Configuring QoS

- Finding Feature Information, on page 1
- Prerequisites for QoS, on page 1
- Restrictions for QoS, on page 3
- Information About QoS, on page 4
- How to Configure QoS, on page 29
- Monitoring Standard QoS, on page 82
- Configuration Examples for QoS, on page 83
- Where to Go Next, on page 91
- Additional References, on page 91
- Feature History and Information for QoS, on page 92

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for 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. For example, is the traffic on your network 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.

QoS ACL Guidelines

Follow these guidelines when configuring QoS with access control lists (ACLs):

- 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 per class map and only one **match** class-map configuration command per class map are supported. The ACL can have multiple ACEs, which match fields against the contents of the packet.
- A trust statement in a policy map requires multiple hardware entries per ACL line. If an input service policy map contains a trust statement in an ACL, the access list might be too large to fit into the available QoS hardware memory, and an error can occur when you apply the policy map to a port. Whenever possible, you should minimize the number of lines is a QoS ACL.

Policing Guidelines

Note To use policing, the switch must be running the LAN Base image.

- The port ASIC device, which controls more than one physical port, supports 256 policers (255 user-configurable policers plus 1 policer reserved for system internal use). The maximum number of user-configurable policers supported per port is 63. 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.
- Only one policer is applied to a packet on an ingress port. Only the average rate and committed burst parameters are configurable.
- 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.
- 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. You must decide whether the QoS configuration should match on all ports in the EtherChannel.
- If you need to modify a policy map of an existing QoS policy, first remove the policy map from all interfaces, and then modify or copy the policy map. After you finish the modification, apply the modified policy map to the interfaces. If you do not first remove the policy map from all interfaces, high CPU usage can occur, which, in turn, can cause the console to pause for a very long time.

General QoS Guidelines

These are the general QoS guidelines:

- You configure QoS only on physical ports; there is no support for it at the VLAN level.
- 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.
- The switch supports homogeneous stacking and mixed stacking. Mixed stacking is supported only with the Catalyst 2960-S switches. A homogenous stack can have up to eight stack members, while a mixed stack can have up to four stack members. All switches in a switch stack must be running the LAN Base image.

Restrictions for QoS

The following are the restrictions for QoS:

- To use these features, the switch must be running the LAN Base image: stacking, DSCP, auto-QoS, trusted boundary, policing, marking, mapping tables, and weighted tail drop.
- Ingress queueing is not supported.
- The switch supports 4 default egress queues, with the option to enable an additional 4 egress queues for a total of 8. This option is only available on a standalone switch running the LAN Base image.
- We recommend that you do not enable 8 egress queues by using the **mls qos srr-queue output queues 8** command, when running the following features in your configuration:
  - Auto-QoS
  - Auto SmartPort
  - EnergyWise

  Running these features with 8 egress queue enabled in a single configuration is not supported on the switch.
- You can configure QoS only on physical ports. VLAN-based QoS is not supported. You configure the QoS settings, such as classification, queueing, and scheduling, and apply the policy map to a port. When configuring QoS on a physical port, you apply a nonhierarchical policy map to a port.
- If the switch is running the LAN Lite image you can:
  - Configure ACLs, but you cannot attach them to physical interfaces. You can attach them to VLAN interfaces to filter traffic to the CPU.
  - Enable only cos trust at interface level.
  - Enable SRR shaping and sharing at interface level.
  - Enable Priority queueing at interface level.
  - Enable or disable **mls qos rewrite ip dscp**.
- The switch must be running the LAN Base image to use the following QoS features:
  - Policy maps
  - Policing and marking
  - Mapping tables

- WTD

Information About QoS

QoS Implementation

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.

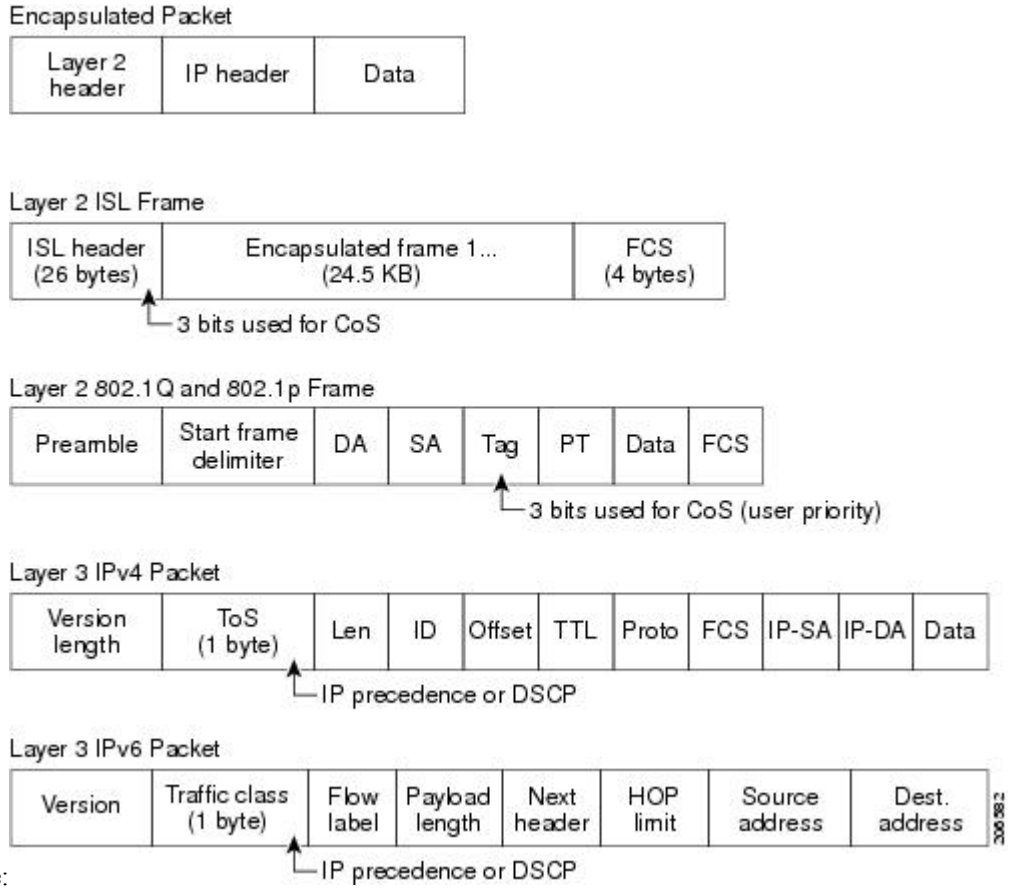
When you configure the QoS feature, you can select specific network traffic, prioritize it according to its relative importance, and use congestion-management and congestion-avoidance techniques to provide preferential treatment. Implementing QoS in your network makes network performance more predictable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

The QoS implementation is based on the Differentiated Services (Diff-Serv) architecture, a standard from the Internet Engineering Task Force (IETF). This architecture specifies that each packet is classified upon entry into the network.

The classification is carried in the IP packet header, using 6 bits from the deprecated IP type of service (ToS) field to carry the classification (*class*) information. Classification can also be carried in the Layer 2 frame.

Figure 1: QoS Classification Layers in Frames and Packets

The special bits in the Layer 2 frame or a Layer 3 packet are shown in the following figure:



Layer 2 Frame Prioritization Bits

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

Layer 3 Packet Prioritization Bits

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

IP precedence values range from 0 to 7. DSCP values range from 0 to 63.

End-to-End QoS Solution Using Classification

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 occur closer to the edge of the network, so that the core switches and routers are not overloaded with this task.

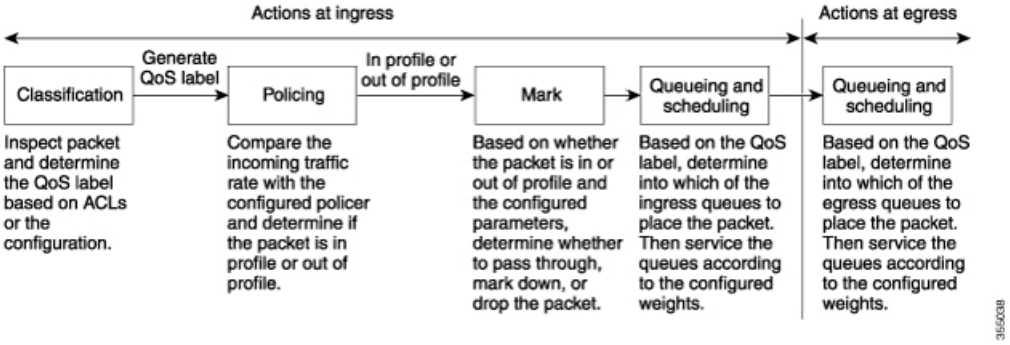
Switches and routers along the path can use the class information to limit the amount of resources allocated per traffic class. The behavior of an individual device when handling traffic in the Diff-Serv architecture is called per-hop behavior. If all devices along a path provide a consistent per-hop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

Implementing QoS in your network can be a simple task 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 incoming and outgoing traffic.

QoS Basic Model

To implement QoS, the switch must distinguish packets or flows from one another (classify), 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 2: QoS Basic Wired Model



Actions at Ingress Port

Actions at the ingress port include classifying traffic, policing, marking, and scheduling:

- Classifying 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.
- Policing determines whether a packet is in or out of profile by comparing the rate of the incoming traffic to the configured policer. The policer limits the bandwidth consumed by a flow of traffic. The result is passed to the marker.

