches a packet based on a Layer 2 class of service (CoS) marking.
match destination-address mac	Uses the destination MAC address as a match criterion.

Supported classification actions	Descriptions
match source-address mac	Uses the source MAC address as a match criterion.
match [ip] dscp	Identifies a specific IP differentiated service code point (DSCP) value as a match criterion. Up to eight DSCP values can be included in one match statement.
match [ip] precedence	Identifies IP precedence values as match criteria.
match protocol	Configures the match criteria for a class map on the basis of the specified protocol.
match qos-group	Identifies a specific QoS group value as a match criterion. Applies only on the egress direction.
Supported Qos Features	**Descriptions**
police	Configures traffic policing.
police (percent)	Configures traffic policing on the basis of a percentage of bandwidth available on an interface.
police (two rates)	Configures traffic policing using two rates, the committed information rate (CIR) and the peak information rate (PIR).
set cos	Sets the Layer 2 class of service (CoS) value of an outgoing packet.
set dscp	Marks a packet by setting the differentiated services code point (DSCP) value in the type of service (ToS) byte of IPv4 or traffic class byte of IPv6 packet.
set precedence	Sets the precedence value in the packet header.
set qos-group	Sets a QoS group identifier (ID) that can be used later to classify packets.
table map support	Unconditional marking of one packet field based on another packet field.
priority	Gives priority to a class of traffic belonging to a policy map.
shape	Shapes traffic to the indicated bit rate according to the algorithm specified.
bandwidth	Provides a guaranteed minimum bandwidth to each of the eight queues.
dbl	Dynamic buffer limit.
queue-limit	Specifies the maximum number of packets a transmit queue can hold.

Platform Hardware Capabilities

Qos Actions	Numbers of entries supported
Classification	64k input and 64k output classification entries are supported. A given policy can use at most 24k ACLs
Policing	16K policers are supported. Policers are allocated to given direction in blocks of 2k. For example, 2k policers can be used in for input and 14k policers can be used for output. Single rate policers uses one policer entry. Single Rate Three Color Marker (srTCM) (RFC 2697) and Two Rate Three Color Marker (trTCM) (RFC 2698) uses two policer entries
Marking	Marking of Cos and DSCP/Precedence is supported through two marking tables, each capable of supporting 512 entries. There are separate tables for each direction.

Qos Actions	Numbers of entries supported
Queuing	The queue size is configuraable with the maximum number of entries configurable per port depending on the chassis and line card type.
DBL	You can enable DBL action on all configured class-maps.

Prerequisites for Applying a QoS Service Policy

Unlike the Switch QoS model, there is no prerequisite for enabling QoS on various targets. Just the attachmenta of a serive policy enables QoS and detachment of that policy disables QoS on that target.

Restrictions for Applying a QoS Service Policy

Traffic marking can be configured on an interface, a VLAN, or a port and VLAN. An interface can be a Layer 2 access port, a Layer 2 switch trunk, a Layer 3 routed port, or an EtherChannel. A policy is attached to a VLAN using the *vlan configuration* mode.

Attaching QoS service policy to VLANs and EtherChannel is described in the "“Policy Associations” section on page 35-91".

Classification

Supervisor Engine 6-E supports classification of Layer 2, IP, IPv6 packets, and ARP packetsPacket marking performed on input can be matched in the output direction. The previous table lists the full set of capabilities. By default, the Supervisor Engine 6-E also supports classification resources sharing.

By default, when the same policy is attached to a port or a VLAN or on per-port per-vlan targets, ACL entries are shared on the Supervisor Engine 6-E. Even though CAM entries are shared, QoS actions is unique on each target.

For example:

```
    class-map c1
      match ip any

Policy Map p1
    class ipp5
      police rate 1 m burst 200000
```

If policy-map p1 is applied to interfaces Gig 1/1 and Gig 1/2, 1 CAM entry is used (one ACE that matches IP packets), but 2 policers are allocated (one per target). So, all IP packets are policed to 1 mbps on interface Gig 1/1 and packets on interface Gig 1/2 are policed to 1 mbps.

Note With Cisco IOS Release 12.2(46)SG, you can issue the **match protocol arp** command. For details, see the *Catalyst 4500 Series Switch Cisco IOS Command Reference*.

Policing

Supervisor Engine 6-E supports policers in the following operation modes:

- Single Rate Policer Two Color Marker

This kind of policer is configured with just the committed rate (CIR) and normal burst and it has only conform and exceed actions.

This is the only form supported in the Supervisor Engine II-Plus to V-10GE based systems.

- Single Rate Three Color Marker (srTCM) (RFC 2697)
- Two Rate Three Color Marker (trTCM) (RFC 2698)
- Color Blind Mode

  Policing accuracy of 0.75% of configured policer rate.

  Supervisor Engine 6-E supports 16384 (16 x 1024, 16K) single rate, single burst policers. 16K policers are organized as 8 banks of 2K policers. The policer banks are dynamically assigned (input or output policer bank) by the software depending on the QoS configuration. So, the 16K policers are dynamically partitioned by software as follows:

  - 0 Input Policers and 16K Output Policers
  - 2K Input Policers and 14K Output Policers
  - 4K Input Policers and 12K Output Policers
  - 6K Input Policers and 10K Output Policers
  - 8K Input Policers and 8K Output Policers
  - 10K Input Policers and 6K Output Policers
  - 12K Input Policers and 4K Output Policers
  - 14K Input Policers and 2K Output Policers
  - 16K Input Policers and 0 Output Policers

These numbers represent individual policer entries in the hardware that support a single rate and burst parameter. Based on this, Supervisor Engines 6-E supports the following number of policers:

- 16K Single Rate Policer with Single Burst (Two Color Marker)
- 8K Single Rate Three Color Marker (srTCM)
- 8K Two Rate Three Color Marker (trTCM)

These policers are partitioned between Input and Output in chunks of 2K policer banks. The different types of policers can all co-exist in the system. However, a given type of policer (srTCM, trTCM etc.) is configurable as a block of 128 policers.

