Configuring QoS

This chapter describes how to use different methods to configure quality of service (QoS) on the Catalyst 3750 Metro switch. With QoS, you can provide preferential treatment to certain traffic at the expense of others. Without QoS, the switch offers best-effort service to each packet, regardless of the packet contents or size. It sends the packets without any assurance of reliability, delay bounds, or throughput.

You can use auto-QoS to identify ports connected to Cisco IP phones and ports that receive trusted voice over IP (VoIP) traffic.

You can use standard QoS to classify, police, mark, queue, and schedule inbound traffic on any port as well as queue and schedule outbound traffic. On ingress, standard QoS offers classification based on the class of service (CoS), Differentiated Services Code Point (DSCP), or IP precedence value in the inbound packet. You can perform the classification based on Layer 2 MAC, IP-standard, or IP-extended access control lists (ACLs). Standard QoS also offers single-rate or aggregate traffic policing on ingress. Drop policy actions are passing through the packet without modification, marking down the assigned DSCP in the packet, or dropping the packet. Standard QoS performs ingress queueing based on the weighted tail drop (WTD) algorithm and ingress scheduling based on shaped round robin (SRR). On egress, standard QoS offers queueing based on WTD and scheduling based on SRR shared or shaped weights.

You can use hierarchical QoS to classify, police, mark, queue, and schedule outbound traffic on an enhanced-services (ES) port. On egress, hierarchical QoS offers classification based on the CoS, DSCP, IP precedence, or the multiprotocol label switching (MPLS) experimental (EXP) bits in the outbound packet. You also can classify a packet based on its VLAN. Hierarchical QoS offers two-rate traffic policing on egress. Drop policy actions are passing through the packet without modification; marking down the CoS, DSCP, IP precedence, or the MPLS EXP bits in the packet; or dropping the packet. Hierarchical QoS performs egress queueing based on tail drop or Weighted Random Early Detection (WRED). The queue scheduling management features is class-based weighted fair queueing (CBWFQ), and the scheduling congestion-management features is low-latency queueing (LLQ). You can use traffic shaping to decrease the burstiness of traffic.

For information about multiprotocol label switching (MPLS), Ethernet over MPLS (EoMPLS), and QoS, see the "Configuring MPLS and EoMPLS QoS" section on page 30-18.

Note For complete syntax and usage information for the commands used in this chapter, refer to the command reference this release.

This chapter consists of these sections:

Understanding 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.

When you configure the QoS feature, you can select specific network traffic, prioritize it according to its relative importance, and use congestion-management and congestion-avoidance techniques to provide preferential treatment. Implementing QoS in your network makes network performance more predictable and bandwidth utilization more effective.

The QoS implementation is based on the Differentiated Services (Diff-Serv) 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 also can be carried in the Layer 2 frame. These special bits in the Layer 2 frame or in the Layer 3 packet are described here and shown in Figure 26-1:

- Prioritization bits in Layer 2 frames:

  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. On ports 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 ports 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.

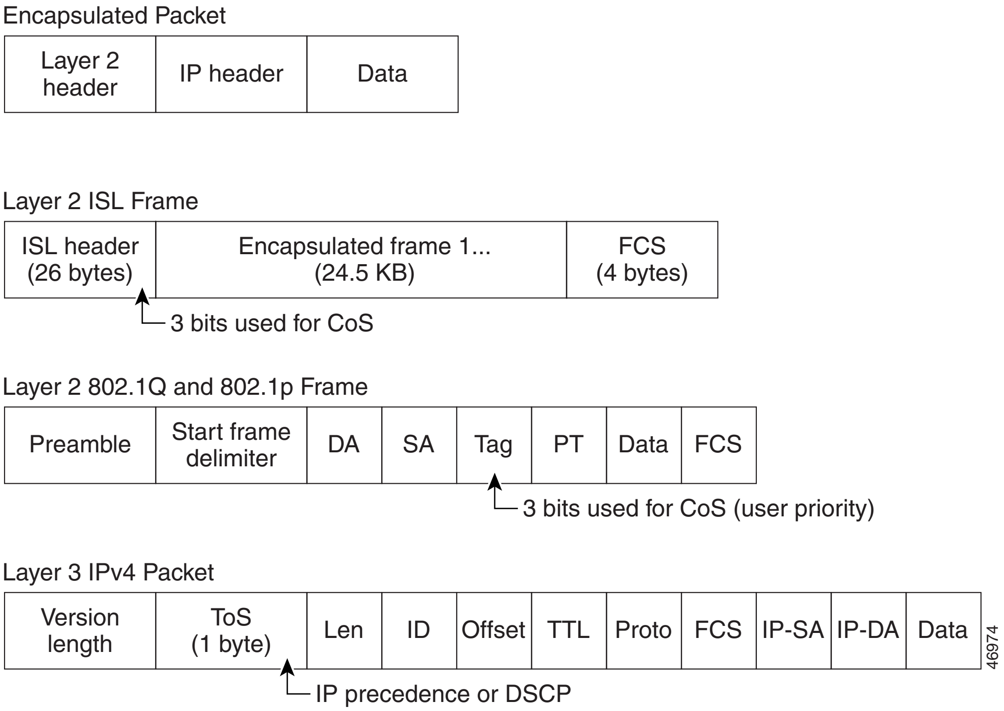
- Prioritization bits in Layer 3 packets:

  Layer 3 IP packets can carry either an IP precedence value or a 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 26-1 QoS Classification Layers in Frames and Packets



Note Layer 3 IPv6 packets are treated as non-IP packets and are bridged by the switch.

All switches and routers that access 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 with this task.

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.

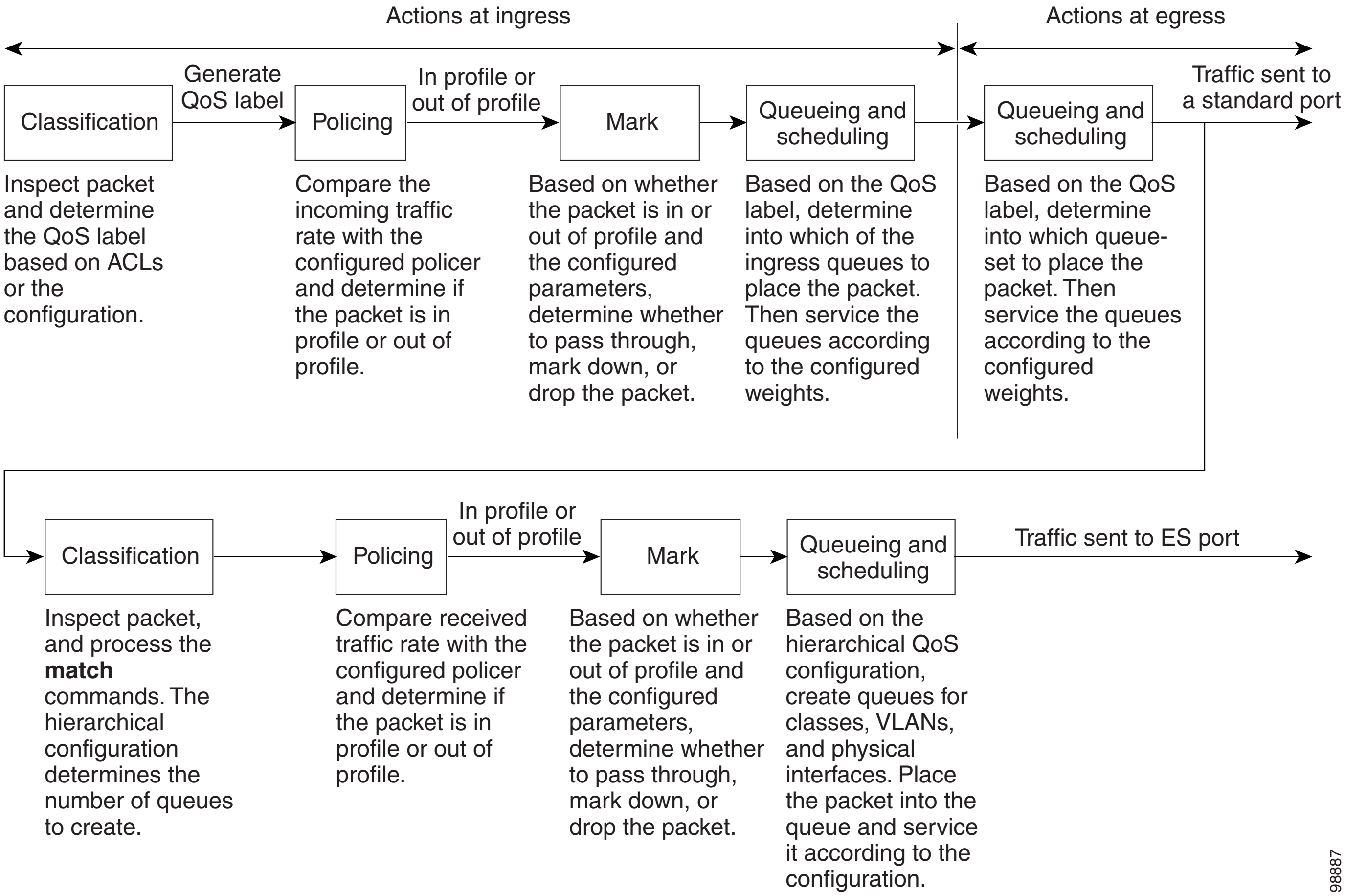
Implementing QoS in your network can be a simple or complex task and depends on the QoS features offered by your internetworking devices, the traffic types and patterns in your network, and the granularity of control that you need over inbound and outbound traffic.

Basic QoS Model

To implement QoS, the switch must distinguish (classify) packets or flows from one another, assign a label to indicate the given quality of service as the packets move through the switch, make the packets comply with the configured resource usage limits (police and mark), and provide different treatment (queue and schedule) in all situations where resource contention exists. The switch also needs to ensure that traffic sent from it meets a specific traffic profile (shape).

Figure 26-2 shows the basic QoS model.

Figure 26-2 Basic QoS Model


Actions at ingress
Actions at egress
Classification
Generate QoS label
Policing
In profile or out of profile
Mark
Queueing and scheduling
Queueing and scheduling
Traffic sent to a standard port
Inspect packet and determine the QoS label based on ACLs or the configuration.
Compare the incoming traffic rate with the configured policer and determine if the packet is in profile or out of profile.
Based on whether the packet is in or out of profile and the configured parameters, determine whether to pass through, mark down, or drop the packet.
Based on the QoS label, determine into which of the ingress queues to place the packet. Then service the queues according to the configured weights.
Based on the QoS label, determine into which queue-set to place the packet. Then service the queues according to the configured weights.
Classification
Policing
In profile or out of profile
Mark
Queueing and scheduling
Traffic sent to ES port
Inspect packet, and process the **match** commands. The hierarchical configuration determines the number of queues to create.
Compare received traffic rate with the configured policer and determine if the packet is in profile or out of profile.
Based on whether the packet is in or out of profile and the configured parameters, determine whether to pass through, mark down, or drop the packet.
Based on the hierarchical QoS configuration, create queues for classes, VLANs, and physical interfaces. Place the packet into the queue and service it according to the configuration.
98887

These are the actions when traffic is received by the switch:

- Classification is the process of generating a distinct path for a packet by associating it with a QoS label. The switch maps the CoS or DSCP in the packet to a QoS label to distinguish one kind of traffic from another. The QoS label that is generated identifies all future QoS actions to be performed on this packet. For more information, see the "Ingress Classification" section on page 26-6.
- Policing decides whether a packet is in or out of profile by comparing the rate of the inbound traffic to the configured policer. The policer limits the bandwidth consumed by a flow of traffic. The result is passed to the marker. For more information, see the "Ingress Policing and Marking" section on page 26-9.

- Marking evaluates the policer configuration information for the action to take when a packet is out of profile. Marking actions are to pass through a packet without modification, to mark down the QoS label in the packet, or to drop the packet. For more information, see the "Ingress Policing and Marking" section on page 26-9.
- Queueing evaluates the QoS label and the corresponding DSCP or CoS value to select into which of the two ingress queues to place a packet. Queueing is enhanced with the WTD algorithm, a congestion-avoidance mechanism. If the threshold is exceeded, the packet is dropped. For more information, see the "Queueing and Scheduling Overview" section on page 26-12.
- Scheduling services the queues based on their configured (SRR) weights. One of the ingress queues is the priority queue, and SRR services it for its configured share before servicing the other queue. For more information, see the "SRR Shaping and Sharing" section on page 26-14.

These are the actions when the traffic is sent out a standard port:

- Queueing evaluates the QoS label and the corresponding DSCP or CoS value to select into which queue-set (a set of four queues per port) to place a packet. Because congestion can occur when multiple ingress ports simultaneously send data to an egress port, WTD is used to differentiate traffic classes and to subject the packets to different thresholds based on the QoS label. If the threshold is exceeded, the packet is dropped. For more information, see the "Queueing and Scheduling Overview" section on page 26-12.
- Scheduling services the four egress queues based on their configured SRR shared or shaped weights.

These are the actions when the traffic is sent out an ES port:

- Classification is the process of generating a distinct path for a packet by matching the CoS, DSCP, IP precedence, MPLS EXP bits in the header, or matching a packet based on the inner and the outer VLAN IDs. The hierarchical configuration controls the number of class-level, VLAN-level, and physical-interface-level queues to create. For information, see the "Understanding Hierarchical QoS" section on page 26-19 and the "Hierarchical Levels" section on page 26-20. For classification information, see the "Egress Classification Based on Traffic Classes and Traffic Policies" section on page 26-23.
- Policing decides whether a packet is in or out of profile by comparing the rate of the outbound traffic to the configured policer. The policer limits the bandwidth consumed by a flow of traffic. The result is passed to the marker. For more information, see the "Egress Policing and Marking" section on page 26-24.
- Marking evaluates the policer and configuration information for 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 QoS label in the packet, or drop the packet). For more information, see the "Egress Policing and Marking" section on page 26-24.
- Queueing is accomplished through a hierarchical queueing framework, in which the switch assigns each packet to a queue based on the packet class, VLAN, and physical interface. Tail drop or WRED can be configured per queue as the congestion-avoidance mechanism. With tail drop, packets are queued until the maximum threshold is exceeded, and then all the packets are dropped. WRED reduces the chances of tail drop by selectively dropping packets when the port begins to show signs of congestion. For more information, see the "Queueing and Scheduling of Hierarchical Queues" section on page 26-26.
- Scheduling is accomplished through CBWFQ or LLQ (strict priority queueing). CBWFQ is a mechanism that provides guaranteed bandwidth to a particular traffic class while still fairly serving all other traffic in the network. LLQ is another scheduling mechanism, which ensures that delay-sensitive traffic is queued and sent before the traffic in other queues. Scheduling services the queues through average-rate shaping. For more information, see the "Queueing and Scheduling of Hierarchical Queues" section on page 26-26.

Ingress Classification

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

During ingress classification, the switch performs a lookup and assigns a QoS label to the packet. The QoS label identifies all QoS actions to be performed on the packet and from which queue the packet is sent.

The QoS label is based on the DSCP or the CoS value in the packet and decides the queueing and scheduling actions to perform on the packet. The label is mapped according to the trust setting and the packet type as shown in Figure 26-3 on page 26-7.

You specify which fields in the frame or packet that you want to use to classify inbound traffic. For non-IP traffic, you have these ingress classification options as shown in Figure 26-3:

- Trust the CoS value in the inbound frame (configure the port to trust CoS). Then use the configurable CoS-to-DSCP map to generate a DSCP value for the packet. 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.
- Trust the DSCP or trust IP precedence value in the inbound frame. These configurations are meaningless for non-IP traffic. If you configure a port with either of these options and non-IP traffic is received, the switch assigns a CoS value and generates a DSCP value from the CoS-to-DSCP map.
- Perform the classification based on a configured Layer 2 MAC ACL, which can examine the MAC source address, the MAC destination address, and other fields. If no ACL is configured, the packet is assigned 0 as the DSCP and CoS values, which means best-effort traffic. Otherwise, the policy-map action specifies a DSCP or a CoS value to assign to the inbound frame.

For IP traffic, you have these ingress classification options as shown in Figure 26-3:

- Trust the DSCP value in the inbound packet (configure the port to trust DSCP), and assign the same DSCP value to the packet. The IETF defines the six most-significant bits of the 1-byte ToS field as the DSCP. The priority represented by a particular DSCP value is configurable. DSCP values range from 0 to 63.

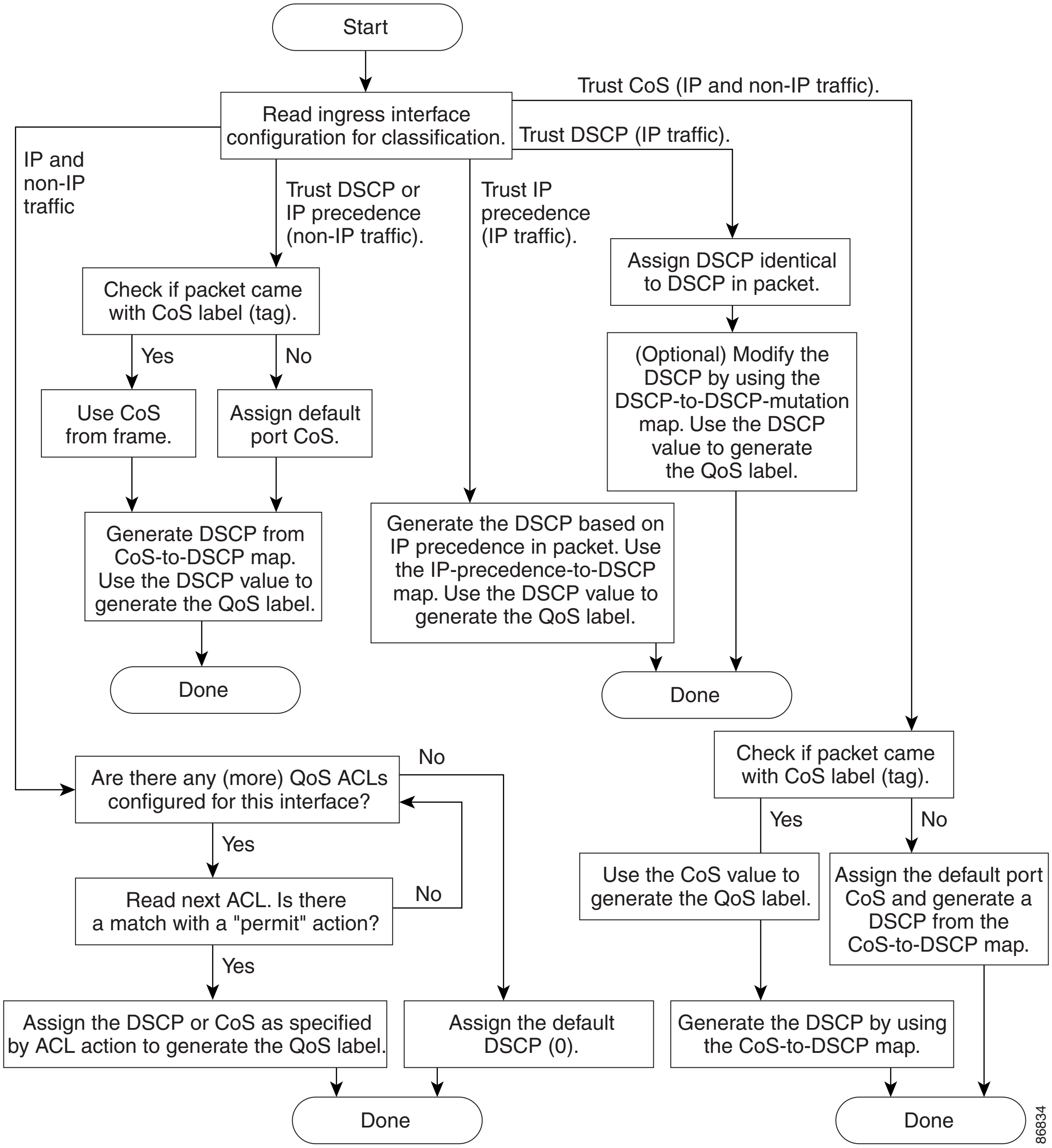
  For ports that are on the boundary between two QoS administrative domains, you can modify the DSCP to another value through the configurable DSCP-to-DSCP-mutation map.
- Trust the IP precedence value in the inbound packet (configure the port to trust IP precedence), and generate a DSCP value for the packet through the configurable IP-precedence-to-DSCP map. The IP version 4 specification defines the three most-significant bits of the 1-byte ToS field as the IP precedence. IP precedence values range from 0 for low priority to 7 for high priority.
- Trust the CoS value (if present) in the inbound packet, and generate a DSCP value for the packet through the CoS-to-DSCP map. If the CoS value is not present, use the default port CoS value.
- Perform the classification based on a configured IP standard or an extended ACL, which examines various fields in the IP header. If no ACL is configured, the packet is assigned 0 as the DSCP and CoS values, which means best-effort traffic. Otherwise, the policy-map action specifies a DSCP or a CoS value to assign to the inbound frame.

For information on the maps described in this section, see the "Mapping Tables" section on page 26-11. For configuration information on port trust states, see the "Configuring Ingress Classification by Using Port Trust States" section on page 26-42.

You can configure classification through ACLs, traffic classes, and traffic policies. For more information, see the "Ingress Classification Based on QoS ACLs" section on page 26-8 and the "Ingress Classification Based on Traffic Classes and Traffic Policies" section on page 26-8.

After ingress classification, the packet is sent to the policing, marking, and the ingress queueing and scheduling stages.

Figure 26-3 Ingress Classification Flowchart


Start
Read ingress interface configuration for classification.
Trust CoS (IP and non-IP traffic).
Trust DSCP (IP traffic).
IP and non-IP traffic
Trust DSCP or IP precedence (non-IP traffic).
Trust IP precedence (IP traffic).
Assign DSCP identical to DSCP in packet.
(Optional) Modify the DSCP by using the DSCP-to-DSCP-mutation map. Use the DSCP value to generate the QoS label.
Check if packet came with CoS label (tag).
Yes
No
Use CoS from frame.
Assign default port CoS.
Generate DSCP from CoS-to-DSCP map. Use the DSCP value to generate the QoS label.
Generate the DSCP based on IP precedence in packet. Use the IP-precedence-to-DSCP map. Use the DSCP value to generate the QoS label.
Done
Done
Are there any (more) QoS ACLs configured for this interface?
No
Yes
Read next ACL. Is there a match with a "permit" action?
No
Yes
Assign the DSCP or CoS as specified by ACL action to generate the QoS label.
Assign the default DSCP (0).
Done
Check if packet came with CoS label (tag).
Yes
No
Use the CoS value to generate the QoS label.
Assign the default port CoS and generate a DSCP from the CoS-to-DSCP map.
Generate the DSCP by using the CoS-to-DSCP map.
Done
86834

Ingress Classification Based on QoS ACLs

You can use IP standard, IP extended, or Layer 2 MAC ACLs to define a group of packets with the same characteristics (*class*). In the QoS context, the permit and deny actions in the access control entries (ACEs) have different meanings than with security ACLs:

- If a match with a permit action is encountered (first-match principle), the specified QoS-related action is taken.
- If a match with a deny action is encountered, the ACL being processed is skipped, and the next ACL is processed.
- If no match with a permit action is encountered and all the ACEs have been examined, no QoS processing occurs on the packet, and the switch offers best-effort service to the packet.
- If multiple ACLs are configured on a port, the lookup stops after the packet matches the first ACL with a permit action, and QoS processing begins.

Note When you create 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 is defined with the ACL, you can attach a policy to it. 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 to rate-limit the class. This ingress policy is then attached to a port.

You implement IP ACLs to classify IP traffic by using the **access-list** global configuration command; you implement Layer 2 MAC ACLs to classify non-IP traffic by using the **mac access-list extended** global configuration command. For configuration information, see the "Configuring an Ingress QoS Policy" section on page 26-48.

Ingress Classification Based on Traffic Classes and Traffic Policies

You define a traffic class to classify traffic, use a traffic policy to decide how to treat the classified traffic, and attach the ingress policy to a port to create a service policy.

You use the class map to define a specific traffic flow (or class) and to isolate it from all other traffic. The class map defines the criteria 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.

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 criterion for the traffic by using the **match** class-map configuration command. Inbound packets are checked against the match criteria configured for a class map to decide if the packet belongs to that class. If a packet matches the specified criteria, the packet is considered a member of the class and is forwarded according to the QoS specifications set in the traffic policy. If a packet fails to meet any of the matching criteria, it is classified as a member of the default traffic class if one is configured.

You use the policy map to create the traffic policy, to specify the traffic class to act on, and to configure the QoS features associated with the traffic class. Actions on ingress can include trusting the received CoS, DSCP, or IP precedence 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.

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 use the **class** policy-map configuration command to name the traffic class associated with the traffic policy. If you specify **class-default** as the class name in the **class** policy-map configuration command, packets that fail to meet any of the matching criteria are classified as members of the default traffic class. You can manipulate this class (for example, police it and mark it) just like any traffic class, but you cannot delete it. After you name the traffic class with the **class** command, the switch enters policy-map class configuration mode, and you can specify the actions to take on this traffic class.

An ingress policy-map can include **police**, **police aggregate**, **trust**, or **set** policy-map class configuration commands. You attach the ingress policy-map to a port by using the **service-policy input** interface configuration command.

For more information, see the "Ingress Policing and Marking" section on page 26-9. For configuration information, see the "Configuring an Ingress QoS Policy" section on page 26-48.

Ingress Policing and Marking

After a packet is classified and has a DSCP-based or CoS-based QoS label assigned to it, the traffic policing and marking process can begin. Figure 26-4 on page 26-11 shows an ingress, single-rate traffic policer.

Ingress traffic policing controls the maximum rate of traffic received on a port. The policer defines the bandwidth limitations of the traffic, and the action to take if the limits are exceeded. It is often configured on ports at the edge of a network to limit traffic into or out of the network. In most policing configurations, traffic that falls within the rate parameters is sent. Traffic that exceeds the parameters is considered to be *out of profile* or *nonconforming* and is dropped or sent with a different priority.

Note All traffic, regardless of whether it is bridged or routed, is subjected to a configured policer. As a result, bridged packets might be dropped or might have their DSCP or CoS fields modified when they are policed and marked.

You can create these types of ingress policers:

- Individual (single-rate)

  QoS applies the bandwidth limits specified in the policer separately to each matched traffic class. You can configure a single-rate policer within a policy map by using the **police** policy-map class configuration command.

- 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 class configuration command. You specify the bandwidth limits of the policer by using the **mls qos aggregate-policer** global configuration command. In this way, the aggregate policer is shared by multiple classes of traffic within a policy map.

A single-rate traffic policer decides on a packet-by-packet basis whether the packet is in or out of profile and specifies the actions on the packet. These actions, carried out by the marker, include passing through the packet without modification, dropping the packet, or modifying (marking down) the assigned DSCP of the packet and allowing the packet to pass through. The configurable policed-DSCP map provides the packet with a new DSCP-based QoS label. For information on the policed-DSCP map, see the "Mapping Tables" section on page 26-11. Marked-down packets use the same queues as the original QoS label to prevent packets in a flow from getting out of order.

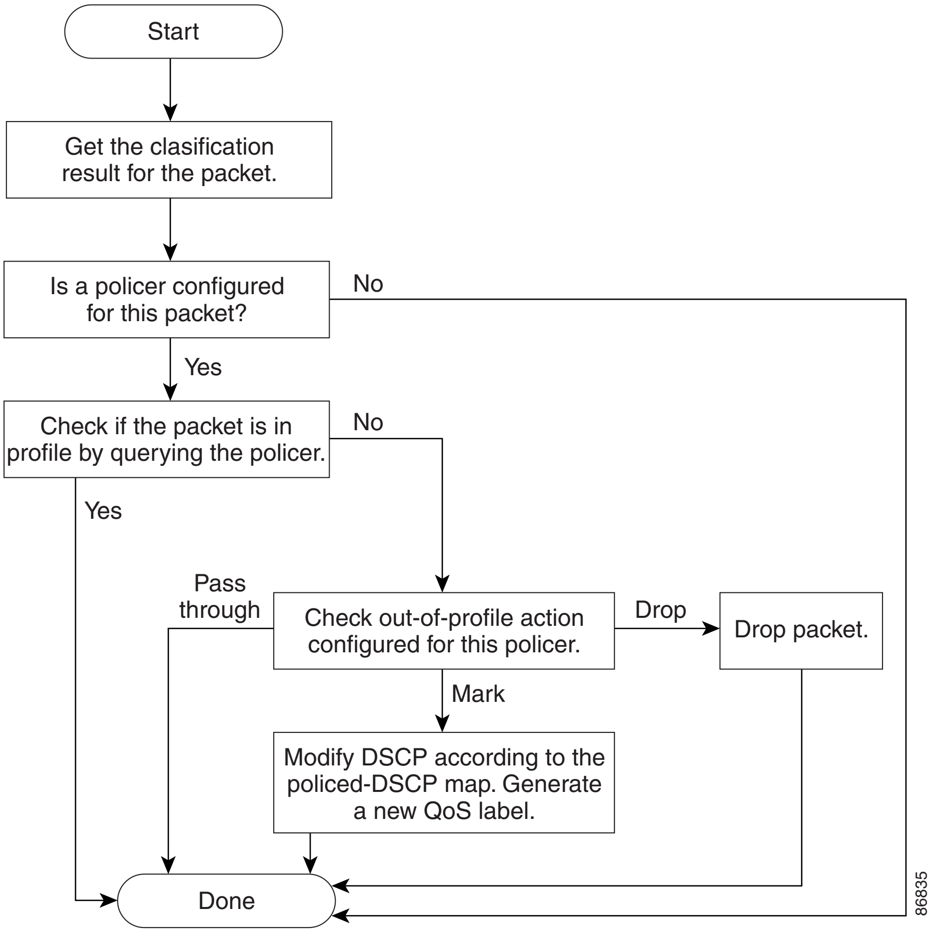
Single-rate policing uses a token-bucket algorithm. As each frame is received by the switch, a token is added to the bucket. The bucket has a hole in it and leaks at a rate that you specify as the average traffic rate in bps. Each time a token is added to the bucket, the switch verifies that there is enough room in the bucket. If there is not enough room, the packet is marked as nonconforming, and the specified policer action is taken (dropped or marked down).