- Marking evaluates the policer and configuration information for the action to be taken when a packet is out of profile and determines what to do with the packet (pass through a packet without modification, marking down the QoS label in the packet, or dropping the packet).

Note Queueing and scheduling are only supported at egress and not at ingress on the switch.

Actions at Egress Port

Actions at the egress port include queueing and scheduling:

- Queueing evaluates the QoS packet label and the corresponding DSCP or CoS value before selecting which of the four egress queues to use. Because congestion can occur when multiple ingress ports simultaneously send data to an egress port, WTD differentiates traffic classes and subjects the packets to different thresholds based on the QoS label. If the threshold is exceeded, the packet is dropped.
- Scheduling services the four egress queues based on their configured SRR shared or shaped weights. One of the queues (queue 1) can be the expedited queue, which is serviced until empty before the other queues are serviced.

Classification Overview

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

During 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 queuing and scheduling actions to perform on the packet. The label is mapped according to the trust setting and the packet type as shown in the Classification Flowchart.

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

Non-IP Traffic Classification

The following table describes the non-IP traffic classification options for your QoS configuration.

Table 1: Non- IP Traffic Classifications

Non-IP Traffic Classification	Description
Trust the CoS value	Trust the CoS value in the incoming frame (configure the port to trust CoS), and 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 3 least-significant bits of the 1-byte User field. Layer 2 802.1Q frame headers carry the CoS value in the 3 most-significant bits of the Tag Control Information field. CoS values range from 0 for low priority to 7 for high priority.

Non-IP Traffic Classification	Description
Trust the DSCP or trust IP precedence value	Trust the DSCP or trust IP precedence value in the incoming 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 an internal DSCP value from the CoS-to-DSCP map. The switch uses the internal DSCP value to generate a CoS value representing the priority of the traffic.
Perform classification based on configured Layer 2 MAC ACL	Perform the classification based on a configured Layer 2 MAC access control list (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 CoS value to assign to the incoming frame.

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

IP Traffic Classification

The following table describes the IP traffic classification options for your QoS configuration.

Table 2: IP Traffic Classifications

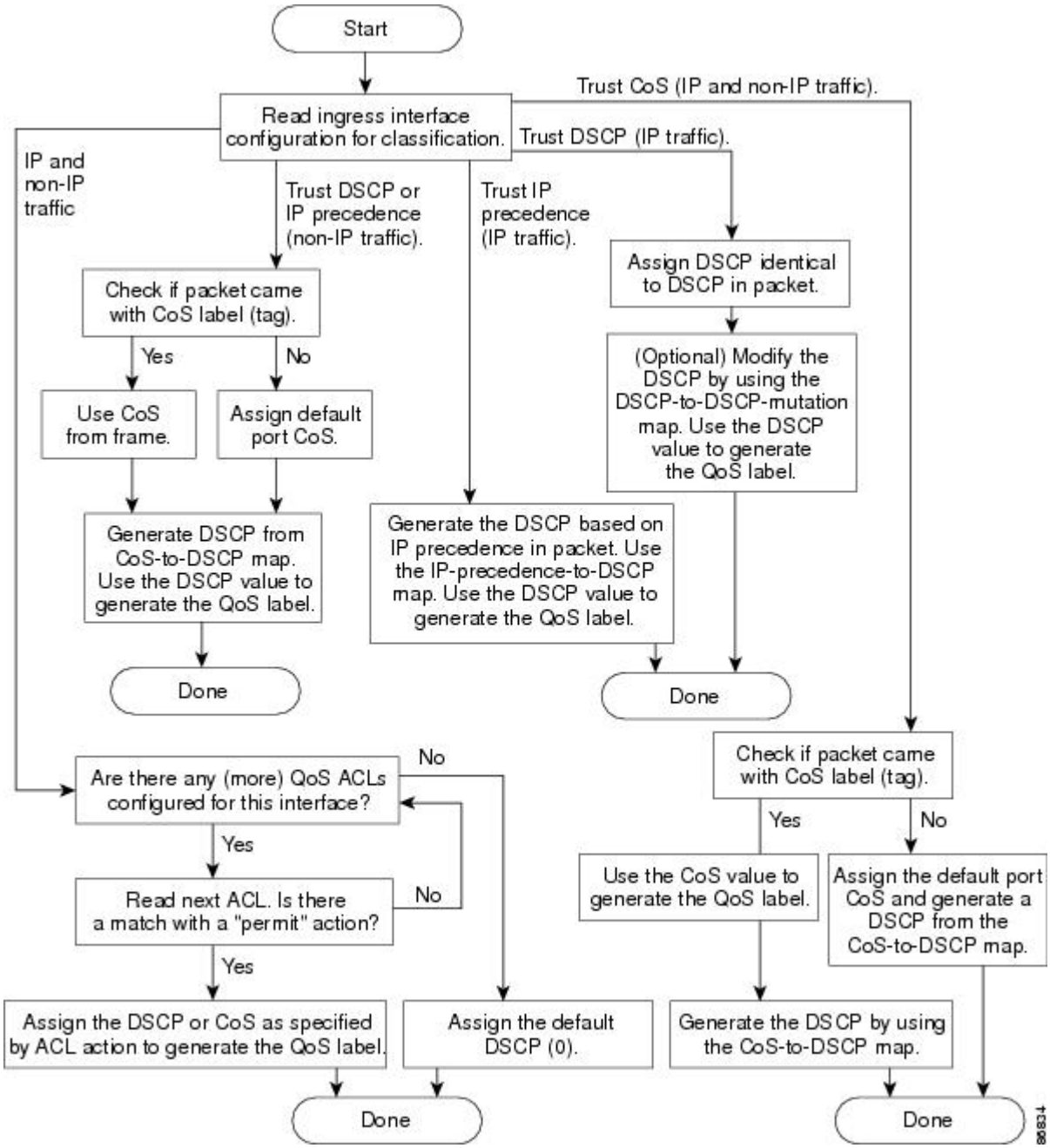
IP Traffic Classification	Description
Trust the DSCP value	Trust the DSCP value in the incoming packet (configure the port to trust DSCP), and assign the same DSCP value to the packet. The IETF defines the 6 most-significant bits of the 1-byte ToS field as the DSCP. The priority represented by a particular DSCP value is configurable. DSCP values range from 0 to 63. You can also classify IP traffic based on IPv6 DSCP. For ports that are on the boundary between two QoS administrative domains, you can modify the DSCP to another value by using the configurable DSCP-to-DSCP-mutation map.
Trust the IP precedence value	Trust the IP precedence value in the incoming packet (configure the port to trust IP precedence), and generate a DSCP value for the packet by using the configurable IP-precedence-to-DSCP map. The IP Version 4 specification defines the 3 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. You can also classify IP traffic based on IPv6 precedence.
Trust the CoS value	Trust the CoS value (if present) in the incoming packet, and generate a DSCP value for the packet by using the CoS-to-DSCP map. If the CoS value is not present, use the default port CoS value.

IP Traffic Classification	Description
IP standard or an extended ACL	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 CoS value to assign to the incoming frame.
Override configured CoS	Override the configured CoS of incoming packets, and apply the default port CoS value to them. For IPv6 packets, the DSCP value is rewritten by using the CoS-to-DSCP map and by using the default CoS of the port. You can do this for both IPv4 and IPv6 traffic.

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

Classification Flowchart

Figure 3: Classification Flowchart



Access Control Lists

You can use IP standard, IP extended, or Layer 2 MAC ACLs to define a group of packets with the same characteristics (class). You can also classify IP traffic based on IPv6 ACLs.

In the QoS context, the permit and deny actions in the access control entries (ACEs) have different meanings from 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.

Note Deny action is supported in Cisco IOS Release 3.7.4E and later releases.

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

After a traffic class has been defined with the 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 rate-limit the class. This policy is then attached to a particular port on which it becomes effective.

You implement IP ACLs to classify IP traffic by using the **access-list** global configuration command; you implement Layer 2 MAC ACLs to classify non-IP traffic by using the **mac access-list extended** global configuration command.

Classification Based on Class Maps and Policy Maps

To use policy maps, the switch must be running the LAN Base image.

A class map is a mechanism that you use to name 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. After a packet is matched against the class-map criteria, you further classify it through the use of a policy map.

A policy map specifies which traffic class to act on. Actions can include trusting the 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. Before a policy map can be effective, you must attach it to a port.

You create a class map by using the **class-map** global configuration command or the **class** policy-map configuration command. You should use the **class-map** command when the map is shared among many ports. 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.

You can configure a default class by using the **class class-default** policy-map configuration command. Unclassified traffic (traffic specified in the other traffic classes configured on the policy-map) is treated as default traffic.

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

The policy map can contain the **police** and **police aggregate** policy-map class configuration commands, which define the policer, the bandwidth limitations of the traffic, and the action to take if the limits are exceeded.

To enable the policy map, you attach it to a port by using the **service-policy** interface configuration command.

Policing and Marking Overview

After a packet is classified and has a DSCP-based or CoS-based QoS label assigned to it, the policing and marking process can begin.

Policing involves creating a policer that specifies the bandwidth limits for the traffic. Packets that exceed the limits are *out of profile* or *nonconforming*. Each policer 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. Marked-down packets use the same queues as the original QoS label to prevent packets in a flow from getting out of order.

