CHAPTER 28

Configuring Quality of Service

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 also describes how to specify different QoS configurations on different VLANs on a given interface (per-port per-VLAN QoS).

This chapter consists of these sections:

- Overview of QoS, page 28-1
- Configuring Auto-QoS, page 28-17
- Configuring QoS, page 28-23

Note For complete syntax and usage information for the switch commands used in this chapter, see 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 in the *Cisco Catalyst 4500 Command Reference*, you can locate it 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

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, page 28-2
- QoS Terminology, page 28-3
- Basic QoS Model, page 28-5

- Classification, page 28-6
- Policing and Marking, page 28-10
- Mapping Tables, page 28-14
- Queueing and Scheduling, page 28-14
- Packet Modification, page 28-16
- Per Port Per VLAN QoS, page 28-16
- QoS and Software Processed Packets, page 28-16

Prioritization

QoS implementation is based on the DiffServ architecture, an emerging standard from the Internet Engineering Task Force (IETF). 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 28-1:

- 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.

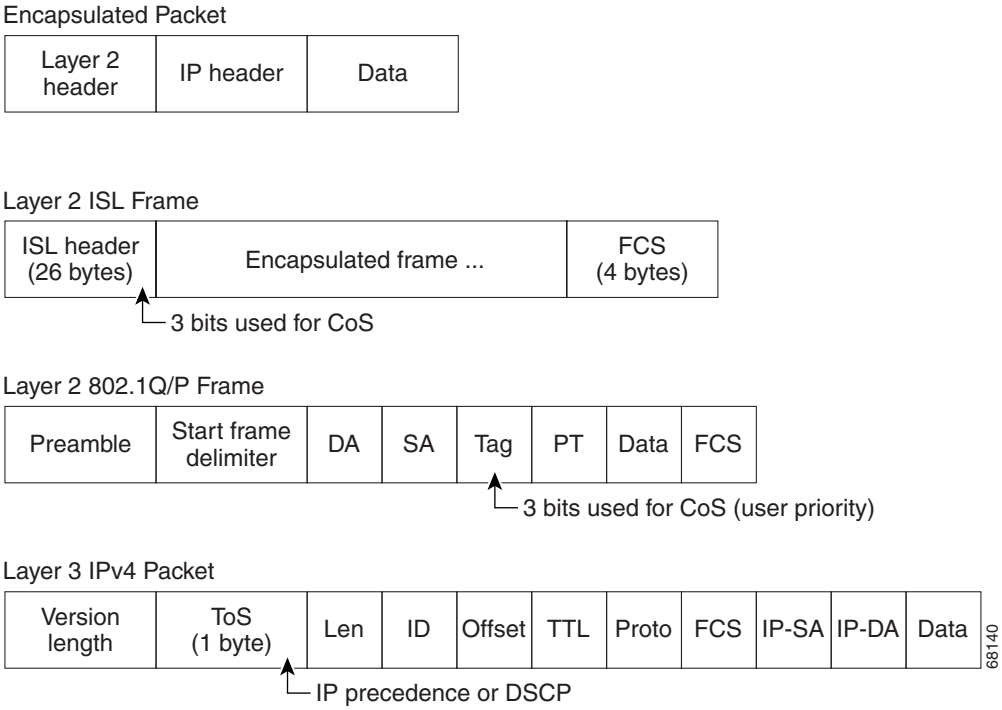
- Prioritization bits in Layer 3 packets:

  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 28-1 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 28-52.

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 28-1.

Table 28-1 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 28-1 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 28-13.
- *Policing* is limiting bandwidth used by a flow of traffic. Policing can mark or drop traffic.