How quickly the bucket fills is a function of the bucket depth (burst-byte), the rate at which the tokens are removed (rate-bps), and the duration of the burst above the average rate. The size of the bucket imposes an upper limit on the burst length and limits the number of frames that can be sent back-to-back. If the burst is short, the bucket does not overflow, and no action is taken against the traffic flow. However, if a burst is long and at a higher rate, the bucket overflows, and the policing actions are taken against the frames in that burst.

You configure the bucket depth (the maximum burst that is tolerated before the bucket overflows) by using the *burst-byte* option of the **police** policy-map class configuration command or the **mls qos aggregate-policer** global configuration command. You configure how fast (the average rate) that the tokens are removed from the bucket by using the *rate-bps* option of the **police** policy-map class configuration command or the **mls qos aggregate-policer** global configuration command.

After you configure the policy map and policing actions, attach the ingress policy to a port by using the **service-policy input** interface configuration command. For configuration information, see the "Classifying, Policing, and Marking Ingress Traffic by Using Policy Maps" section on page 26-54 and the "Classifying, Policing, and Marking Ingress Traffic by Using Aggregate Policers" section on page 26-57.

Figure 26-4 Ingress, Single-Rate Policing and Marking Flowchart


Start
Get the clasification result for the packet.
Is a policer configured for this packet?
No
Yes
Check if the packet is in profile by querying the policer.
No
Yes
Pass through
Check out-of-profile action configured for this policer.
Drop
Drop packet.
Mark
Modify DSCP according to the policed-DSCP map. Generate a new QoS label.
Done
86835

Mapping Tables

During QoS ingress processing, the switch represents the priority of all traffic (including non-IP traffic) with an QoS label based on the DSCP or CoS value from the classification stage:

- During ingress classification, QoS uses configurable mapping tables to derive a corresponding DSCP or CoS value from a received CoS, DSCP, or IP precedence value. These maps include the CoS-to-DSCP map and the IP-precedence-to-DSCP map. You configure these maps by using the **mls qos map cos-dscp** and the **mls qos map ip-prec-dscp** global configuration commands.

  On an ingress port configured in the DSCP-trusted state, if the DSCP values are different between the QoS domains, you can apply the configurable DSCP-to-DSCP-mutation map to the port that is on the boundary between the two QoS domains. You configure this map by using the **mls qos map dscp-mutation** global configuration command.

- During ingress policing, QoS can assign another DSCP value to an IP or a non-IP packet (if the packet is out of profile and the policer specifies a marked-down value). This configurable map is called the policed-DSCP map. You configure this map by using the **mls qos map policed-dscp** global configuration command.

- Before the traffic reaches the scheduling stage, QoS stores the packet in an ingress and an egress queue according to the QoS label. The QoS label is based on the DSCP or the CoS value in the inbound packet. The QoS label selects an ingress queue and an egress queue from the queue-set through the DSCP input and output queue threshold maps or through the CoS input and output queue threshold maps. You configure these maps by using the **mls qos srr-queue** {**input** | **output**} **dscp-map** and the **mls qos srr-queue** {**input** | **output**} **cos-map** global configuration commands.

The CoS-to-DSCP, DSCP-to-CoS, and the IP-precedence-to-DSCP maps have default values that might or might not be appropriate for your network.

The default DSCP-to-DSCP-mutation map and the default policed-DSCP map are null maps; they map an inbound DSCP value to the same DSCP value. The DSCP-to-DSCP-mutation map is the only map you apply to a specific port. All other maps apply to the entire switch.

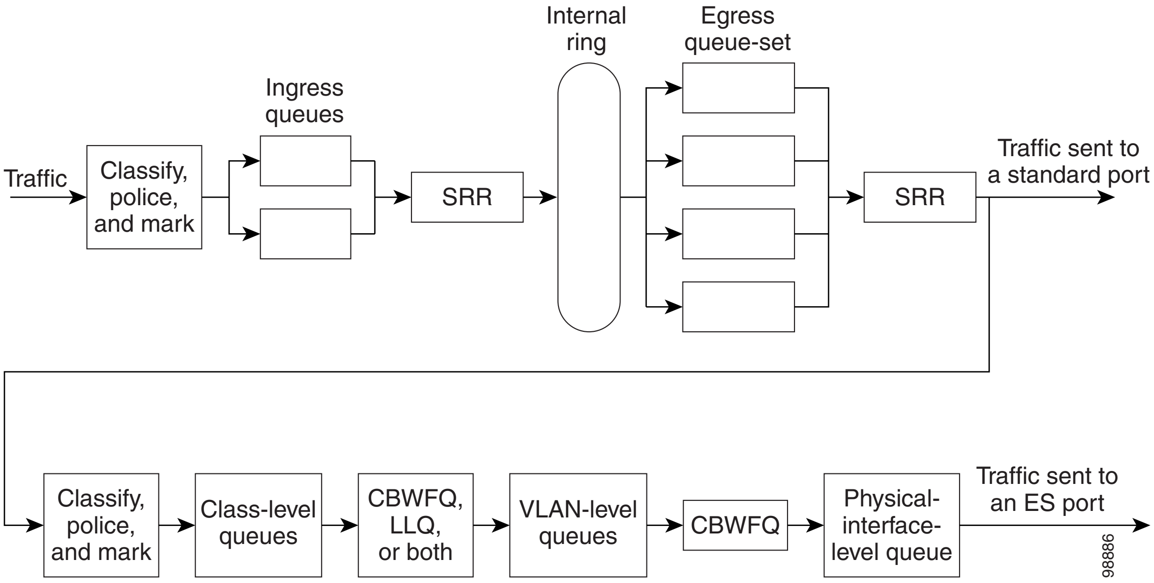
For configuration information, see the "Configuring DSCP Maps" section on page 26-59.

For information about the DSCP and CoS input queue threshold maps, see the "Queueing and Scheduling of Ingress Queues" section on page 26-15. For information about the DSCP and CoS output queue threshold maps, see the "Queueing and Scheduling of Egress Queue-Sets" section on page 26-17.

Queueing and Scheduling Overview

The switch has queues at specific points to help prevent congestion, as shown in Figure 26-5.

Figure 26-5 Ingress and Egress Queue Location


Internal ring
Egress queue-set
Ingress queues
Traffic
Classify, police, and mark
SRR
SRR
Traffic sent to a standard port
Classify, police, and mark
Class-level queues
CBWFQ, LLQ, or both
VLAN-level queues
CBWFQ
Physical-interface-level queue
Traffic sent to an ES port
98886

Because the total ingress bandwidth of all ports can exceed the bandwidth of the internal ring, ingress queues are located after the packet is classified, policed, and marked and before packets are forwarded into the switch fabric. Because multiple ingress ports can simultaneously send packets to an egress port and cause congestion, egress queue-sets are located after the internal ring.

Each port belongs to an egress queue-set, which defines all the characteristics of the four queues per port. The ES ports also use a hierarchical queueing model in which each packet is assigned to a hierarchical queue based on the physical interface, VLAN, or class. Traffic destined for an ES port passes through the queue-set before reaching the hierarchical queues. If congestion occurs in the hierarchical queues and backs up to the queue-sets, the queue-set configuration controls how traffic is dropped.

Ingress queues use WTD for congestion management. The egress queue-sets also use WTD. For more information, see the next section.

Ingress queues support SRR in shared mode for scheduling. The egress queue-sets also support SRR in shared or shaped mode for scheduling. For more information, see the "SRR Shaping and Sharing" section on page 26-14.

For information about ingress queueing and scheduling, see the "Queueing and Scheduling of Ingress Queues" section on page 26-15. For information about egress queue-set queueing and scheduling, see the "Queueing and Scheduling of Egress Queue-Sets" section on page 26-17.

The egress hierarchical queues for ES ports use tail drop or WRED for congestion management. These ports also use CBWFQ or LLQ for scheduling. For more information, see the "Queueing and Scheduling of Hierarchical Queues" section on page 26-26.

Weighted Tail Drop

The ingress queues and the egress queue-sets use an enhanced version of the tail-drop congestion-avoidance mechanism called WTD. WTD manages the queue lengths and provides drop precedences for different traffic classifications.

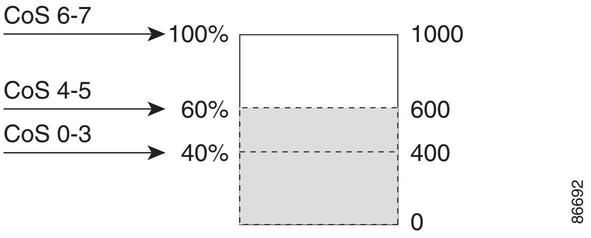
As a frame is sent to a particular queue, WTD uses the frame's assigned QoS label to subject it to different thresholds. If the threshold is exceeded for that QoS label (the space available in the destination queue is less than the size of the frame), the switch drops the frame.

Figure 26-6 shows an example of WTD operating on a queue whose size is 1000 frames. Three drop percentages are configured: 40, 60, and 100 percent. These percentages mean that up to 400 frames can be queued at the 40-percent threshold, up to 600 frames at the 60-percent threshold, and up to 1000 frames at the 100-percent threshold.

In this example, CoS values 6 and 7 have a greater importance than the other CoS values, and they are assigned to the 100-percent drop threshold (queue-full state). CoS values 4 and 5 are assigned to the 60-percent threshold, and CoS values 0 to 3 are assigned to the 40-percent threshold.

Suppose the queue is already filled with 600 frames, and a new frame arrives. It contains CoS values 4 and 5 and is subjected to the 60-percent threshold. If this frame is added to the queue, the threshold will be exceeded, so the switch drops it.

Figure 26-6 WTD and Queue Operation



For more information, see the "Mapping DSCP or CoS Values to an Ingress Queue and Setting WTD Thresholds" section on page 26-65, the "Allocating Buffer Space to and Setting WTD Thresholds for an Egress Queue-Set" section on page 26-69, and the "Mapping DSCP or CoS Values to an Egress Queue-Set and to a Threshold ID" section on page 26-71.

SRR Shaping and Sharing

The ingress queues and egress queue-sets are serviced by SRR, which controls the rate at which packets are sent. On the ingress queues, SRR sends packets to the internal ring. On the egress queue-sets, SRR sends packets to a standard port.

You can configure SRR on the egress queue-sets for sharing or for shaping. However, for ingress queues, sharing is the default mode and is the only mode supported.

In shaped mode, the queues are guaranteed a percentage of the bandwidth, and they are rate-limited to that amount. Shaped traffic does not use more than the allocated bandwidth even if the link is idle. Shaping provides a more even flow of traffic over time and reduces the peaks and valleys of bursty traffic. With shaping, the absolute value of each weight is used to compute the bandwidth available for the queues.

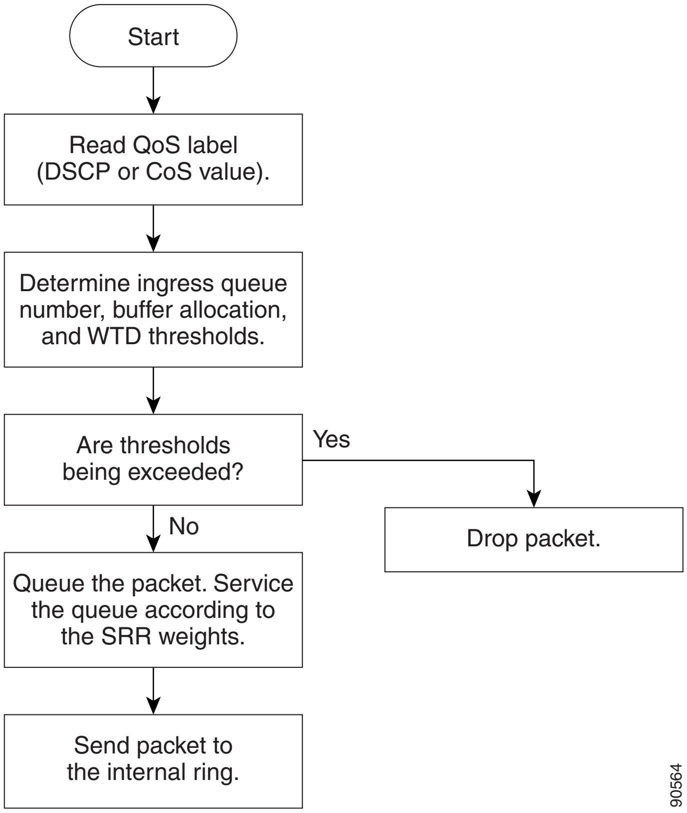
In shared mode, the queues share the bandwidth among them according to the configured weights. The bandwidth is guaranteed at this level but not limited to it. For example, if a queue is empty and no longer requires a share of the link, the remaining queues can expand into the unused bandwidth and share it among them. With sharing, the ratio of the weights controls the frequency of dequeuing; the absolute values are meaningless.

For more information, see the "Allocating Bandwidth Between the Ingress Queues" section on page 26-67, the "Configuring SRR Shaped Weights on an Egress Queue-Set" section on page 26-72, and the "Configuring SRR Shared Weights on an Egress Queue-Set" section on page 26-73.

Queueing and Scheduling of Ingress Queues

Figure 26-7 shows the ingress queueing and scheduling flowchart.

Figure 26-7 Ingress Queueing and Scheduling Flowchart


Start
Read QoS label (DSCP or CoS value).
Determine ingress queue number, buffer allocation, and WTD thresholds.
Are thresholds being exceeded?
Yes
Drop packet.
No
Queue the packet. Service the queue according to the SRR weights.
Send packet to the internal ring.
90564

Note SRR services the priority queue for its configured share before servicing the other queue.

The switch supports two configurable ingress queues, which are serviced by SRR in shared mode only. Table 26-1 describes the queues.

Table 26-1 Ingress Queue Types

Queue Type[1]	Function
Normal	User traffic that is considered to be normal priority. You can configure three different thresholds to differentiate among the flows. You can use the **mls qos srr-queue input threshold**, the **mls qos srr-queue input dscp-map**, and the **mls qos srr-queue input cos-map** global configuration commands.
Expedite	High-priority user traffic such as differentiated services (DF) expedited forwarding or voice traffic. You can configure the bandwidth required for this traffic as a percentage of the total traffic by using the **mls qos srr-queue input priority-queue** global configuration command. The expedite queue has guaranteed bandwidth.

[1] The switch uses two nonconfigurable queues for traffic that is essential for proper network operation.

You assign each packet that flows through the switch to a queue and to a threshold. Specifically, you map DSCP or CoS values to an ingress queue and map DSCP or CoS values to a threshold ID. You use the **mls qos srr-queue input dscp-map queue** *queue-id* {*dscp1...dscp8* | **threshold** *threshold-id dscp1...dscp8*} or the **mls qos srr-queue input cos-map queue** *queue-id* {*cos1...cos8* | **threshold** *threshold-id cos1...cos8*} global configuration command. You can display the DSCP input queue threshold map and the CoS input queue threshold map by using the **show mls qos maps** privileged EXEC command.

WTD Thresholds

The queues use WTD to support distinct drop percentages for different traffic classes. Each queue has three drop thresholds: two configurable (*explicit*) WTD thresholds and one nonconfigurable (*implicit*) threshold preset to the queue-full state. You assign the two explicit WTD threshold percentages for threshold ID 1 and ID 2 to the ingress queues by using the **mls qos srr-queue input threshold** *queue-id threshold-percentage1 threshold-percentage2* global configuration command. Each threshold value is a percentage of the total number of allocated buffers for the queue. The drop threshold for threshold ID 3 is preset to the queue-full state, and you cannot modify it. For more information about how WTD works, see the "Weighted Tail Drop" section on page 26-13.

Buffer and Bandwidth Allocation

You define the ratio (allocate the amount of space) with which to divide the ingress buffers between the two queues by using the **mls qos srr-queue input buffers** *percentage1 percentage2* global configuration command. The buffer allocation together with the bandwidth allocation control how much data can be buffered and sent before packets are dropped. You allocate bandwidth as a percentage by using the **mls qos srr-queue input bandwidth** *weight1 weight2* global configuration command. The ratio of the weights is the ratio of the frequency in which the SRR scheduler sends packets from each queue to the internal ring.

Priority Queueing

You can configure one ingress queue as the priority queue by using the **mls qos srr-queue input priority-queue** *queue-id* **bandwidth** *weight* global configuration command. The priority queue should be used for traffic (such as voice) that requires guaranteed delivery because this queue is guaranteed part of the bandwidth regardless of the load on the internal ring.

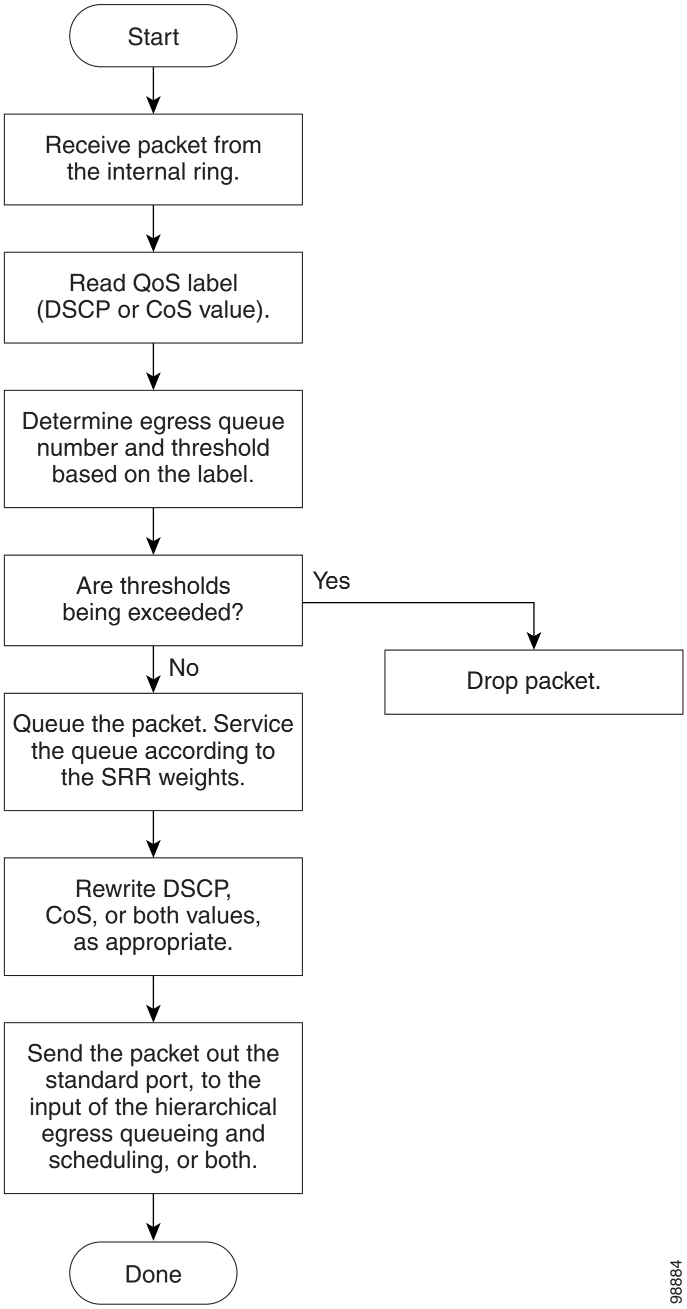
SRR services the priority queue for its configured weight as specified by the **bandwidth** keyword in the **mls qos srr-queue input priority-queue** *queue-id* **bandwidth** *weight* global configuration command. Then, SRR shares the remaining bandwidth with both ingress queues and services them as specified by the weights configured with the **mls qos srr-queue input bandwidth** *weight1 weight2* global configuration command.

You can combine the commands described in this section to prioritize traffic by placing packets with particular DSCP or CoS values into certain queues, by allocating a large queue size or by servicing the queue more frequently, and by adjusting queue thresholds so that packets with lower priorities are dropped. For configuration information, see the "Configuring Ingress Queue Characteristics" section on page 26-64.

Queueing and Scheduling of Egress Queue-Sets

Figure 26-8 shows the egress queue-set queueing and scheduling flowchart.

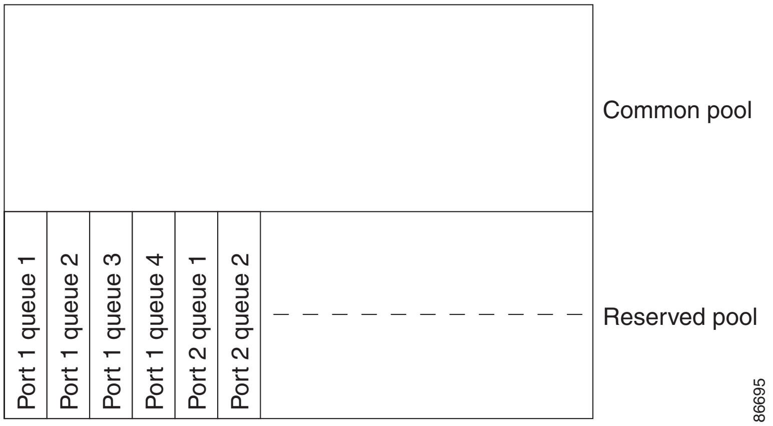
Figure 26-8 Egress Queue-Set Queueing and Scheduling Flowchart


Start
Receive packet from the internal ring.
Read QoS label (DSCP or CoS value).
Determine egress queue number and threshold based on the label.
Are thresholds being exceeded?
Yes
Drop packet.
No
Queue the packet. Service the queue according to the SRR weights.
Rewrite DSCP, CoS, or both values, as appropriate.
Send the packet out the standard port, to the input of the hierarchical egress queueing and scheduling, or both.
Done
98884

Each port supports four egress queues. These queues are assigned to a queue-set. All traffic exiting the switch on a standard port flows through one of these four queues and is subjected to a threshold based on the QoS label assigned to the packet. Traffic destined for an ES port passes through the queue-set before reaching the hierarchical queues. If congestion occurs in the hierarchical queues that backs up to the queue-sets, the queue-set configuration controls how traffic is dropped.

Figure 26-9 shows the egress queue-set buffer. The buffer space is divided between the common pool and the reserved pool. The switch uses a buffer allocation scheme to reserve a minimum amount of buffers for each egress queue, to prevent any queue or port from consuming all the buffers and depriving other queues, and to control whether to grant buffer space to a requesting queue. The switch detects whether or not the target queue has consumed more buffers than its reserved amount (under-limit), whether it has consumed all of its maximum buffers (over limit), and whether the common pool is empty (no free buffers) or not empty (free buffers). If the queue is not over-limit, the switch can allocate buffer space from the reserved pool or from the common pool (if it is not empty). If there are no free buffers in the common pool or if the queue is over-limit, the switch drops the frame.

Figure 26-9 Egress Queue-Set Buffer Allocation



Buffer and Memory Allocation

You guarantee the availability of buffers, set drop thresholds, and configure the maximum memory allocation for a queue-set by using the **mls qos queue-set output** *qset-id* **threshold** *queue-id drop-threshold1 drop-threshold2 reserved-threshold maximum-threshold* global configuration command. Each threshold value is a percentage of the queue's allocated memory, which you specify by using the **mls qos queue-set output** *qset-id* **buffers** *allocation1 ... allocation4* global configuration command. The sum of all the allocated buffers represents the reserved pool, and the remaining buffers are part of the common pool.

Through buffer allocation, you can ensure that high-priority traffic is buffered. For example, if the buffer space is 400, you can allocate 70 percent of it to queue 1 and 10 percent to queues 2 through 4. Queue 1 then has 280 buffers allocated to it, and queues 2 through 4 each have 40 buffers allocated to them.

You can guarantee that the allocated buffers are reserved for a specific queue in a queue-set. For example, if there are 100 buffers for a queue, you can reserve 50 percent (50 buffers). The switch returns the remaining 50 buffers to the common pool. You also can enable a queue in the full condition to obtain more buffers than are reserved for it by setting a maximum threshold. The switch can allocate the needed buffers from the common pool if the common pool is not empty.

WTD Thresholds

You can assign each packet that flows through the switch to a queue and to a threshold. Specifically, you map DSCP or CoS values to an egress queue-set and map DSCP or CoS values to a threshold ID. You use the **mls qos srr-queue output dscp-map queue** *queue-id* {*dscp1...dscp8* | **threshold** *threshold-id dscp1...dscp8*} or the **mls qos srr-queue output cos-map queue** *queue-id* {*cos1...cos8* | **threshold** *threshold-id cos1...cos8*} global configuration command. You can display the DSCP output queue threshold map and the CoS output queue threshold map by using the **show mls qos maps** privileged EXEC command.

The queue-set uses WTD to support distinct drop percentages for different traffic classes. Each queue has three drop thresholds: two configurable (*explicit*) WTD thresholds and one nonconfigurable (*implicit*) threshold preset to the queue-full state. You assign the two WTD threshold percentages for threshold ID 1 and ID 2. The drop threshold for threshold ID 3 is preset to the queue-full state, and you cannot modify it. For more information about how WTD works, see the "Weighted Tail Drop" section on page 26-13.

Shaped or Shared Mode

SRR services each queue-set in shared or shaped mode.You map a port to a queue-set by using the **queue-set** *qset-id* interface configuration command. You assign shared or shaped weights to a standard port by using the **srr-queue bandwidth share** *weight1 weight2 weight3 weight4* or the **srr-queue bandwidth shape** *weight1 weight2 weight3 weight4* interface configuration command. You can assign only shared weights to an ES port. For an explanation of the differences between shaping and sharing, see the "SRR Shaping and Sharing" section on page 26-14.

The buffer allocation together with the SRR weight ratios control how much data can be buffered and sent before packets are dropped. The weight ratio is the ratio of the frequency in which the SRR scheduler sends packets from each queue.

You can combine the commands described in this section to prioritize traffic by placing packets with particular DSCP or CoS values into certain queues, by allocating a large queue size or by servicing the queue more frequently, and by adjusting queue thresholds so that packets with lower priorities are dropped. For configuration information, see the "Configuring Egress Queue-Set Characteristics" section on page 26-69.

Note The egress queue-set default settings are suitable for most situations. You should change them only when you have a thorough understanding of the queues and only if these settings do not meet your QoS solution.

Understanding Hierarchical QoS

The switch supports a hierarchical QoS configuration (traffic classification, CBWFQ, LLQ, shaping, and two-rate three-color policing) that is applied to the output of the ES ports.

Hierarchical QoS configuration is based on the concept of a bandwidth-limited stream of traffic, which is a stream of packets that has its departure rate constrained in some manner. At each level of the hierarchy, the switch must classify each packet to select into which traffic stream in that level the packet belongs. When the stream is classified, if its arrival rate exceeds its departure rate, queueing can become congested. To compensate for this, you can configure policies that contain policer drops, configure tail drop or WRED, a congestion-avoidance technique, or to influence whether the packet is queued. You also can implement scheduling policies (CBWFQ, LLQ, and shaping) to influence how quickly a packet is sent out the port.

Hierarchical Levels

Hierarchical QoS configuration involves traffic classification, policing, queueing, and scheduling. You can create a hierarchy by associating a class-level policy-map with a VLAN-level policy-map, by associating that VLAN-level policy-map with a physical-level policy-map, and by attaching the physical-level policy-map to the output of an ES port. You can skip hierarchical levels, but the order of the levels (class level, VLAN level, and then the physical level) must be preserved.

You can configure these three QoS levels in the hierarchy:

- Class level—You configure this level of the hierarchy by matching CoS, DSCP, IP precedence, or MPLS EXP bits in the outbound packet through the **match** {**cos** *cos-list* | **ip dscp** *dscp-list* | **ip precedence** *ip-precedence-list* | **mpls experimental** *exp-list*} class-map configuration command. At the class level, you can:
  - Configure policer drops by using the **police cir** or **police cir percent** policy-map class configuration command.
  - Configure tail drop or WRED drop policies by using the **queue-limit** or the **random-detect** policy-map class configuration command.
  - Modify the traffic class by setting Layer 2 and Layer 3 QoS fields through the **set** {**cos** *new-cos* | **ip** {**dscp** *new-dscp* | **precedence** *new-precedence*} | **mpls experimental** *exp-number*} policy-map class configuration command.
  - Configure CBWFQ or LLQ scheduling by using the **bandwidth** or the **priority** policy-map class configuration command.
  - Configure egress traffic shaping by using the **shape** policy-map class configuration command.