Note All traffic, regardless of whether it is bridged or routed, is subjected to a policer, if one is configured. 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 configure policing on a physical port. After you configure the policy map and policing actions, attach the policy to a port by using the **service-policy** interface configuration command.

Physical Port Policing

In policy maps on physical ports, you can create the following types of policers:

- Individual—QoS applies the bandwidth limits specified in the policer separately to each matched traffic class. You configure this type of 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.

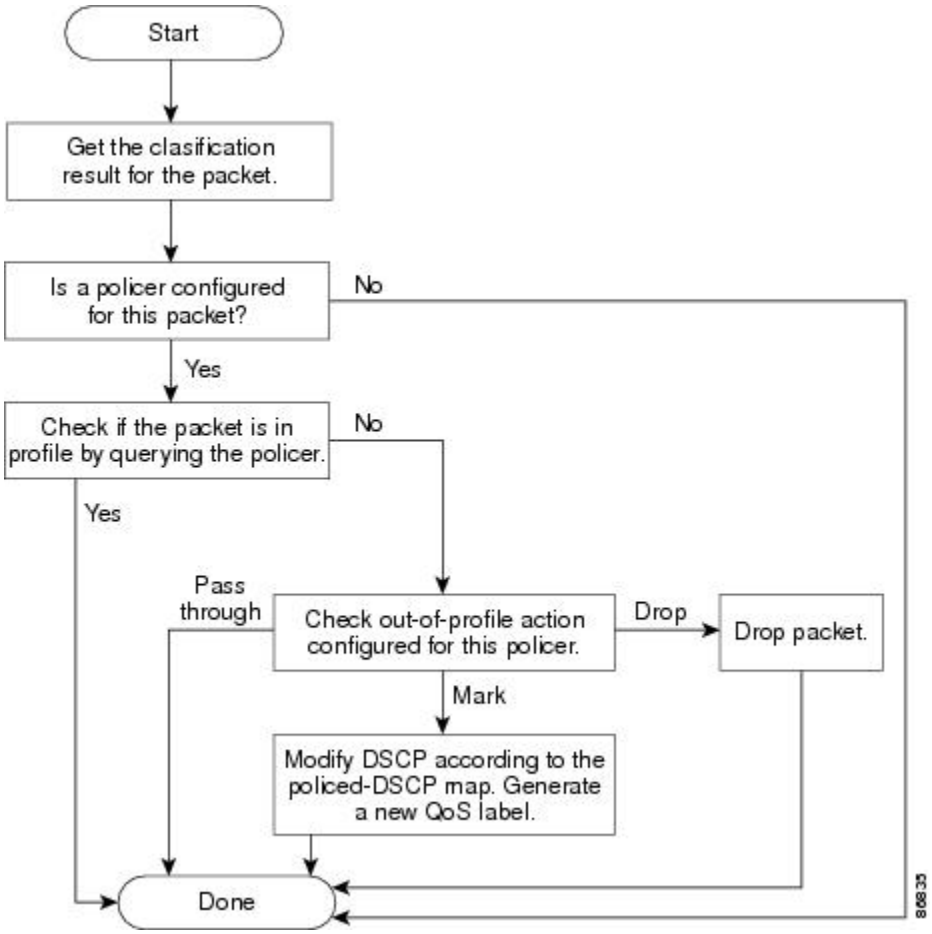
Policing uses a token-bucket algorithm. As each frame is received by the switch, a token is added to the bucket. The bucket has a hole in it and leaks at a rate that you specify as the average traffic rate in bits per second. Each time a token is added to the bucket, the switch verifies that there is 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 transmitted 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.

Figure 4: Policing and Marking Flowchart on Physical Ports



Mapping Tables Overview

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

The following table describes QoS processing and mapping tables.

Table 3: QoS Processing and Mapping Tables

QoS Processing Stage	Mapping Table Usage
Classification	During the classification stage, 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.
Policing	During policing stage, 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.
Pre-scheduling	Before the traffic reaches the scheduling stage, QoS stores the packet in an egress queue according to the QoS label. The QoS label is based on the DSCP or the CoS value in the packet and selects the queue through the DSCP output queue threshold maps or through the CoS output queue threshold maps. In addition to an egress queue, the QOS label also identifies the WTD threshold value. You configure these maps by using the **mls qos srr-queue** { **output**} **dscp-map** and the **mls qos srr-queue** { **output**} **cos-map** global configuration commands.

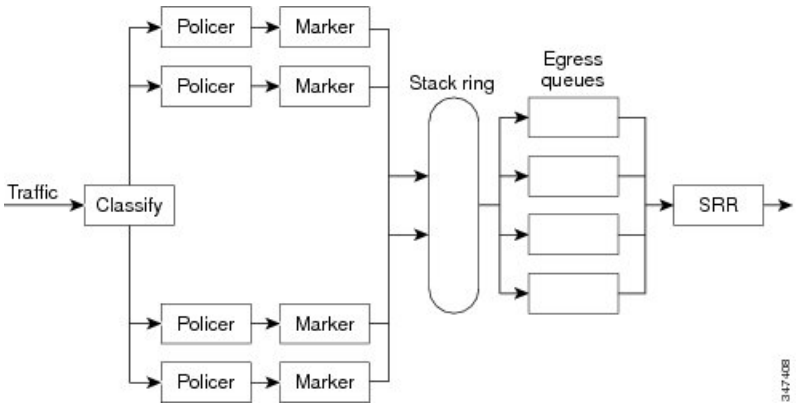
The CoS-to-DSCP, DSCP-to-CoS, and the IP-precedence-to-DSCP maps have default values that might or might not be appropriate for your network.

The default DSCP-to-DSCP-mutation map and the default policed-DSCP map are null maps; they map an incoming 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.

Queueing and Scheduling Overview

The switch has queues at specific points to help prevent congestion.

Figure 5: Egress Queue Location on Switch



Note The switch supports 4 egress queues by default and there is an option to enable a total of 8 egress queues. The 8 egress queue configuration is only supported on a standalone switch.

Weighted Tail Drop

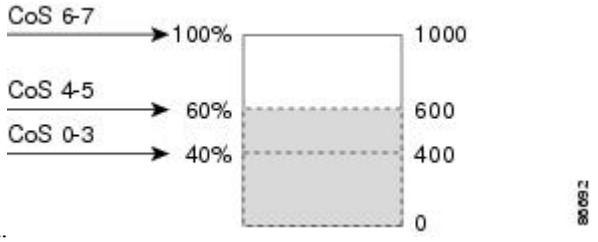
Egress queues use an enhanced version of the tail-drop congestion-avoidance mechanism called weighted tail drop (WTD). WTD is implemented on queues to manage the queue lengths and to provide drop precedences for different traffic classifications.

As a frame is enqueued 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.

Each queue has three threshold values. The QoS label determines which of the three threshold values is subjected to the frame. Of the three thresholds, two are configurable (explicit) and one is not (implicit).

Figure 6: WTD and Queue Operation

The following figure shows an example of WTD operating on a queue whose size is 1000 frames. Three drop percentages are configured: 40 percent (400 frames), 60 percent (600 frames), and 100 percent (1000 frames). These percentages indicate 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 the 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.

SRR Shaping and Sharing

Egress queues are serviced by shaped round robin (SRR), which controls the rate at which packets are sent. On the egress queues, SRR sends packets to the egress port.

You can configure SRR on egress queues for sharing or for shaping.

In shaped mode, the egress 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. Shaping and sharing is configured per interface. Each interface can be uniquely configured.

Queueing and Scheduling on Ingress Queues

Figure 7: Queueing and Scheduling Flowchart for Ingress Ports on Catalyst 3750-E and 3750-X Switches

The following figure shows queueing and scheduling flowcharts for ingress ports on Catalyst 3750-E and 3750-X switches.