How to Implement Policing

For details on how to implement the policing features on a Catalyst 4500 series switch, refer to the Cisco IOS documentation at the following link:

http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_book09186a0080435d50.html

Platform Restrictions

Platform restrictions include the following:

- Multi-policer actions can be specified (setting CoS and IP DSCP is supported).
- Simultaneous unconditional and policer based marking is not supported.

- If policer based service-policy is attached to both a port and a VLAN, port-based policed is preferred by default. To over-ride a specific VLAN policy on a given port, then you must configure a per-port per-vlan policy.
- When you delete a port-channel with a per-port per-VLAN QoS policy, the switch crashes.

  Workaround: Before deleting the port-channel, do the following:

  1. Remove any per-port per-VLAN QoS policies, if any.
  2. Remove the VLAN configuration on the port-channel with the **no vlan-range** command.

Marking Network Traffic

Marking network traffic allows you to set or modify the attributes of 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.

Contents

Information About Marking Network Traffic

To mark network traffic, you should understand the following concepts:

Purpose of Marking Network Traffic

Traffic marking is 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 or marking fields.

Attributes that can be set and modified include the following:

- CoS value of a tagged Ethernet frame

- DSCP/Precedence value in the Type of Service (ToS) byte of IPv4.
- QoS group identifier (ID)
- DSCP /Precedence value in the traffic class byte of IPv6

Benefits of Marking Network Traffic

Traffic marking allows you to fine-tune the attributes for traffic on your network. This increased granularity helps isolate 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 strict priority can then be configured to put all packets of that marking into that queue. In this case, the marking was used to identify traffic for strict priority queue.
- 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 switch. The switch 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 64 different individual markings, similar to DSCP.
  - If changing the Precedence or DSCP value is undesirable.

Two Methods for Marking Traffic Attributes

Note This section describes *Unconditional* marking, which differs from *Policer-based* marking. Unconditional marking is based solely on classification.

Method One: Unconditional Explicit Marking (using the set command)

You specify the traffic attribute you want to change with a set command configured in a policy map. The following table lists the available set commands and the corresponding attribute. For details on the set command, refer to the *Catalyst 4500 Series Switch Command Reference*.

Table 35-8 set Commands and Applicable Packet Types

set Commands	Traffic Attribute	Packet Type
set cos	Layer 2 CoS value of the outgoing traffic	Ethernet IPv4, IPv6
set dscp	DSCP value in the ToS byte	IPv4, IPv6
set precendence	precendence value in the packet header	IPv4, IPv6
set qos-group	QoS group ID	Ethernet, IPv4, IPv6

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 Table 35-8.

In this sample configuration, the **set cos** command has been configured in the policy map (policy1) to mark the CoS attribute:

```
enable
configure terminal
policy map p1
    class class1
      set cos 3
end
```

For information on configuring a policy map, see the "Creating a Policy Map" section on page 35-37.

The final task is to attach the policy map to the interface. For information on attaching the policy map to the interface, see the "Attaching a Policy Map to an Interface" section on page 35-41.

Method Two: Unconditional Tablemap-based Marking

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.

The following is a sample table map configuration:

```
table-map table-map1
map from 0 to 1
map from 2 to 3
exit
```

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

Table 35-9 Traffic Attributes for Which a To-From Relationship Can Be Established

The "To" Attribute	The "From" Attribute
Precedence	CoS, QoS group, DSCP, Precedence
DSCP	COS, QoS group, DSCP, Precendence
CoS	DSCP, QoS group, CoS, Precedence

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

```
Policy map policy
    class class-default
      set cos dscp table table-map

exit
```

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

For information on configuring a policy map to use a table map, "Configuring a Policy Map" section on page 35-36.

The final task is to attach the policy map to the interface. For information on attaching the policy map to the interface, see the "Attaching a Policy Map to an Interface" section on page 35-41.

Marking Action Drivers

A marking action can be triggered based on one of the two QoS processing steps.