  The switch supports eight classes (including the default class) per policy map at this level. The default class is reserved for packets that do not meet any of the matching criteria.

  This is an example of a class-level classification and its naming convention:

```
Switch(config)# class-map match-any class-level-class-map-name
Switch(config-cmap)# match ip dscp 10 11 12
```

  This is an example of a class-level policy-map and its naming convention:

```
Switch(config)# policy-map class-level-policy-map-name
Switch(config-pmap)# class class-level-class-name
Switch(config-pmap-c)# bandwidth percent 20
Switch(config-pmap-c)# shape average 20000000
```

  This is a class-level configuration example that combines a class-level classification and a class-level policy-map to create a service policy:

```
Switch(config)# class-map c1
Switch(config-cmap)# match ip precedence 4
Switch(config-cmap)# exit
Switch(config)# policy-map policy1
Switch(config-pmap)# class c1
Switch(config-pmap-c)# police cir 500000 bc 10000 pir 1000000 be 10000 conform-action
transmit exceed-action set-prec-transmit 2 violate-action drop
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1/1
Switch(config-if)# service-policy output policy1
```

- VLAN level—You start configuration of per-VLAN QoS by entering the **match vlan** *vlan-id* class-map configuration command on one or more VLANs or by entering the **match vlan** *vlan-id* and the **match vlan inner** *vlan-id* class-map configuration command on one or more 802.1Q tunnels. At this level, you can configure the VLANs or 802.1Q tunnels to police, to share the available port bandwidth and to enable CBWFQ, and to shape the traffic. You configure these features by using the **police cir**, **police cir percent**, **bandwidth**, and **shape** policy-map class configuration commands.

  For a finer level of control, you also can associate a previously defined child policy at the class level with a new service policy by using the **service-policy** policy-map class configuration command. In the class-level child policy, you can configure tail drop or WRED drop policies, set Layer 2 and Layer 3 QoS fields, or enable the priority queue. These features are available only at the class level. By using a child policy, you apply a class-level policy only to traffic that matches the VLAN class.

  You cannot mix VLAN-level and class-level matches within a class map.

  You can attach up to 2045 user-created VLAN-level classes. This means that you can have 1022 unique classes and can associate them with the two ES ports (and have one left over), or you can add more classes to one ES port and can subtract from the other one. You can shape every class that you configure. You can create up to 4093 class maps.

  This is an example of a VLAN-level classification and its naming convention:

```
Switch(config)# class-map match-all vlan-level-class-map-name
Switch(config-cmap)# match vlan 5
Switch(config-cmap)# match vlan inner 3 - 8
```

  This is an example of a VLAN-level policy-map and its naming convention:

```
Switch(config)# policy-map vlan-level-policy-map-name
Switch(config-pmap)# class vlan-level-class-name
Switch(config-pmap-c)# police cir 500000 bc 10000 pir 1000000 be 10000 conform-action
transmit exceed-action set-prec-transmit 2 violate-action drop
```

  This is an example of a VLAN-level policy-map and its naming convention when a previously defined child policy is associated at the class level:

```
Switch(config)# policy-map vlan-level-policy-map-name
Switch(config-pmap)# class vlan-level-class-name
Switch(config-pmap-c)# bandwidth percent 30
Switch(config-pmap-c)# service-policy class-level-policy-map-name
```

  This is a VLAN-level configuration example that combines a VLAN-level classification and a VLAN-level policy-map:

```
Switch(config)# class-map match-all vlan203
Switch(config-cmap)# match vlan 203
Switch(config-cmap)# exit
Switch(config)# policy-map vlan-policy
Switch(config-pmap)# class vlan203
Switch(config-pmap-c)# police cir 500000 bc 10000 pir 1000000 be 10000 conform-action
transmit exceed-action set-prec-transmit 2 violate-action drop
```

This is an example of a VLAN-level policy-map that combines a VLAN-level classification with a VLAN-level policy-map and associates a previously defined child policy at the class level:

```
Switch(config)# class-map cls-class
Switch(config-cmap)# match mpls experimental 2
Switch(config-cmap)# exit
Switch(config)# class-map log-class
Switch(config-cmap)# match vlan 203
Switch(config-cmap)# exit
Switch(config)# policy-map cls-policy
Switch(config-pmap)# class cls-class
Switch(config-pmap-c)# set mpls experimental 5
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# policy-map log-policy
Switch(config-pmap)# class log-class
Switch(config-pmap-c)# service-policy cls-policy
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1/2
Switch(config-if)# switchport trunk encapsulation isl
Switch(config-if)# switchport mode trunk
Switch(config-if)# service-policy output log-policy
```

- Physical level—You can shape or police only the class-default class at the physical level of the hierarchy by using the **shape**, **police cir**, or **police cir percent** policy-map class configuration command.

  Within a policy-map, the class-default applies to all traffic that is not explicitly matched within the policy map but does match the parent policy. If no parent policy is configured, the parent policy represents the physical port. In a physical-level policy-map, class-default is the only class that you can configure.

  You use the **service-policy output** interface configuration command to attach an egress policy to an ES port.

  This is an example of a physical-level configuration. All hierarchical levels exist, and the order is preserved. The class level at the bottom, the VLAN level in the middle, and the physical level at the top.

```
Switch(config)# class-map my-class
Switch(config-cmap)# match ip precedence 1
Switch(config-cmap)# exit
Switch(config)# class-map my-logical-class
Switch(config-cmap)# match vlan 5
Switch(config-cmap)# exit
Switch(config)# policy-map my-class-policy
Switch(config-pmap)# class my-class
Switch(config-pmap-c)# set ip precedence 2
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# policy-map my-logical-policy
Switch(config-pmap)# class my-logical-class
Switch(config-pmap-c)# shape average 400000000
Switch(config-pmap-c)# service-policy my-class-policy
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# policy-map my-physical-policy
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# shape average 500000000
Switch(config-pmap-c)# service-policy my-logical-policy
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1/1
Switch(config-if)# service-policy output my-physical-policy
```

Egress Classification Based on Traffic Classes and Traffic Policies

Egress classification is the process of distinguishing one kind of traffic from another by examining the fields in the packet, processing the **match** commands, and creating the egress queues before the switch sends the packet out the ES port.

You use the class map to define a specific traffic flow (or class) and to isolate it from all other traffic. The class map defines the criteria used to match against a specific traffic flow to further classify it. At the class level, the criteria can include matching CoS, DSCP, IP precedence, or MPLS EXP bits in the header. At the VLAN level, the criteria can include matching a packet based on the inner and the outer VLAN IDs. 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.

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 criterion for the traffic by using the **match** class-map configuration command. Outbound packets are checked against the match criteria configured for a class map. If a packet matches the specified criteria, the packet is considered a member of the class, the switch creates a queue for it, and the packet is forwarded according to the QoS specifications set in the traffic policy. If a packet fails to meet any of the matching criteria, it is classified as a member of the default traffic class if one is configured.

You use the policy map to create the traffic policy, to specify the traffic class to act on, and to configure the QoS features associated with the traffic class. Actions can include trusting the received CoS, DSCP, or IP precedence bits in the traffic class; setting specific CoS, DSCP, IP precedence, or MPLS EXP bits in the traffic class; or specifying the traffic bandwidth limitations and the action to take when the traffic is out of profile.

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 use the **class** policy-map configuration command to name the traffic class associated with the traffic policy. If you specify **class-default** as the class name in the **class** policy-map configuration command, packets that fail to meet any of the matching criteria are classified as members of the default traffic class. You can manipulate this class (for example, police it and mark it) just like any traffic class, but you cannot delete it.

Within a policy-map, the class-default designates all traffic that is not explicitly matched within the policy-map but does match the policy map of the parent policy. If no parent policy is configured, the parent policy represents the physical port. In the physical-level policy-map, class-default is the only class that can be configured.

After you name the traffic class with the **class** command, the switch enters policy-map class configuration mode, and you can specify the actions to take on this traffic class.

You attach an egress policy-map to an ES port by using the **service-policy output** interface configuration command. The egress policy-map can include the **bandwidth**, **police cir**, **police cir percent**, **priority**, **queue-limit**, **random-detect**, **shape**, or **set** policy-map class configuration commands. If the policy map contains the class-default class, you can configure settings only through the **police cir**, **police cir percent**, and **shape** commands.

For more information, see the "Egress Policing and Marking" section on page 26-24 and the "Queueing and Scheduling of Hierarchical Queues" section on page 26-26.

Egress Policing and Marking

Egress traffic policing controls the maximum rate of traffic sent on a port. A policer defines the bandwidth limitations of the traffic and the action to take if the limits are exceeded. It is often configured on ports at the edge of a network to limit traffic into or out of the network. In most policing configurations, traffic that falls within the rate parameters is sent. Traffic that exceeds the parameters is considered to be *out of profile* or *nonconforming* and is dropped or sent with a different priority.

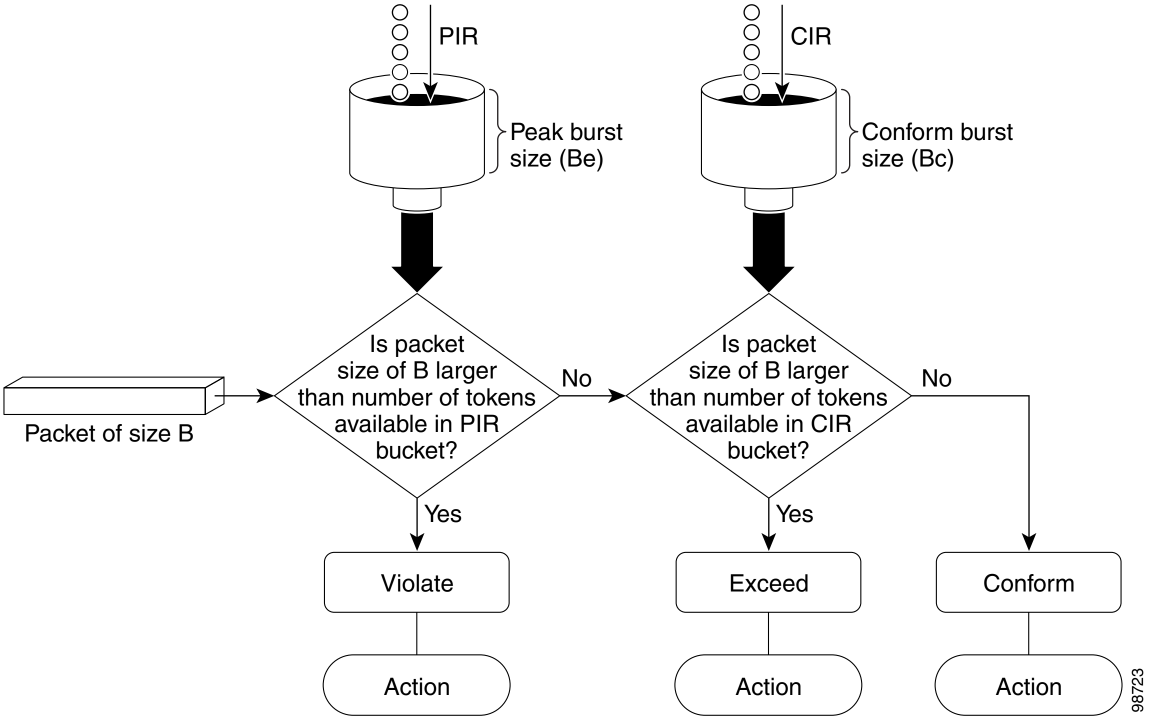
You can configure a two-rate traffic policer within a policy map at the class level, at the VLAN level, and at the physical level by using the **police cir** or the **police cir percent** policy-map class configuration command. At the physical level of the hierarchy, you can police only the class-default class in an egress policy attached to an ES port.

You can configure a two-rate traffic policer to limit the transmission rate of a traffic class and mark actions (conform, exceed, and violate) for each packet. Within the conform, exceed, and violate categories, you decide packet treatments. In the most common configurations, you configure packets that conform to be sent, packets that exceed to be sent with a decreased priority, and packets that violate to be dropped. You can decrease the priority of the CoS, the DSCP, the IP precedence, or the MPLS EXP bits in the packet.

The two-rate policer manages the maximum rate of traffic through a token-bucket algorithm. The algorithm uses the configured committed information rate (CIR) and the peak information rate (PIR) rate values to control the maximum rate of traffic allowed on a port at a given moment in time. The algorithm is affected by all traffic leaving the port, and it manages network bandwidth when several large packets are sent in the same traffic stream.

A token bucket is provided for the CIR and the PIR as shown in Figure 26-10.

Figure 26-10 Egress Two-Rate Policing and Marking Flowchart


PIR
CIR
Peak burst size (Be)
Conform burst size (Bc)
Packet of size B
Is packet size of B larger than number of tokens available in PIR bucket?
No
Is packet size of B larger than number of tokens available in CIR bucket?
No
Yes
Yes
Violate
Exceed
Conform
Action
Action
Action
98723

You configure the CIR and PIR rates in bps (or as a percentage of the bandwidth available on an ES port), and these rates control how fast the bucket fills (is updated) with tokens. The conform burst size (bc) and the peak burst size (be) represent the depth of the CIR and PIR buckets in bytes. This depth limits the number of tokens that the bucket can accumulate. If the bucket fills to capacity, newly arriving tokens are discarded.

Each token is permission for the source to send a certain number of bits into the network. To send a packet, the number of tokens equal to the packet size must be drained from the bucket. If there are enough tokens in the bucket, the packet conforms and can pass to the next stage. Otherwise, the exceed action associated with the bucket is applied to the packet. The packet might be dropped, or its priority value might be marked down.

In this token-bucket example, if the CIR rate is 2 kbps, 2000 tokens are added to the bucket every second (for this example, consider each token to represent a single bit of information). If a 1500-byte packet arrives, 12000 tokens (1500 bytes x 8 bits per byte) must be in the bucket for the packet to pass to the next state without triggering the exceed action. If enough tokens are in the bucket, they are drained, and the packet conforms and passes to the next stage. If there are less than 12000 tokens in the bucket, the exceed action is applied to the packet. The deeper the bucket, the more data can burst through at a rate greater than the rate at which the bucket is filling. For example, if the CIR bucket holds 6000 tokens, 750 bytes of traffic can instantaneously burst without draining the bucket (and without triggering an exceed action), even though the instantaneous burst is at a greater rate than the CIR rate of 2000 bps.

If the burst sizes approach the system maximum transmission unit (MTU), the policer strictly enforces the CIR and PIR. Normal traffic jitter can cause some percentage of inbound traffic to be flagged as nonconforming even if the average inbound rate appears to conform. If the burst size is very large, on the other hand, large traffic bursts at nonconforming data rates can be passed through the policer and flagged as conforming. Setting the burst sizes too low can result in less traffic than expected, and setting them too high can result in more traffic than expected.

For packet marking actions, if the CIR is 100 kbps, the PIR is 200 kbps, and a data stream with a rate of 250 kbps arrives at the two-rate policer:

- 100 kbps is marked as conforming to the rate.
- 100 kbps is marked as exceeding the rate.
- 50 kbps is marked as violating the rate.

If you set the CIR equal to the PIR, a traffic rate that is less than the CIR or that meets the CIR is in the conform range. Traffic that exceeds the CIR rate is in the violate range.

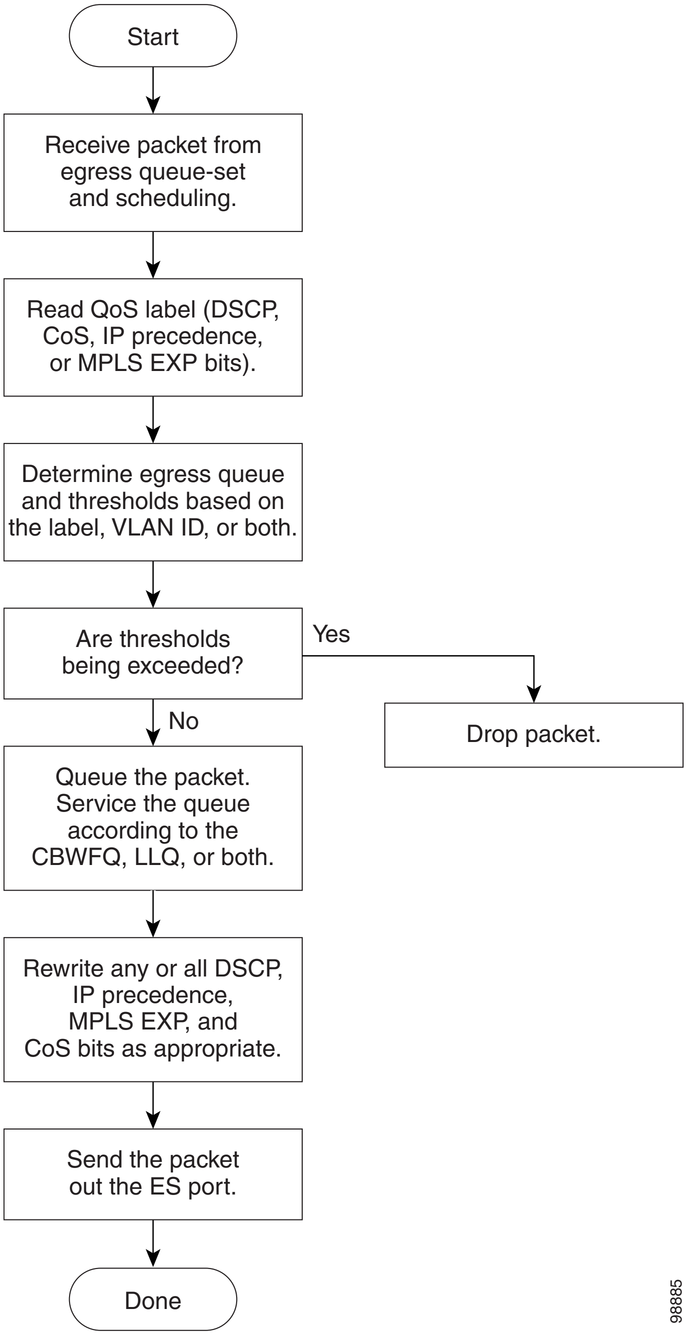
If you set the PIR greater than the CIR, a traffic rate less than the CIR is in the conform range. A traffic rate that exceeds the CIR but is less than or equal to the PIR is in the exceed range. A traffic rate that exceeds the PIR is in the violate range.

After you configure the policy map and policing actions, attach the egress policy to an ES port by using the **service-policy output** interface configuration command. For configuration information, see the "Configuring Hierarchical QoS" section on page 26-76.

Queueing and Scheduling of Hierarchical Queues

Figure 26-11 shows the egress queueing and scheduling flowchart for a hierarchical queue.

Figure 26-11 Egress Queueing and Scheduling Flowchart for a Hierarchical Queue


Start
Receive packet from egress queue-set and scheduling.
Read QoS label (DSCP, CoS, IP precedence, or MPLS EXP bits).
Determine egress queue and thresholds based on the label, VLAN ID, or both.
Are thresholds being exceeded?
Yes
Drop packet.
No
Queue the packet. Service the queue according to the CBWFQ, LLQ, or both.
Rewrite any or all DSCP, IP precedence, MPLS EXP, and CoS bits as appropriate.
Send the packet out the ES port.
Done
98885

Hierarchical Queues

The switch uses a hierarchical queueing model for traffic sent from an ES port. Each packet is assigned a queue based on its physical interface, VLAN, or class:

- At the class level, a packet is queued to one of four queues according to its CoS, DSCP, IP precedence, or MPLS EXP classification. Packets can be classified by any combination of these values, but if a packet matches more than one, the classification occurs in the order listed. The last queue in each set of four queues is reserved as the default queue. Packets that are not classified into one of the other three queues are assigned to the default queue. You can configure traffic in the default queue with congestion-avoidance features and scheduling congestion-management features as described in the "Queueing and Scheduling of Hierarchical Queues" section on page 26-26.
- At the VLAN level, the switch supports 2045 VLAN classes divided between the two ES ports. One queue is reserved as the default queue. Packets that are not classified into one of the other VLAN queues are assigned to the default queue. You can configure traffic in the default queue with congestion-avoidance features and scheduling congestion-management features as described in the "Queueing and Scheduling of Hierarchical Queues" section on page 26-26.
- At the physical level, the switch reserves one queue per port.

The switch creates the default queue and uses it to send all traffic when an egress service policy is not attached to an ES port. User traffic on a physical port without an attached service policy bypasses the QoS classification and is queued to the default queue. The minimum and maximum bandwidth for the default queue is the same as the port bandwidth.

Under congested conditions, the switch discards packets for all classes configured for the same sending queue with equal probability. To achieve the full queueing capacity, there must be an equal division of traffic among the classes for each sending queue.

Congestion-Management and Congestion-Avoidance Features

You use congestion-management features to control congestion and to control the order in which outbound packets are sent from an ES port based on the priorities assigned to those packets. You manage congestion by creating queues, by assigning packets based on the packet classification, and by scheduling the packets to be sent from the queue.

During periods with light traffic (when no congestion exists), the switch sends packets as soon as they arrive. During periods of congestion at the outbound port, packets arrive faster than the port can send them. If you use congestion-management features, the switch queues accumulating packets at a port until it is free to send them. They are then scheduled for transmission according to their assigned priorities and the queueing mechanism for the port.

You can configure either tail drop or WRED. You cannot configure both tail drop and WRED in the same class policy, but they can be used in two different class policies in the same policy map.

You can configure CBWFQ as a queue scheduling management feature, LLQ as a scheduling congestion-management feature, and traffic shaping to decrease the burstiness of traffic.

Tail Drop

With tail drop, packets are queued for the class until the maximum threshold is exceeded, and then all the packets destined for the class queue are dropped. You enable tail drop at the class level by using the **queue-limit** policy-map class configuration command. For configuration information, see the "Configuring an Egress Hierarchical QoS Policy" section on page 26-77.

WRED

Cisco Systems implements a version of Random Early Detection (RED), called WRED, differently from other congestion-avoidance techniques. WRED attempts to anticipate and avoid congestion, rather than controlling congestion when it occurs. WRED takes advantage of the TCP congestion control to try to control the average queue size by signaling end hosts when they should temporarily stop sending packets. By randomly dropping packets before periods of high congestion, it tells the packet source to decrease its sending rate. Assuming the packet source is using TCP, WRED tells it to decrease its sending rate until all the packets reach their destination, meaning that the congestion is cleared. By dropping some packets early rather than waiting until the queue is full, WRED avoids dropping large numbers of packets at once.

When a packet arrives and WRED is enabled, these events occur:

- The average queue size is calculated based on the previous average and the current size of the queue. The average queue-size calculation is affected by the exponential weight constant setting in the **random-detect exponential-weight-constant** policy-map class configuration command.
- If the average queue size is less than the minimum queue threshold, the arriving packet is queued. The minimum queue threshold is configured through the *min-threshold* option in the **random-detect** {**dscp** | **precedence**} policy-map class configuration command.
- If the average queue size is between the minimum queue threshold and the maximum queue threshold, the packet is either dropped or queued, depending on the packet-drop probability. The packet-drop probability is based on the minimum threshold, the maximum threshold, and the mark-probability denominator. The maximum queue threshold is configured through the *max-threshold* option, and the mark-probability denominator is configured through the *mark-prob-denominator* option in the **random-detect dscp** {**dscp** | **precedence**} policy-map class configuration command.
- If the average queue size is greater than the maximum queue threshold, the packet is automatically dropped.

You enable WRED by using the **random-detect** policy-map class configuration command at the class level. This command allows for preferential drop treatment among packets with different IP precedence or DSCP values. The WRED algorithm discards or marks packets destined for a queue when that queue is congested. It discards packets fairly and before the queue is full. Packets with high IP-precedence values are preferred over packets with low IP-precedence values. For configuration information, see the "Configuring an Egress Hierarchical QoS Policy" section on page 26-77.

CBWFQ

CBWFQ provides guaranteed bandwidth to particular traffic classes, such as voice, that are delay sensitive, while still fairly serving all other traffic in the network. You define traffic classes based on match criteria. Packets satisfying the match criteria for a class constitute the traffic for that class. A queue is reserved for each class, and traffic belonging to a class is directed to the queue for that class.

The bandwidth assigned to a class is the minimum bandwidth that is delivered to the class during congestion. CBWFQ uses the bandwidth weight to ensure that the queue for the class is serviced fairly.

You enable CBWFQ and specify the minimum bandwidth as a rate in kbps or as a percentage of the available bandwidth by using the **bandwidth** policy-map class configuration command at the class level or at the VLAN level. During periods of congestion, the classes are serviced in proportion to their configured bandwidth. The amount of bandwidth available to a class is dependent on the amount of bandwidth reserved by the parent class. For configuration information, see the "Configuring an Egress Hierarchical QoS Policy" section on page 26-77.

LLQ

LLQ provides strict-priority queueing for a traffic class. It enables delay-sensitive data, such as voice, to be sent before packets in other queues are sent. The priority queue is serviced first until it is empty. Only one traffic stream can be destined for the priority queue per class-level policy. The priority queue restricts all traffic streams in the same hierarchy, and you should use care when configuring this feature. You enable the priority queue for a traffic class by using the **priority** policy-map class configuration command at the class level. For configuration information, see the "Configuring an Egress Hierarchical QoS Policy" section on page 26-77.

Shaping

Shaping provides a process for delaying out-of-profile packets in queues so that they conform to a specified profile. Shaping is distinct from policing. Policing drops packets that exceed a configured threshold, but shaping buffers packets so that traffic remains within a threshold. Shaping offers greater smoothness in handling traffic than policing. You enable average-rate traffic shaping on a traffic class by using the **shape** policy-map class configuration command at the class level or at the VLAN level. At the physical level of the hierarchy, you can shape only the class-default class by using the **shape** policy-map class configuration command in an egress policy attached to an ES port. For configuration information, see the "Configuring an Egress Hierarchical QoS Policy" section on page 26-77.

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 ingress and 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 contents or size, and sends it from a single queue.)

When you enable auto-QoS, it automatically classifies traffic based on the traffic type and 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 26-30
- Effects of Auto-QoS on the Configuration, page 26-33
- Auto-QoS Configuration Guidelines, page 26-33
- Enabling Auto-QoS for VoIP, page 26-34
- Auto-QoS Configuration Example, page 26-35

Generated Auto-QoS Configuration

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

When auto-QoS is enabled, it uses the ingress packet label to categorize traffic, to assign packet labels, and to configure the ingress queues and egress queue-sets as shown in Table 26-2.

Table 26-2 Traffic Types, Ingress Packet Labels, Assigned Packet Labels, and Queues

	VoIP Data Traffic	VoIP Control Traffic	Routing Protocol Traffic	STP BPDU Traffic	All Other Traffic	
Ingress DSCP	46	26	–	–	–	
Ingress CoS	5	3	6	7	–	
DiffServ	EF	AF31	–	–	–	
Assigned DSCP	46	26	48	56	0	
Assigned CoS	5	3	6	7	0	
CoS-to-ingress queue map	2, 3, 4, 5, 6, 7 (queue 2)				0, 1 (queue 1)	
CoS-to-egress queue-set map	5 (queue 1)	3, 6, 7 (queue 2)			2, 4 (queue 3)	0, 1 (queue 4)

Table 26-3 shows the generated auto-QoS configuration for the ingress queues.

Table 26-3 Auto-QoS Configuration for the Ingress Queues

Ingress Queue	Queue Number	CoS-to-Queue Map	Queue Weight (Bandwidth)	Queue (Buffer) Size
SRR shared	1	0, 1	90 percent	90 percent
Priority	2	2, 3, 4, 5, 6, 7	10 percent	10 percent

Table 26-4 shows the generated auto-QoS configuration for the egress queue-set.

Table 26-4 Auto-QoS Configuration for the Egress Queue-Set

Egress Queue	Queue Number in the Queue-Set	CoS-to-Queue Map	Queue Weight (Bandwidth)	Queue (Buffer) Size
Priority (shaped)	1	5	10 percent	20 percent
SRR shared	2	3, 6, 7	10 percent	20 percent
SRR shared	3	2, 4	60 percent	20 percent
SRR shared	4	0, 1	20 percent	40 percent

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

- QoS is globally enabled (**mls qos** global configuration command), and other global configuration commands are added.
- When you enter the **auto qos voip cisco-phone** interface configuration command on a port at the edge of the network that is connected to a Cisco IP phone, the switch enables the trusted boundary feature. The switch 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 port is set to trust the QoS label received in the packet. When a Cisco IP phone is absent, the ingress classification is set to not trust the QoS label in the packet. The switch configures ingress queues and the egress queue-set on the port according to the settings in Table 26-3 and Table 26-4.
- When you enter the **auto qos voip trust** interface configuration command on a port connected to the interior of the network, the switch trusts the CoS value for nonrouted ports or the DSCP value for routed ports in ingress packets (the assumption is that traffic has already been classified by other edge devices). The switch configures the ingress queues and the egress queue-set on the port according to the settings in Table 26-3 and Table 26-4.

  For information about the trusted boundary feature, see the "Configuring a Trusted Boundary to Ensure Port Security" section on page 26-46.

When you enable auto-QoS by using the **auto qos voip cisco-phone** or the **auto qos voip trust** interface configuration command, the switch automatically generates a QoS configuration based on the traffic type and the ingress packet label and applies the commands listed in Table 26-5 to the port.