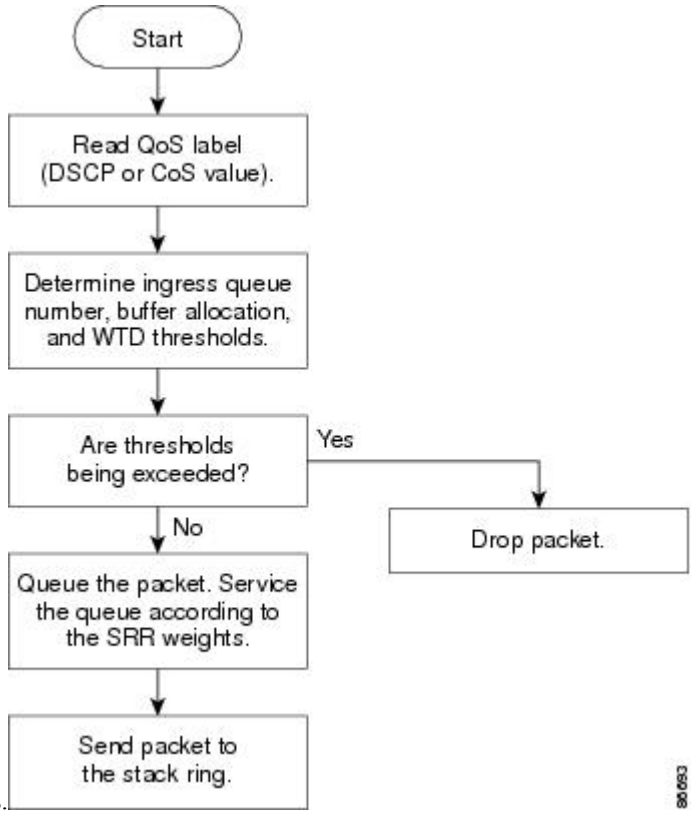
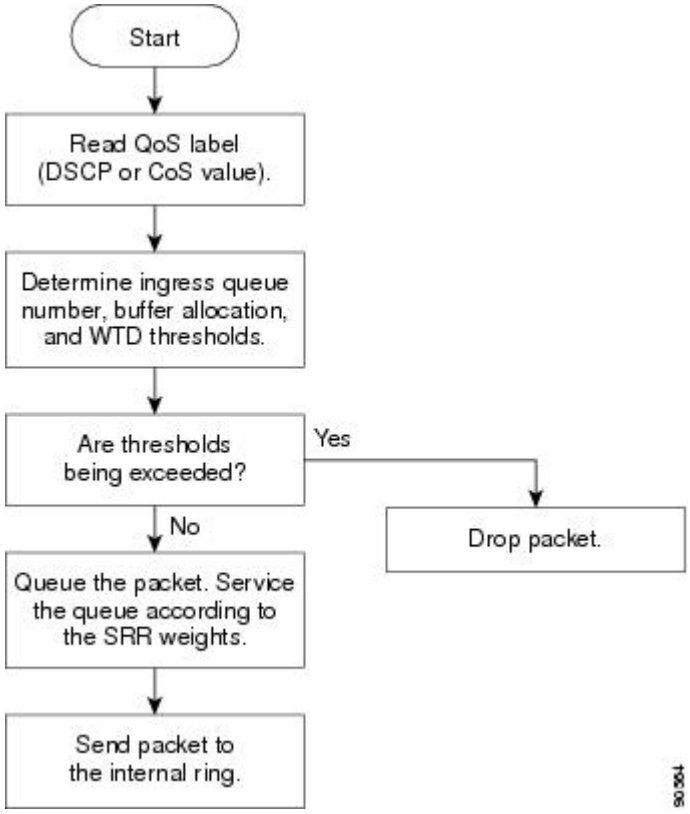


Figure 8: Queueing and Scheduling Flowchart for Ingress Ports on Catalyst 3560-E and 3560-X Switches

The following figure shows queueing and scheduling flowcharts for ingress ports on Catalyst 3560-E and 3560-X switches.



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

Configurable Ingress Queue Types

The switch supports two configurable ingress queue types, which are serviced by SRR in shared mode only.

Note The switch also uses two nonconfigurable queues for traffic that are essential for proper network and stack operation.

The following table describes the two configurable ingress queues.

Table 4: Configurable Ingress Queue Types

Queue Type	Function
Normal	User traffic that is considered to be normal priority. You can configure three different thresholds to differentiate among the flows. Use the following global configuration commands: • **mls qos srr-queue input threshold** • **mls qos srr-queue input dscp-map** • **mls qos srr-queue input cos-map**
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 or total stac the **mls qos srr-queue input priority-queue** global configuration command. The expedite queue has guaranteed bandwidth.

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.

Buffer and Bandwidth Allocation

You define the ratio (allocate the amount of space) with which to divide the ingress buffers between the two queues (normal and expedite) 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.

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 stack or internal ring.

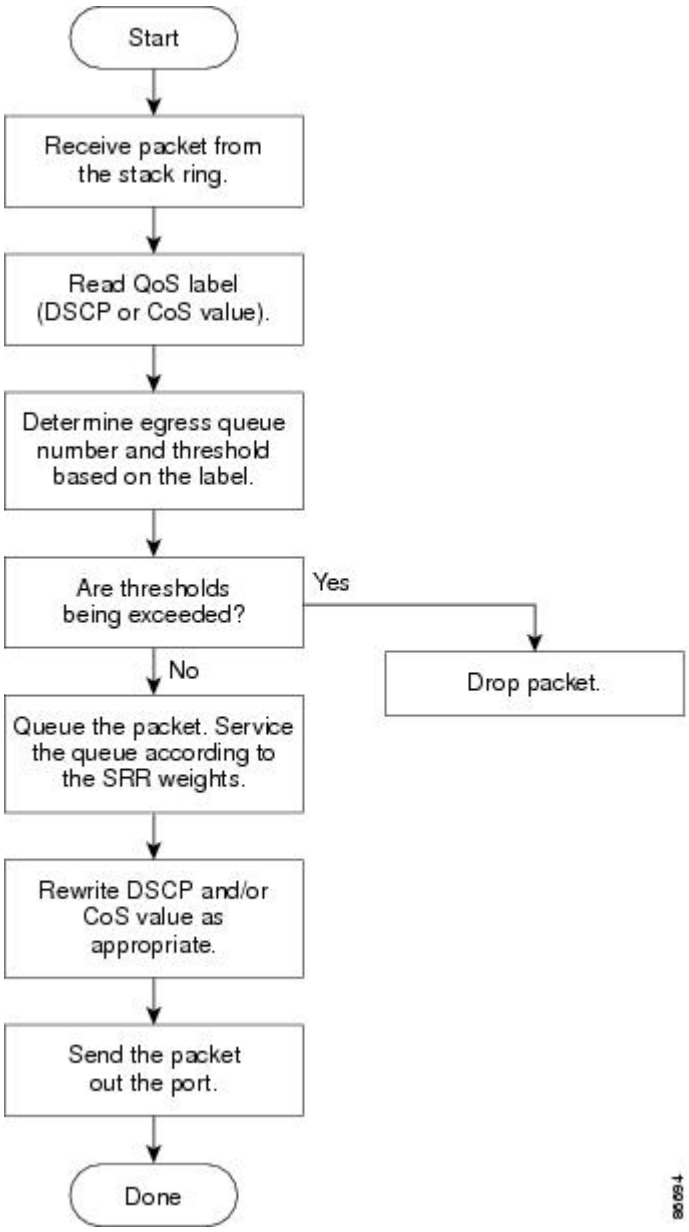
SRR services the priority queue for its configured weight as specified by the **bandwidth** keyword in the **mls qos srr-queue input priority-queue** *queue-id* **bandwidth** *weight* global configuration command. Then, SRR shares the remaining bandwidth with both ingress queues and services them as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

You can combine the above commands to prioritize traffic by placing packets with particular DSCPs or CoSs 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.

Queueing and Scheduling on Egress Queues

The following figure shows queueing and scheduling flowcharts for egress ports on the switch.

Figure 9: Queueing and Scheduling Flowchart for Egress Ports on the Switch



Note If the expedite queue is enabled, SRR services it until it is empty before servicing the other three queues.

Egress Expedite Queue

Each port supports four egress queues, one of which (queue 1) can be the egress expedite queue. These queues are assigned to a queue-set. All traffic exiting the switch flows through one of these four queues and is subjected to a threshold based on the QoS label assigned to the packet.

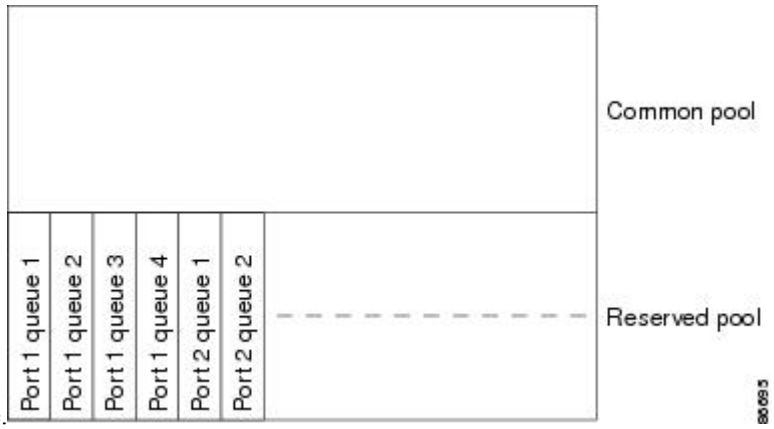
Note If the expedite queue is enabled, SRR services it until it is empty before servicing the other three queues.

Egress Queue Buffer Allocation

The following figure shows the egress queue buffer.

Figure 10: Egress Queue Buffer Allocation

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 the target queue has not 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.



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.

Note The switch supports 4 egress queues by default, although there is an option to enable a total of 8 egress queues. Use the **mls qos srr-queue output queues 8** global configuration command to enable all 8 egress queues. Once 8 egress queues are enabled, you are able to configure thresholds and buffers for all 8 queues. The 8 egress queue configuration is only supported on a standalone switch.

Queues and 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 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 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 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. You map a port to queue-set by using the **queue-set qset-id** interface configuration command. Modify the queue-set configuration to change the WTD threshold percentages.

Note The switch supports 4 egress queues by default, although there is an option to enable a total of 8 egress queues. Use the **mls qos srr-queue output queues 8** global configuration command to enable all 8 egress queues. Once 8 egress queues are enabled, you are able to configure thresholds and buffers for all 8 queues. The 8 egress queue configuration is only supported on a standalone switch.

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 the 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.

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.

All four queues participate in the SRR unless the expedite queue is enabled, in which case the first bandwidth weight is ignored and is not used in the ratio calculation. The expedite queue is a priority queue, and it is serviced until empty before the other queues are serviced. You enable the expedite queue by using the **priority-queue out** interface configuration command.

You can combine the commands described in this section to prioritize traffic by placing packets with particular DSCPs or CoSs 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.