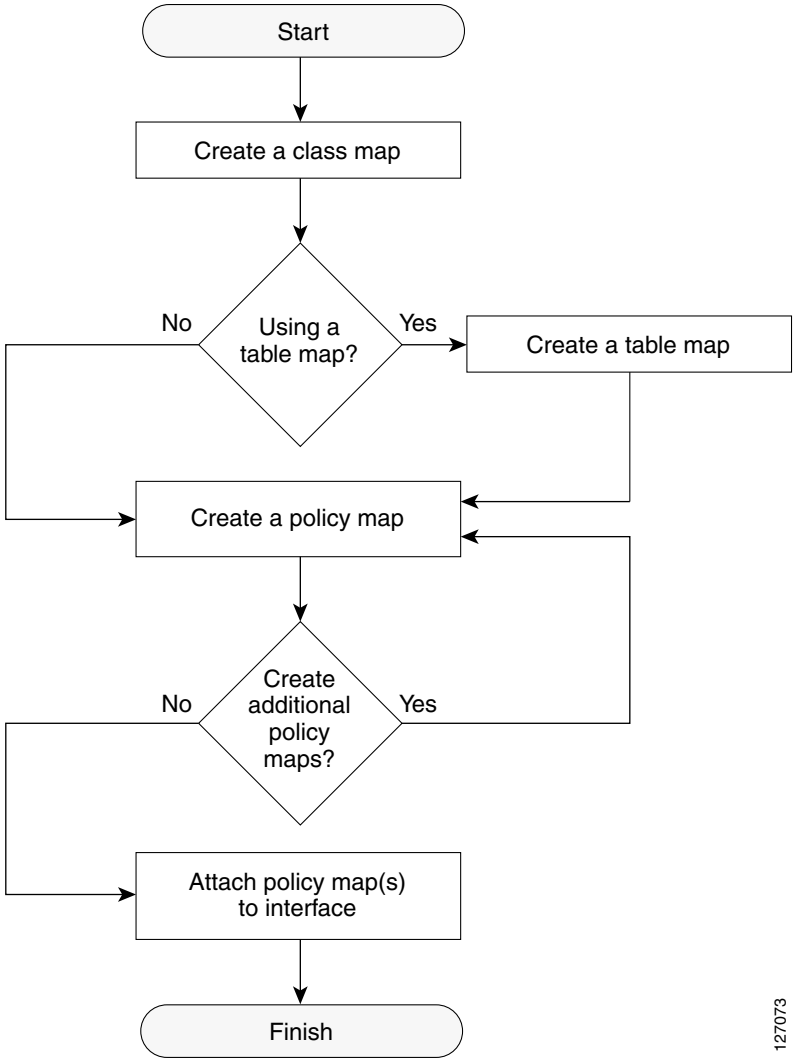
Classification based: In this case, all the traffic matching a class is marked using either explicit or tablemap based method. This method is referred to as *unconditional* marking.

Policer result-based: In this case, a class of traffic is marked differently based on the policer result (conform/exceed/violate) applicable to that packet. This method is referred to as *conditional* marking.

Traffic Marking Procedure Flowchart

Figure 35-8 illustrates the order of the procedures for configuring traffic marking.

Figure 35-8 Traffic marking Procedure Flowchart



Restrictions for Marking Network Traffic

The following restrictions apply to packet marking actions:

- QoS-group can be marked only in the input direction and can only support unconditional explicit marking.
- Only explicit marking is supported for policer-based marking.

Multi-attribute Marking Support

Supervisor Engine 6-E can mark more than one QoS attribute of a packet matching a class of traffic. For example, DSCP, CoS, and QoS-group can all be set together, using either explicit or tablemap-based marking.

Note When using unconditional explicit marking of multiple fields or policer-based multifield, multi-region (conform/exceed/violate) marking the number of tablemaps that can be setup in TOS or COS marking tables will be less than the maximum supported.

Hardware Capabilities for Marking

Supervisor Engine 6-E provides a 128 entry marking action table where each entry specifies the type of marking actions on COS and DSCP/precdence fields as well as policer action to transmit/markdown/drop a packet. One such table is supported for each direction, input and output. This table is used for both unconditional marking as well as policer-based marking. It can be used to support 128 unique marking actions or 32 unique policer-based actions or a combinations of the two.

For each of the marking fields (COS and DSCP), the Supervisor Engine 6-E provides 512 entry marking tables for each direction. These are similar to mapping tables available on supervisor engines that support the switch QoS model. However, these provide an ability to have multiple unique mapping tables that are setup by the user.

For example, the TOS marking table provides marking of DSCP/Precedence fields and can be used as one of the following:

- 8 different tablemaps with each mapping the 64 DSCP or qos-group values to another DSCP
- 64 (32) different tablemaps with each one mapping 8 CoS (16 CoS and CFi) values to DSCP in input (output) direction
- a combination of above two types of tablemaps

Similar mappings are available on the 512 entry COS marking table.

Configuring the Policy Map Marking Action

This section describes how to establish unconditional marking action for network traffic.

Prerequisites

Perform the following:

- Create a class map (*ipp5)* and a policy map. (Refer to the"Configuring a QoS Policy" section on page 35-33)
- Configure the marking action. (Refer to the "Configuring Policy-Map Class Actions" section on page 35-37)

Note On the Supervisor Engine 6-E, the marking action command options have been extended (refer to Table 35-8 on page 35-75 andTable 35-9 on page 35-76).

Configuring Tablemap-based Unconditional Marking

To configure table-map based unconditional marking, perform this task:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# table-map name`	Configure a tablemap.

	Command	Purpose
Step 3	`Switch(config-tablemap)# map from from_value to to_value`	Create a map from a *from_value* to a *to_value*
Step 4	`Switch(config-tablemap)# exit`	Exit table-map configuration mode.
Step 5	`Switch(config)# policy-map name`	Enter policy-map configuration mode.
Step 6	`Switch(config-p)# class name`	Selects the class for QoS actions.
Step 7	`Switch(config-p-c)# set cos | dscp | prec cos | dscp | prec | qos-group [table name]`	Selects the marking action based on an implicit or explicit table-map.
Step 8	`Switch(config-p-c)# end`	Exits configuration mode.
Step 9	`Switch# show policy-map name`	Verifies the configuration of the policy-map.
Step 10	`Switch# show table-map name`	Verifies the configuration of the table-map.