Note On an ES port, the **srr-queue bandwidth shape** interface configuration command is not part of the generated **auto qos voip** command list.

Table 26-5 Generated Auto-QoS Configuration

Description	Automatically Generated Command

The switch automatically enables standard QoS and configures the CoS-to-DSCP map (maps CoS values in inbound packets to a DSCP value).	`Switch(config)# mls qos`
	`Switch(config)# mls qos map cos-dscp 0 8 16 26 32 46 48 56`

The switch automatically maps CoS values to an ingress queue and to a threshold ID.	`Switch(config)# no mls qos srr-queue input cos-map`
	`Switch(config)# mls qos srr-queue input cos-map queue 1 threshold 3 0`
	`Switch(config)# mls qos srr-queue input cos-map queue 1 threshold 2 1`
	`Switch(config)# mls qos srr-queue input cos-map queue 2 threshold 1 2`
	`Switch(config)# mls qos srr-queue input cos-map queue 2 threshold 2 4 6 7`
	`Switch(config)# mls qos srr-queue input cos-map queue 2 threshold 3 3 5`

The switch automatically maps CoS values to an egress queue in the queue-set and to a threshold ID.	`Switch(config)# no mls qos srr-queue output cos-map`
	`Switch(config)# mls qos srr-queue output cos-map queue 1 threshold 3 5`
	`Switch(config)# mls qos srr-queue output cos-map queue 2 threshold 3 3 6 7`
	`Switch(config)# mls qos srr-queue output cos-map queue 3 threshold 3 2 4`
	`Switch(config)# mls qos srr-queue output cos-map queue 4 threshold 2 1`
	`Switch(config)# mls qos srr-queue output cos-map queue 4 threshold 3 0`

Table 26-5 Generated Auto-QoS Configuration (continued)

Description	Automatically Generated Command
The switch automatically maps DSCP values to an ingress queue and to a threshold ID.	Switch(config)# **no mls qos srr-queue input dscp-map** Switch(config)# **mls qos srr-queue input dscp-map queue 1 threshold 2 9 10 11 12 13 14 15** Switch(config)# **mls qos srr-queue input dscp-map queue 1 threshold 3 0 1 2 3 4 5 6 7** Switch(config)# **mls qos srr-queue input dscp-map queue 1 threshold 3 32** Switch(config)# **mls qos srr-queue input dscp-map queue 2 threshold 1 16 17 18 19 20 21 22 23** Switch(config)# **mls qos srr-queue input dscp-map queue 2 threshold 2 26 33 34 35 36 37 38 39** Switch(config)# **mls qos srr-queue input dscp-map queue 2 threshold 2 48 49 50 51 52 53 54 55** Switch(config)# **mls qos srr-queue input dscp-map queue 2 threshold 2 56 57 58 59 60 61 62 63** Switch(config)# **mls qos srr-queue input dscp-map queue 2 threshold 3 24 25 27 28 29 30 31 40** Switch(config)# **mls qos srr-queue input dscp-map queue 2 threshold 3 41 42 43 44 45 46 47**
The switch automatically maps DSCP values to an egress queue in the queue-set and to a threshold ID.	Switch(config)# **no mls qos srr-queue output dscp-map** Switch(config)# **mls qos srr-queue output dscp-map queue 1 threshold 3 40 41 42 43 44 45 46 47** Switch(config)# **mls qos srr-queue output dscp-map queue 2 threshold 3 24 25 27 28 29 30 31 48** Switch(config)# **mls qos srr-queue output dscp-map queue 2 threshold 3 49 50 51 52 53 54 55 56** Switch(config)# **mls qos srr-queue output dscp-map queue 2 threshold 3 57 58 59 60 61 62 63** Switch(config)# **mls qos srr-queue output dscp-map queue 3 threshold 3 16 17 18 19 20 21 22 23** Switch(config)# **mls qos srr-queue output dscp-map queue 3 threshold 3 26 32 33 34 35 36 37 38** Switch(config)# **mls qos srr-queue output dscp-map queue 3 threshold 3 39** Switch(config)# **mls qos srr-queue output dscp-map queue 4 threshold 1 8** Switch(config)# **mls qos srr-queue output dscp-map queue 4 threshold 2 9 10 11 12 13 14 15** Switch(config)# **mls qos srr-queue output dscp-map queue 4 threshold 3 0 1 2 3 4 5 6 7**
The switch automatically sets up the ingress queues, with queue 2 as the priority queue and queue 1 in shared mode. The switch also configures the bandwidth and buffer size for the ingress queues.	Switch(config)# **no mls qos srr-queue input priority-queue 1** Switch(config)# **no mls qos srr-queue input priority-queue 2** Switch(config)# m**ls qos srr-queue input bandwidth 90 10** Switch(config)# **no mls qos srr-queue input buffers**
The switch automatically configures the egress queue-set buffer sizes. It configures the bandwidth and the SRR mode (shaped or shared).	Switch(config)# **mls qos queue-set output 1 buffers 20 20 20 40** Switch(config-if)# **srr-queue bandwidth shape 10 0 0 0** Switch(config-if)# **srr-queue bandwidth share 10 10 60 20**

Table 26-5 Generated Auto-QoS Configuration (continued)

Description	Automatically Generated Command

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

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)# mls qos trust device cisco-phone`

Effects of Auto-QoS on the Configuration

When you enable auto-QoS, the switch adds the **auto qos voip** interface configuration command and the generated configuration to the running configuration.

The switch applies the auto-QoS-generated commands as if the commands were entered from the CLI. An existing user configuration can cause the application of the generated commands to fail, or the user configuration might be overridden by the generated commands. These actions occur without warning. If all the generated commands are successfully applied, any user-entered configuration that was not overridden remains in the running configuration. Any user-entered configuration that was overridden can be retrieved by reloading the switch without saving the current configuration to memory. If the generated commands are not applied, the previous running configuration is restored.

Auto-QoS 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, you should enable auto-QoS before you configure other 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. For more information, see the "Effects of Auto-QoS on the Configuration" section on page 26-33.
- You can enable auto-QoS on static, dynamic-access, voice VLAN access, and trunk ports.
- By default, the CDP is enabled on all ports. For auto-QoS to function properly, do not disable the CDP.
- Policing is not enabled with auto-QoS. You can manually enable policing, as described in the "Configuring an Ingress QoS Policy" section on page 26-48.

Enabling Auto-QoS for VoIP

Beginning in privileged EXEC mode, follow these steps to enable auto-QoS for VoIP within a QoS domain:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Specify the port that is connected to a Cisco IP phone or the uplink port that is connected to another switch or router in the interior of the network, and enter interface configuration mode.
Step 3	**auto qos voip** {**cisco-phone** | **trust**}	Enable auto-QoS. The keywords have these meanings: • **cisco-phone**—If the port is connected to a Cisco IP phone, the QoS labels of inbound packets are trusted only when the telephone is detected. • **trust**—The uplink port is connected to a trusted switch or router, and the VoIP traffic classification in the ingress packet is trusted.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show auto qos interface** *interface-id*	Verify your entries. This command displays the initial auto-QoS configuration that was applied; it does not display any user changes to the configuration that might be in effect. You can use the **show running-config** privileged EXEC command to display the auto-QoS configuration and the user modifications.

To display the QoS commands that are automatically generated when auto-QoS is enabled or disabled, enter the **debug autoqos** privileged EXEC command *before* enabling auto-QoS. For more information, refer to the “debug autoqos” command in the command reference for this release.

To disable auto-QoS on a port, use the **no auto qos voip** interface configuration command. Only the auto-QoS-generated interface configuration commands for this port are removed. If this is the last port on which auto-QoS is enabled and you enter the **no auto qos voip** command, auto-QoS is considered disabled even though the auto-QoS-generated global configuration commands remain (to avoid disrupting traffic on other ports affected by the global configuration). You can use the **no mls qos** global configuration command to disable the auto-QoS-generated global configuration commands. With QoS disabled, there is no concept of trusted or untrusted ports because the packets are not modified (the CoS, DSCP, and IP precedence values in the packet are not changed). Traffic is switched in pass-through mode (packets are switched without any rewrites and classified as best effort without any policing).

This example shows how to enable auto-QoS on a port and to trust the QoS labels received in inbound packets when the switch or router connected to a port is a trusted device:

```
Switch(config)# interface gigabitethernet1/0/1
Switch(config-if)# auto qos voip trust
```

Auto-QoS Configuration Example

This section describes how you could implement auto-QoS in a network, as shown in Figure 26-12.

Figure 26-12 Auto-QoS Configuration Example Network

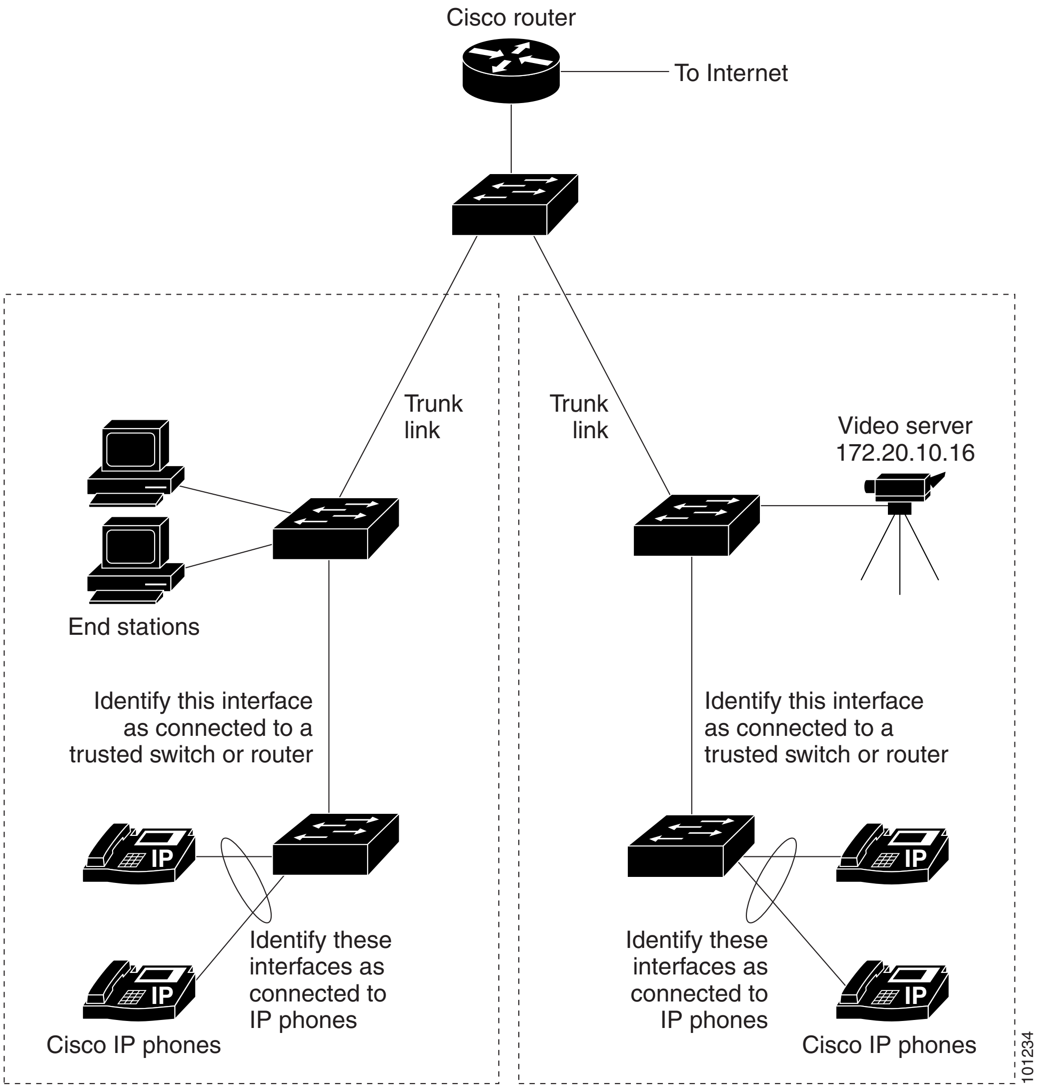

Cisco router
To Internet
Trunk link
Trunk link
Video server 172.20.10.16
End stations
Identify this interface as connected to a trusted switch or router
Identify this interface as connected to a trusted switch or router
IP
IP
IP
IP
Identify these interfaces as connected to IP phones
Identify these interfaces as connected to IP phones
Cisco IP phones
Cisco IP phones
101234

Figure 26-12 shows a network in which the VoIP traffic is prioritized over all other traffic. Auto-QoS is enabled on the switches in the wiring closets at the edge of the QoS domains.

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.

Beginning in privileged EXEC mode, follow these steps to configure the switch at the edge of the QoS domain to prioritize the VoIP traffic over all other traffic:

	Command	Purpose
Step 1	**debug autoqos**	Enable 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	**configure terminal**	Enter global configuration mode.
Step 3	**cdp enable**	Enable CDP globally. By default, it is enabled.
Step 4	**interface** *interface-id*	Specify the switch port connected to the Cisco IP phone, and enter interface configuration mode.
Step 5	**auto qos voip cisco-phone**	Enable auto-QoS on the port, and specify that the port is connected to a Cisco IP phone. The QoS labels of inbound packets are trusted only when the Cisco IP phone is detected.
Step 6	**exit**	Return to global configuration mode.
Step 7		Repeat Steps 4 to 6 for as many ports as are connected to the Cisco IP phone.
Step 8	**auto qos voip cisco-phone**	Enable auto-QoS on the port, and specify that the port is connected to a Cisco IP phone.
Step 9	**exit**	Return to global configuration mode.
Step 10	**interface** *interface-id*	Specify the port identified as connected to a trusted switch or router. See Figure 26-12. Enter interface configuration mode.
Step 11	**auto qos voip trust**	Enable auto-QoS on the port, and specify that the port is connected to a trusted router or switch.
Step 12	**end**	Return to privileged EXEC mode.
Step 13	**show auto qos**	Verify 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 26-12.
Step 14	**copy running-config startup-config**	Save the **auto qos voip** interface configuration commands and the generated auto-QoS configuration in the configuration file.

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 displays 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 mls qos**
- **show mls qos maps cos-dscp**
- **show mls qos interface** [*interface-id*] [**buffers** | **queueing**]
- **show mls qos maps** [**cos-dscp** | **cos-input-q** | **cos-output-q** | **dscp-cos** | **dscp-input-q** | **dscp-output-q**]
- **show mls qos input-queue**
- **show running-config**

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

Configuring Standard QoS

Before configuring standard 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 standard QoS on your switch:

- Default Standard QoS Configuration, page 26-38
- Standard QoS Configuration Guidelines, page 26-40
- Packet Modification, page 26-41
- Enabling QoS Globally, page 26-42 (required)
- Configuring Ingress Classification by Using Port Trust States, page 26-42 (required
- Configuring an Ingress QoS Policy, page 26-48 (required)
- Configuring DSCP Maps, page 26-59 (optional, unless you need to use the DSCP-to-DSCP-mutation map or the policed-DSCP map)
- Configuring Ingress Queue Characteristics, page 26-64 (optional)
- Configuring Egress Queue-Set Characteristics, page 26-69 (optional)

If you need to configure outbound traffic on an ES port, see the "Configuring Hierarchical QoS" section on page 26-76.

Default Standard QoS Configuration

QoS is disabled. There is no concept of trusted or untrusted ports because the packets are not modified (the CoS, DSCP, and IP precedence values in the packet are not changed). Traffic is switched in pass-through mode (packets are switched without any rewrites and classified as best effort without any policing).

When QoS is enabled with the **mls qos** global configuration command and all other QoS settings are at their defaults, traffic is classified as best effort (the DSCP and CoS value is set to 0) without any policing. No policy maps are configured. The default port trust state on all ports is untrusted. The default ingress queues and egress queue-set settings are described in the "Default Ingress Queue Configuration" section on page 26-38 and the "Default Egress Queue-Set Configuration" section on page 26-39. For outbound traffic on an ES port, see the "Default Hierarchical QoS Configuration" section on page 26-76.

Default Ingress Queue Configuration

Table 26-6 shows the default ingress queue configuration when QoS is enabled.

Table 26-6 Default Ingress Queue Configuration

Feature	Queue 1	Queue 2
Buffer allocation	90 percent	10 percent
Bandwidth allocation [1]	4	4
Priority queue bandwidth [2]	0	10
WTD drop threshold 1	100 percent	100 percent
WTD drop threshold 2	100 percent	100 percent

[1] The bandwidth is equally shared between the queues. SRR sends packets in shared mode only.

[2] Queue 2 is the priority queue. SRR services the priority queue for its configured share before servicing the other queue.

Table 26-7 shows the default CoS input queue threshold map when QoS is enabled.

Table 26-7 Default CoS Input Queue Threshold Map

CoS Value	0–4	5	6, 7
Queue ID - Threshold ID	1 - 1	2 - 1	1 - 1

Table 26-8 shows the default DSCP input queue threshold map when QoS is enabled.

Table 26-8 Default DSCP Input Queue Threshold Map

DSCP Value	0–39	40–47	48–63
Queue ID - Threshold ID	1 - 1	2 - 1	1 - 1

Default Egress Queue-Set Configuration

Table 26-9 shows the default egress queue-set configuration when QoS is enabled. All ports are mapped to queue-set 1. The port bandwidth limit is set to 100 percent, and the rate is unlimited.

Table 26-9 Default Egress Queue-Set Configuration

Feature	Queue 1	Queue 2	Queue 3	Queue 4
Buffer allocation	25 percent	25 percent	25 percent	25 percent
WTD drop threshold 1	100 percent	50 percent	100 percent	100 percent
WTD drop threshold 2	100 percent	50 percent	100 percent	100 percent
Reserved threshold	50 percent	100 percent	50 percent	50 percent
Maximum threshold	400 percent	400 percent	400 percent	400 percent
SRR shaped weights (absolute) [1]	25	0	0	0
SRR shared weights [2]	25	25	25	25

[1] A shaped weight of zero means that this queue is operating in shared mode.

[2] One quarter of the bandwidth is allocated to each queue.

Table 26-10 shows the default CoS output queue threshold map when QoS is enabled.

Table 26-10 Default CoS Output Queue Threshold Map

CoS value	0, 1	2, 3	4	5	6, 7
Queue ID - threshold ID	2 - 1	3 - 1	4 - 1	1 -1	4 - 1

Table 26-11 shows the default DSCP output queue threshold map when QoS is enabled.

Table 26-11 Default DSCP Output Queue Threshold Map

DSCP Value	0–15	16–31	32–39	40–47	48–63
Queue ID - Threshold ID	2 - 1	3 - 1	4 - 1	1 - 1	4 - 1

Default Mapping Table Configuration

The default CoS-to-DSCP map is shown in Table 26-12 on page 26-59.

The default IP-precedence-to-DSCP map is shown in Table 26-13 on page 26-60.

The default DSCP-to-CoS map is shown in Table 26-14 on page 26-62.

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

The default policed-DSCP map is a null map, which maps an inbound DSCP value to the same DSCP value (no markdown).

Standard QoS Configuration Guidelines

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

- It is not possible to match IP fragments against configured IP extended ACLs to enforce QoS. IP fragments are sent as best-effort. IP fragments are denoted by fields in the IP header.
- Only one ACL can be configured per class map. The ACL can have multiple ACEs, which match fields against the contents of the packet. Class maps that contain ACLs are supported only in an ingress policy-map.
- Only one policy map per port is supported. You can attach one ingress service-policy per port and one egress service-policy per ES port.
- Inbound traffic is classified, policed, and marked down (if configured) regardless of whether the traffic is bridged, routed, or sent to the CPU. Bridged frames can be dropped or have their DSCP and CoS values modified.
- Only one ingress policer is applied to a packet on a port. Only the average-rate and committed-burst parameters are configurable.
- You can create an aggregate policer that is shared by multiple traffic classes within the same policy map. However, you cannot use the aggregate policer across different policy maps.
- On a standard port and on the input of an ES port, the port ASIC device, which controls more than one physical port, supports 256 ingress policers (255 policers plus 1 **no** policer). The maximum number of policers supported per is 64. For example, you could configure 32 policers on a Gigabit Ethernet port and 8 policers on a Fast Ethernet port, or you could configure 64 policers on a Gigabit Ethernet port and 5 policers on a Fast Ethernet port. Policers are allocated on demand by the software and are constrained by the hardware and ASIC boundaries. You cannot reserve policers per port; there is no guarantee that a port will be assigned to any policer. These limitations do not apply to egress policers configured on the ES ports.
- On a port configured for QoS, all traffic received through the port is classified, policed, and marked according to the policy map attached to the port. On a trunk port configured for QoS, traffic in *all* VLANs received through the port is classified, policed, and marked according to the policy map attached to the port.
- Because the switch does not support attaching a service policy to a logical interface (such as an EtherChannel), if you have EtherChannel ports configured on your switch, you must configure QoS classification, policing, mapping, and queueing on the individual physical ports that comprise the EtherChannel.
- Control traffic (such as spanning-tree bridge protocol data units [BPDUs] and routing update packets) received by the switch are subject to all ingress QoS processing.
- You are likely to lose data when you change queue settings; therefore, try to make changes when traffic is at a minimum.

For outbound traffic on an ES port, see the "Hierarchical QoS Configuration Guidelines" section on page 26-76.

Packet Modification

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

- For IP and non-IP packets, ingress classification involves assigning a QoS label to a packet based on the DSCP or CoS of the received packet. However, the packet is not modified at this stage; only an indication of the assigned DSCP or CoS value is carried along. The reason for this is that QoS classification and forwarding lookups occur in parallel, and it is possible that the packet is forwarded with its original DSCP to the CPU where it is again processed through software.

  The ES ports classify traffic on egress, and this classification can be used for queuing or for marking the CoS, DSCP, IP precedence, or MPLS EXP bits. Any packet modifications that result from ingress classification are applied before the packet reaches the egress classification stage. For example, if the switch receives traffic with a CoS value of 2 and an ingress action resets the CoS to 4, the packet will have a CoS of 4 (instead of a CoS of 2 and an indicator that the CoS should be set to 4) when it moves to the egress classification stage.
- During ingress 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; for non-IP packets the DSCP is converted to CoS and used for queueing and scheduling decisions.

  During egress policing on the ES ports, marking actions can set the CoS, DSCP, IP precedence, or the MPLS EXP bits. Any markings performed by an ingress policer are applied before the packet reaches the egress classification stage.
- Depending on the QoS label assigned to a frame and the mutation chosen, the DSCP and CoS values of the frame are rewritten. If you do not configure the mutation map and if you configure the port to trust the DSCP of the inbound frame, the DSCP value in the frame is not changed, but the CoS is rewritten according to the DSCP-to-CoS map. If you configure the port to trust the CoS of the inbound frame and it is an IP packet, the CoS value in the frame is not changed, but the DSCP might be changed according to the CoS-to-DSCP map.

  The input mutation causes the DSCP to be rewritten depending on the new value of DSCP chosen. The set action in a policy map also causes the DSCP to be rewritten.

  This information applies to both standard and ES ports. On the ES ports, the switch also applies trust policies to 802.1Q tunneling frames at egress.

Enabling QoS Globally

By default, QoS is disabled on the switch.

Beginning in privileged EXEC mode, follow these steps to enable QoS. This procedure is required.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**mls qos**	Enable QoS globally. QoS runs from the default settings described in the "Default Standard QoS Configuration" section on page 26-38 and the "Default Hierarchical QoS Configuration" section on page 26-76.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show mls qos**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To disable QoS, use the **no mls qos** global configuration command.

Configuring Ingress Classification by Using Port Trust States

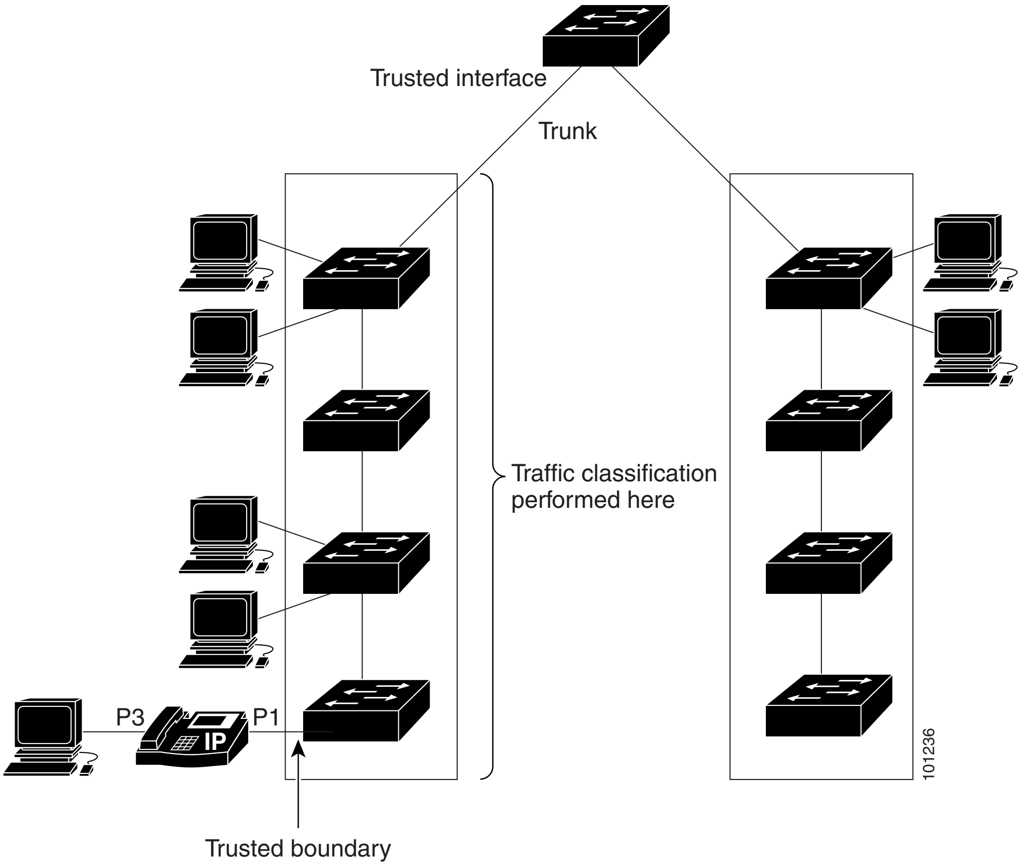
These sections describe how to classify inbound traffic through port trust states. Depending on your network configuration, you must perform one or more of these tasks or one or more of the tasks in the "Configuring an Ingress QoS Policy" section on page 26-48:

- Configuring the Trust State on Ports Within the QoS Domain, page 26-42
- Configuring the CoS Value for an Interface, page 26-45
- Configuring a Trusted Boundary to Ensure Port Security, page 26-46
- Configuring the DSCP Trust State on a Port Bordering Another QoS Domain, page 26-47

Configuring the Trust State on Ports Within the QoS Domain

Packets entering a QoS domain are classified at the edge of the QoS domain. The switch port within the QoS domain can then be configured to one of the trusted states because there is no need to classify the packets at every switch within the domain. Figure 26-13 shows a sample network topology.

Figure 26-13 Port Trusted States Within the QoS Domain



Beginning in privileged EXEC mode, follow these steps to configure the port to trust the classification of the traffic that it receives:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Specify the port to be trusted, and enter interface configuration mode. Valid interfaces include physical ports.

	Command	Purpose
Step 3	**mls qos trust** [**cos** | **dscp** | **ip-precedence**]	Configure the port trust state. By default, the port is not trusted. If no keyword is specified, the default is **dscp**. For 802.1Q tunnels, the switch processes inbound traffic on a standard port according to the trusted setting applied to this port. The switch configures the inner and outer tags for packets sent over the ES trunk port. The keywords have these meanings: • **cos**—Classifies an ingress packet by using the packet CoS value. For an untagged packet, the port default CoS value is used. The default port CoS value is 0. For 802.1Q tunnels, the switch copies the inner CoS value to the outer CoS value and sends the packet out an ES port. • **dscp**—Classifies an ingress packet by using the packet DSCP value if the packet is an IP packet. For a non-IP packet, the packet CoS value is used if the packet is tagged; for an untagged packet, the default port CoS is used. Internally, the switch maps the CoS value to a DSCP value by using the CoS-to-DSCP map. For 802.1Q tunnels, for a non-IP packet that is untagged, the switch configures the outer CoS value from the DSCP-to-CoS map, does not modify the inner CoS value, and sends the packet out an ES port. For an IP packet, the switch modifies the DSCP value in the packet if there is a DSCP-to-DSCP mutation map configured on the standard port. The switch uses the mutated DSCP value to configure the outer CoS value from the DSCP-to-CoS map and sends the packet out an ES port • **ip-precedence**—Classifies an ingress packet by using the packet IP-precedence value. For a non-IP packet, the packet CoS value is used if the packet is tagged; for an untagged packet, the default port CoS is used. Internally, the switch maps the CoS value to a DSCP value through the CoS-to-DSCP map. For 802.1Q tunnels, the switch converts the generated DSCP value from the DSCP-to-CoS map and uses it as the outer CoS value in the packet. The switch does not modify the inner CoS value in the packet and sends the packet out an ES port. **Note** When port trust policies are used with 802.1Q tunneling, all ports sharing the same tunnel VLAN must be configured with the same trust policy, and the ports involved must use the same DSCP-to-DSCP mutation map. For more information, see the "Configuring the DSCP-to-DSCP-Mutation Map" section on page 26-63.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show mls qos interface**	Verify your entries.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return a port to its untrusted state, use the **no mls qos trust** interface configuration command.