Basic QoS Model

Figure 28-2 shows the basic QoS model. Actions at the ingress and egress interfaces include classifying traffic, policing, and marking:

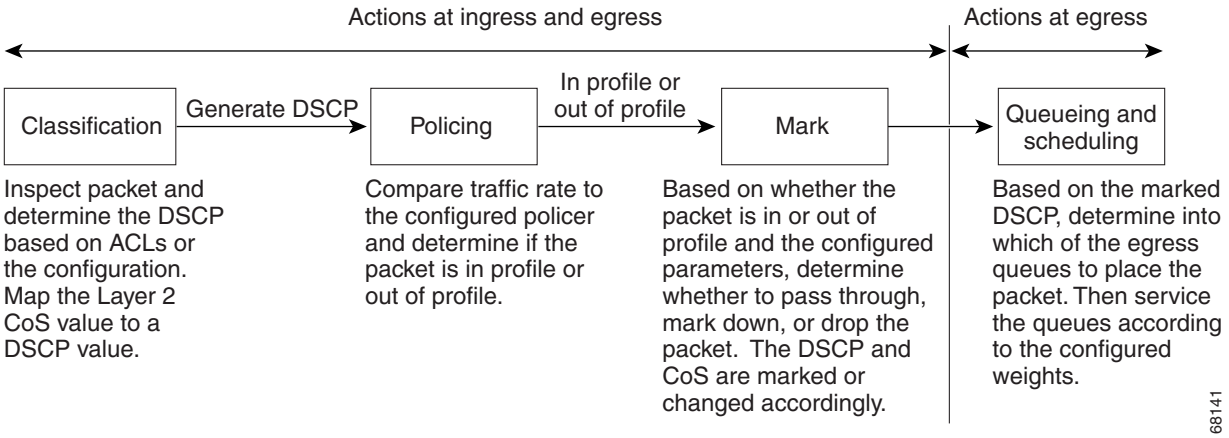
- Classifying distinguishes one kind of traffic from another. The process generates an internal DSCP for a packet, which identifies all the future QoS actions to be performed on this packet. For more information, see the "Classification" section on page 28-6.
- Policing determines whether a packet is 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 28-10.
- 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 28-10.

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 28-14.

Figure 28-2 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 28-3.

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:

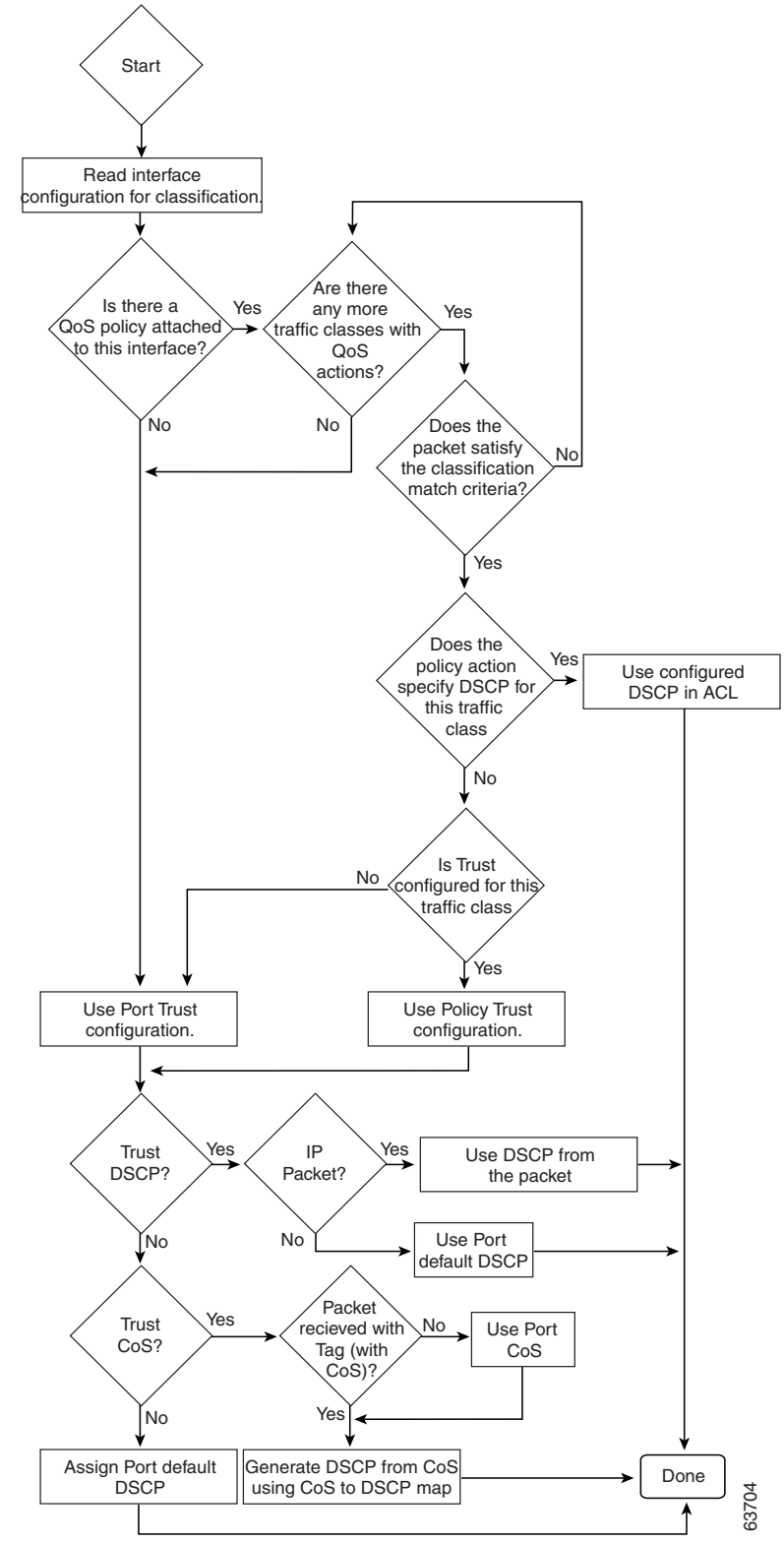
- Trust the IP DSCP in the incoming packet (configure the port to trust DSCP), and assign the same DSCP to the packet for internal use. The IETF defines the six most-significant bits of the 1-byte Type of Service (ToS) field as the DSCP. The priority represented by a particular DSCP value is configurable. 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 28-14. For configuration information on port trust states, see the "Configuring the Trust State of Interfaces" section on page 28-46.

Figure 28-3 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 28-9.

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 28-29.

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 or IP precedence 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 28-10.

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 28-29.

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 28-4.

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 28-14.

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 36.

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.

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.