The following example shows how to enable marking action using table-map.

```
Switch(config)# table-map dscp2Cos
Switch(config-tablemap)# map from 8 to 1
Switch(config-tablemap)# exit
Switch(config)# policy-map p1
Switch(config-pmap)# class ipp5
Switch(config-pmap-c)# set cos dscp table dscp2Cos
Switch(config-pmap-c)# end
Switch# show policy-map p1

Policy Map p1
    Class ipp5
      set cos dscp table dscp2Qos

Switch# show table-map dscp2Cos

 Table Map dscp2Cos
    from 8 to 1
    default copy
```

Configuring Policer Result-based Conditional Marking

To configure policer result-based conditional marking, setup a single rate or dual rate policer. Refer to the "How to Implement Policing" section on page 35-73.

This example shows how to configure a two rate three-color policer with explicit actions for each policer region:

```
Switch# configure terminal
Switch(config-pmap-c)# policer cir percent 20 pir percent 30
Switch(config-pmap-c-policer)# conform-action set-cos-transmit 3 set-dscp-transmit 10
Switch(config-pmap-c-policer)# exceed-action set-cos-transmit 4 set-dscp-transmit 20
Switch(config-pmap-c-policer)# violate action drop
Switch# show policy-map p1

  Policy Map police
    Class ipp5
     police cir percent 20 pir percent 30
       conform-action set-cos-transmit 3
       conform-action set-dscp-transmit af11
       exceed-action set-cos-transmit 4
       exceed-action set-dscp-transmit af22
       violate-action drop
```

Marking Statistics

The marking statistics indicate the number of packets that are *marked*.

For unconditional marking, the *classification entry* points to an entry in the marking action table that in turn indicates the fields in the packet that are marked. Therefore, the classification statistics by itself indicates the unconditional marking statistics.

For a conditional marking using policer, provided the policer is a packet rate policer, you cannot determine the number packets marked because the policer only provides byte statistics for different policing results.

Shaping, Sharing (Bandwidth), Priority Queuing, Queue-limiting and DBL

Supervisor Engine 6-E supports the Classification-based (class-based) mode for transmit queue selection. In this mode, the transmit queue selection is based on the Output QoS classification lookup.

Note Only output (egress) queuing is supported.

The Supervisor Engine 6-E hardware supports 8 transmit queues per port. Once the forwarding decision has been made to forward a packet out a port, the output QoS classification determines the transmit queue into which the packet needs to be enqueued.

By default, in Supervisor Engine 6-E, without any service policies associated with a port, there are two queues (a control packet queue and a default queue) with no guarantee as to the bandwidth or kind of prioritization. The only exception is that system generated control packets are enqueued into control packet queue so that control traffic receives some minimum link bandwidth.

Queues are assigned when an output policy attached to a port with one or more queuing related actions for one or more classes of traffic. Because there are only eight queues per port, there can be at most eight classes of traffic (including the reserved class, class-default) with queuing action(s). Classes of traffic that do not have any queuing action are referred to as *non-queuing* classes. Non-queuing class traffic ends up using the queue corresponding to class class-default.

When a queuing policy (a policy with queuing action) is attached, the control packet queue is deleted and the control packets are enqueued into respective queue per their classification. Note that this differs from the way control-traffic was prioritized in the Catalyst 4924, Catalyst 4948, Catalyst 4948-10GE, and the Supervisor Engines II+, II+10GE, IV, V, and V-10GE. On these platforms, by default, control traffic was guaranteed 25 per cent of the link bandwidth whether or not QoS was configured. If this same behavior is required on Supervisor Engine 6-E, an egress QoS class must be configured to match IP Precedence 6 and 7 traffic, and a bandwidth guarantee must be configured.

Dynamic resizing of queues (queue limit class-map action) is supported through the use of the **queue-limit** command. Based on the chassis and line card type, all eight queues on a port are configured with equal queue size.

Shaping

Shaping enables you to delay out-of-profile packets in queues so that they conform to a specified profile. Shaping is distinct from policing. Policing drops packets that exceed a configured threshold, whereas shaping *buffers* packets so that traffic remains within a given threshold. Shaping offers greater *smoothness* in handling traffic than policing. You enable average-rate traffic shaping on a traffic class with the **policy-map** class configuration command.

Supervisor Engine 6-E supports a range of 32kbps to 10 gbps for shaping, with a precision of approximately with a precision of approximately +/- 0.75 per cent.

When a queuing class is configured without any explicit shape configuration, the queue shape is set to the link rate.

To configure class-level shaping in a service policy, follow these steps:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# policy-map policy-map-name`	Create a policy map by entering the policy-map name, and enter policy-map configuration mode. By default, no policy maps are defined.
Step 3	`Switch(config-pmap)# class class-name`	Specify the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.
Step 4	`Switch(config-pmap-class)# shape average {cir-bps kbps | percent percent}`	Enable average-rate traffic shaping. You can specify the bandwidth in kbps or as a percentage: • For *cir-bps*, specify the committed information rate, the bit rate that traffic is shaped to, in bps. The range is 32000 to 10000000000 bps. • For *percent*, specify the percentage of link rate to shape the class of traffic. The range is 1 to 100. By default, average-rate traffic shaping is disabled.
Step 5	`Switch(config-pmap-class)# exit`	Return to policy-map configuration mode.
Step 6	`Switch(config-pmap)# exit`	Return to global configuration mode.
Step 7	`Switch(config)# interface interface-id`	Specify a physical port and enter interface configuration mode.
Step 8	`Switch(config-interface)# service-policy output policy-map-name`	Specify the policy-map name, and apply it a physical interface.
Step 9	`Switch(config-interface)# end`	Return to privileged EXEC mode.