For information on how to change the default CoS value, see the "Configuring the CoS Value for an Interface" section on page 26-45. For information on how to configure the CoS-to-DSCP map, see the "Configuring the CoS-to-DSCP Map" section on page 26-59.

Configuring the CoS Value for an Interface

QoS assigns the CoS value specified with the **mls qos cos** interface configuration command to untagged frames received on trusted and untrusted ports.

Beginning in privileged EXEC mode, follow these steps to define the default CoS value of a port or to assign the default CoS to all inbound packets on the port:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Specify the port to be configured, and enter interface configuration mode. Valid interfaces include physical ports.
Step 3	**mls qos cos** {*default-cos* | **override**}	Configure the default CoS value for the port. For 802.1Q tunnels, the switch processes inbound traffic on a standard port according to the trusted setting applied to this port. The switch configures the inner and outer tags for packets sent over the ES trunk port. • For *default-cos*, specify a default CoS value to be assigned to a port. If the packet is untagged, the default CoS value becomes the packet CoS value. The CoS range is 0 to 7. The default is 0. For 802.1Q tunnels, the switch copies the inner CoS value to the outer CoS value and sends the packet out an ES port. • Use the **override** keyword to override the previously configured trust state of the inbound packet and to apply the default port CoS value to the port on all inbound packets. By default, CoS override is disabled. Use the **override** keyword when all inbound packets on specified ports deserve higher or lower priority than packets entering from other ports. Even if a port was previously set to trust DSCP, CoS, or IP precedence, this command overrides the previously configured trust state, and all the inbound CoS values are assigned the default CoS value configured with this command. If an inbound packet is tagged, the CoS value of the packet is modified with the default CoS of the port at the ingress port. For 802.1Q tunnels, the switch copies the port CoS value to the outer CoS value, does not modify the inner CoS value, and sends the packet out an ES port.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show mls qos interface**	Verify your entries.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return to the default setting, use the **no mls qos cos** {*default-cos* | **override**} interface configuration command.

Configuring a Trusted Boundary to Ensure Port Security

In a typical network, you connect a Cisco IP phone to a switch port, as shown in Figure 26-13 on page 26-43, and cascade devices that generate data packets from the back of the telephone. The Cisco IP phone guarantees the voice quality through a shared data link by marking the CoS level of the voice packets as high priority (CoS = 5) and by marking the data packets as low priority (CoS = 0). 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 CoS 3-bit field, which is the priority of the packet.

For most Cisco IP phone configurations, the traffic sent from the telephone to the switch should be trusted to ensure that voice traffic is properly prioritized over other types of traffic in the network. By using the **mls qos trust cos** interface configuration command, you configure the switch port to which the telephone is connected to trust the CoS labels of all traffic received on that port.

With the trusted setting, you also can use the trusted boundary feature to prevent misuse of a high-priority queue if a user bypasses the telephone and connects the PC directly to the switch. Without trusted boundary, the CoS labels generated by the PC are trusted by the switch (because of the trusted CoS setting). By contrast, trusted boundary uses 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. If the telephone is not detected, the trusted boundary feature disables the trusted setting on the switch port and prevents misuse of a high-priority queue. Note that the trusted boundary feature is not effective if the PC and Cisco IP phone are connected to a hub that is connected to the switch.

In some situations, you can prevent a PC connected to the IP phone from taking advantage of a high-priority data queue. 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.

Beginning in privileged EXEC mode, follow these steps to enable trusted boundary on a port:

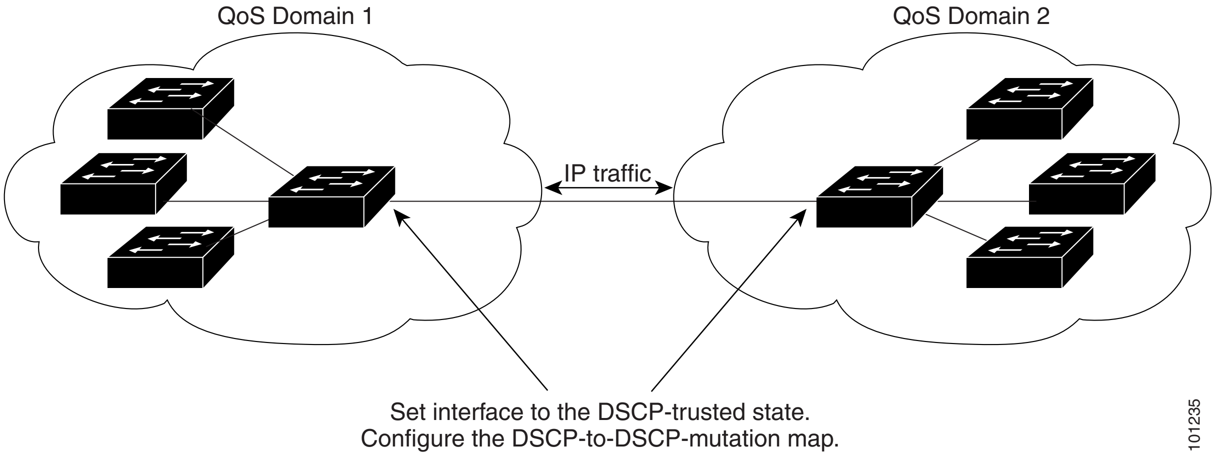
	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**cdp run**	Enable CDP globally. By default, CDP is enabled.
Step 3	**interface** *interface-id*	Specify the port connected to the IP phone, and enter interface configuration mode. Valid interfaces include physical ports.
Step 4	**cdp enable**	Enable CDP on the port. By default, CDP is enabled.
Step 5	**mls qos trust cos**	Configure the port to trust the CoS value in traffic received from the Cisco IP phone. By default, the port is not trusted.
Step 6	**mls qos trust device cisco-phone**	Specify that the Cisco IP phone is a trusted device. You cannot enable both trusted boundary and auto-QoS (**auto qos voip** interface configuration command) at the same time; they are mutually exclusive.
Step 7	**end**	Return to privileged EXEC mode.
Step 8	**show mls qos interface**	Verify your entries.
Step 9	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

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

Configuring the DSCP Trust State on a Port Bordering Another QoS Domain

If you are administering two separate QoS domains between which you want to implement QoS features for IP traffic, you can configure the switch ports bordering the domains to a DSCP-trusted state as shown in Figure 26-14. Then the receiving port accepts the DSCP-trusted value and avoids the classification stage of QoS. If the two domains use different DSCP values, you can configure the DSCP-to-DSCP-mutation map to translate a set of DSCP values to match the definition in the other domain.

Figure 26-14 DSCP-Trusted State on a Port Bordering Another QoS Domain


QoS Domain 1
QoS Domain 2
IP traffic
Set interface to the DSCP-trusted state.
Configure the DSCP-to-DSCP-mutation map.
101235

Beginning in privileged EXEC mode, follow these steps to configure the DSCP-trusted state on a port and modify the DSCP-to-DSCP-mutation map. To ensure a consistent mapping strategy across both QoS domains, you must perform this procedure on the ports in both domains:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**mls qos map dscp-mutation** *dscp-mutation-name in-dscp* **to** *out-dscp*	Modify the DSCP-to-DSCP-mutation map. The default DSCP-to-DSCP-mutation map is a null map, which maps an inbound DSCP value to the same DSCP value. • For *dscp-mutation-name*, enter the mutation map name. You can create more than one map by specifying a new name. • For *in-dscp*, enter up to eight DSCP values separated by spaces. Then enter the **to** keyword. • For *out-dscp*, enter a single DSCP value. The DSCP range is 0 to 63.
Step 3	**interface** *interface-id*	Specify the port to be trusted, and enter interface configuration mode. Valid interfaces include physical ports.
Step 4	**mls qos trust dscp**	Configure the ingress port as a DSCP-trusted port. By default, the port is not trusted.

	Command	Purpose
Step 5	**mls qos dscp-mutation** *dscp-mutation-name*	Apply the map to the specified ingress DSCP-trusted port. For *dscp-mutation-name*, specify the mutation map name created in Step 2. You can configure multiple DSCP-to-DSCP-mutation maps on an ingress port.
Step 6	**end**	Return to privileged EXEC mode.
Step 7	**show mls qos maps dscp-mutation**	Verify your entries.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return a port to its non-trusted state, use the **no mls qos trust** interface configuration command. To return to the default DSCP-to-DSCP-mutation map values, use the **no mls qos map dscp-mutation** *dscp-mutation-name* global configuration command.

This example shows how to configure a port to the DSCP-trusted state and to modify the DSCP-to-DSCP-mutation map (named *gi1/1/1-mutation*) so that inbound DSCP values 10 to 13 are mapped to DSCP 30:

```
Switch(config)# mls qos map dscp-mutation gi1/1/1-mutation 10 11 12 13 to 30
Switch(config)# interface gigabitethernet1/1/1
Switch(config-if)# mls qos trust dscp
Switch(config-if)# mls qos dscp-mutation gi1/1/1-mutation
Switch(config-if)# end
```

Configuring an Ingress QoS Policy

Configuring an ingress QoS policy typically requires classifying traffic into classes, configuring policies applied to those traffic classes, and attaching the policies to a port.

For background information, see the "Ingress Classification" section on page 26-6 and the "Ingress Policing and Marking" section on page 26-9. For configuration guidelines, see the "Standard QoS Configuration Guidelines" section on page 26-40.

These sections describe how to classify, police, and mark inbound traffic. Depending on your network configuration, you must perform one or more of these tasks:

- Classifying Ingress Traffic by Using ACLs, page 26-49
- Classifying Ingress Traffic by Using Class Maps, page 26-52
- Classifying, Policing, and Marking Ingress Traffic by Using Policy Maps, page 26-54
- Classifying, Policing, and Marking Ingress Traffic by Using Aggregate Policers, page 26-57

For information on configuring egress policies for the ES ports, see the "Configuring an Egress Hierarchical QoS Policy" section on page 26-77. This section describes how to classify egress traffic by using class maps, how to configure an egress two-rate traffic policer, how to configure class-based packet marking in an egress traffic policy, how to configure CBWFQ, tail drop, DSCP-based WRED, and IP precedence-based WRED, how to enable LLQ, and how to configure shaping.

Classifying Ingress Traffic by Using ACLs

You can classify ingress IP traffic by using IP standard or IP extended ACLs. You also can classify ingress non-IP traffic by using Layer 2 MAC ACLs.

Beginning in privileged EXEC mode, follow these steps to create an IP standard ACL for inbound IP traffic:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]	Create an IP standard ACL, repeating the command as many times as necessary. • For *access-list-number*, enter the access list number. The range is 1 to 99 and 1300 to 1999. • Use the **permit** keyword to permit a certain type of traffic if the conditions are matched. Use the **deny** keyword to deny a certain type of traffic if conditions are matched. • For *source*, enter the network or host from which the packet is being sent. You can use the **any** keyword as an abbreviation for 0.0.0.0 255.255.255.255. • (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. **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.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show access-lists**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To delete an access list, use the **no access-list** *access-list-number* global configuration command.

This example shows how to allow access for only those hosts on the three specified networks. The wildcard bits apply to the host portions of the network addresses. Any host with a source address that does not match the access list statements is rejected.

```
Switch(config)# access-list 1 permit 192.5.255.0 0.0.0.255
Switch(config)# access-list 1 permit 128.88.0.0 0.0.255.255
Switch(config)# access-list 1 permit 36.0.0.0 0.0.0.255
! (Note: all other access implicitly denied)
```

Beginning in privileged EXEC mode, follow these steps to create an IP extended ACL for inbound IP traffic:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard*	Create an IP extended ACL, repeating the command as many times as necessary. • For *access-list-number*, enter the access list number. The range is 100 to 199 and 2000 to 2699. • Use the **permit** keyword to permit a certain type of traffic if the conditions are matched. Use the **deny** keyword to deny a certain type of traffic if conditions are matched. • For *protocol*, enter the name or number of an IP protocol. Use the question mark (?) to see a list of available protocol keywords. • For *source*, enter the network or host from which the packet is being sent. You specify this by using dotted decimal notation, by using the **any** keyword as an abbreviation for *source* 0.0.0.0 *source-wildcard* 255.255.255.255, or by using the **host** keyword for *source* 0.0.0.0. • For *source-wildcard*, enter the wildcard bits by placing ones in the bit positions that you want to ignore. You specify the wildcard by using dotted decimal notation, by using the **any** keyword as an abbreviation for *source* 0.0.0.0 *source-wildcard* 255.255.255.255, or by using the **host** keyword for *source* 0.0.0.0. • For *destination*, enter the network or host to which the packet is being sent. You have the same options for specifying the *destination and destination-wildcard* as those described by *source* and *source-wildcard*. **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.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show access-lists**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To delete an access list, use the **no access-list** *access-list-number* global configuration command.

This example shows how to create an ACL that permits IP traffic from any source to any destination that has the DSCP value set to 32:

```
Switch(config)# access-list 100 permit ip any any dscp 32
```

This example shows how to create an ACL that permits IP traffic from a source host at 10.1.1.1 to a destination host at 10.1.1.2 with a precedence value of 5:

```
Switch(config)# access-list 100 permit ip host 10.1.1.1 host 10.1.1.2 precedence 5
```

This example shows how to create an ACL that permits PIM traffic from any source to a destination group address of 224.0.0.2 with a DSCP set to 32:

```
Switch(config)# access-list 102 permit pim any 224.0.0.2 dscp 32
```

Beginning in privileged EXEC mode, follow these steps to create a Layer 2 MAC ACL for inbound non-IP traffic:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**mac access-list extended** *name*	Create a Layer 2 MAC ACL by specifying the name of the list. After entering this command, the mode changes to extended MAC ACL configuration.
Step 3	{**permit** | **deny**} {**host** *src-MAC-addr mask* | **any** | **host** *dst-MAC-addr* | *dst-MAC-addr mask*} [*type mask*]	Specify the type of traffic to permit or deny if the conditions are matched, entering the command as many times as necessary. • For *src-MAC-addr*, enter the MAC address of the host from which the packet is being sent. You specify this by using the hexadecimal format (H.H.H), by using the **any** keyword as an abbreviation for *source* 0.0.0, *source-wildcard* 255.255.255, or by using the **host** keyword for *source* 0.0.0. • For *mask,* enter the wildcard bits by placing ones in the bit positions that you want to ignore. • For *dst-MAC-addr*, enter the MAC address of the host to which the packet is being sent. You specify this by using the hexadecimal format (H.H.H), by using the **any** keyword as an abbreviation for *source* 0.0.0, *source-wildcard* 255.255.255, or by using the **host** keyword for *source* 0.0.0. • (Optional) For *type mask*, specify the Ethertype number of a packet with Ethernet II or SNAP encapsulation to identify the protocol of the packet. For *type*, the range is from 0 to 65535, typically specified in hexadecimal. For *mask*, enter the *don't care* bits applied to the Ethertype before testing for a match. **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.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show access-lists** [*access-list-number* | *access-list-name*]	Verify your entries.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To delete an access list, use the **no mac access-list extended** *access-list-name* global configuration command.

This example shows how to create a Layer 2 MAC ACL with two permit statements. The first statement allows traffic from the host with MAC address 0001.0000.0001 to the host with MAC address 0002.0000.0001. The second statement allows only Ethertype XNS-IDP traffic from the host with MAC address 0001.0000.0002 to the host with MAC address 0002.0000.0002.

```
Switch(config)# mac access-list extended maclist1
Switch(config-ext-macl)# permit 0001.0000.0001 0.0.0 0002.0000.0001 0.0.0
Switch(config-ext-macl)# permit 0001.0000.0002 0.0.0 0002.0000.0002 0.0.0 xns-idp
! (Note: all other access implicitly denied)
```

Classifying Ingress Traffic by Using Class Maps

You use the **class-map** global configuration command to create a class map for matching packets to the class whose name you specify. The class map isolates a specific ingress traffic flow (class) from all other traffic by defining the criteria to use to match against a specific flow. A match criterion is defined with a match statement entered within the class-map configuration mode. Packets are checked against the match criteria configured for a class map. If a packet matches the specified criteria, the packet is considered a member of the class and is forwarded according to the QoS specifications set in the traffic policy.

Note You cannot use ACL matches in service policies attached to ES ports.

For information on how to classify egress traffic on an ES port, see the "Classifying Egress Traffic by Using Class Maps" section on page 26-78.

Beginning in privileged EXEC mode, follow these steps to create a class map and to define the match criterion to classify inbound traffic:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] or **access-list** *access-list-number* {**deny** | **permit**} *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] or **mac access-list extended** *name* {**permit** | **deny**} {**host** *src-MAC-addr mask* | **any** | **host** *dst-MAC-addr* | *dst-MAC-addr mask*} [*type mask*]	Create an IP standard or extended ACL for IP traffic or a Layer 2 MAC ACL for non-IP traffic, repeating the command as many times as necessary. For more information, see the "Classifying Ingress Traffic by Using ACLs" section on page 26-49. **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.

	Command	Purpose
Step 3	**class-map** [**match-all** | **match-any**] *class-map-name*	Create a class map, and enter class-map configuration mode. By default, no class maps are defined. • (Optional) Use the **match-all** keyword to perform a logical-AND of all matching statements under this class map. All criteria in the class map must be matched. • (Optional) Use the **match-any** keyword to perform a logical-OR of all matching statements under this class map. One or more criteria must be matched. • For *class-map-name*, specify the name of the class map. If neither the **match-all** nor the **match-any** keyword is specified, the default is **match-all**.
Step 4	**match** {**access-group** *acl-index-or-name* | **ip dscp** *dscp-list* | **ip precedence** *ip-precedence-list*}	Define the match criterion to classify traffic. By default, no match criterion is defined. Only one ACL per class map is supported. • For **access-group** *acl-index-or-name*, specify the number or name of the ACL created in Step 2. • For **ip dscp** *dscp-list*, enter a list of up to eight IP DSCP values to match against inbound packets. Separate each value with a space. The range is 0 to 63. • For **ip precedence** *ip-precedence-list*, enter a list of up to eight IP-precedence values to match against inbound packets. Separate each value with a space. The range is 0 to 7.
Step 5	**end**	Return to privileged EXEC mode.
Step 6	**show class-map**	Verify your entries.
Step 7	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To delete an existing class map, use the **no class-map** [**match-all** | **match-any**] *class-map-name* global configuration command. To remove a match criterion, use the **no match** {**access-group** *acl-index-or-name* | **ip dscp** | **ip precedence**} class-map configuration command.

This example shows how to configure the class map called *class1*. The *class1* has one match criterion, which is access list 103. It permits traffic from any host to any destination that matches a DSCP value of 10.

```
Switch(config)# access-list 103 permit any any dscp 10
Switch(config)# class-map class1
Switch(config-cmap)# match access-group 103
Switch(config-cmap)# exit
```

This example shows how to create a class map called *class2*, which matches inbound traffic with DSCP values of 10, 11, and 12:

```
Switch(config)# class-map class2
Switch(config-cmap)# match ip dscp 10 11 12
Switch(config-cmap)# exit
```

This example shows how to create a class map called *class3*, which matches inbound traffic with IP-precedence values of 5, 6, and 7:

```
Switch(config)# class-map class3
Switch(config-cmap)# match ip precedence 5 6 7
Switch(config-cmap)# exit
```

Classifying, Policing, and Marking Ingress Traffic by Using Policy Maps

A policy map specifies which inbound traffic class to act on. You can specify which CoS, DSCP, or IP precedence values in the traffic class to trust. You can specify which DSCP or IP precedence values in the traffic class to set. You can specify the traffic bandwidth limitations for each matched traffic class (policer) and the action to take when the traffic is out of profile (marking).

A policy map also has these characteristics:

- A policy map can contain multiple class statements, each with different match criteria and policers.
- A separate policy-map class can exist for each type of traffic received through a port.
- A policy-map trust state and a port trust state are mutually exclusive. The last one configured takes affect.

You can attach only one ingress policy-map per port.

Before beginning this procedure, make sure that you have created the class map to isolate traffic. For more information, see the "Classifying Ingress Traffic by Using ACLs" section on page 26-49 and the "Classifying Ingress Traffic by Using Class Maps" section on page 26-52.

For information about configuring an egress policy-map for an ES port, see the "Configuring an Egress Two-Rate Traffic Policer" section on page 26-80 and the "Configuring Class-Based Packet Marking in an Egress Traffic Policy" section on page 26-84.

Beginning in privileged EXEC mode, follow these steps to create an ingress policy-map:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**policy-map** *policy-map-name*	Create a policy map by entering the policy-map name, and enter policy-map configuration mode. By default, no policy maps are defined. The default behavior of a policy map is to set the DSCP to 0 if the packet is an IP packet and to set the CoS to 0 if the packet is tagged. No policing is performed.
Step 3	**class** *class-name*	Specify the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.

	Command	Purpose
Step 4	**trust** [**cos** | **dscp** | **ip-precedence**]	Configure the trust state, which QoS uses to generate a CoS-based or DSCP-based QoS label. By default, the port is not trusted. If no keyword is specified when the command is entered, the default is **dscp**. The keywords have these meanings: • **cos**—QoS derives the DSCP value by using the received or default port CoS value and the CoS-to-DSCP map. • **dscp**—QoS derives the DSCP value by using the DSCP value from the ingress packet. For non-IP packets that are tagged, QoS derives the DSCP value by using the received CoS value; for non-IP packets that are untagged, QoS derives the DSCP value by using the default port CoS value. In either case, the DSCP value is derived from the CoS-to-DSCP map. • **ip-precedence**—QoS derives the DSCP value by using the IP precedence value from the ingress packet and the IP-precedence-to-DSCP map. For non-IP packets that are tagged, QoS derives the DSCP value by using the received CoS value; for non-IP packets that are untagged, QoS derives the DSCP value by using the default port CoS value. In either case, the DSCP value is derived from the CoS-to-DSCP map. For more information, see the "Configuring the CoS-to-DSCP Map" section on page 26-59.
Step 5	**set** {**ip dscp** *new-dscp* | **ip precedence** *new-precedence*}	Mark IP traffic by setting a new value in the packet: • For **ip dscp** *new-dscp*, enter a new DSCP value to be assigned to the classified traffic. The range is 0 to 63. • For **ip precedence** *new-precedence*, enter a new IP-precedence value to be assigned to the classified traffic. The range is 0 to 7.
Step 6	**police** *rate-bps burst-byte* [**exceed-action** {**drop** | **policed-dscp-transmit**}]	Define a policer for the classified traffic. By default, no policer is defined. For information on the number of policers supported, see the "Standard QoS Configuration Guidelines" section on page 26-40. • For *rate-bps*, specify average traffic rate in bps. The range is 8000 to 1000000000. • For *burst-byte*, specify the normal burst size in bytes. The range is 8000 to 1000000. • (Optional) Specify the action to take when the rates are exceeded. Use the **exceed-action drop** keywords to drop the packet. Use the **exceed-action policed-dscp-transmit** keywords to mark down the DSCP value (through the policed-DSCP map) and send the packet. For more information, see the "Configuring the Policed-DSCP Map" section on page 26-61.
Step 7	**exit**	Return to policy-map configuration mode.
Step 8	**exit**	Return to global configuration mode.

	Command	Purpose
Step 9	**interface** *interface-id*	Specify the port to attach to the policy map, and enter interface configuration mode. Valid interfaces include physical ports.
Step 10	**service-policy input** *policy-map-name*	Specify the ingress policy-map name, and apply it to a port. Only one policy map per port is supported.
Step 11	**end**	Return to privileged EXEC mode.
Step 12	**show policy-map** [*policy-map-name* [**class** *class-map-name*]]	Verify your entries.
Step 13	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To delete an existing policy map, use the **no policy-map** *policy-map-name* global configuration command. To delete an existing class, use the **no class** *class-name* policy-map configuration command. To return to the untrusted state, use the **no trust** policy-map class configuration command. To remove an assigned DSCP or IP precedence value, use the **no set** {**ip dscp** *new-dscp* | **ip precedence** *new-precedence*} policy-map class configuration command. To remove an existing policer, use the **no police** *rate-bps burst-byte* [**exceed-action** {**drop** | **policed-dscp-transmit**}] policy-map class configuration command. To remove the policy map and interface association, use the **no service-policy input** *policy-map-name* interface configuration command.

This example shows how to create an ingress policy-map and attach it to a port. In the configuration, the IP standard ACL permits traffic from network 10.1.0.0. For traffic matching this classification, the DSCP value in the inbound packet is trusted. If the matched traffic exceeds an average traffic rate of 48000 bps and a normal burst size of 8000 bytes, its DSCP is marked down (based on the policed-DSCP map) and sent.

```
Switch(config)# access-list 1 permit 10.1.0.0 0.0.255.255
Switch(config)# class-map ipclass1
Switch(config-cmap)# match access-group 1
Switch(config-cmap)# exit
Switch(config)# policy-map flow1t
Switch(config-pmap)# class ipclass1
Switch(config-pmap-c)# trust dscp
Switch(config-pmap-c)# police 48000 8000 exceed-action policed-dscp-transmit
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/0/1
Switch(config-if)# service-policy input flow1t
```

This example shows how to create a Layer 2 MAC ACL with two permit statements and attach it to a port. The first permit statement allows traffic from the host with MAC address 0001.0000.0001 destined for the host with MAC address 0002.0000.0001. The second permit statement allows only Ethertype XNS-IDP traffic from the host with MAC address 0001.0000.0002 destined for the host with MAC address 0002.0000.0002.

```
Switch(config)# mac access-list extended maclist1
Switch(config-ext-mac)# permit 0001.0000.0001 0.0.0 0002.0000.0001 0.0.0
Switch(config-ext-mac)# permit 0001.0000.0002 0.0.0 0002.0000.0002 0.0.0 xns-idp
Switch(config-ext-mac)# exit
Switch(config)# mac access-list extended maclist2
Switch(config-ext-mac)# permit 0001.0000.0003 0.0.0 0002.0000.0003 0.0.0
Switch(config-ext-mac)# permit 0001.0000.0004 0.0.0 0002.0000.0004 0.0.0 aarp
Switch(config-ext-mac)# exit
Switch(config)# class-map macclass1
Switch(config-cmap)# match access-group maclist1
Switch(config-cmap)# exit
Switch(config)# policy-map macpolicy1
Switch(config-pmap)# class macclass1
Switch(config-pmap-c)# set ip dscp 63
Switch(config-pmap-c)# exit
Switch(config-pmap)# class macclass2 maclist2
Switch(config-pmap-c)# set ip dscp 45
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/0/1
Switch(config-if)# mls qos trust cos
Switch(config-if)# service-policy input macpolicy1
```

Classifying, Policing, and Marking Ingress Traffic by Using Aggregate Policers

By using an aggregate policer, you can create a policer that is shared by multiple traffic classes within the same policy map. However, you cannot use the aggregate policer across different policy maps or ports.

Before beginning this procedure, make sure that you have created the class map to isolate traffic. For more information, see the "Classifying Ingress Traffic by Using ACLs" section on page 26-49 and the "Classifying Ingress Traffic by Using Class Maps" section on page 26-52.

Beginning in privileged EXEC mode, follow these steps to create an aggregate policer for inbound traffic:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**mls qos aggregate-policer** *aggregate-policer-name rate-bps burst-byte* **exceed-action** {**drop** | **policed-dscp-transmit**}	Define the policer parameters that can be applied to multiple traffic classes within the same policy map. By default, no aggregate policer is defined. For information on the number of policers supported, see the "Standard QoS Configuration Guidelines" section on page 26-40 • For *aggregate-policer-name*, specify the name of the aggregate policer. • For *rate-bps*, specify average traffic rate in bps. The range is 8000 to 1000000000. • For *burst-byte*, specify the normal burst size in bytes. The range is 8000 to 1000000. • Specify the action to take when the rates are exceeded. Use the **exceed-action drop** keywords to drop the packet. Use the **exceed-action policed-dscp-transmit** keywords to mark down the DSCP value (through the policed-DSCP map) and send the packet. For more information, see the "Configuring the Policed-DSCP Map" section on page 26-61.
Step 3	**policy-map** *policy-map-name*	Create a policy map by entering the policy-map name, and enter policy-map configuration mode. For more information, see the "Classifying, Policing, and Marking Ingress Traffic by Using Policy Maps" section on page 26-54.

	Command	Purpose
Step 4	**class** *class-name*	Specify the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.
Step 5	**police aggregate** *aggregate-policer-name*	Apply an aggregate policer to multiple classes in the same policy map. For *aggregate-policer-name*, enter the name specified in Step 2.
Step 6	**exit**	Return to global configuration mode.
Step 7	**interface** *interface-id*	Specify the port to attach to the policy map, and enter interface configuration mode. Valid interfaces include physical ports.
Step 8	**service-policy input** *policy-map-name*	Specify the ingress policy-map name, and apply it to a port. Only one policy map per port is supported.
Step 9	**end**	Return to privileged EXEC mode.
Step 10	**show mls qos aggregate-policer** [*aggregate-policer-name*]	Verify your entries.
Step 11	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To remove the specified aggregate policer from a policy map, use the **no police aggregate** *aggregate-policer-name* policy-map class configuration command. To delete an aggregate policer and its parameters, use the **no mls qos aggregate-policer** *aggregate-policer-name* global configuration command.