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

Note The switch supports 4 egress queues by default, although there is an option to enable a total of 8 egress queues. Use the **mls qos srr-queue output queues 8** global configuration command to enable all 8 egress queues. Once 8 egress queues are enabled, you are able to configure thresholds, buffers, bandwidth share weights, and bandwidth shape weights for all 8 queues. The 8 egress queue configuration is only supported on a standalone switch.

Packet Modification

A packet is classified, policed, and queued to provide QoS. The following packet modifications can occur during the process to provide QoS:

- For IP and non-IP packets, 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.
- During policing, IP and non-IP packets can have another DSCP assigned to them (if they are out of profile and the policer specifies a markdown DSCP). Once again, the DSCP in the packet is not modified, but an indication of the marked-down value is carried along. For IP packets, the packet modification occurs at a later stage; for non-IP packets the DSCP is converted to CoS and used for queueing and scheduling decisions.
- 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 a table map and if you configure the port to trust the DSCP of the incoming 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 incoming 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.

Standard QoS Default Configuration

Standard QoS is disabled by default.

When 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. The packets are switched without any rewrites and classified as best effort without any policing.

When QoS is enabled using 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.

Note Starting Cisco IOS Release 15.2(1)E, IPv6 QoS is supported on switches running the LAN base license with lanbase-routing template.

Default Ingress Queue Configuration

The following tables describe the default ingress queue configurations.

The following table shows the default ingress queue configuration when QoS is enabled. For the bandwidth allocation feature, bandwidth is equally shared between the queues. SRR sends packets in shared mode only. Queue 2 is the priority queue. SRR services the priority queue for its configured share before servicing the other queue.

Table 5: Default Ingress Queue Configuration

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

The following table shows the default CoS input queue threshold map when QoS is enabled.

Table 6: Default CoS Input Queue Threshold Map

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

The following table shows the default DSCP input queue threshold map when QoS is enabled.

Table 7: Default DSCP Input Queue Threshold Map

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

Default Egress Queue Configuration

The following tables describe the default egress queue configurations.

Note The switch supports 4 egress queues by default, although there is an option to enable a total of 8 egress queues. Use the **mls qos srr-queue output queues 8** global configuration command to enable all 8 egress queues. Once 8 egress queues are enabled, you are able to configure thresholds and buffers for all 8 queues. The 8 egress queue configuration is only supported on a standalone switch.

The following table shows the default egress queue configuration for each queue-set when QoS is enabled. All ports are mapped to queue-set 1. The port bandwidth limit is set to 100 percent and rate unlimited. Note that for the SRR shaped weights (absolute) feature, a shaped weight of zero indicates that the queue is operating in shared mode. Note that for the SRR shared weights feature, one quarter of the bandwidth is allocated to each queue.

Table 8: Default Egress Queue 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	200 percent	100 percent	100 percent
WTD drop threshold 2	100 percent	200 percent	100 percent	100 percent
Reserved threshold	50 percent	50 percent	50 percent	50 percent
Maximum threshold	400 percent	400 percent	400 percent	400 percent
SRR shaped weights (absolute)	25	0	0	0
SRR shared weights	25	25	25	25

The following table shows the default CoS output queue threshold map when QoS is enabled.

Table 9: Default CoS Output Queue Threshold Map

CoS Value	Queue ID–Threshold ID
0, 1	2–1
2, 3	3–1
4	4–1
5	1–1
6, 7	4–1

The following table shows the default DSCP output queue threshold map when QoS is enabled.

Table 10: Default DSCP Output Queue Threshold Map

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

The following table displays the default egress queue configuration when the 8 egress queue configuration is enabled using the **mls qos srr-queue output queues 8** command.

Table 11: Default 8 Egress Queue Configuration

Feature	Queue 1	Queue 2	Queue 3	Queue 4	Queue 5	Queue 6	Queue 7	Queue 8
Buffer allocation	10	30	10	10	10	10	10	10
WTD drop threshold 1	100	1600	100	100	100	100	100	100
WTD drop threshold 2	100	2000	100	100	100	100	100	100
Reserved threshold	100	100	100	100	100	100	100	100
Maximum threshold	400	2400	400	400	400	400	400	400
SRR shaped weights	25	0	0	0	0	0	0	0
SRR shared weights	25	25	25	25	25	25	25	25

The following table displays the default CoS output queue threshold map when QoS is enabled and the 8 egress queue configuration is enabled using the **mls qos srr-queue output queues 8** command.

Table 12: Default CoS Output 8 Queue Threshold Map

CoS	Egress Queue	Threshold ID	4 Egress Queue Mapping
0	2	1	2
1	3	1	2
2	4	1	3
3	5	1	3
4	6	1	4
5	1	1	1
6	7	1	4
7	8	1	4

The following table displays the default DSCP output queue threshold map when QoS is enabled and the 8 egress queue configuration is enabled using the **mls qos srr-queue output queues 8** command.

Table 13: Default DSCP Output 8 Queue Threshold Map

DSCP	Egress Queue	Threshold ID	4 Egress Queue Mapping
0-7	2	1	2
8-15	3	1	2
16-23	4	1	3
24-31	5	1	3
32-39	6	1	4
40-47	1	1	1
48-55	7	1	4
56-63	8	1	4

Default Mapping Table Configuration

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

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

DSCP Maps

Default CoS-to-DSCP Map

When DSCP transparency mode is disabled, the DSCP values are derived from CoS as per the following table. If these values are not appropriate for your network, you need to modify them.

Note The DSCP transparency mode is disabled by default. If it is enabled (**no mls qos rewrite ip dscp** interface configuration command), DSCP rewrite will not happen.

Table 14: Default CoS-to-DSCP Map

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

Default IP-Precedence-to-DSCP Map

You use the IP-precedence-to-DSCP map to map IP precedence values in incoming packets to a DSCP value that QoS uses internally to represent the priority of the traffic. The following table shows the default IP-precedence-to-DSCP map. If these values are not appropriate for your network, you need to modify them.

Table 15: Default IP-Precedence-to-DSCP Map

IP Precedence Value	DSCP Value
0	0
1	8
2	16
3	24
4	32
5	40
6	48

IP Precedence Value	DSCP Value
7	56

Default 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 egress queues. The following table shows the default DSCP-to-CoS map. If these values are not appropriate for your network, you need to modify them.

Table 16: Default DSCP-to-CoS Map

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

How to Configure QoS

Enabling QoS Globally

By default, QoS is disabled on the switch.

The following procedure to enable QoS globally is required.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**mls qos** **Example:**	Enables QoS globally.

	Command or Action	Purpose
	`Device(config)# mls qos`	QoS operates with the default settings described in the related topic sections below. **Note** To disable QoS, use the **no mls qos** global configuration command.
Step 3	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 4	**show mls qos** **Example:** `Device# show mls qos`	Verifies the QoS configuration.
Step 5	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Enabling VLAN-Based QoS on Physical Ports

By default, VLAN-based QoS is disabled on all physical switch ports. You can enable VLAN-based QoS on a switch port.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Specifies the physical port, and enter interface configuration mode.
Step 3	**mls qos vlan-based**	Enables VLAN-based QoS on the port.

	Command or Action	Purpose
	Example: `Device(config-if)# mls qos vlan-based`	**Note** Use the **no mls qos vlan-based** interface configuration command to disable VLAN-based QoS on the physical port.
Step 4	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 5	**show mls qos interface** *interface-id* **Example:** `Device# show mls qos interface gigabitethernet 1/0/1`	Verifies if VLAN-based QoS is enabled on the physical port.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring Classification Using Port Trust States

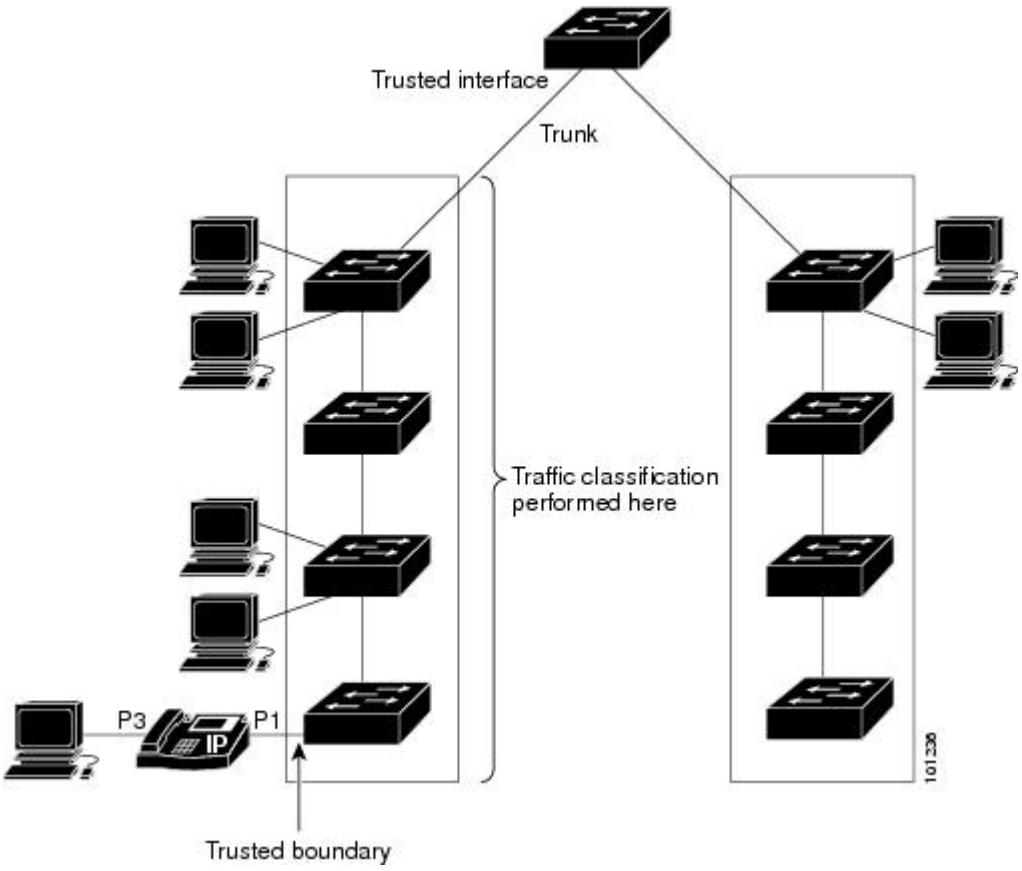
These sections describe how to classify incoming traffic by using port trust states.