Step 10	`Switch# show policy-map [policy-map-name [class class-map-name]]`
	or
	`Switch# show policy-map interface interface-id`
		Verifies your entries.

Step 11	`Switch# copy running-config startup-config`	(Optional) Save your entries in the configuration file.

To delete an existing policy map, use the **no policy-map policy-map-name** global configuration command. To delete an existing class, use the **no class class-name policy-map** configuration command. To disable the average-rate traffic shaping, use the **no shape average policy-map** class configuration command.

This example shows how to configure class-level, average-rate shaping. It limits traffic class class1 to a data transmission rate of 256 kbps:

```
Switch# configure terminal
Switch(config)# policy-map policy1
Switch(config-pmap)# class class1
Switch(config-pmap-c)# shape average 256000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy output policy1
Switch(config-if)# end
Switch#

Switch# show policy-map policy1
  Policy Map policy1
    Class class1
       shape average 256000
```

This example shows how to configure class-level, average shape percentage to 32% of link bandwidth for queuing-class traffic:

```
Switch# configure terminal
Switch(config)# policy-map queuing-policy
Switch(config-pmap)# class queuing-class
Switch(config-pmap-c)# shape average percent 32
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy output policy1
Switch(config-if)# end
Switch #

Switch# show policy-map queuing-policy
  Policy Map queuing-policy
    Class queuing-class
       Average Rate Traffic Shaping
       cir 32%
```

Sharing(bandwidth)

The bandwidth assigned to a class of traffic is the minimum bandwidth that is guaranteed to the class during congestion. Transmit Queue Sharing is the processs by which output link bandwidth is shared among multiple queues of a given port.

Supervisor Engine 6-E supports a range of 32 kbps to 10 gbps for sharing, with a precision of approximately +/- 0.75 per cent. The sum of configured bandwidth across all queuing classes should not exceed the link bandwidth.

To configure class-level bandwidth action in a service policy, follow these steps:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# policy-map` *policy-map-name*	Create a policy map by entering the policy-map name, and enter policy-map configuration mode. By default, no policy maps are defined.

	Command	Purpose
Step 3	`Switch(config-pmap)# class class-name`	Specify the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.
Step 4	`Switch(config-pmap-class)# bandwidth {bandwidth-kbps | percent percent}`	Specify the minimum bandwidth provided to a class belonging to the policy map when there is traffic congestion in the switch. If the switch is not congested, the class receives more bandwidth than you specify with the **bandwidth** command. By default, no bandwidth is specified. You can specify the bandwidth in kbps or as a percentage: o For *bandwidth-kbps*, specify the bandwidth amount in kbps assigned to the class. The range is 32 to 10000000. o For *percent*, specify the percentage of available bandwidth assigned to the class. The range is 1 to 100. Specify all the class bandwidths in either kbps or in percentages, but not a mix of both.
Step 5	`Switch(config-pmap-class)# exit`	Return to policy-map configuration mode.
Step 6	`Switch(config-pmap)# exit`	Return to global configuration mode.
Step 7	`Switch(config)# interface interface-id`	Specify a physical port and enter interface configuration mode.
Step 8	`Switch(config-interface)# service-policy output policy-map-name`	Specify the policy-map name, and apply it a physical interface.
Step 9	`Switch(config-interface)# end`	Return to privileged EXEC mode.

Step 10	`Switch# show policy-map [policy-map-name [class class-map-name]]`
	or
	`Switch# show policy-map interface interface-id`
		Verifies your entries.

Step 11	`Switch# copy running-config startup-config`	(Optional) Save your entries in the configuration file.

To delete an existing policy map, use the **no policy-map policy-map-name** global configuration command. To delete an existing class, use the **no class class-name policy-map** configuration command. To return to the default bandwidth, use the **no bandwidth policy-map** class configuration command.

This example shows how to create a class-level policy map called policy11 for three classes called prec1, prec2, and prec3. In the policy for these classes, 30 percent of the available bandwidth is assigned to the queue for the first class, 20 percent is assigned to the queue for the second class, and 10 percent is assigned to the queue for the third class.

```
Switch # configure terminal
Switch(config)# policy-map policy11
Switch(config-pmap)# class prec1
Switch(config-pmap-c)# bandwidth percent 30
Switch(config-pmap-c)# exit
Switch(config-pmap)# class prec2
Switch(config-pmap-c)# bandwidth percent 20
```

```
Switch(config-pmap-c)# exit
Switch(config-pmap)# class prec3
Switch(config-pmap-c)# bandwidth percent 10
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy output policy11
Switch(config-if)# end
Switch #

Switch# show policy-map policy11
  Policy Map policy11
    Class prec1
      bandwidth percent 30
    Class prec2
      bandwidth percent 20
    Class prec3
      bandwidth percent 10
```

This example shows how to create a class-level policy map called policy11 for three classes called prec1, prec2, and prec3. In the policy for these classes, 300 mbps of the available bandwidth is assigned to the queue for the first class, 200 mbps is assigned to the queue for the second class, and 100 mbps is assigned to the queue for the third class.

```
Switch # configure terminal
Switch(config)# policy-map policy11
Switch(config-pmap)# class prec1
Switch(config-pmap-c)# bandwidth 300000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class prec2
Switch(config-pmap-c)# bandwidth 200000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class prec3
Switch(config-pmap-c)# bandwidth 100000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy output policy11
Switch(config-if)# end
Switch #

Switch# show policy-map policy11
  Policy Map policy11
    Class prec1
      bandwidth 300000 (kbps)
    Class prec2
      bandwidth 200000 (kbps)
    Class prec3
      bandwidth 100000 (kbps)
```

When a queuing class is configured without any explicit share/bandwidth configuration, because the queue is not guaranteed any minimum bandwidth, the hardware queue is programmed to get a share of any unallocated bandwidth on the port as shown in the following example.