This example shows how to create an aggregate policer and attach it to multiple classes within a policy map. In the configuration, the IP ACLs permit traffic from network 10.1.0.0 and from host 11.3.1.1. For traffic coming from network 10.1.0.0, the DSCP in the inbound packets is trusted. For traffic coming from host 11.3.1.1, the DSCP in the packet is changed to 56. The traffic rate from the 10.1.0.0 network and from host 11.3.1.1 is policed. If the traffic exceeds an average rate of 48000 bps and a normal burst size of 8000 bytes, its DSCP is marked down (based on the policed-DSCP map) and sent. The ingress policy-map is attached to a port.

```
Switch(config)# access-list 1 permit 10.1.0.0 0.0.255.255
Switch(config)# access-list 2 permit 11.3.1.1
Switch(config)# mls qos aggregate-police transmit1 48000 8000 exceed-action 
policed-dscp-transmit
Switch(config)# class-map ipclass1
Switch(config-cmap)# match access-group 1
Switch(config-cmap)# exit
Switch(config)# class-map ipclass2
Switch(config-cmap)# match access-group 2
Switch(config-cmap)# exit
Switch(config)# policy-map aggflow1
Switch(config-pmap)# class ipclass1
Switch(config-pmap-c)# trust dscp
Switch(config-pmap-c)# police aggregate transmit1
Switch(config-pmap-c)# exit
Switch(config-pmap)# class ipclass2
Switch(config-pmap-c)# set ip dscp 56
Switch(config-pmap-c)# police aggregate transmit1
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/0/1
Switch(config-if)# service-policy input aggflow1
Switch(config-if)# exit
```

Configuring DSCP Maps

These sections describe how to configure the DSCP maps:

- Configuring the CoS-to-DSCP Map, page 26-59 (optional)
- Configuring the IP-Precedence-to-DSCP Map, page 26-60 (optional)
- Configuring the Policed-DSCP Map, page 26-61 (optional, unless the null settings in the map are not appropriate)
- Configuring the DSCP-to-CoS Map, page 26-62 (optional)
- Configuring the DSCP-to-DSCP-Mutation Map, page 26-63 (optional, unless the null settings in the map are not appropriate)

All the maps, except the DSCP-to-DSCP-mutation map, 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 inbound packets to a DSCP value that QoS uses internally to represent the priority of the traffic.

Table 26-12 shows the default CoS-to-DSCP map.

Table 26-12 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.

Beginning in privileged EXEC mode, follow these steps to modify the CoS-to-DSCP map. This procedure is optional.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**mls qos map cos-dscp** *dscp1...dscp8*	Modify the CoS-to-DSCP map. For *dscp1...dscp8*, enter eight DSCP values that correspond to CoS values 0 to 7. Separate each DSCP value with a space. The DSCP range is 0 to 63.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show mls qos maps cos-dscp**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

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

This example shows how to modify and display the CoS-to-DSCP map:

```
Switch(config)# mls qos map cos-dscp 10 15 20 25 30 35 40 45
Switch(config)# end
Switch# show mls qos maps cos-dscp

Cos-dscp map:
        cos:   0  1  2  3  4  5  6  7
     --------------------------------
       dscp:   10 15 20 25 30 35 40 45
```

Configuring the IP-Precedence-to-DSCP Map

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

Table 26-13 shows the default IP-precedence-to-DSCP map:

Table 26-13 Default IP-Precedence-to-DSCP Map

IP precedence 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.

Beginning in privileged EXEC mode, follow these steps to modify the IP-precedence-to-DSCP map. This procedure is optional.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**mls qos map ip-prec-dscp** *dscp1...dscp8*	Modify the IP-precedence-to-DSCP map. For *dscp1...dscp8*, enter eight DSCP values that correspond to the IP precedence values 0 to 7. Separate each DSCP value with a space. The DSCP range is 0 to 63.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show mls qos maps ip-prec-dscp**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return to the default map, use the **no mls qos ip-prec-dscp** global configuration command.

This example shows how to modify and display the IP-precedence-to-DSCP map:

```
Switch(config)# mls qos map ip-prec-dscp 10 15 20 25 30 35 40 45
Switch(config)# end
Switch# show mls qos maps ip-prec-dscp

IpPrecedence-dscp map:
     ipprec:   0  1  2  3  4  5  6  7
     --------------------------------
       dscp:   10 15 20 25 30 35 40 45
```

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 an ingress policing and marking action.

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

Beginning in privileged EXEC mode, follow these steps to modify the policed-DSCP map. This procedure is optional.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**mls qos map policed-dscp** *dscp-list* **to** *mark-down-dscp*	Modify the policed-DSCP map. • For *dscp-list*, enter up to eight DSCP values separated by spaces. Then enter the **to** keyword. • For *mark-down-dscp*, enter the corresponding policed (marked down) DSCP value.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show mls qos maps policed-dscp**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

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

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

```
Switch(config)# mls qos map policed-dscp 50 51 52 53 54 55 56 57 to 0
Switch(config)# end
Switch# show mls qos maps policed-dscp
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 this 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, which is used to select one of the four queues in the egress queue-set.

Table 26-14 shows the default DSCP-to-CoS map.

Table 26-14 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 these values are not appropriate for your network, you need to modify them.

Beginning in privileged EXEC mode, follow these steps to modify the DSCP-to-CoS map. This procedure is optional.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**mls qos map dscp-cos** *dscp-list* **to** *cos*	Modify the DSCP-to-CoS map. • For *dscp-list*, enter up to eight DSCP values separated by spaces. Then enter the **to** keyword. • For *cos*, enter the CoS value to which the DSCP values correspond. The DSCP range is 0 to 63; the CoS range is 0 to 7.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show mls qos maps dscp-to-cos**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return to the default map, use the **no mls qos dscp-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(config)# mls qos map dscp-cos 0 8 16 24 32 40 48 50 to 0
Switch(config)# end
Switch# show mls 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.

Configuring the DSCP-to-DSCP-Mutation Map

If two QoS domains have different DSCP definitions, use the DSCP-to-DSCP-mutation map to translate one set of DSCP values to match the definition of another domain. You apply the DSCP-to-DSCP-mutation map to the receiving port (ingress mutation) at the boundary of a QoS administrative domain.

With ingress mutation, the new DSCP value overwrites the one in the packet, and QoS treats the packet with this new value. The switch sends the packet out the port with the new DSCP value.

You can configure multiple DSCP-to-DSCP-mutation maps and apply them to traffic received on a port. The default DSCP-to-DSCP-mutation map is a null map, which maps an inbound DSCP value to the same DSCP value.

Beginning in privileged EXEC mode, follow these steps to modify the DSCP-to-DSCP-mutation map. This procedure is optional.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**mls qos map dscp-mutation** *dscp-mutation-name in-dscp* **to** *out-dscp*	Modify the DSCP-to-DSCP-mutation map. • For *dscp-mutation-name*, enter the mutation map name. You can create more than one map by specifying a new name. • For *in-dscp*, enter up to eight DSCP values separated by spaces. Then enter the **to** keyword. • For *out-dscp*, enter a single DSCP value. The DSCP range is 0 to 63.
Step 3	**interface** *interface-id*	Specify the port to which to attach the map, and enter interface configuration mode. Valid interfaces include physical ports.
Step 4	**mls qos trust dscp**	Configure the ingress port as a DSCP-trusted port. By default, the port is not trusted.
Step 5	**mls qos dscp-mutation** *dscp-mutation-name*	Apply the map to the specified ingress DSCP-trusted port. For *dscp-mutation-name*, enter the mutation map name specified in Step 2.
Step 6	**end**	Return to privileged EXEC mode.
Step 7	**show mls qos maps dscp-mutation**	Verify your entries.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return to the default map, use the **no mls qos dscp-mutation** *dscp-mutation-name* global configuration command.

This example shows how to define the DSCP-to-DSCP-mutation map. All the entries that are not explicitly configured are not modified (remains as specified in the null map).

```
Switch(config)# mls qos map dscp-mutation mutation1 1 2 3 4 5 6 7 to 0
Switch(config)# mls qos map dscp-mutation mutation1 8 9 10 11 12 13 to 10
Switch(config)# mls qos map dscp-mutation mutation1 20 21 22 to 20
Switch(config)# mls qos map dscp-mutation mutation1 30 31 32 33 34 to 30
Switch(config)# interface gigabitethernet1/0/1
Switch(config-if)# mls qos trust dscp
Switch(config-if)# mls qos dscp-mutation mutation1
Switch(config-if)# end
Switch# show mls qos maps dscp-mutation mutation1
Dscp-dscp mutation map:
   mutation1:
     d1 :  d2 0  1  2  3  4  5  6  7  8  9
     ---------------------------------------
      0 :    00 00 00 00 00 00 00 00 10 10
      1 :    10 10 10 10 14 15 16 17 18 19
      2 :    20 20 20 23 24 25 26 27 28 29
      3 :    30 30 30 30 30 35 36 37 38 39
      4 :    40 41 42 43 44 45 46 47 48 49
      5 :    50 51 52 53 54 55 56 57 58 59
      6 :    60 61 62 63
```

Note In the above DSCP-to-DSCP-mutation map, the mutated 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 mutated value. For example, a DSCP value of 12 corresponds to a mutated value of 10.

Configuring Ingress Queue Characteristics

Depending on the complexity of your network and your QoS solution, you might need to perform all of the tasks in the next sections. You will need to make decisions about these characteristics:

- Which packets are assigned (by DSCP or CoS value) to each queue?
- What drop percentage thresholds apply to each queue, and which CoS or DSCP values map to each threshold?
- How much of the available buffer space is allocated between the queues?
- How much of the available bandwidth is allocated between the queues?
- Is there traffic (such as voice) that should be given high priority?

These sections describe how to configure ingress queue characteristics:

- Mapping DSCP or CoS Values to an Ingress Queue and Setting WTD Thresholds, page 26-65 (optional)
- Allocating Buffer Space Between the Ingress Queues, page 26-66 (optional)
- Allocating Bandwidth Between the Ingress Queues, page 26-67 (optional)
- Configuring the Ingress Priority Queue, page 26-68 (optional)

Mapping DSCP or CoS Values to an Ingress Queue and Setting WTD Thresholds

You can prioritize inbound traffic by placing packets with particular DSCP or CoS values into certain queues and by adjusting the queue thresholds so that packets with lower priorities are dropped.

Beginning in privileged EXEC mode, follow these steps to map DSCP or CoS values to an ingress queue and to set WTD thresholds. This procedure is optional.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**mls qos srr-queue input dscp-map queue** *queue-id* **threshold** *threshold-id dscp1...dscp8* or **mls qos srr-queue input cos-map queue** *queue-id* **threshold** *threshold-id cos1...cos8*	Map DSCP or CoS values to an ingress queue and to a threshold ID. By default, DSCP values 0–39 and 48–63 are mapped to queue 1 and threshold 1. DSCP values 40–47 are mapped to queue 2 and threshold 1. By default, CoS values 0–4, 6, and 7 are mapped to queue 1 and threshold 1. CoS value 5 is mapped to queue 2 and threshold 1. • For *queue-id*, the range is 1 to 2. • For *threshold-id*, the range is 1 to 3. The drop-threshold percentage for threshold 3 is predefined. It is set to the queue-full state. • For *dscp1...dscp8*, enter up to eight values, and separate each value with a space. The range is 0 to 63. • For *cos1...cos8*, enter up to eight values, and separate each value with a space. The range is 0 to 7.
Step 3	**mls qos srr-queue input threshold** *queue-id threshold-percentage1 threshold-percentage2*	Assign the two WTD threshold percentages for (threshold 1 and 2) to an ingress queue. The default, both thresholds are set to 100 percent. • For *queue-id*, the range is 1 to 2. • For *threshold-percentage1 threshold-percentage2*, the range is 1 to 100. Separate each value with a space. Each threshold value is a percentage of the total number of queue descriptors allocated for the queue.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show mls qos maps**	Verify your entries. The DSCP input queue threshold map appears as a matrix. The d1 column specifies the most-significant digit of the DSCP number; the d2 row specifies the least-significant digit in the DSCP number. The intersection of the d1 and the d2 values provides the queue ID and threshold ID; for example, queue 2 and threshold 1 (02-01). The CoS input queue threshold map shows the CoS value in the top row and the corresponding queue ID and threshold ID in the second row; for example, queue 2 and threshold 2 (2-2).
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return to the default CoS input queue threshold map or the default DSCP input queue threshold map, use the **no mls qos srr-queue input cos-map** or the **no mls qos srr-queue input dscp-map** global configuration command. To return to the default WTD threshold percentages, use the **no mls qos srr-queue input threshold** *queue-id* global configuration command.

This example shows how to map DSCP values 0 to 6 to ingress queue 1 and to threshold 1 with a drop threshold of 50 percent. It maps DSCP values 20 to 26 to ingress queue 1 and to threshold 2 with a drop threshold of 70 percent.

```
Switch(config)# mls qos srr-queue input dscp-map queue 1 threshold 1 0 1 2 3 4 5 6
Switch(config)# mls qos srr-queue input dscp-map queue 1 threshold 2 20 21 22 23 24 25 26
Switch(config)# mls qos srr-queue input threshold 1 50 70
```

In this example, the DSCP values (0 to 6) are assigned the WTD threshold of 50 percent and will be dropped sooner than the DSCP values (20 to 26) assigned to the WTD threshold of 70 percent.

Allocating Buffer Space Between the Ingress Queues

You define the ratio (allocate the amount of space) with which to divide the ingress buffers between the two queues. The buffer and the bandwidth allocation control how much data can be buffered before packets are dropped.

Beginning in privileged EXEC mode, follow these steps to allocate the buffers between the ingress queues. This procedure is optional.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**mls qos srr-queue input buffers** *percentage1 percentage2*	Allocate the buffers between the ingress queues By default 90 percent of the buffers are allocated to queue 1, and 10 percent of the buffers are allocated to queue 2. For *percentage1 percentage2,* the range is 0 to 100. Separate each value with a space. You should allocate the buffers so that the queues can handle any inbound bursty traffic.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show mls qos interface buffer** or **show mls qos input-queue**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return to the default setting, use the **no mls qos srr-queue input buffers** global configuration command.

This example shows how to allocate 60 percent of the buffer space to ingress queue 1 and 40 percent of the buffer space to ingress queue 2:

```
Switch(config)# mls qos srr-queue input buffers 60 40
```

Allocating Bandwidth Between the Ingress Queues

You need to specify how much of the available bandwidth is allocated between the ingress queues. The ratio of the weights is the ratio of the frequency in which the SRR scheduler sends packets from each queue to the internal ring. The bandwidth and the buffer allocation control how much data can be buffered before packets are dropped. On ingress queues, SRR operates only in shared mode.

Beginning in privileged EXEC mode, follow these steps to allocate bandwidth between the ingress queues. This procedure is optional.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**mls qos srr-queue input bandwidth** *weight1 weight2*	Assign shared round robin weights to the ingress queues. The default setting for *weight1 and weight2* is 4 (1/2 of the bandwidth is equally shared between the two queues). For *weight1* and *weight2*, the range is 1 to 100. Separate each value with a space. SRR services the priority queue for its configured weight as specified by the **bandwidth** keyword in the **mls qos srr-queue input priority-queue** *queue-id* **bandwidth** *weight* global configuration command. Then, SRR shares the remaining bandwidth with both ingress queues and services them as specified by the weights configured with the **mls qos srr-queue input bandwidth** *weight1 weight2* global configuration command. For more information, see the "Configuring the Ingress Priority Queue" section on page 26-68.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show mls qos interface queueing** or **show mls qos input-queue**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return to the default setting, use the **no mls qos srr-queue input bandwidth** global configuration command.

This example shows how to assign the ingress bandwidth to the queues. Priority queueing is disabled, and the shared bandwidth ratio allocated to queue 1 is 25/(25+75) and to queue 2 is 75/(25+75).

```
Switch(config)# mls qos srr-queue input priority-queue 2 bandwidth 0
Switch(config)# mls qos srr-queue input bandwidth 25 75
```

Configuring the Ingress Priority Queue

You should use the ingress priority queue only for traffic that needs to be expedited (for example, voice traffic, which needs minimum delay and jitter).

The priority queue is guaranteed part of the bandwidth to reduce the delay and jitter under heavy network traffic on an oversubscribed ring (when there is more traffic than the backplane can carry, and the queues are full and dropping frames).

SRR services the priority queue for its configured weight as specified by the **bandwidth** keyword in the **mls qos srr-queue input priority-queue** *queue-id* **bandwidth** *weight* global configuration command. Then, SRR shares the remaining bandwidth with both ingress queues and services them as specified by the weights configured with the **mls qos srr-queue input bandwidth** *weight1 weight2* global configuration command.

Beginning in privileged EXEC mode, follow these steps to configure the ingress priority queue. This procedure is optional.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**mls qos srr-queue input priority-queue** *queue-id* **bandwidth** *weight*	Assign a queue as the priority queue and guarantee bandwidth on the internal ring if the ring is congested. By default, the priority queue is queue 2, and 10 percent of the bandwidth is allocated to it. • For *queue-id*, the range is 1 to 2. • For **bandwidth** *weight*, assign the bandwidth percentage of the internal ring. The range is 0 to 40. The amount of bandwidth that can be guaranteed is restricted because a large value affects the entire ring and can degrade performance.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show mls qos interface queueing** or **show mls qos input-queue**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return to the default setting, use the **no mls qos srr-queue input priority-queue** *queue-id* global configuration command. To disable priority queueing, set the bandwidth weight to 0, for example, **mls qos srr-queue input priority-queue** *queue-id* **bandwidth 0**.

This example shows how to assign the ingress bandwidths to the queues. Queue 1 is the priority queue with 10 percent of the bandwidth allocated to it. The bandwidth ratios allocated to queues 1 and 2 is 4/(4+4). SRR services queue 1 (the priority queue) first for its configured 10 percent bandwidth. Then SRR equally shares the remaining 90 percent of the bandwidth between queues 1 and 2 by allocating 45 percent to each queue.

```
Switch(config)# mls qos srr-queue input priority-queue 1 bandwidth 10
Switch(config)# mls qos srr-queue input bandwidth 4 4
```

Configuring Egress Queue-Set Characteristics

Depending on the complexity of your network and your QoS solution, you might need to perform all of the tasks in the next sections. You will need to make decisions about these characteristics:

- Which packets are mapped by DSCP or CoS value to each queue and threshold ID?
- What drop percentage thresholds apply to the queue-set (four egress queues per port), and how much reserved and maximum memory is needed for the traffic type?
- How much of the fixed buffer space is allocated to the queue-set?
- Does the bandwidth of the port need to be rate limited?
- How often should the egress queue-set be serviced and which technique (shaped, shared, or both) should be used?

These sections describe how to configure egress queue-set characteristics:

- Allocating Buffer Space to and Setting WTD Thresholds for an Egress Queue-Set, page 26-69 (optional)
- Mapping DSCP or CoS Values to an Egress Queue-Set and to a Threshold ID, page 26-71 (optional)
- Configuring SRR Shaped Weights on an Egress Queue-Set, page 26-72 (optional)
- Configuring SRR Shared Weights on an Egress Queue-Set, page 26-73 (optional)
- Limiting the Egress Bandwidth on a Queue-Set, page 26-74 (optional)

Allocating Buffer Space to and Setting WTD Thresholds for an Egress Queue-Set

You can guarantee the availability of buffers, set WTD thresholds, and configure the maximum memory allocation for a queue-set by using the **mls qos queue-set output** *qset-id* **threshold** *queue-id drop-threshold1 drop-threshold2 reserved-threshold maximum-threshold* global configuration command.

Each threshold value is a percentage of the queue's allocated memory, which you specify by using the **mls qos queue-set output** *qset-id* **buffers** *allocation1* ... *allocation4* global configuration command. The queues use WTD to support distinct drop percentages for different traffic classes.

Note The egress queue-set default settings are suitable for most situations. You should change them only when you have a thorough understanding of the queues and only if these settings do not meet your QoS solution.

Beginning in privileged EXEC mode, follow these steps to configure the memory allocation and drop thresholds for a queue-set. This procedure is optional.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**mls qos queue-set output** *qset-id* **buffers** *allocation1 ... allocation4*	Allocate buffers to a queue-set. By default, all allocation values are equally mapped among the four queues (25, 25, 25, 25). Each queue has 1/4 of the buffer space. • For *qset-id,* enter the ID of the queue-set. The range is 1 to 2. Each port belongs to a queue-set, which defines all the characteristics of the four egress queues per port. • For *allocation1 ... allocation4*, specify four percentages, one for each queue in the queue-set. The range is 0 to 100. Separate each value with a space. Allocate buffers according to the importance of the traffic. For example, give a larger percentage of the buffer to the queue with the highest-priority traffic.
Step 3	**mls qos queue-set output** *qset-id* **threshold** *queue-id drop-threshold1 drop-threshold2 reserved-threshold maximum-threshold*	Configure the WTD thresholds, guarantee the availability of buffers, and configure the maximum memory allocation for the queue-set (four egress queues per port). By default, the WTD thresholds for queues 1, 3, and 4 are set to 100 percent. The thresholds for queue 2 are set to 50 percent. The reserved thresholds for queues 1, 3, and 4 are set to 50 percent. The reserved threshold for queue 2 is set to 100 percent. The maximum thresholds for all queues are set to 400 percent. • For *qset-id*, enter the ID of the queue-set specified in Step 2. The range is 1 to 2. • For *queue-id*, enter the specific queue in the queue-set on which the command is performed. The range is 1 to 4. • For *drop-threshold1 drop-threshold2*, specify the two WTD thresholds expressed as a percentage of the queue's allocated memory. The range is 1 to 400 percent. • For *reserved-threshold*, enter the amount of memory to be guaranteed (reserved) for the queue expressed as a percentage of the allocated memory. The range is 1 to 100 percent. • For *maximum-threshold*, enable a full queue to obtain more buffers than are reserved for it. This is the maximum memory the queue can have before the packets are dropped if the common pool is not empty. The range is 1 to 400 percent.
Step 4	**interface** *interface-id*	Specify the port, and enter interface configuration mode.
Step 5	**queue-set** *qset-id*	Map the port to a queue-set. For *qset-id*, enter the ID of the queue-set specified in Step 2. The range is 1 to 2. The default is 1.
Step 6	**end**	Return to privileged EXEC mode.
Step 7	**show mls qos interface** [*interface-id*] **buffers**	Verify your entries.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return to the default setting, use the **no mls qos queue-set output** *qset-id* **buffers** global configuration command. To return to the default WTD threshold percentages, use the **no mls qos queue-set output** *qset-id* **threshold** [*queue-id*] global configuration command.

This example shows how to map a port to queue-set 2. It allocates 40 percent of the buffer space to egress queue 1 and 20 percent each to egress queues 2, 3, and 4. It configures the drop thresholds for queue 2 to 40 and 60 percent of the allocated memory, guarantees (reserves) 100 percent of the allocated memory, and configures 200 percent as the maximum memory that this queue can have before packets are dropped.

```
Switch(config)# mls qos queue-set output 2 buffers 40 20 20 20
Switch(config)# mls qos queue-set output 2 threshold 2 40 60 100 200
Switch(config)# interface fastethernet1/0/1
Switch(config-if)# queue-set 2
```

Mapping DSCP or CoS Values to an Egress Queue-Set and to a Threshold ID

You can prioritize traffic by placing packets with particular DSCP or CoS values into certain queues and by adjusting the queue thresholds so that packets with lower priorities are dropped.

Note The egress queue-set default settings are suitable for most situations. You should change them only when you have a thorough understanding of the queues and only if these settings do not meet your QoS solution.

Beginning in privileged EXEC mode, follow these steps to map DSCP or CoS values to an egress queue and to a threshold ID. This procedure is optional.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**mls qos srr-queue output dscp-map queue** *queue-id* **threshold** *threshold-id dscp1...dscp8* or **mls qos srr-queue output cos-map queue** *queue-id* **threshold** *threshold-id cos1...cos8*	Map DSCP or CoS values to an egress queue and to a threshold ID for a port. By default, DSCP values 0–15 are mapped to queue 2 and threshold 1. DSCP values 16–31 are mapped to queue 3 and threshold 1. DSCP values 32–39 and 48–63 are mapped to queue 4 and threshold 1. DSCP values 40–47 are mapped to queue 1 and threshold 1. By default, CoS values 0 and 1 are mapped to queue 2 and threshold 1. CoS values 2 and 3 are mapped to queue 3 and threshold 1. CoS values 4, 6, and 7 are mapped to queue 4 and threshold 1. CoS value 5 is mapped to queue 1 and threshold 1. • For *queue-id*, the range is 1 to 4. • For *threshold-id*, the range is 1 to 3. The drop-threshold percentage for threshold 3 is predefined. It is set to the queue-full state. • For *dscp1...dscp8*, enter up to eight values, and separate each value with a space. The range is 0 to 63. • For *cos1...cos8*, enter up to eight values, and separate each value with a space. The range is 0 to 7.
Step 3	**end**	Return to privileged EXEC mode.

	Command	Purpose
Step 4	**show mls qos maps**	Verify your entries. The DSCP output queue threshold map appears as a matrix. The d1 column specifies the most-significant digit of the DSCP number; the d2 row specifies the least-significant digit in the DSCP number. The intersection of the d1 and the d2 values provides the queue ID and threshold ID; for example, queue 2 and threshold 1 (02-01). The CoS output queue threshold map shows the CoS value in the top row and the corresponding queue ID and threshold ID in the second row; for example, queue 2 and threshold 2 (2-2).
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return to the default DSCP output queue threshold map or the default CoS output queue threshold map, use the **no mls qos srr-queue output dscp-map** or the **no mls qos srr-queue output cos-map** global configuration command.

This example shows how to map DSCP values 10 and 11 to egress queue 1 and to threshold 2:

```
Switch(config)# mls qos srr-queue output dscp-map queue 1 threshold 2 10 11
```

Configuring SRR Shaped Weights on an Egress Queue-Set

You can specify how much of the available bandwidth is allocated to each queue in the queue-set. The ratio of the weights is the ratio of frequency in which the SRR scheduler sends packets from a standard port.

Note SRR shaping is not supported on the ES ports; however, you can configure average-rate shaping. For more information, see the "Configuring Shaping" section on page 26-99.

You can configure the egress queues for shaped weights, shared weights, or both. Use shaping to smooth bursty traffic or to provide a smoother output over time. For conceptual information, see the "SRR Shaping and Sharing" section on page 26-14. For configuration information, see the "Configuring SRR Shared Weights on an Egress Queue-Set" section on page 26-73.

Note The egress queue-set default settings are suitable for most situations. You should change them only when you have a thorough understanding of the queues and only if these settings do not meet your QoS solution.

Beginning in privileged EXEC mode, follow these steps to assign the shaped weights and to enable bandwidth shaping on a standard port mapped to the four egress queues. This procedure is optional.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Specify a standard port, and enter interface configuration mode.

	Command	Purpose
Step 3	**srr-queue bandwidth shape** *weight1 weight2 weight3 weight4*	Assign SRR weights to the egress queues. This command is not supported on an ES port. By default, weight1 is set to 25; weight2, weight3, and weight4 are set to 0, and these queues are in shared mode. For *weight1 weight2 weight3 weight4*, enter the weights to control the percentage of the port that is shaped. The inverse ratio (1/*weight*) controls the shaping bandwidth for this queue. Separate each value with a space. The range is 0 to 65535. If you configure a weight of 0, the corresponding queue operates in shared mode. The weight specified with the **srr-queue bandwidth shape** command is ignored, and the weights specified with the **srr-queue bandwidth share** interface configuration command for a queue come into effect. When configuring queues in the same queue-set for both shaping and sharing, make sure that you configure the lowest number queue for shaping. The shaped mode overrides the shared mode.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show mls qos interface** *interface-id* **queueing**	Verify your entries.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return to the default setting, use the **no srr-queue bandwidth shape** interface configuration command.

This example shows how to configure bandwidth shaping on queue 1. Because the weight ratios for queues 2, 3, and 4 are set to 0, these queues operate in shared mode. The bandwidth weight for queue 1 is 1/8, which is 12.5 percent.

```
Switch(config)# interface gigabitethernet1/0/1
Switch(config-if)# srr-queue bandwidth shape 8 0 0 0
```

Configuring SRR Shared Weights on an Egress Queue-Set

In shared mode, the queues in the queue-set share the bandwidth among them according to the configured weights. The bandwidth is guaranteed at this level but not limited to it. For example, if a queue empties and does not require a share of the link, the remaining queues can expand into the unused bandwidth and share it among them. With sharing, the ratio of the weights controls the frequency of dequeuing; the absolute values are meaningless.

Note The egress queue-set default settings are suitable for most situations. You should change them only when you have a thorough understanding of the queues and only if these settings do not meet your QoS solution.

Beginning in privileged EXEC mode, follow these steps to assign the shared weights and to enable bandwidth sharing on a port mapped to the four egress queues. This procedure is optional.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Specify a port, and enter interface configuration mode.
Step 3	**srr-queue bandwidth share** *weight1 weight2 weight3 weight4*	Assign SRR weights to the egress queues. By default, all four weights are 25 (1/4 of the bandwidth is allocated to each queue). For *weight1 weight2 weight3 weight4*, enter the weights to control the ratio of the frequency in which the SRR scheduler sends packets. Separate each value with a space. The range is 1 to 255.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show mls qos interface** *interface-id* **queueing**	Verify your entries.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return to the default setting, use the **no srr-queue bandwidth share** interface configuration command.

This example shows how to configure the weight ratio of the SRR scheduler running on an egress port. Four queues are used, and the bandwidth ratio allocated for each queue in shared mode is 1/(1+2+3+4), 2/(1+2+3+4), 3/(1+2+3+4), and 4/(1+2+3+4), which is 10 percent, 20 percent, 30 percent, and 40 percent for queues 1, 2, 3, and 4. This means that queue 4 has four times the bandwidth of queue 1, twice the bandwidth of queue 2, and one-and-a-third times the bandwidth of queue 3.