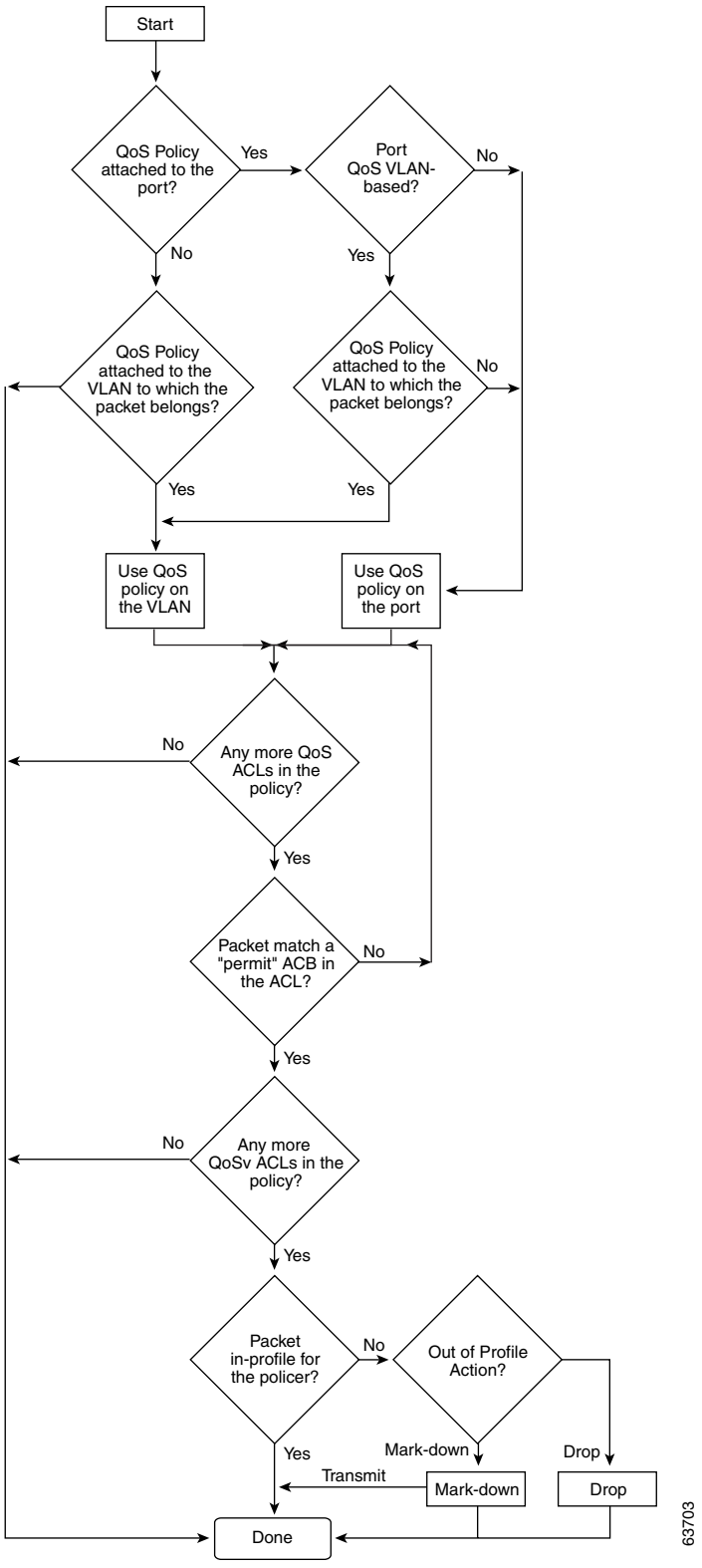
  - Greater than 100,000 flows can be microflow policed.

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” on page 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 28-29 and the “Creating Named Aggregate Policers” section on page 28-27.

Figure 28-4 Policing and Marking Flowchart



Internal DSCP Values

The following sections describe the internal DSCP values:

- Internal DSCP Sources, page 28-13
- Egress ToS and CoS Sources, page 28-13

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 28-52).

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 28-1 on page 28-4).

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 28-54). 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 28-52.

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 28-49.

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.

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.

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 49 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 54 for details on how to configure the DSCP to CoS mapping.

Configuring 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, page 28-17
- Effects of Auto-QoS on the Configuration, page 28-18
- Configuration Guidelines, page 28-18
- Enabling Auto-QoS for VoIP, page 28-19

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 28-2.)
- 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.

For information about the trusted boundary feature, see the "Configuring a Trusted Boundary to Ensure Port Security" section on page 28-26.

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 28-2 to the interface.

Table 28-2 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 26`
	`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`

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.
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:

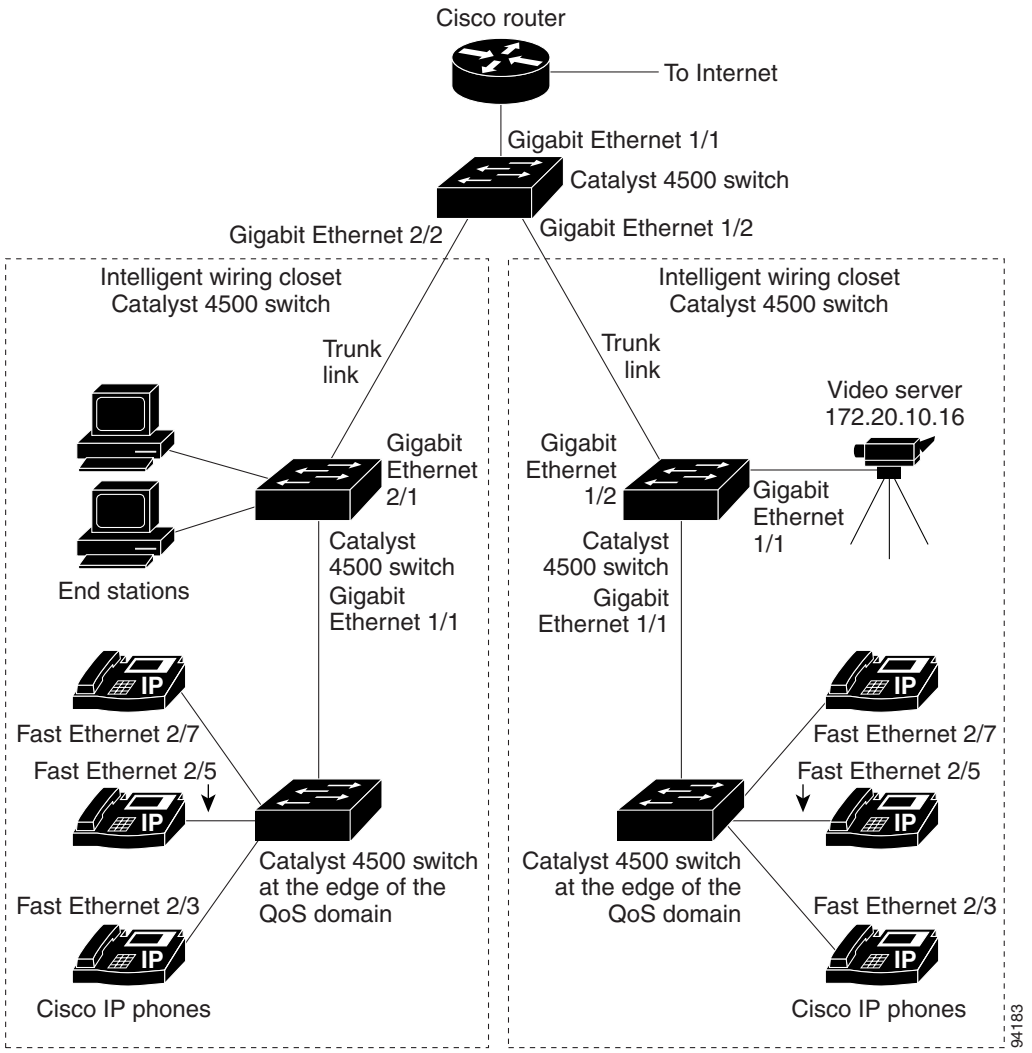
- **show qos**
- **show qos map**
- **show qos interface** [*interface-id*]

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 28-5.

Figure 28-5 Auto-QoS Configuration Example Network



The intelligent wiring closets in Figure 28-5 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 28-20.
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

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 4500 series switch:

- Default QoS Configuration, page 28-23
- Configuration Guidelines, page 28-25
- Enabling QoS Globally, page 28-25
- Configuring a Trusted Boundary to Ensure Port Security, page 28-26
- Enabling Dynamic Buffer Limiting, page 28-27
- Creating Named Aggregate Policers, page 28-27
- Configuring a QoS Policy, page 28-29
- Configuring User Based Rate Limiting, page 28-36
- Enabling Per-Port Per-VLAN QoS, page 28-42
- Enabling or Disabling QoS on an Interface, page 28-45
- Configuring VLAN-Based QoS on Layer 2 Interfaces, page 28-46
- Configuring the Trust State of Interfaces, page 28-46
- Configuring the CoS Value for an Interface, page 28-47
- Configuring DSCP Values for an Interface, page 28-48
- Configuring Transmit Queues, page 28-49
- Configuring DSCP Maps, page 28-52

Default QoS Configuration

Table 28-3 shows the QoS default configuration.

Table 28-3 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

Table 28-3 QoS Default Configuration (continued)

Feature	Default Value
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.
- If you want to use the set command in the policy map, you must enable IP routing (disabled by default) and configure an IP default route to send traffic to the next-hop device that is capable of forwarding.