If there is no bandwidth remaining for the new queue or if the unallocated bandwidth is not sufficient to meet the minimum configurable rate (32kbps) for all queues which do not have any explicit share/bandwidth configuration, then the policy association is rejected.

For example, if there are two queues as given below

```
policy-map queue-policy
   class q1
     bandwidth percent 10

   class q2
     bandwidth percent 20
```

then the bandwidth allocation for the queues is as follows

```
q1 = 10%
           q2 = 20%
class-default = 70%
```

Similarly, when another queuing class (say q3) is added without any explicit bandwidth (say, just a shape command), then the bandwidth allocation is

```
q1 = 10%
           q2 = 20%
           q3 = min(35%, q3-shape-rate)
class-default = max(35%, (100 - (q1 + q2 + q3 )))
```

Priority queuing

On Supervisor Engine 6-E only one transmit queue on a port can be configured as *strict priority* (termed Low Latency Queue, or LLQ).

LLQ provides strict-priority queuing for a traffic class. It enables delay-sensitive data, such as voice, to be sent *before* packets in other queues. The priority queue is serviced first until it is empty or until it is under its shape rate. Only one traffic stream can be destined for the priority queue per class-level policy. You enable the priority queue for a traffic class with the **priority policy-map class** configuration command at the class mode.

A LLQ can starve other queues unless it is rate limited. Supervisor Engine 6-E does not support *conditional policing* where a 2-parameter policer (rate, burst) becomes effective when the queue is *congested* (based on queue length). However, it supports application of an unconditional policer to rate limit packets enqueued to the strict priority queue.

When a priority queue is configured on one class of a policy map, only *bandwidth remaining* is accepted on other classes, guaranteeing a minimum bandwidth for other classes from the remaining bandwidth of what is left after using the priority queue. When a priority queue is configured with a policer, then either *bandwidth* or *bandwidth remaining* is accepted on other classes.

To enable class-level priority queuing in a service policy, follow these steps:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# policy-map` *policy-map-name*	Create a policy map by entering the policy-map name, and enter policy-map configuration mode. By default, no policy maps are defined.
Step 3	`Switch(config-pmap)# class` *class-name*	Specify the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.

	Command	Purpose
Step 4	`Switch(config-pmap-class)# priority`	Enable the strict-priority queue, and give priority to a class of traffic. By default, strict-priority queueing is disabled.
Step 5	`Switch(config-pmap-class)# exit`	Return to policy-map configuration mode.
Step 6	`Switch(config-pmap)# exit`	Return to global configuration mode.
Step 7	`Switch(config)# interface` *interface-id*	Specify a physical port and enter interface configuration mode.
Step 8	`Switch(config-interface)# service-policy output` *policy-map-name*	Specify the policy-map name, and apply it a physical interface.
Step 9	`Switch(config-interface)# end`	Return to privileged EXEC mode.
Step 10	`Switch# show policy-map [`*policy-map-name* `[class` *class-map-name*`]]` or `Switch# show policy-map interface` *interface-id*	Verifies your entries.
Step 11	`Switch# copy running-config startup-config`	(Optional) Save your entries in the configuration file.

To delete an existing policy map, use the **no policy-map policy-map-name** global configuration command. To delete an existing class, use the **no class class-name policy-map** configuration command. To disable the priority queue, use the **no priority policy-map class** configuration command.

This example shows how to configure a class-level policy called policy1. Class 1 is configured as the priority queue, which is serviced first until it is empty.

```
Switch# configure terminal
Switch(config)# policy-map policy1
Switch(config-pmap)# class class1
Switch(config-pmap-c)# priority
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy output policy1
Switch(config-if)# end
Switch #

Switch# show policy-map policy1
  Policy Map policy1
    Class class1
      priority
```

Queue-limiting

When a class-based queue is instantiated on a physical port, it is set up with a default size. This size represents the number of queue entries in which packets belonging to that class of traffic can be queued. The scheduler moves packets from the queue that are ready for transmission, based on the queue shape, bandwidth, and priority configuration.

The queue-limit provides the maximum number of packets that can be in the queue at any given time. When the queue is full, an attempt to enqueue any further packets results in tail drop. However, if dynamic buffer limiting (DBL) is enabled on the queue, packets get a probabilistic drop based on the DBL algorithm, even when the queue is not full.

The **queue-limit** command can be configured under a class only when queue scheduling, such as bandwidth, shape, or priority is already configured. The only exception to this requirement is the support of the stand-alone **queue-limit** command on the class-default class.

Queue Memory

The number of queue entries that can be allocated has to be a multiple of 8 and can range from 16 to 8184. When a class-based queue is instantiated on a physical port, it is given a default number of entries. This default queue size is based on the number of slots in the chassis and the number of front-panel ports in each slot.

The Supervisor Engine 6-E has 512K (524,288) queue entries of which the system sets aside 100K (102,400) queue entries in a free reserve pool. Of the remaining 412K (421,88), the drop port is provided 8184 entries and the CPU ports are assigned 11704 entries. The remaining entries are divided equally among the slots in the chassis. In a redundant chassis the two supervisor slots are treated as one for the purpose of this entries distribution. Within each slot the number of queue entries are equally divided among the front-panel ports present on the line card in that slot.