```
Switch(config)# interface gigabitethernet1/0/1
Switch(config-if)# srr-queue bandwidth share 1 2 3 4
```

Limiting the Egress Bandwidth on a Queue-Set

You can limit the egress bandwidth on a standard port mapped to a queue-set. For example, if a customer pays only for a small percentage of a high-speed link, you can limit the bandwidth to that amount.

Note The egress queue-set default settings are suitable for most situations. You should change them only when you have a thorough understanding of the queues and only if these settings do not meet your QoS solution.

Beginning in privileged EXEC mode, follow these steps to limit the egress bandwidth on a standard port. This procedure is optional.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Specify a standard port to be rate-limited, and enter interface configuration mode.
Step 3	**srr-queue bandwidth limit** *weight1*	Specify the percentage of the port speed to which the port should be limited. The range is 10 to 90. This command is not supported on an ES port. By default, the port is not rate limited and is set to 100 percent.

	Command	Purpose
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show mls qos interface** [*interface-id*] **queueing**	Verify your entries.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return to the default setting, use the **no srr-queue bandwidth limit** interface configuration command.

This example shows how to limit the bandwidth on a standard port to 80 percent:

```
Switch(config)# interface gigabitethernet1/0/1
Switch(config-if)# srr-queue bandwidth limit 80
```

When you configure this command to 80 percent, the port is idle 20 percent of the time. The line rate drops to 80 percent of the connected speed, which is 800 Mbps. These values are not exact because the hardware adjusts the line rate in increments of six.

Displaying Standard QoS Information

To display standard QoS information, use one or more of the privileged EXEC commands in Table 26-15:

Table 26-15 Commands for Displaying Standard QoS Information

Command	Purpose
show class-map [*class-map-name*]	Display QoS class maps, which define the match criteria to classify traffic.
show mls qos	Display global QoS configuration information.
show mls qos aggregate-policer [*aggregate-policer-name*]	Display the aggregate policer configuration.
show mls qos input-queue	Display QoS settings for the ingress queues.
show mls qos interface [*interface-id*] [**buffers** | **policers** | **queueing** | **statistics**]	Display QoS information at the port level, including the buffer allocation, which ports have policers, the queueing strategy, and the ingress and egress statistics.
show mls qos maps [**cos-dscp** | **cos-input-q** | **cos-output-q** | **dscp-cos** | **dscp-input-q** | **dscp-mutation** *dscp-mutation-name* | **dscp-output-q** | **ip-prec-dscp** | **policed-dscp**]	Display QoS mapping information.
show mls qos queue-set [*qset-id*]	Display the egress queue-set settings.
show policy-map [*policy-map-name* [**class** *class-map-name*]] **show policy-map interface** *interface-id* [**input**]	Display QoS policy-maps, which define the traffic policy for a traffic class. Display the ingress policy-map name applied to the specified port.

Configuring Hierarchical QoS

You can configure hierarchical QoS (traffic classification, CBWFQ, LLQ, shaping, and two-rate, three-color policing) and apply it to outbound traffic on an ES port.

Before configuring hierarchical 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 hierarchical QoS on your switch. Read these sections if traffic is flowing from a standard or an ES port to an ES port:

- Default Hierarchical QoS Configuration, page 26-76
- Hierarchical QoS Configuration Guidelines, page 26-76
- Configuring an Egress Hierarchical QoS Policy, page 26-77 (required)

Default Hierarchical QoS Configuration

QoS is disabled.

No traffic classes, class maps, policy maps, or policers are defined.

LLQ is disabled.

CBWFQ is disabled.

Tail drop is enabled.

WRED is disabled.

Average-rate traffic shaping is disabled.

Hierarchical QoS Configuration Guidelines

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

- QoS must be enabled with the **mls qos** global configuration command for any hierarchical configuration to take effect. When enabled, QoS uses the default settings described in the "Default Standard QoS Configuration" section on page 26-38 and the "Default Hierarchical QoS Configuration" section on page 26-76. For detailed steps, see the "Enabling QoS Globally" section on page 26-42.
- Because the switch does not support attaching a service policy to a logical interface (such as an EtherChannel), if you have EtherChannel ports configured on your switch, you must configure QoS classification, policing, mapping, and queueing on the individual physical ports that comprise the EtherChannel.
- Class maps that contain ACLs are not supported in an egress policy attached to an ES port. You cannot use the **match access-group** *acl-index-or-name* class-map configuration command.

- To configure a class map that matches an 802.1Q tunneling pair (instead of matching a single VLAN), you must configure the **class-map** global configuration command with the **match-all** keyword. You must enter the **match vlan** command before the **match vlan inner** command.
- You cannot have **match vlan** {*vlan-id* | **inner** *vlan-id*} and **match** {**cos** *cos-list* | **ip dscp** *dscp-list* | **ip precedence** *ip-precedence-list* | **mpls experimental** *exp-list*} class-map configuration commands in the same class map.
- You cannot mix VLAN-level and class-level matches within a class map.
- When port trust policies are used with 802.1Q tunneling, all ports sharing the same tunnel VLAN must be configured with the same trust policy.
- Only one policy map per port is supported. You can attach one ingress service-policy per port and one egress service-policy per ES port.
- On an ES port, there is no limit to the number of egress policers.
- You can define a class policy to use either tail drop through the **queue-limit** policy-map class configuration command or to use WRED packet drop through the **random-detect** policy-map class configuration command. You cannot use the **queue-limit** and **random-detect** commands in the same class policy, but they can be used in two class policies in the same policy map.
- You cannot use **bandwidth**, **queue-limit**, **random-detect**, and **shape** policy-map class configuration commands with the **priority** policy-map class configuration command in the same class within the same policy map. However, you can use these commands in different classes in the same policy map. Within a policy map, you can give priority status to only one class.
- You must configure the **bandwidth** or the **shape** policy-map class configuration command before you configure either the **queue-limit** or the **random-detect** policy-map class configuration command in a class policy. You must configure the **bandwidth** or the **shape** command in the same policy map as the **queue-limit** or the **random-detect** command if the policy is not using the default traffic class. If the policy is using the default traffic class, you do not need to specify the **bandwidth** and **shape** commands in the policy map.
- A policy map can have all the class bandwidths specified in either kbps or in percentages, but not a mix of both. You cannot specify bandwidth in kbps in a child policy (configured through the **service-policy** policy-map class configuration command) and then specify bandwidth as a percentage in the parent policy.

Configuring an Egress Hierarchical QoS Policy

These sections describe how to configure an egress hierarchical QoS policy to create a service policy that is attached to an ES port.

For background information, see the "Hierarchical Levels" section on page 26-20, the "Egress Classification Based on Traffic Classes and Traffic Policies" section on page 26-23, the "Egress Policing and Marking" section on page 26-24, and the "Queueing and Scheduling of Hierarchical Queues" section on page 26-26. For configuration guidelines, see the "Hierarchical QoS Configuration Guidelines" section on page 26-76.

Depending on your network configuration and QoS solution, you must perform one or more of these tasks if you want classify, policy, mark, queue, or schedule outbound traffic:

- Classifying Egress Traffic by Using Class Maps, page 26-78 (required)
- Configuring an Egress Two-Rate Traffic Policer, page 26-80 (optional)
- Configuring Class-Based Packet Marking in an Egress Traffic Policy, page 26-84 (optional)

- Configuring CBWFQ and Tail Drop, page 26-86 (optional)
- Configuring CBWFQ and DSCP-Based WRED, page 26-89 (optional)
- Configuring CBWFQ and IP Precedence-Based WRED, page 26-93 (optional)
- Enabling LLQ, page 26-97 (optional)
- Configuring Shaping, page 26-99 (optional)

Classifying Egress Traffic by Using Class Maps

You use the **class-map** global configuration command to create a class map for matching packets to the class whose name you specify. The class map isolates a specific outbound traffic flow (class) from all other traffic by defining the criteria to use to match against a specific flow. The match criterion is defined with a match statement entered within the class-map configuration mode. Packets are checked against the match criteria configured for a class map. If a packet matches the specified criteria, the packet is considered a member of the class and is forwarded according to the QoS specifications set in the traffic policy.

Beginning in privileged EXEC mode, follow these steps to create a class-level class-map and to define the match criterion to classify outbound traffic. This procedure is required. The examples that follow the procedure show how to create a class-level and a VLAN-level class-map.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**class-map** [**match-all** | **match-any**] *class-map-name*	Create a class map, and enter class-map configuration mode. By default, no class maps are defined. • (Optional) Use the **match-all** keyword to perform a logical-AND of all matching statements under this class map. All criteria in the class map must be matched. • (Optional) Use the **match-any** keyword to perform a logical-OR of all matching statements under this class map. One or more criteria must be matched. • For *class-map-name*, specify the name of the class map. If neither the **match-all** nor the **match-any** keyword is specified, the default is **match-all**. You must use the **match-all** keyword if you are matching an 802.1Q tunneling pair (instead of matching a single VLAN).

	Command	Purpose
Step 3	**match** {**cos** *cos-list* | **ip dscp** *dscp-list* | **ip precedence** *ip-precedence-list* | **mpls experimental** *exp-list* | **vlan** {*vlan-id* | **inner** *vlan-id*}}	Define the match criterion to classify traffic. By default, no match criterion is defined. • For **cos** *cos-list*, specify up to four Layer 2 CoS values to match against the packet. Separate each value with a space. The range is 0 to 7. • For **ip dscp** *dscp-list*, specify up to eight IP DSCP values to match against the packet. Separate each value with a space. The range is 0 to 63. • For **ip precedence** *ip-precedence-list*, specify up to eight IP-precedence values to match against the packet. Separate each value with a space. The range is 0 to 7. • For **mpls experimental** *exp-list*, specify up to eight MPLS EXP values to match against the packet. Separate each value with a space. The range is 0 to 7. • For **vlan** {*vlan-id* | **inner** *vlan-id*}, specify packet match based on the VLAN ID. – For **vlan** *vlan-id*, match a packet based on the VLAN ID, or match a packet based on the outer VLAN ID if an 802.1Q tunnel is configured. You can specify a single VLAN identified by a VLAN number or a range of VLANs separated by a hyphen. The range is 1 to 4094. – For **vlan inner** *vlan-id*, match a packet based on the inner VLAN ID of an 802.1Q tunnel.You can specify a single VLAN identified by a VLAN number or a range of VLANs separated by a hyphen. The range is 1 to 4094. When matching an 802.1Q tunneling pair (instead of matching a single VLAN), you must enter the **match vlan** command before the **match vlan inner** command. **Note** You cannot mix VLAN-level and class-level matches within a class map.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show class-map**	Verify your entries.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To delete an existing class map, use the **no class-map** [**match-all** | **match-any**] *class-map-name* global configuration command. To remove a match criterion, use the **no match** {**cos** *cos-list* | **ip dscp** *dscp-list* | **ip precedence** *ip-precedence-list* | **mpls experimental** *exp-list* | **vlan** {*vlan-id* | **inner** *vlan-id*}} class-map configuration command.

This example shows how to create a class-level class-map called *class3*, which matches traffic with IP-precedence values of 5, 6, and 7:

```
Switch(config)# class-map class3
Switch(config-cmap)# match ip precedence 5 6 7
```

This example shows how to create a VLAN-level class-map for 802.1Q tunneling called *dot1q*, which matches all traffic with an outer VLAN ID of 5 and an inner VLAN ID of 3 to 8:

```
Switch(config)# class-map match-all dot1q
Switch(config-cmap)# match vlan 5
Switch(config-cmap)# match vlan inner 3 - 8
```

This example shows how to create a VLAN-level class-map called *vlan203*, which matches traffic in VLAN 203:

```
Switch(config)# class-map match-all vlan203
Switch(config-cmap)# match vlan 203
```

Configuring an Egress Two-Rate Traffic Policer

You can configure a two-rate traffic policer within a policy map at the class level, at the VLAN level, and at the physical level by using the **police cir** or the **police cir percent** policy-map class configuration command. At the physical level of the hierarchy, you can police only the class-default class in an egress policy attached to an ES port.

The policer limits the transmission rate of a traffic class and marks actions (conform, exceed, and violate) for each packet. Within the conform, exceed, and violate categories, you decide packet treatments. In the most common configurations, you configure packets that conform to be sent, packets that exceed to be sent with a decreased priority, and packets that violate to be dropped. You can decrease the priority of the CoS, the DSCP, the IP precedence, or the MPLS EXP bits in the packet. You configure the policer by using the **police cir** policy-map class configuration command.

You also can use the **police cir percent** policy-map class configuration command to configure an egress traffic policer that uses two rates, the CIR and the PIR. The switch calculates the CIR and PIR based on a percentage of the maximum amount of bandwidth assigned to the parent class. The maximum bandwidth is controlled by the **shape** policy-map class configuration command if it is configured. Otherwise, the maximum bandwidth is the physical port bandwidth (1 Gbps). For more information about the interaction between the **police cir percent** command and the **shape** command, refer to the **police cir percent** command in the command reference for this release.

Before beginning this procedure, make sure that you have created the class map to isolate traffic. For more information, see the "Classifying Egress Traffic by Using Class Maps" section on page 26-78.

Beginning in privileged EXEC mode, follow these steps to configure a class-level, two-rate traffic policer in a service policy for outbound traffic. This procedure is optional. The examples that follow the procedure show how to configure a class-level, a VLAN-level, and a physical-level policer.

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

	Command	Purpose
Step 4	**police cir** *cir* [**bc** *conform-burst*] **pir** *pir* [**be** *peak-burst*] [**conform-action** *action* [**exceed-action** *action* [**violate-action** *action*]]] or **police cir percent** *percent* [**bc** *conform-burst* **ms**] | **pir percent** *percent* [**be** *peak-burst* **ms**] [**conform-action** *action* [**exceed-action** *action* [**violate-action** *action*]]] **Note** For the syntax description of the **police cir percent** command, refer to the command reference for this release.	Configure an egress traffic policer that uses two rates, the CIR and the peak information rate PIR. By default, no policer is defined. • For **cir** *cir*, specify the CIR at which the first token bucket is updated. The range is 64000 to 990000000 bps. • (Optional) For **bc** *conform-burst*, specify the conform burst size used by the first token bucket for policing. The range is 1536 to 16776960 bytes. The default is 8192. • For **pir** *pir*, specify the PIR at which the second token bucket for policing is updated. The range is 64000 to 990000000 bps. • (Optional) For **be** *peak-burst*, specify the peak burst size used by the second token bucket. The range is 1536 to 16776960 bytes. The default is 8192. • (Optional) For **conform-action**, specify the action to perform on packets that conform to the CIR and PIR. • (Optional) For **exceed-action**, specify the action to perform on packets that conform to the PIR but not the CIR. • (Optional) For **violate-action**, specify the action to perform on packets that exceed the PIR. **Note** If you do not specify an action, the default **conform-action** is transmit, the default **exceed-action** is drop, and the default **violate-action** is drop. • (Optional) For *action*, specify the action to perform on packets: – **drop**—drop the packet. – **set-cos-transmit** *new-cos*—set the CoS value to a new value, and send the packet. The range is 0 to 7. – **set-dscp-transmit** *new-dscp*—set the IP DSCP value to a new value, and send the packet. The range is 0 to 63. You also can enter a mnemonic name for a commonly used value. – **set-mpls-exp-transmit** *new-exp*—set the MPLS EXP bits, and send the packet. The range is 0 to 7. – **set-prec-transmit** *new-prec*—set the IP precedence value to a new value, and send the packet. The range is 0 to 7. – **transmit**—send the packet without altering it. If you set the CIR equal to the PIR, a traffic rate that is less than the CIR or that meets the CIR is in the conform range. Traffic that exceeds the CIR rate is in the violate range. If you set the PIR greater than the CIR, a traffic rate less than the CIR is in the conform range. A traffic rate that exceeds the CIR but is less than or equal to the PIR is in the exceed range. A traffic rate that exceeds the PIR is in the violate range. Setting the burst sizes too low can result in less traffic than expected, and setting them too high can result in more traffic than expected.
Step 5	**exit**	Return to policy-map configuration mode.

	Command	Purpose
Step 6	**exit**	Return to global configuration mode.
Step 7	**interface** *interface-id*	Specify an ES port to attach to the policy map, and enter interface configuration mode.
Step 8	**service-policy output** *policy-map-name*	Specify the egress policy-map name, and apply it to the ES port. Only one policy map per port is supported.
Step 9	**end**	Return to privileged EXEC mode.
Step 10	**show policy-map** *policy-map-name* or **show policy-map interface** *interface-id*	Verify your entries.
Step 11	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To delete an existing policy map, use the **no policy-map** *policy-map-name* global configuration command. To delete an existing class, use the **no class** *class-name* policy-map configuration command. To remove the two-rate policer, use the **no police cir** *cir* [**bc** *conform-burst*] **pir** *pir* [**be** *peak-burst*] [**conform-action** *action* [**exceed-action** *action* [**violate-action** *action*]]] policy-map class configuration command. To remove the policy map and interface association, use the **no service-policy output** *policy-map-name* interface configuration command.

This example shows how to configure a class-level, two-rate traffic policer to limit outbound traffic to an average committed rate of 500 kbps and a peak rate of 1 Mbps. Traffic marked as conforming to the average committed rate (500 kbps) is sent as is. Traffic marked as exceeding 500 kbps, but not exceeding 1 Mbps, is marked with IP precedence 2 and then sent. All traffic marked as exceeding 1 Mbps is dropped. The burst parameters are set to 10000 bytes.

```
Switch(config)# class-map c1
Switch(config-cmap)# match ip precedence 4
Switch(config-cmap)# exit
Switch(config)# policy-map policy1
Switch(config-pmap)# class c1
Switch(config-pmap-c)# police cir 500000 bc 10000 pir 1000000 be 10000 conform-action
transmit exceed-action set-prec-transmit 2 violate-action drop
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1/1
Switch(config-if)# service-policy output policy1
```

This example shows how to configure a class-level, two-rate traffic policer that uses a CIR and a PIR based on a percentage of bandwidth. A CIR of 20 percent and a PIR of 40 percent are specified. The optional **bc** and **be** values (300 ms and 400 ms, respectively) are specified.

```
Switch(config)# class-map c1
Switch(config-cmap)# match ip precedence 4
Switch(config-cmap)# exit
Switch(config)# policy-map policy1
Switch(config-pmap)# class class1
Switch(config-pmap-c)# police cir percent 20 bc 300 ms pir percent 40 be 400 ms
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1/1
Switch(config-if)# service-policy output policy1
```

This example shows how to configure a VLAN-level, two-rate traffic policer to limit outbound traffic to an average committed rate of 500 kbps and a peak rate of 1 Mbps. Traffic marked as conforming to the average committed rate (500 kbps) is sent as is. Traffic marked as exceeding 500 kbps, but not exceeding 1 Mbps, is marked with IP precedence 2 and then sent. All traffic marked as exceeding 1 Mbps is dropped. The burst parameters are set to 10000 bytes.

```
Switch(config)# class-map match-all vlan203
Switch(config-cmap)# match vlan 203
Switch(config-cmap)# match vlan inner 206
Switch(config-cmap)# exit
Switch(config)# policy-map vlan-policy
Switch(config-pmap)# class vlan203
Switch(config-pmap-c)# police cir 500000 bc 10000 pir 1000000 be 10000 conform-action
transmit exceed-action set-prec-transmit 2 violate-action drop
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1/2
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# service-policy output vlan-policy
```

This example shows how to create a hierarchical service-policy in which all levels are present. This configuration associates a class-level policy-map with a VLAN-level policy-map, associates the VLAN-level policy-map with a physical-level policy-map, and attaches the physical-level policy-map to a physical port.

Within a policy map, the class-default applies to all traffic that is not explicitly matched within the policy map but does match the parent policy. If no parent policy is configured, the parent policy represents the physical port. In a physical-level policy-map, class-default is the only class that can be configured.

```
Switch(config)# class-map my-class
Switch(config-cmap)# match ip precedence 1
Switch(config-cmap)# exit
Switch(config)# class-map my-logical-class
Switch(config-cmap)# match vlan 5
Switch(config-cmap)# exit
Switch(config)# policy-map my-class-policy
Switch(config-pmap)# class my-class
Switch(config-pmap-c)# set ip precedence 2
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# policy-map my-logical-policy
Switch(config-pmap)# class my-logical-class
Switch(config-pmap-c)# service-policy my-class-policy
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# policy-map my-physical-policy
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# police cir 500000 bc 10000 pir 1000000 be 10000 conform-action
transmit exceed-action set-prec-transmit 2 violate-action drop
Switch(config-pmap-c)# service-policy my-logical-policy
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1/1
Switch(config)# service-policy output my-physical-policy
```

Configuring Class-Based Packet Marking in an Egress Traffic Policy

You can perform class-based packet marking in an traffic policy by configuring the **set** {**cos** *new-cos* | **ip** {**dscp** *new-dscp* | **precedence** *new-precedence*} | **mpls experimental** *exp-number*} policy-map class configuration command in an egress policy-map attached to an ES port.

Before beginning this procedure, make sure that you have created the class map to isolate traffic. For more information, see the "Classifying Egress Traffic by Using Class Maps" section on page 26-78.

Beginning in privileged EXEC mode, follow these steps to configure class-level, class-based packet marking in a service policy for outbound traffic. This procedure is optional. The examples that follow the procedure show how to configure class-level and VLAN-level class-based packet marking.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**policy-map** *policy-map-name*	Create a policy map by entering the policy-map name, and enter policy-map configuration mode. By default, no policy maps are defined.
Step 3	**class** *class-name*	Specify the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.
Step 4	**set** {**cos** *new-cos* | **ip** {**dscp** *new-dscp* | **precedence** *new-precedence*} | **mpls experimental** *exp-number*}	Designate the value to set in the traffic class if the packets match the criteria. • For **cos** *new-cos*, enter the new CoS value assigned to the classified traffic. The range is 0 to 7. • For **ip dscp** *new-dscp*, enter the new DSCP value assigned to the classified traffic. The range is 0 to 63. The specified value sets the ToS byte in the packet header. • For **ip precedence** *new-precedence*, enter the new IP-precedence value assigned to the classified traffic. The range is 0 to 7. The specified value sets the precedence bit in the IP header. • For **mpls experimental** *exp-number*, enter the new MPLS EXP value assigned to the classified traffic. The range is 0 to 7. The specified value sets the MPLS EXP 3-bit field in the packet header.
Step 5	**exit**	Return to policy-map configuration mode.
Step 6	**exit**	Return to global configuration mode.
Step 7	**interface** *interface-id*	Specify an ES port to attach to the policy map, and enter interface configuration mode.
Step 8	**service-policy output** *policy-map-name*	Specify the egress policy-map name, and apply it to the ES port. Only one policy map per port is supported.
Step 9	**end**	Return to privileged EXEC mode.

	Command	Purpose
Step 10	**show policy-map** [*policy-map-name* [**class** *class-map-name*]] or **show policy-map interface** *interface-id*	Verify your entries.
Step 11	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To delete an existing policy map, use the **no policy-map** *policy-map-name* global configuration command. To delete an existing class, use the **no class** *class-name* policy-map configuration command. To remove an assigned value, use the **no set** {**cos** *new-cos* | **ip** {**dscp** *new-dscp* | **precedence** *new-precedence*} | **mpls experimental** *exp-number*} policy-map class configuration command. To remove the policy map and interface association, use the **no service-policy output** *policy-map-name* interface configuration command.

This example shows how to create a class-level policy-map called *out_pmap*. When it is attached to the ES port, it matches packets with MPLS EXP field 2 and resets this field to 3.

```
Switch(config)# class-map mpls_2
Switch(config-cmap)# match mpls experimental 2
Switch(config-cmap)# exit
Switch(config)# policy-map out-pmap
Switch(config-pmap)# class mpls_2
Switch(config-pmap-c)# set mpls experimental 3
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1/1
Switch(config-if)# service-policy output out-pmap
```

This example shows how to create a VLAN-level policy-map called *log-policy*. It matches packets with VLAN 203 and associates a class-level child policy called *cls-policy.* The child policy matches packets with MPLS EXP 2 and resets them to 5.

```
Switch(config)# class-map cls-class
Switch(config-cmap)# match mpls experimental 2
Switch(config-cmap)# exit
Switch(config)# class-map log-class
Switch(config-cmap)# match vlan 203
Switch(config-cmap)# exit
Switch(config)# policy-map cls-policy
Switch(config-pmap)# class cls-class
Switch(config-pmap-c)# set mpls experimental 5
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# policy-map log-policy
Switch(config-pmap)# class log-class
Switch(config-pmap-c)# service-policy cls-policy
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1/2
Switch(config-if)# switchport trunk encapsulation isl
Switch(config-if)# switchport mode trunk
Switch(config-if)# service-policy output log-policy
```

Configuring CBWFQ and Tail Drop

CBWFQ creates a queue for every class for which a class map is defined. Packets satisfying the match criteria for a class accumulate in the queue reserved for the class until they are sent, which occurs when the queue is serviced by the fair queue process. When the maximum packet threshold that you defined for the class is reached, any more packets destined for the class queue are dropped according to the tail drop or WRED mechanism.

You configure tail drop by using the **queue-limit** policy-map class configuration command in an egress policy-map attached to an ES port. This command configures the maximum threshold for tail drop. Packets are queued until the maximum threshold is exceeded, and then all the packets are dropped.

Before beginning this procedure, make sure that you have reviewed the configuration guidelines and have created the class map to isolate traffic. For more information, see the "Hierarchical QoS Configuration Guidelines" section on page 26-76 and the "Classifying Egress Traffic by Using Class Maps" section on page 26-78. For information on how to configure WRED, see the "Configuring CBWFQ and DSCP-Based WRED" section on page 26-89.

Beginning in privileged EXEC mode, follow these steps to configure class-level CBWFQ and tail drop in a service policy for outbound traffic. This procedure is optional. The examples that follow the procedure show how to configure class-level and VLAN-level CBWFQ and tail drop.

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

	Command	Purpose
Step 4	**bandwidth** {*bandwidth-kbps* | **percent** *percent*}	Specify the minimum bandwidth provided to a class belonging to the egress policy-map when there is traffic congestion in the switch. If the switch is not congested, the class receives more bandwidth than you specify with the **bandwidth** command. CBWFQ derives the weight for packets belonging to the class from the bandwidth allocated to the class. CBWFQ then uses the weight to ensure that the queue for the class is serviced fairly. By default, no bandwidth is specified. You can specify the bandwidth in kbps or as a percentage: • For *bandwidth-kbps*, specify the bandwidth amount in kbps assigned to the class. The range is 200 to 2000000. Allocate the bandwidth in 100-kbps increments; otherwise, the software rounds down the bandwidth to the nearest 100-kbps increment. • For **percent** *percent*, specify the percentage of available bandwidth assigned to the class. The range is 1 to 100. The sum of the class bandwidth percentages within a single policy map cannot exceed 99 percent. Percentage calculations are based on the bandwidth available at the parent class (or the physical level if it is the parent). Specify all the class bandwidths in either kbps or in percentages, but not a mix of both. The amount of bandwidth configured should be large enough to accommodate Layer 2 overhead.
Step 5	**queue-limit** *limit*	Configure the maximum threshold for tail drop. For *limit*, the range is 1 to 32768 packets. The default is 128 packets.
Step 6	**exit**	Return to policy-map configuration mode.
Step 7	**exit**	Return to global configuration mode.
Step 8	**interface** *interface-id*	Specify an ES port to attach to the policy map, and enter interface configuration mode.
Step 9	**service-policy output** *policy-map-name*	Specify the egress policy-map name, and apply it to the ES port. Only one policy map per port is supported.
Step 10	**end**	Return to privileged EXEC mode.
Step 11	**show policy-map** [*policy-map-name* [**class** *class-map-name*]] or **show policy-map interface** *interface-id*	Verify your entries.
Step 12	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

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

This example shows how to create a class-level policy-map called *policy11* for three classes called *prec1*, *prec2*, and *prec3*. In the policy for these classes, 30 percent of the available bandwidth is assigned to the queue for the first class, 20 percent is assigned to the queue for the second class, and 10 percent is assigned to the queue for the third class. Tail drop is enabled on each class queue, packets are queued until the maximum threshold of 2000 packets is exceeded, and then all the packets are dropped.