Note Depending on your network configuration, you must perform one or more of these tasks in this module or one or more of the tasks in the Configuring a QoS Policy.

Configuring the Trust State on Ports Within the QoS Domain

Packets entering a QoS domain are classified at the edge of the QoS domain. When the packets are classified at the edge, the switch port within the QoS domain can be configured to one of the trusted states because there is no need to classify the packets at every switch within the QoS domain.

Figure 11: Port Trusted States on Ports Within the QoS Domain



Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 1/0/2`	Specifies the port to be trusted, and enters interface configuration mode. Valid interfaces are physical ports.
Step 3	**mls qos trust** [**cos** | **dscp** | **ip-precedence**] **Example:** `Device(config-if)# mls qos trust cos`	Configures the port trust state. By default, the port is not trusted. If no keyword is specified, the default is **dscp**. The keywords have these meanings:

	Command or Action	Purpose
		• **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. • **dscp**—Classifies an ingress packet by using the packet DSCP 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 by using the CoS-to-DSCP map. • **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 by using the CoS-to-DSCP map. To return a port to its untrusted state, use the **no mls qos trust** interface configuration command.
Step 4	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 5	**show mls qos interface** **Example:** `Device# show mls qos interface`	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

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 incoming packets on the port.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 1/1/1`	Specifies the port to be configured, and enters interface configuration mode. Valid interfaces include physical ports.
Step 3	**mls qos cos** {*default-cos* | **override**} **Example:** `Device(config-if)# mls qos override`	Configures the default CoS value for the 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. • Use the **override** keyword to override the previously configured trust state of the incoming packet and to apply the default port CoS value to the port on all incoming packets. By default, CoS override is disabled. Use the **override** keyword when all incoming 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 incoming CoS values are assigned the default CoS value configured with this command. If an incoming packet is tagged, the CoS value of the packet is modified with the default CoS of the port at the ingress port. **Note** To return to the default setting, use the **no mls qos cos** {*default-cos* | **override**} interface configuration command.

	Command or Action	Purpose
Step 4	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 5	**show mls qos interface** **Example:** `Device# show mls qos interface`	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring a Trusted Boundary to Ensure Port Security

In a typical network, you connect a Cisco IP Phone to a port 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 is typically marked with a tag that uses the 802.1Q header. The header contains the VLAN information and the class of service (CoS) 3-bit field, which is the priority of the packet.

For most Cisco IP Phone configurations, the traffic sent from the telephone to the 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 port to which the telephone is connected to trust the CoS labels of all traffic received on that port. Use the **mls qos trust dscp** interface configuration command to configure a routed port to which the telephone is connected to trust the DSCP 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 . Without trusted boundary, the CoS labels generated by the PC are trusted by the (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 port. If the telephone is not detected, the trusted boundary feature disables the trusted setting on the 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 .

In some situations, you can prevent a PC connected to the Cisco 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 CLI to override the priority of the traffic received from the PC.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**cdp run** **Example:** `Device(config)# cdp run`	Enables CDP globally. By default, CDP is enabled.
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 2/1/1`	Specifies the port connected to the Cisco IP Phone, and enters interface configuration mode. Valid interfaces include physical ports.
Step 4	**cdp enable** **Example:** `Device(config-if)# cdp enable`	Enables CDP on the port. By default, CDP is enabled.
Step 5	Use one of the following: • **mls qos trust cos** • **mls qos trust dscp** **Example:** `Device(config-if)# mls qos trust cos`	Configures the port to trust the CoS value in traffic received from the Cisco IP Phone. or Configures the routed port to trust the DSCP value in traffic received from the Cisco IP Phone. By default, the port is not trusted.
Step 6	**mls qos trust device cisco-phone** **Example:** `Device(config-if)# mls qos trust device cisco-phone`	Specifies that the Cisco IP Phone is a trusted device. You cannot enable both trusted boundary and auto-QoS (**auto qos voip** interface configuration command) at the same time; they are mutually exclusive. **Note** To disable the trusted boundary feature, use the **no mls qos trust device** interface configuration command.

	Command or Action	Purpose
Step 7	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 8	**show mls qos interface** **Example:** `Device# show mls qos interface`	Verifies your entries.
Step 9	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Enabling DSCP Transparency Mode

The switch supports the DSCP transparency feature. It affects only the DSCP field of a packet at egress. By default, DSCP transparency is disabled. The switch modifies the DSCP field in an incoming packet, and the DSCP field in the outgoing packet is based on the quality of service (QoS) configuration, including the port trust setting, policing and marking, and the DSCP-to-DSCP mutation map.

If DSCP transparency is enabled by using the **no mls qos rewrite ip dscp** command, the switch does not modify the DSCP field in the incoming packet, and the DSCP field in the outgoing packet is the same as that in the incoming packet.

Regardless of the DSCP transparency configuration, the switch modifies the internal DSCP value of the packet, which the switch uses to generate a class of service (CoS) value that represents the priority of the traffic. The switch also uses the internal DSCP value to select an egress queue and threshold.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**mls qos** **Example:** `Device(config)# mls qos`	Enables QoS globally.

	Command or Action	Purpose
Step 3	**no mls qos rewrite ip dscp** **Example:** `Device(config)# no mls qos rewrite ip dscp`	Enables DSCP transparency. The switch is configured to not modify the DSCP field of the IP packet.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show mls qos interface** [*interface-id*] **Example:** `Device# show mls qos interface gigabitethernet 2/1/1`	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

DSCP Transparency Mode

To configure the switch to modify the DSCP value based on the trust setting or on an ACL by disabling DSCP transparency, use the **mls qos rewrite ip dscp** global configuration command.

If you disable QoS by using the **no mls qos** global configuration command, the CoS and DSCP values are not changed (the default QoS setting).

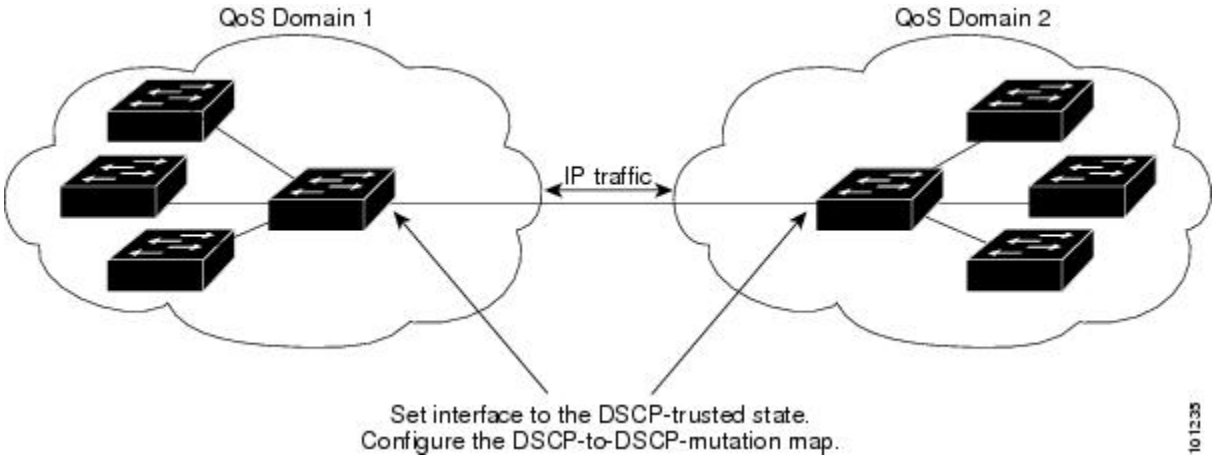
If you enter the **no mls qos rewrite ip dscp** global configuration command to enable DSCP transparency and then enter the **mls qos trust** [**cos** | **dscp**] interface configuration command, DSCP transparency is still enabled.

Note For Catalyst 2960-L switches, DSCP transparency is enabled by default.

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 ports bordering the domains to a DSCP-trusted state. 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 12: DSCP-Trusted State on a Port Bordering Another QoS Domain



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.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**mls qos map dscp-mutation** *dscp-mutation-name in-dscp* **to** *out-dscp* **Example:** `Device(config)# mls qos map dscp-mutation gigabitethernet1/0/2-mutation 10 11 12 13 to 30`	Modifies the DSCP-to-DSCP-mutation map. The default DSCP-to-DSCP-mutation map is a null map, which maps an incoming 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* **Example:** `Device(config)# interface`	Specifies the port to be trusted, and enter interface configuration mode. Valid interfaces include physical ports.