When the user configuration for queue entries on an interface exceeds its dedicated quota, the system attempts to satisfy the configuration from the free reserve pool. The entries from the free reserve pool are allocated to interfaces on a first-come first-served basis.

Service Policy Association

When a QoS service-policy with queuing actions is configured, but no explicit queue-limit command is attached in the egress direction on a physical interface, each of the class-based queues gets the same number of queue entries from within the dedicated quota for that physical port. When a queue is explicitly given a size using the queue-limit command, the switch tries to allocate all the entries from within the dedicated quota for the interface. If the required number of entries is greater than the dedicated quota for the interface, the switch tries to allocate the entries from the free reserve.

The queue entries associated with a queue always have to be consecutive. This requirement can result in fragmentation of the 512K of the queue entries that are shared across the switch. For example, an interface may not have enough entries for a queue in its dedicated quota and thus have to use the free reserve to set up that queue. In this case, the queue entries from the dedicated quota remain unused because they cannot be shared with any other port or slot.

When the QoS service-policy associated with an interface is removed, any queue entries taken from the free reserve are returned to the free reserve pool. The interface queuing configuration reverts to two queues — class-default and the control-packet queue with default shape, bandwidth, and size. The control-packet queue is set up with size 16 whereas the default queue is set up with the maximum size possible based on the dedicated quota for that interface.

Queue Allocation Failure

The switch might not be able to satisfy the explicit queue size required on one or more queues on an interface because of fragmentation of queue memory or lack of enough free reserve entries. In this scenario, the switch logs an error message to notify you of the failure. The QoS service-policy is left configured on the interface. You can fix the error by removing the QoS service-policy and examining the current usage of the queue entries from the free reserve by other ports on the switch.

To configure class-level queue-limit in a service policy, follow these steps:

To remove the explicit queue size use the **no queue-limit** command under the class in a policy-map.

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# policy-map policy-map-name`	Create a policy map by entering the policy-map name, and enter policy-map configuration mode. By default, no policy maps are defined.
Step 3	`Switch(config-pmap)# class class-name`	Specify the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.
Step 4	`Switch(config-pmap-class)# shape average {cir-bps kbps | percent percent}`	Enable average-rate traffic shaping. You can specify the bandwidth in kbps or as a percentage: • For *cir-bps*, specify the committed information rate, the bit rate that traffic is shaped to, in bps. The range is 32000 to 10000000000 bps. • For *percent*, specify the percentage of link rate to shape the class of traffic. The range is 1 to 100. By default, average-rate traffic shaping is disabled.
Step 5	`Switch(config-pmap-class)# queue-limit number-of-packets`	Provide an explicit queue size in packets. The size must be a multiple of 8 and ranging from 16 to 8184.
Step 6	`Switch(config-pmap-class)# exit`	Return to policy-map configuration mode.
Step 7	`Switch(config-pmap)# exit`	Return to global configuration mode.
Step 8	`Switch(config)# interface interface-id`	Specify a physical port and enter interface configuration mode.
Step 9	`Switch(config-interface)# service-policy output policy-map-name`	Specify the policy-map name, and apply it a physical interface.
Step 10	`Switch(config-interface)# end`	Return to privileged EXEC mode.

Step 11	`Switch# show policy-map [policy-map-name [class class-map-name]]`
	
	`or`
	
	`Switch# show policy-map interface interface-id`
		Verifies your entries.

Step 12	`Switch# copy running-config startup-config`	(Optional) Save your entries in the configuration file.

This example shows how to configure a class-based queue with an explicit **queue-limit** command. It limits traffic class class1 to a queue of size 4048:

```
Switch# configure terminal
Switch(config)# policy-map policy1
Switch(config-pmap)# class class1
Switch(config-pmap-c)# shape average 256000
Switch(config-pmap-c)# queue-limit 4048
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy output policy1
Switch(config-if)# end
Switch#

Switch# show policy-map policy1
  Policy Map policy1
    Class class1
       shape average 256000
       queue-limit 4048
Switch#
```

Active Queue Management (AQM) via Dynamic Buffer Limiting (DBL)

AQM provides buffering control of traffic flows prior to queuing a packet into a transmit queue of a port. This is of significant interest in a shared memory switch, ensuring that certain flows do not hog the switch packet memory.

Note Supervisor Engine 6-E supports active switch buffer management via DBL.

Except for the default class of traffic (class class-default), you can configure DBL action only when at least one of the other queuing action is configured.

To configure class-level dbl action along with shaping in a service policy, follow these steps:

	Command	Purpose
Step 1	`Switch# configure terminal`	Enters global configuration mode.
Step 2	`Switch(config)# policy-map` *policy-map-name*	Create a policy map by entering the policy-map name, and enter policy-map configuration mode. By default, no policy maps are defined.
Step 3	`Switch(config-pmap)# class` *class-name*	Specify the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.
Step 4	`Switch(config-pmap-class)# shape average` *cir-bps*	Enable average-rate traffic shaping. Specify the committed information rate, the bit rate that traffic is shaped to, in bps. The range is 32000 to 10000000000 bps. By default, average-rate traffic shaping is disabled.
Step 5	`Switch(config-pmap-class)# dbl`	Enable DBL on the queue associated with this class of traffic
Step 6	`Switch(config-pmap-class)# exit`	Return to policy-map configuration mode.
Step 7	`Switch(config-pmap)# exit`	Return to global configuration mode.

	Command	Purpose
Step 8	`Switch(config)# interface interface-id`	Specify a physical port and enter interface configuration mode.
Step 9	`Switch(config-interface)# service-policy output policy-map-name`	Specify the policy-map name, and apply it a physical interface.
Step 10	`Switch(config-interface)# end`	Return to privileged EXEC mode.