```
Switch(config)# class-map prec1
Switch(config-cmap)# match ip precedence 1
Switch(config-cmap)# exit
Switch(config)# class-map prec2
Switch(config-cmap)# match ip precedence 2
Switch(config-cmap)# exit
Switch(config)# class-map prec3
Switch(config-cmap)# match ip precedence 3
Switch(config-cmap)# exit
Switch(config)# policy-map policy11
Switch(config-pmap)# class prec1
Switch(config-pmap-c)# bandwidth percent 30
Switch(config-pmap-c)# queue-limit 2000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class prec2
Switch(config-pmap-c)# bandwidth percent 20
Switch(config-pmap-c)# queue-limit 2000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class prec3
Switch(config-pmap-c)# bandwidth percent 10
Switch(config-pmap-c)# queue-limit 2000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1/1
Switch(config-if)# service-policy output policy11
```

This example shows how to create a VLAN-level policy-map called *vlan-policy* for two classes called *vlan203* and *vlan202.* In the policy for these classes, the minimum bandwidth is set to 2000 kbps.

```
Switch(config)# class-map match-all vlan203
Switch(config-cmap)# match vlan 203
Switch(config-cmap)# match vlan inner 206
Switch(config-cmap)# exit
Switch(config)# class-map match-all vlan202
Switch(config-cmap)# match vlan 202
Switch(config-cmap)# match vlan inner 204
Switch(config-cmap)# exit
Switch(config)# policy-map vlan-policy
Switch(config-pmap)# class vlan203
Switch(config-pmap-c)# bandwidth 2000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class vlan202
Switch(config-pmap-c)# bandwidth 2000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1/2
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# service-policy output vlan-policy
```

This example shows how to create a VLAN-level policy-map called *log-policy*. It matches packets with VLAN 203 and associates a class-level child policy called *cls-policy*. The child policy matches packets with a CoS value of 2, sets 50 percent as the available bandwidth for this class, and sets 2000 packets as the maximum threshold for tail drop. Packets are queued until the maximum threshold is exceeded, and then all the packets are dropped. Note that when you configure the **bandwidth** command in a class policy, you also must configure the **bandwidth** or **shape** policy-map class configuration command in the parent VLAN-level policy.

```
Switch(config)# class-map cls-class
Switch(config-cmap)# match cos 2
Switch(config-cmap)# exit
Switch(config)# class-map log-class
Switch(config-cmalp)# match vlan 203
Switch(config-cmap)# exit
Switch(config)# policy-map cls-policy
Switch(config-pmap)# class cls-class
Switch(config-pmap-c)# bandwidth percent 50
Switch(config-pmap-c)# queue-limit 2000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# policy-map log-policy
Switch(config-pmap)# class log-class
Switch(config-pmap-c)# bandwidth percent 50
Switch(config-pmap-c)# service-policy cls-policy
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1/2
Switch(config-if)# switchport trunk encapsulation isl
Switch(config-if)# switchport mode trunk
Switch(config-if)# service-policy output log-policy
```

Configuring CBWFQ and DSCP-Based WRED

CBWFQ creates a queue for every class for which a class map is defined. Packets satisfying the match criteria for a class accumulate in the queue reserved for the class until they are sent, which occurs when the queue is serviced by the fair queue process. When the maximum packet threshold that you defined for the class is reached, any more packets destined for the class queue are dropped according to the tail drop or the WRED mechanism.

WRED reduces the chances of tail drop by selectively dropping packets when the port begins to show signs of congestion. By dropping some packets early rather than waiting until the queue is full, WRED avoids dropping large numbers of packets at once.

When a packet arrives and WRED is enabled, these events occur:

- The average queue size is calculate based on the previous average and the current size of the queue. The average queue-size calculation is affected by the exponential weight constant setting in the **random-detect exponential-weight-constant** policy-map class configuration command.
- If the average queue size is less than the minimum queue threshold, the arriving packet is queued. The minimum queue threshold is configured through the *min-threshold* option in the **random-detect dscp** policy-map class configuration command.

- If the average queue size is between the minimum queue threshold and the maximum queue threshold, the packet is either dropped or queued, depending on the packet-drop probability. The packet-drop probability is based on the minimum threshold, the maximum threshold, and the mark-probability denominator. The maximum queue threshold is configured through the *max-threshold* option, and the mark-probability denominator is configured through the *mark-prob-denominator* option in the **random-detect dscp** policy-map class configuration command.
- If the average queue size is greater than the maximum queue threshold, the packet is automatically dropped.

You enable DSCP-based WRED by using the **random-detect dscp-based** policy-map class configuration command in an egress policy-map attached to an ES port. This command allows for preferential drop treatment among packets with different DSCP values. The WRED algorithm discards or marks packets destined for a queue when that queue is congested. It discards packets fairly and before the queue is full. If you want to enable IP precedence-based WRED instead of DSCP-based WRED, see the "Configuring CBWFQ and IP Precedence-Based WRED" section on page 26-93.

Before beginning this procedure, make sure that you have reviewed the configuration guidelines and have created the class map to isolate traffic. For more information, see the "Hierarchical QoS Configuration Guidelines" section on page 26-76 and the "Classifying Egress Traffic by Using Class Maps" section on page 26-78. For information on how to configure tail drop, see the "Configuring CBWFQ and Tail Drop" section on page 26-86.

Beginning in privileged EXEC mode, follow these steps to configure class-level CBWFQ and DSCP-based WRED in a service policy for outbound traffic. This procedure is optional. The examples that follow the procedure show how to configure class-level and VLAN-level CBWFQ and DSCP-based WRED.

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

	Command	Purpose
Step 4	**bandwidth** {*bandwidth-kbps* | **percent** *percent*}	Specify the minimum bandwidth provided to a class belonging to the egress policy-map when there is traffic congestion in the switch. If the switch is not congested, the class receives more bandwidth than you specify with the **bandwidth** command. CBWFQ derives the weight for packets belonging to the class from the bandwidth allocated to the class. CBWFQ then uses the weight to ensure that the queue for the class is serviced fairly. By default, no bandwidth is specified. You can specify the bandwidth in kbps or as a percentage: • For *bandwidth-kbps*, specify the bandwidth amount in kbps assigned to the class. The range is 200 to 2000000. Allocate the bandwidth in 100-kbps increments; otherwise, the software rounds down the bandwidth to the nearest 100-kbps increment. • For **percent** *percent*, specify the percentage of available bandwidth assigned to the class. The range is 1 to 100. The sum of the class bandwidth percentages within a single policy map cannot exceed 99 percent. Percentage calculations are based on the bandwidth available at the parent class (or the physical level if it is the parent). Specify all the class bandwidths in either kbps or in percentages, but not a mix of both. The amount of bandwidth configured should be large enough to accommodate Layer 2 overhead.
Step 5	**random-detect dscp-based**	Enable DSCP-based WRED as a drop policy. By default, WRED is disabled.
Step 6	**random-detect dscp** *dscp min-threshold max-threshold mark-prob-denominator*	Specify packet threshold and mark-probability values for a specific DSCP value: • For *dscp*, enter a DSCP value. The range is 0 to 63. You also can enter a mnemonic name for a commonly used value. • For *min-threshold*, enter the minimum threshold in packets. The range is 1 to 32768. When the average queue size reaches the minimum threshold, WRED randomly drops some packets with the specified DSCP value. • For *max-threshold*, enter the maximum threshold in packets. The range is 1 to 32768. The default is 128. When the average queue size exceeds the maximum threshold, WRED drops all packets with the specified DSCP value. • For *mark-prob-denominator*, enter the denominator for the fraction of packets dropped when the average queue size is at the maximum threshold. The range is 1 to 65535. The default is 10. For example, if the denominator is 512, one out of every 512 packets is dropped when the queue is at the maximum threshold. For a list of the default settings for a specified DSCP value, see the command reference for this release.

	Command	Purpose
Step 7	**random-detect exponential-weighting-constant** *weight*	(Optional) Configure the exponential weight factor for the average queue-size calculation for WRED. The range is 1 to 16. The default is 9.
Step 8	**exit**	Return to policy-map configuration mode.
Step 9	**exit**	Return to global configuration mode.
Step 10	**interface** *interface-id*	Specify an ES port to attach to the policy map, and enter interface configuration mode.
Step 11	**service-policy output** *policy-map-name*	Specify the egress policy-map name, and apply it to the ES port. Only one policy map per port is supported.
Step 12	**end**	Return to privileged EXEC mode.
Step 13	**show policy-map** [*policy-map-name* [**class** *class-map-name*]] or **show policy-map interface** *interface-id*	Verify your entries.
Step 14	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To delete an existing policy map, use the **no policy-map** *policy-map-name* global configuration command. To delete an existing class, use the **no class** *class-name* policy-map configuration command. To return to the default bandwidth, use the **no bandwidth** policy-map class configuration command. To disable DSCP-based WRED, use the **no random-detect dscp-based** policy-map class configuration command. To return to the default WRED settings, use the **no random-detect dscp** *dscp* policy-map class configuration command.

This example shows how to configure a class-level policy called *policy10*. Class *c1* has these characteristics: a minimum of 2000 kbps of bandwidth is expected to be delivered to this class in the event of congestion, and a weight factor of 10 is used to calculate the average queue size. To avoid congestion, DSCP-based WRED packet drop is used instead of tail drop. The minimum threshold for DSCP value 8 is 24, the maximum threshold is 40, and the mark-probability denominator is 512.

```
Switch(config)# class-map c1
Switch(config-cmap)# match ip dscp 8
Switch(config-cmap)# exit
Switch(config)# policy-map policy10
Switch(config-pmap)# class c1
Switch(config-pmap-c)# bandwidth 2000
Switch(config-pmap-c)# random-detect dscp-based
Switch(config-pmap-c)# random-detect exponential-weighting-constant 10
Switch(config-pmap-c)# random-detect dscp 8 24 40 512
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1/1
Switch(config-if)# switchport trunk encapsulation isl
Switch(config-if)# switchport mode trunk
Switch(config-if)# service-policy output policy10
```

This example shows how to configure a VLAN-level policy called *parent*. It matches packets with VLAN 101 and associates a class-level child policy called *policy1*. The child policy matches two DSCP values in two classes. Thirty percentage of the available bandwidth is assigned to the gold class, 20 percent is assigned to the silver class, and 20 percent is assigned to vlan101. DSCP-based WRED is used as the drop policy. For the af11 value in the gold class, WRED randomly drops packets with this DSCP when the minimum threshold reaches 30. When the average queue size exceeds the maximum threshold of 40, WRED drops all packets with DSCP af11. The mark-probability denominator is set to 10, which means that one out of every 10 packets is dropped when the average queue is at the maximum threshold. The configuration has similar settings for af12 in the gold class and for af21 and af22 in the silver class. Note that when you configure the **bandwidth** command in a class policy, you also must configure the **bandwidth** or **shape** policy-map class configuration command in the parent VLAN-level policy.

```
Switch(config)# class-map gold
Switch(config-cmap)# match ip dscp af11 af12
Switch(config-cmap)# exit
Switch(config)# class-map silver
Switch(config-cmap)# match ip dscp af21 af22
Switch(config-cmap)# exit
Switch(config)# class-map vlan101
Switch(config-cmap)# match vlan 101
Switch(config-cmap)# exit
Switch(config)# policy-map policy1
Switch(config-pmap)# class gold
Switch(config-pmap-c)# bandwidth percent 30
Switch(config-pmap-c)# random-detect dscp-based
Switch(config-pmap-c)# random-detect dscp af11 30 40 10
Switch(config-pmap-c)# random-detect dscp af12 25 40 10
Switch(config-pmap-c)# exit
Switch(config-pmap)# class silver
Switch(config-pmap-c)# bandwidth percent 20
Switch(config-pmap-c)# random-detect dscp-based
Switch(config-pmap-c)# random-detect dscp af21 28 35 10
Switch(config-pmap-c)# random-detect dscp af22 26 35 10
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# policy-map parent
Switch(config-pmap)# class vlan101
Switch(config-pmap-c)# bandwidth percent 30
Switch(config-pmap-c)# service-policy policy1
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1/1
Switch(config-if)# service-policy output parent
```

Configuring CBWFQ and IP Precedence-Based WRED

CBWFQ creates a queue for every class for which a class map is defined. Packets satisfying the match criteria for a class accumulate in the queue reserved for the class until they are sent, which occurs when the queue is serviced by the fair queue process. When the maximum packet threshold you defined for the class is reached, any more packets destined for the class queue are dropped according to the tail drop or the WRED mechanism.

WRED reduces the chances of tail drop by selectively dropping packets when the port begins to show signs of congestion. By dropping some packets early rather than waiting until the queue is full, WRED avoids dropping large numbers of packets at once.

You enable IP precedence-based WRED by using the **random-detect precedence-based** policy-map class configuration command in an egress policy-map attached to an ES port. This command allows for preferential drop treatment among packets with different IP precedence values. The WRED algorithm discards or marks packets destined for a queue when that queue is congested. It discards packets fairly and before the queue is full. Packets with high IP-precedence values are preferred over packets with low IP-precedence values. If you want to enable DSCP-based WRED instead of IP precedence-based WRED, see the "Configuring CBWFQ and DSCP-Based WRED" section on page 26-89

Before beginning this procedure, make sure that you have reviewed the configuration guidelines and have created the class map to isolate traffic. For more information, see the "Hierarchical QoS Configuration Guidelines" section on page 26-76 and the "Classifying Egress Traffic by Using Class Maps" section on page 26-78. For information on how to configure tail drop, see the "Configuring CBWFQ and Tail Drop" section on page 26-86.

Beginning in privileged EXEC mode, follow these steps to configure class-level CBWFQ and IP precedence-based WRED in a service policy for outbound traffic. This procedure is optional. The examples that follow the procedure show how to configure class-level and VLAN-level CBWFQ and IP precedence-based WRED.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**policy-map** *policy-map-name*	Create a policy map by entering the policy-map name, and enter policy-map configuration mode. By default, no policy maps are defined.
Step 3	**class** *class-name*	Specify the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.
Step 4	**bandwidth** {*bandwidth-kbps* | **percent** *percent*}	Specify the minimum bandwidth provided to a class belonging to the egress policy-map when there is traffic congestion in the switch. If the switch is not congested, the class receives more bandwidth than you specify with the **bandwidth** command. CBWFQ derives the weight for packets belonging to the class from the bandwidth allocated to the class. CBWFQ then uses the weight to ensure that the queue for the class is serviced fairly. By default, no bandwidth is specified. You can specify the bandwidth in kbps or as a percentage: • For *bandwidth-kbps*, specify the bandwidth amount in kbps assigned to the class. The range is 200 to 2000000. Allocate the bandwidth in 100-kbps increments; otherwise, the software rounds down the bandwidth to the nearest 100-kbps increment. • For **percent** *percent*, specify the percentage of available bandwidth assigned to the class. The range is 1 to 100. The sum of the class bandwidth percentages within a single policy map cannot exceed 99 percent. Percentage calculations are based on the bandwidth available at the parent class (or the physical level if it is the parent). Specify all the class bandwidths in either kbps or in percentages, but not a mix of both. The amount of bandwidth configured should be large enough to accommodate Layer 2 overhead.
Step 5	**random-detect precedence-based**	Enable IP precedence-based WRED as a drop policy. By default, WRED is disabled.

	Command	Purpose
Step 6	**random-detect precedence** *ip-precedence min-threshold max-threshold mark-prob-denominator*	Specify packet threshold and mark-probability values for a specific IP precedence value: • For *ip-precedence*, specify an IP precedence value. The range is 0 to 7. • For *min-threshold*, specify the minimum threshold in packets. The range is 1 to 32768. When the average queue size reaches the minimum threshold, WRED randomly drops some packets with the specified precedence value. • For *max-threshold*, specify the maximum threshold in packets. The range is 1 to 32768. The default is 128. When the average queue size exceeds the maximum threshold, WRED drops all packets with the specified precedence value. • For *mark-prob-denominator*, specify the denominator for the fraction of packets dropped when the average queue size is at the maximum threshold. The range is 1 to 65535. The default is 10. For example, if the denominator is 512, one out of every 512 packets is dropped when the queue is at the maximum threshold. For a list of the default settings for a specified IP precedence value, see the command reference for this release.
Step 7	**random-detect exponential-weighting-constant** *weight*	(Optional) Configure the exponential weight factor for the average queue-size calculation for WRED. The range is 1 to 16. The default is 9.
Step 8	**exit**	Return to policy-map configuration mode.
Step 9	**exit**	Return to global configuration mode.
Step 10	**interface** *interface-id*	Specify an ES port to attach to the policy map, and enter interface configuration mode.
Step 11	**service-policy output** *policy-map-name*	Specify the egress policy-map name, and apply it to the ES port. Only one policy map per port is supported.
Step 12	**end**	Return to privileged EXEC mode.
Step 13	**show policy-map** [*policy-map-name* [**class** *class-map-name*]] or **show policy-map interface** *interface-id*	Verify your entries.
Step 14	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To delete an existing policy map, use the **no policy-map** *policy-map-name* global configuration command. To delete an existing class, use the **no class** *class-name* policy-map configuration command. To return to the default bandwidth, use the **no bandwidth** policy-map class configuration command. To disable IP precedence-based WRED, use the **no random-detect precedence-based** policy-map class configuration command. To return to the default WRED settings, use the **no random-detect precedence** *ip-precedence* policy-map class configuration command.

This example shows how to configure a class-level policy called *policy10*. Class *c1* has these characteristics: a minimum of 2000 kbps of bandwidth is expected to be delivered to this class in the event of congestion, and a weight factor of 10 is used to calculate the average queue size. To avoid congestion, WRED packet drop is used instead of tail drop. IP precedence is reset for levels 0 to 2. The minimum threshold for IP precedence value 0 is 32, the maximum threshold is 256, and the mark-probability denominator is 100. IP precedence values 1 and 2 have similar thresholds and probability denominators.

```
Switch(config)# class-map c1
Switch(config-cmap)# match ip precedence 0 1 2
Switch(config-cmap)# exit
Switch(config)# policy-map policy10
Switch(config-pmap)# class c1
Switch(config-pmap-c)# bandwidth 2000
Switch(config-pmap-c)# random-detect precedence-based
Switch(config-pmap-c)# random-detect exponential-weighting-constant 10
Switch(config-pmap-c)# random-detect precedence 0 32 256 100
Switch(config-pmap-c)# random-detect precedence 1 64 256 100
Switch(config-pmap-c)# random-detect precedence 2 96 256 100
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1/1
Switch(config-if)# switchport trunk encapsulation isl
Switch(config-if)# switchport mode trunk
Switch(config-if)# service-policy output policy10
```

This example shows how to configure a VLAN-level policy called *parent*. It matches packets with VLAN 101 and associates a class-level child policy called *policy1*. The child policy matches two IP precedence values in two classes. Thirty percentage of the available bandwidth is assigned to the gold class, 20 percent is assigned to the silver class, and 30 percent is assigned to vlan101. IP precedence-based WRED is used as the drop policy. For the IP precedence 0 in the gold class, WRED randomly drops packets with this IP precedence when the minimum threshold reaches 30. When the average queue size exceeds the maximum threshold of 40, WRED drops all packets with IP precedence 0. The mark-probability denominator is set to 10, which means that one out of every 10 packets is dropped when the average queue is at the maximum threshold. The configuration has similar settings for IP precedence 3 in the silver class. Note that when you configure the **bandwidth** command in a class policy, you also must configure the **bandwidth** or **shape** policy-map class configuration command in the parent VLAN-level policy.

```
Switch(config)# class-map gold
Switch(config-cmap)# match ip precedence 0
Switch(config-cmap)# exit
Switch(config)# class-map silver
Switch(config-cmap)# match ip precedence 3
Switch(config-cmap)# exit
Switch(config)# class-map vlan101
Switch(config-cmap)# match vlan 101
Switch(config-cmap)# exit
Switch(config)# policy-map policy1
Switch(config-pmap)# class gold
Switch(config-pmap-c)# bandwidth percent 30
Switch(config-pmap-c)# random-detect precedence-based
Switch(config-pmap-c)# random-detect precedence 0 30 40 10
Switch(config-pmap-c)# exit
Switch(config-pmap)# class silver
Switch(config-pmap-c)# bandwidth percent 20
Switch(config-pmap-c)# random-detect precedence-based
Switch(config-pmap-c)# random-detect precedence 3 28 35 10
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# policy-map parent
Switch(config-pmap)# class vlan101
Switch(config-pmap-c)# bandwidth percent 30
Switch(config-pmap-c)# service-policy policy1
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1/1
Switch(config-if)# service-policy output parent
```

Enabling LLQ

LLQ provides strict-priority queueing for a traffic class. It enables delay-sensitive data, such as voice, to be sent before packets in other queues are sent. The priority queue is serviced first until it is empty. Only one traffic stream can be destined for the priority queue per class-level policy. The priority queue restricts all traffic streams in the same hierarchy, and you should use care when configuring this feature. You enable the priority queue for a traffic class by using the **priority** policy-map class configuration command in an egress policy-map attached to an ES port.

Before beginning this procedure, make sure that you have reviewed the configuration guidelines and have created the class map to isolate traffic. For more information, see the "Hierarchical QoS Configuration Guidelines" section on page 26-76 and the "Classifying Egress Traffic by Using Class Maps" section on page 26-78.

Beginning in privileged EXEC mode, follow these steps to enable class-level priority queueing in a service policy for outbound traffic. This procedure is optional. The examples that follow the procedure show how to enable class-level and VLAN-level priority queueing.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**policy-map** *policy-map-name*	Create a policy map by entering the policy-map name, and enter policy-map configuration mode. By default, no policy maps are defined.
Step 3	**class** *class-name*	Specify the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.
Step 4	**priority**	Enable the strict priority queue, and give priority to a class of traffic. By default, strict priority queueing is disabled.
Step 5	**exit**	Return to policy-map configuration mode.
Step 6	**exit**	Return to global configuration mode.
Step 7	**interface** *interface-id*	Specify an ES port to attach to the policy map, and enter interface configuration mode.
Step 8	**service-policy output** *policy-map-name*	Specify the egress policy-map name, and apply it to the ES port. Only one policy map per port is supported.
Step 9	**end**	Return to privileged EXEC mode.

	Command	Purpose
Step 10	**show policy-map** [*policy-map-name* [**class** *class-map-name*]] or **show policy-map interface** *interface-id*	Verify your entries.
Step 11	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

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

This example shows how to configure a class-level policy called *policy1* that has two classes. Class 1 has 10 percent of the available bandwidth. Class 2 is configured as the priority queue, which is serviced first until it is empty.

```
Switch(config)# class-map class1
Switch(config-cmap)# match mpls experimental 2 3 4
Switch(config-cmap)# exit
Switch(config)# class-map class2
Switch(config-cmap)# match mpls experimental 7
Switch(config-cmap)# exit
Switch(config)# policy-map policy1
Switch(config-pmap)# class class1
Switch(config-pmap-c)# bandwidth percent 10
Switch(config-pmap-c)# exit
Switch(config-pmap)# class class2
Switch(config-pmap-c)# priority
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1/2
Switch(config-if)# service-policy output policy1
```

This example shows how to configure a VLAN-level policy called *parent*. It matches packets with VLAN 101 and associates a class-level child policy called *policy1*. The gold class is configured as the priority queue, whereas the silver class and vlan101 class each have 20 percent of the available bandwidth assigned to them. DSCP-based WRED is used as the drop policy. For the af21 value in the silver class, WRED randomly drops packets with this DSCP when the minimum threshold reaches 28. When the average queue size exceeds the maximum threshold of 35, WRED drops all packets with DSCP af21. The mark-probability denominator is set to 10, which means that one out of every 10 packets is dropped when the average queue is at the maximum threshold. The configuration has similar settings for af22 in the silver class.

```
Switch(config)# class-map gold
Switch(config-cmap)# match ip dscp af11 af12
Switch(config-cmap)# exit
Switch(config)# class-map silver
Switch(config-cmap)# match ip dscp af21 af22
Switch(config-cmap)# exit
Switch(config)# class-map vlan101
Switch(config-cmap)# match vlan 101
Switch(config-cmap)# exit
Switch(config)# policy-map policy1
Switch(config-pmap)# class gold
Switch(config-pmap-c)# priority
Switch(config-pmap-c)# exit
Switch(config-pmap)# class silver
Switch(config-pmap-c)# bandwidth percent 20
Switch(config-pmap-c)# random-detect dscp-based
Switch(config-pmap-c)# random-detect dscp af21 28 35 10
Switch(config-pmap-c)# random-detect dscp af22 26 35 10
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# policy-map parent
Switch(config-pmap)# class vlan101
Switch(config-pmap-c)# bandwidth percent 20
Switch(config-pmap-c)# service-policy policy1
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1/1
Switch(config-if)# service-policy output parent
```

Configuring Shaping

Shaping provides a process for delaying out-of-profile packets in queues so that they conform to a specified profile. Shaping is distinct from policing. Policing drops packets that exceed a configured threshold, but shaping buffers packets so that traffic remains within a threshold. Shaping offers greater smoothness in handling traffic than policing. You can configure average-rate traffic shaping on a traffic class within a policy map at the class level, at the VLAN level, and at the physical level by using the **shape** policy-map class configuration command. At the physical level of the hierarchy, you can shape only the class-default class in an egress policy attached to an ES port.

Before beginning this procedure, make sure that you have reviewed the configuration guidelines and have created the class map to isolate traffic. For more information, see the "Hierarchical QoS Configuration Guidelines" section on page 26-76 and the "Classifying Egress Traffic by Using Class Maps" section on page 26-78.

Beginning in privileged EXEC mode, follow these step to configure class-level shaping in a service policy for outbound traffic. This procedure is optional. The examples that follow the procedure show how to configure class-level, VLAN-level, and physical-level shaping.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**policy-map** *policy-map-name*	Create a policy map by entering the policy-map name, and enter policy-map configuration mode. By default, no policy maps are defined.
Step 3	**class** *class-name*	Specify the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.
Step 4	**shape average** *cir-bps*	Enable average-rate traffic shaping. Specify the committed information rate, the bit rate that traffic is shaped to, in bps. This is the access bit rate that you have contracted for with your service provider or the service levels that you intend to maintain. The range is 64000 to 2000000000 bps. Allocate the shaped rate in 100-kbps increments; otherwise, the software rounds down the bandwidth to the nearest 100-kbps increment. By default, average-rate traffic shaping is disabled.
Step 5	**exit**	Return to policy-map configuration mode.
Step 6	**exit**	Return to global configuration mode.

	Command	Purpose
Step 7	**interface** *interface-id*	Specify an ES port to attach to the policy map, and enter interface configuration mode.
Step 8	**service-policy output** *policy-map-name*	Specify the egress policy-map name, and apply it to the ES port. Only one policy map per port is supported.
Step 9	**end**	Return to privileged EXEC mode.
Step 10	**show policy-map** [*policy-map-name* [**class** *class-map-name*]] or **show policy-map interface** *interface-id*	Verify your entries.
Step 11	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

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

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

```
Switch(config)# class-map class1
Switch(config-cmap)# match cos 0 1 2 3
Switch(config-cmap)# exit
Switch(config)# policy-map policy1
Switch(config-pmap)# class class1
Switch(config-pmap-c)# shape average 256000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1/1
Switch(config-if)# service-policy output policy1
```

This example shows how to configure VLAN-level, average-rate shaping. It limits each traffic class, *vlan101* and *vlan102*, to a data transmission rate of 400 Mbps.

```
Switch(config)# class-map match-all vlan101
Switch(config-cmap)# match vlan 101
Switch(config-cmap)# exit
Switch(config)# class-map match-all vlan102
Switch(config-cmap)# match vlan 102
Switch(config-cmap)# exit
Switch(config)# policy-map vlan-policy
Switch(config-pmap)# class vlan101
Switch(config-pmap-c)# shape average 400000000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class vlan102
Switch(config-pmap-c)# shape average 400000000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1/1
Switch(config-if)# switchport trunk encapsulation isl
Switch(config-if)# switchport mode trunk
Switch(config-if)# service-policy output vlan-policy
```

This example shows how to shape the default class. This configuration associates a class-level policy-map with a VLAN-level policy-map and then associates the VLAN-level policy-map with a physical-level policy-map.

```
Switch(config)# class-map my-class
Switch(config-cmap)# match ip precedence 1
Switch(config-cmap)# exit
Switch(config)# class-map my-logical-class
Switch(config-cmap)# match vlan 5
Switch(config-cmap)# exit
Switch(config)# policy-map my-class-policy
Switch(config-pmap)# class my-class
Switch(config-pmap-c)# set ip precedence 2
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# policy-map my-logical-policy
Switch(config-pmap)# class my-logical-class
Switch(config-pmap-c)# shape average 400000000
Switch(config-pmap-c)# service-policy my-class-policy
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# policy-map my-physical-policy
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# shape average 500000000
Switch(config-pmap-c)# service-policy my-logical-policy
```

Displaying Hierarchical QoS Information

To display hierarchical QoS information, use one or more of the privileged EXEC commands in Table 26-16:

Table 26-16 Commands for Displaying Hierarchical QoS Information

Command	Purpose
show class-map [*class-map-name*]	Display QoS class maps, which define the match criteria to classify traffic.
show mls qos	Display global QoS configuration information.
show policy-map [*policy-map-name* [**class** *class-map-name*]]	Display QoS policy-maps, which define the traffic policy for a traffic class.
show policy-map interface *interface-id* **output** [**class** *class-name*]]	Display QoS policy-map information for the specified ES port, and display statistics for an individual class.