	Command or Action	Purpose
	`gigabitethernet1/0/2`	
Step 4	**mls qos trust dscp** **Example:** `Device(config-if)# mls qos trust dscp`	Configures the ingress port as a DSCP-trusted port. By default, the port is not trusted. **Note** To return a port to its non-trusted state, use the **no mls qos trust** interface configuration command.
Step 5	**mls qos dscp-mutation** *dscp-mutation-name* **Example:** `Device(config-if)# mls qos dscp-mutation gigabitethernet1/0/2-mutation`	Applies 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. **Note** 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.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 7	**show mls qos maps dscp-mutation** **Example:** `Device# show mls qos maps dscp-mutation`	Verifies your entries.
Step 8	**copy running-config startup-config** **Example:** `Device# copy-running-config startup-config`	(Optional) Saves your entries in the configuration file. **Note** 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.

Configuring a QoS Policy

Configuring a QoS policy typically requires the following tasks:

- Classifying traffic into classes
- Configuring policies applied to those traffic classes
- Attaching policies to ports

These sections describe how to classify, police, and mark traffic. Depending on your network configuration, you must perform one or more of the modules in this section.

Classifying Traffic by Using ACLs

You can classify IP traffic by using IPv4 standard ACLS, IPv4 extended ACLs, or IPv6 ACLs.

You can classify non-IP traffic by using Layer 2 MAC ACLs.

Creating an IP Standard ACL for IPv4 Traffic

Before you begin

Before you perform this task, determine which access lists you will be using for your QoS configuration.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] **Example:** `Device(config)# access-list 1 permit 192.2.255.0 10.1.1.255`	Creates 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

	Command or Action	Purpose
		be applied to the source. Place ones in the bit positions that you want to ignore. 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. **Note** To delete an access list, use the **no access-list** *access-list-number* global configuration command.
Step 3	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 4	**show access-lists** **Example:** `Device# show access-lists`	Verifies your entries.
Step 5	**copy running-config startup-config** **Example:** `Device# copy-running-config startup-config`	(Optional) Saves your entries in the configuration file.

Creating an IP Extended ACL for IPv4 Traffic

Before you begin

Before you perform this task, determine which access lists you will be using for your QoS configuration.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard*	Creates an IP extended ACL, repeating the command as many times as necessary.

	Command or Action	Purpose
	Example: `Device(config)# access-list 100 permit ip any any dscp 32`	• 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*. 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. **Note** To delete an access list, use the **no access-list** *access-list-number* global configuration command.
Step 3	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# end`	
Step 4	**show access-lists** **Example:** `Device# show access-lists`	Verifies your entries.
Step 5	**copy running-config startup-config** **Example:** `Device# copy-running-config startup-config`	(Optional) Saves your entries in the configuration file.

Creating an IPv6 ACL for IPv6 Traffic

Before you begin

Before you perform this task, determine which access lists you will be using for your QoS configuration.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**ipv6 access-list** *access-list-name* **Example:** `Device(config)# ipv6 access-list ipv6_Name_ACL`	Creates an IPv6 ACL and enters IPv6 access-list configuration mode. Accesses list names cannot contain a space or quotation mark or begin with a numeric. **Note** To delete an access list, use the **no ipv6 access-list** *access-list-number* global configuration command.
Step 3	{**deny** | **permit**} *protocol* {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix*/ *prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* [*port-number*]] [**dscp** *value*] [**fragments**] [**log**] [**log-input**] [**routing**] [**sequence** *value*] [**time-range** *name*]	Enters **deny** or **permit** to specify whether to deny or permit the packet if conditions are matched. These are the conditions: For *protocol*, enter the name or number of an Internet protocol: **ahp**, **esp**, **icmp**, **ipv6**, **pcp**, **stcp**, **tcp**, or **udp**, or an integer in the range 0 to 255 representing an IPv6 protocol number.

	Command or Action	Purpose
	Example: `Device(config-ipv6-acl)#` **`permit ip host 10::1 host 11::2 host`**	• The *source-ipv6-prefix/prefix-length* or *destination-ipv6-prefix/ prefix-length* is the source or destination IPv6 network or class of networks for which to set deny or permit conditions, specified in hexadecimal and using 16-bit values between colons (see RFC 2373). • Enter **any** as an abbreviation for the IPv6 prefix ::/0. • For **host** *source-ipv6-address* or *destination-ipv6-address*, enter the source or destination IPv6 host address for which to set deny or permit conditions, specified in hexadecimal using 16-bit values between colons. • (Optional) For *operator*, specify an operand that compares the source or destination ports of the specified protocol. Operands are **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range**. If the operator follows the *source-ipv6-prefix/prefix-length* argument, it must match the source port. If the operator follows the *destination-ipv6-prefix/prefix-length* argument, it must match the destination port. • (Optional) The *port-number* is a decimal number from 0 to 65535 or the name of a TCP or UDP port. You can use TCP port names only when filtering TCP. You can use UDP port names only when filtering UDP. • (Optional) Enter **dscp** *value* to match a differentiated services code point value against the traffic class value in the Traffic Class field of each IPv6 packet header. The acceptable range is from 0 to 63. • (Optional) Enter **fragments** to check noninitial fragments. This keyword is visible only if the protocol is IPv6. • (Optional) Enter **log** to cause a logging message to be sent to the console about the packet that matches the entry. Enter **log-input** to include the input interface in

	Command or Action	Purpose
		the log entry. Logging is supported only for router ACLs. • (Optional) Enter **routing** to specify that IPv6 packets be routed. • (Optional) Enter **sequence** *value* to specify the sequence number for the access list statement. The acceptable range is from 1 to 4294967295. • (Optional) Enter **time-range** *name* to specify the time range that applies to the deny or permit statement.
Step 4	**end** **Example:** `Device(config-ipv6-acl)# end`	Returns to privileged EXEC mode.
Step 5	**show ipv6 access-list** **Example:** `Device# show ipv6 access-list`	Verifies the access list configuration.
Step 6	**copy running-config startup-config** **Example:** `Device# copy-running-config startup-config`	(Optional) Saves your entries in the configuration file.

Creating a Layer 2 MAC ACL for Non-IP Traffic

Before you begin

Before you perform this task, determine that Layer 2 MAC access lists are required for your QoS configuration.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**mac access-list extended** *name* **Example:** `Device(config)# mac access-list extended maclist1`	Creates a Layer 2 MAC ACL by specifying the name of the list. After entering this command, the mode changes to extended MAC ACL configuration. **Note** To delete an access list, use the **no mac access-list extended** *access-list-name* global configuration command.

Step 3	{**permit** | **deny**} {**host** *src-MAC-addr mask* | **any** | **host** *dst-MAC-addr* | *dst-MAC-addr mask*} [*type mask*]
	Example:
	`Device(config-ext-mac1) # permit 0001.0000.0001 0.0.0 0002.0000.0001 0.0.0`
	`Device(config-ext-mac1) # permit 0001.0000.0002 0.0.0 0002.0000.0002 0.0.0 xns-idp`
		Specifies 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* ffff.ffff.ffff, 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* ffff.ffff.ffff, 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.
		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** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-ext-mac1)# end`	
Step 5	**show access-lists** [*access-list-number* | *access-list-name*] **Example:** `Device# show access-lists`	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** `Device# copy-running-config startup-config`	(Optional) Saves your entries in the configuration file.

Classifying Traffic by Using Class Maps

You use the **class-map** global configuration command to name and to isolate a specific traffic flow (or class) from all other traffic. The class map defines the criteria to use to match against a specific traffic flow to further classify it. Match statements can include criteria such as an ACL, IP precedence values, or DSCP values. The match criterion is defined with one match statement entered within the class-map configuration mode.

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

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	Use one of the following: • **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] • **access-list** *access-list-number* {**deny** | **permit**} *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] • **ipv6 access-list** *access-list-name* {**deny** | **permit**} *protocol* {*source-ipv6-prefix/prefix-length* | **any** |	Creates an IP standard or extended ACL, an IPv6 ACL for IP traffic, or a Layer 2 MAC ACL for non-IP traffic, repeating the command as many times as necessary. 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 or Action	Purpose
	host *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix/ prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* [*port-number*]] [**dscp** *value*] [**fragments**] [**log**] [**log-input**] [**routing**] [**sequence** *value*] [**time-range** *name*] • **mac access-list extended** *name* {**permit** | **deny**} {**host** *src-MAC-addr mask* | **any** | **host** *dst-MAC-addr* | *dst-MAC-addr mask*} [*type mask*] **Example:** `Device(config)# access-list 103 permit ip any any dscp 10`	
Step 3	**class-map** [**match-all** | **match-any**] *class-map-name* **Example:** `Device(config)# class-map class1`	Creates a class map, and enters 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 match 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 match criteria must be matched. • For *class-map-name*, specify the name of the class map. If neither the **match-all** or **match-any** keyword is specified, the default is **match-all**. **Note** To delete an existing class map, use the **no class-map** [**match-all** | **match-any**] *class-map-name* global configuration command.
Step 4	**match** {**access-group** *acl-index-or-name* | **ip dscp** *dscp-list* | **ip precedence** *ip-precedence-list*} **Example:** `Device(config-cmap)# match ip dscp 10 11 12`	Defines the match criterion to classify traffic. By default, no match criterion is defined. Only one match criterion per class map is supported, and only one ACL per class map is supported.