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:

```
Switch# config terminal
Switch(config)# qos
Switch(config)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show qos
  QoS is enabled globally

Switch#
```

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 29, "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 clssified based on IP DSCP values. Output queue selection is not impacted by this new behaviour. 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 28-49".

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.

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.

	Command	Purpose
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

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:

```
Switch(config)# qos dbl
Global DBL enabled
Switch(config)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show qos dbl
DBL is enabled globally
DBL flow includes vlan
DBL flow includes l4-ports
DBL does not use ecn to indicate congestion
DBL exceed-action mark probability:15%
DBL max credits:15
DBL aggressive credit limit:10
DBL aggressive buffer limit:2 packets
Switch#
```

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 28-25.
- 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 28-29
- Configuring a Class Map (Optional), page 28-30
- Verifying Class Map Configuration, page 28-31
- Configuring a Policy Map, page 28-32
- Verifying Policy-Map Configuration, page 28-35
- Attaching a Policy Map to an Interface, page 28-35

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 35, "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 28-30.)
- **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:

- Creating a Class Map, page 28-30
- Configuring Filtering in a Class Map, page 28-30

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 28-29.
`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 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 The interfaces on the Catalyst 4500 series 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#
```

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 28-32
- Configuring Policy-Map Class Actions, page 28-32

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 Class Trust State, page 28-32
- Configuring the Policy Map Class DBL State, page 28-33
- Configuring Policy-Map Class Policing, page 28-33
- Using a Named Aggregate Policer, page 28-33
- Configuring a Per-Interface Policer, page 28-34

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 28-13). 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 28-14). 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 28-33
- Configuring a Per-Interface Policer, page 28-34

Using a Named Aggregate Policer

To use a named aggregate policer (see the "Creating Named Aggregate Policers" section on page 28-27), 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 28-10), 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 28-25.

- 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 28-53.

– 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
```

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 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" on page 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.
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 protocl 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
```

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.

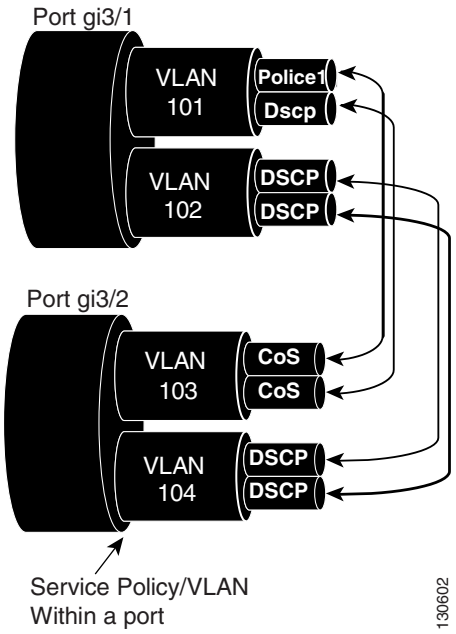
To configure per-port per-VLAN QoS, perform this task:

	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 28-6 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 28-6 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 28-35.)

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 28-49".

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 describes how to configure the transmit queues:

- Mapping DSCP Values to Specific Transmit Queues, page 28-49
- Allocating Bandwidth Among Transmit Queues, page 28-50
- Configuring Traffic Shaping of Transmit Queues, page 28-51
- Configuring a High Priority Transmit Queue, page 28-51

Depending on the complexity of your network and your QoS solution, you might need to perform all of the procedures in the next sections, but first you need to make decisions about these characteristics:

- 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.

Bandwidth can only be configured on these interfaces:

- Uplink ports on Supervisor Engine III (WS-X4014)
- Ports on the WS-X4306-GB 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

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.
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 28-52
- Configuring the Policed-DSCP Map, page 28-53
- Configuring the DSCP-to-CoS Map, page 28-54

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 28-4 shows the default CoS-to-DSCP map.

Table 28-4 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 above 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 28-5 shows the default DSCP-to-CoS map.

Table 28-5 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)# 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.
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.

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

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 above 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.