Step 11	`Switch# show policy-map [policy-map-name [class class-map-name]]`
	
	or
	
	`Switch# show policy-map interface interface-id`
		Verifies your entries.

Step 12	`Switch# copy running-config startup-config`	(Optional) Save your entries in the configuration file.

To delete an existing policy map, use the **no policy-map policy-map-name** global configuration command. To delete an existing class, use the **no class class-name policy-map** configuration command. To disable DBL on the associated queue, use the **no dbl policy-map class** configuration command.

The following example shows how to configure class-level, DBL action along with average-rate shaping. It enables DBL on the queue associated with traffic-class *class1*.

```
Switch# configure terminal
Switch(config)# policy-map policy1
Switch(config-pmap)# class class1
Switch(config-pmap-c)# shape average 256000
Switch(config-pmap-c)# dbl
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy output policy1
Switch(config-if)# end
Switch#

Switch# show policy-map policy1
  Policy Map policy1
    Class class1
       shape average 256000
     dbl
```

Transmit Queue Statistics

Transmit queue statistics are visible via the **show policy-map interface** command.

Policy Associations

Supervisor Engine 6-E supports per-port, per-VLAN policies. The associated policies are attached to the interface, VLAN, and a specific VLAN on a given port, respectively.

A policy can be associated with a variety of objects. The following table lists the objects and the actions allowed.

Table 35-1 Table QoS Policy Associations

Object	Action
Physical port	Policing, marking, and queuing
VLAN	Policing and marking
Port and VLAN (PV)	Policing and marking
EtherChannel	Policing and marking
EtherChannel member port	Queuing

For details, refer to the following link:

http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_book09186a0080435d50.html

Qos Action Restrictions

- The same actions cannot be performed multiple times in a given direction on different targets. In other words, it is not possible to police the packets both on port and VLAN in the input direction. However, the user can police on the input port and on the output VLAN.
- Queuing actions are only allowed in the egress direction and only on the physical port.
- Percentage-based actions like policer cannot be configured on a VLAN.
- Port channel or VLAN configuration can only have a policing or a marking action, not a queueing action.

Qos Policy priorities

- If a policy on a port and a VLAN are configured with conflicting actions (such as policing or marking actions on both a port and VLAN), the port policy is picked.
- If policy on a VLAN on a given port must be over-written, the user can configure PV policy.

Qos Policy merging

Applicable policies are applied to a given packet in given direction. For example, if the user configures egress VLAN-based police and marking, followed by selective queuing on the port, then for this packet, actions from both policies will be applied.

The following policy-map configuration restrictions are imposed on an EtherChannel:

- only policing and marking actions are supported at the EtherChannel level
- only queuing actions are supported at the physical member port level

A packet can be marked (dscp or cos fields) by the EtherChannel policy. If the physical member port policy uses a classification based on dscp or cos fields, it must be based on the marked (modified) value. To ensure proper operation, the following restriction is placed on the EtherChannel.

The classification critieria for the policy-map on the physical member ports has to based only on one type of field:

- dscp
- precedence

- cos
- any non marking field (no dscp or cos based classification)

Classification criteria for the policy-map on the physical member ports cannot be based on a combination of fields. This restriction ensures that if the EtherChannel policy is marking down dscp or cos, the marked (modified) value-based classification can be implemented in hardware.

Auto-QoS is not supported on EtherChannel or its member ports. A physical port configured with Auto-QoS is not allowed to become a member of a physical port.

Software QoS

At the highest level, there are two types of locally sourced traffic (such as control protocol packets, pings, and telnets) from the switch: high priority traffic (typically the control protocol packets like OSPF Hellos and STP) and low priority packets (all other packet types).

The QoS treatment for locally-sourced packets differs for the two types.

Supervisor Engine 6-E provides a way to apply QoS to packets processed in the software path. The packets that get this QoS treatment in software can be classified into two types: software switched packets and software generated packets.

On reception, software switched packets are sent to the CPU that in turn sends them out of another interface. For such packets, input software QoS provides input marking and output software QoS provides output marking and queue selection.

The software generated packets are the ones locally sourced by the switch. The type of output software QoS processing applied to these packets is the same as the one applied to software switched packets. The only difference in the two is that the software switched packets take input marking of the packet into account for output classification purpose.

High Priority Packets

High priority packets are marked as one of the following:

- internally with PAK_PRIORITY
- with IP Precedence of 6 (for IP packets)
- with CoS of 6 (for VLAN Tagged packets)

These packets behave as follows:

- They are not dropped because of any policing, AQM, drop thresholds (or any feature that can drop a packet) configured as per the egress service policy. However, they might be dropped because of hardware resource constraints (packet buffers, queue full, etc.).
- They are classified and marked as per the marking configuration of the egress service policy that could be a port or VLAN (refer to the "Policy Associations" section on page 35-91.
- These high priority packets are enqueued to queue on the egress port based on the following criteria:
  - If there is no egress queuing policy on the port, the packet is queued to a control packet queue that is setup separately from the default queue and has 5 percent of the link bandwidth reserved for it.
  - If there is an egress queuing policy on the port, the queue is selected based on the classification criteria applicable to the packet.

Low Priority Packets

Packets that are not considered high priority (as described previously) are considered *unimportant*. These include locally sourced pings, telnet, and other protocol packets. They undergo the same treatment as any other packet that is transiting the given transmit port including egress classification, marking and queuing.