	Command or Action	Purpose
		• For **access-group** *acl-index-or-name*, specify the number or name of the ACL created in Step 2. • To filter IPv6 traffic with the **match access-group** command, create an IPv6 ACL, as described in Step 2. • For **ip dscp** *dscp-list*, enter a list of up to eight IP DSCP values to match against incoming 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 incoming packets. Separate each value with a space. The range is 0 to 7. **Note** To remove a match criterion, use the **no match** {**access-group** *acl-index-or-name* | **ip dscp** | **ip precedence**} class-map configuration command.
Step 5	**end** **Example:** `Device(config-cmap)# end`	Returns to privileged EXEC mode.
Step 6	**show class-map** **Example:** `Device# show class-map`	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** `Device# copy-running-config startup-config`	(Optional) Saves your entries in the configuration file.

Classifying Traffic by Using Class Maps and Filtering IPv6 Traffic

To apply the primary match criteria to only IPv4 traffic, use the **match protocol** command with the **ip** keyword. To apply the primary match criteria to only IPv6 traffic, use the **match protocol** command with the **ipv6** keyword.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**class-map** {**match-all**} *class-map-name* **Example:** `Device(config)# class-map cm-1`	Creates a class map, and enters class-map configuration mode. By default, no class maps are defined. When you use the **match protocol** command, only the **match-all** keyword is supported. • For *class-map-name*, specify the name of the class map. If neither the **match-all** or **match-any** keyword is specified, the default is **match-all**. **Note** To delete an existing class map, use the **no class-map** [**match-all** | **match-any**] *class-map-name* global configuration command.
Step 3	**match protocol** [*ip* | *ipv6*] **Example:** `Device(config-cmap)# match protocol ip`	(Optional) Specifies the IP protocol to which the class map applies: • Use the argument *ip* to specify IPv4 traffic and *ipv6* to specify IPv6 traffic. • When you use the **match protocol** command, only the **match-all** keyword is supported for the **class-map** command.
Step 4	**match** {**ip dscp** *dscp-list* | **ip precedence** *ip-precedence-list*} **Example:** `Device(config-cmap)# match ip dscp 10`	Defines the match criterion to classify traffic. By default, no match criterion is defined. • For **ip dscp** *dscp-list*, enter a list of up to eight IP DSCP values to match against incoming 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 incoming packets. Separate each value with a space. The range is 0 to 7.

	Command or Action	Purpose
		Note To remove a match criterion, use the no match {**access-group** *acl-index-or-name* | **ip dscp** | **ip precedence**} class-map configuration command.
Step 5	**end** **Example:** `Device(config-cmap)# end`	Returns to privileged EXEC mode.
Step 6	**show class-map** **Example:** `Device# show class-map`	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** `Device# copy-running-config startup-config`	(Optional) Saves your entries in the configuration file.

Classifying, Policing, and Marking Traffic on Physical Ports by Using Policy Maps

You can configure a policy map on a physical port that specifies which traffic class to act on. Actions can include trusting the CoS, DSCP, or IP precedence values in the traffic class; setting a specific DSCP or IP precedence value in the traffic class; and specifying 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 policy map can contain a predefined default traffic class explicitly placed at the end of the map.
- A separate policy-map class can exist for each type of traffic received through a port.

Follow these guidelines when configuring policy maps on physical ports:

- You can attach only one policy map per ingress port.
- If you configure the IP-precedence-to-DSCP map by using the **mls qos map ip-prec-dscp** *dscp1...dscp8* global configuration command, the settings only affect packets on ingress interfaces that are configured to trust the IP precedence value. In a policy map, if you set the packet IP precedence value to a new value by using the **set ip precedence** *new-precedence* policy-map class configuration command, the egress DSCP value is not affected by the IP-precedence-to-DSCP map. If you want the egress DSCP value to be different than the ingress value, use the **set dscp new-dscp** policy-map class configuration command.

- If you enter or have used the **set ip dscp** command, the changes this command to **set dscp** in its configuration.
- You can use the **set ip precedence** or the **set precedence** policy-map class configuration command to change the packet IP precedence value. This setting appears as set ip precedence in the configuration.
- A policy-map and a port trust state can both run on a physical interface. The policy-map is applied before the port trust state.
- When you configure a default traffic class by using the **class class-default** policy-map configuration command, unclassified traffic (traffic that does not meet the match criteria specified in the traffic classes) is treated as the default traffic class (class-default).

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**class-map** [**match-all** | **match-any**] *class-map-name* **Example:** `Device(config)# class-map ipclass1`	Creates a class map, and enters 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 match 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 match criteria must be matched. • For *class-map-name*, specify the name of the class map. If neither the **match-all** or **match-any** keyword is specified, the default is **match-all**.
Step 3	**policy-map** *policy-map-name* **Example:** `Device(config-cmap)# policy-map flowit`	Creates a policy map by entering the policy map name, and enters 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.

	Command or Action	Purpose
		Note To delete an existing policy map, use the **no policy-map** *policy-map-name* global configuration command.
Step 4	**class** [*class-map-name* | **class-default**] **Example:** `Device(config-pmap)# class ipclass1`	Defines a traffic classification, and enters policy-map class configuration mode. By default, no policy map class-maps are defined. If a traffic class has already been defined by using the **class-map** global configuration command, specify its name for *class-map-name* in this command. A **class-default** traffic class is pre-defined and can be added to any policy. It is always placed at the end of a policy map. With an implied **match any** included in the **class-default** class, all packets that have not already matched the other traffic classes will match **class-default**. **Note** To delete an existing class map, use the **no class** *class-map-name* policy-map configuration command.
Step 5	**trust** [**cos** | **dscp** | **ip-precedence**] **Example:** `Device(config-pmap-c)# trust dscp`	Configures the trust state, which QoS uses to generate a CoS-based or DSCP-based QoS label. This command is mutually exclusive with the **set** command within the same policy map. If you enter the **trust** command, go to Step 6. 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

	Command or Action	Purpose
		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. **Note** To return to the untrusted state, use the **no trust** policy-map configuration command
Step 6	**set** {**dscp** *new-dscp* | **ip precedence** *new-precedence*} **Example:** `Device(config-pmap-c)# set dscp 45`	Classifies IP traffic by setting a new value in the packet. • For **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. **Note** To remove an assigned DSCP or IP precedence value, use the **no set** {**dscp** *new-dscp* | **ip** precedence *new-precedence*} policy-map configuration command.
Step 7	**police** *rate-bps burst-byte* [**exceed-action** {**drop** | **policed-dscp-transmit**}] **Example:** `Device(config-pmap-c)# police 100000 80000 drop`	Defines a policer for the classified traffic. By default, no policer is defined. • For *rate-bps,* specify average traffic rate in bits per second (b/s). The range is 8000 to 10000000000. • For *burst-byte,* specify the normal burst size in bytes. The range is 8000 to 1000000. • (Optional) Specifies the action to take when the rates are exceeded. Use the **exceed-action drop** keywords to drop the

	Command or Action	Purpose
		packet. Use the **exceed-action policed-dscp-transmit** keywords to mark down the DSCP value (by using the policed-DSCP map) and to send the packet. **Note** To remove an existing policer, use the **no police** *rate-bps burst-byte* [**exceed-action** {**drop** | **policed-dscp-transmit**}] policy-map configuration command.
Step 8	**exit** **Example:** `Device(config-pmap-c)# exit`	Returns to policy map configuration mode.
Step 9	**exit** **Example:** `Device(config-pmap)# exit`	Returns to global configuration mode.
Step 10	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 2/0/1`	Specifies the port to attach to the policy map, and enters interface configuration mode. Valid interfaces include physical ports.
Step 11	**service-policy input** *policy-map-name* **Example:** `Device(config-if)# service-policy input flowit`	Specifies the policy-map name, and applies it to an ingress port. Only one policy map per ingress port is supported. **Note** To remove the policy map and port association, use the **no service-policy** *input policy-map-name* interface configuration command.
Step 12	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 13	**show policy-map** [*policy-map-name* [**class** *class-map-name*]] **Example:** `Device# show policy-map`	Verifies your entries.
Step 14	**copy running-config startup-config** **Example:** `Device# copy-running-config startup-config`	(Optional) Saves your entries in the configuration file.

Classifying, Policing, and Marking 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.

You can configure aggregate policers only in nonhierarchical policy maps on physical ports.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**mls qos aggregate-policer** *aggregate-policer-name rate-bps burst-byte* **exceed-action** {**drop** | **policed-dscp-transmit**} **Example:** `Device(config)# mls qos aggregate-police transmit1 48000 8000 exceed-action policed-dscp-transmit`	Defines the policer parameters that can be applied to multiple traffic classes within the same policy map. By default, no aggregate policer is defined. • For *aggregate-policer-name*, specify the name of the aggregate policer. • For *rate-bps*, specify average traffic rate in bits per second (b/s). The range is 8000 to 10000000000. • For *burst-byte*, specify the normal burst size in bytes. The range is 8000 to 1000000. • Specifies 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**

	Command or Action	Purpose
		keywords to mark down the DSCP value (by using the policed-DSCP map) and to send the packet.
Step 3	**class-map** [**match-all** | **match-any**] *class-map-name* **Example:** `Device(config)# class-map ipclass1`	Creates a class map to classify traffic as necessary.
Step 4	**policy-map** *policy-map-name* **Example:** `Device(config-cmap)# policy-map aggflow1`	Creates a policy map by entering the policy map name, and enters policy-map configuration mode.
Step 5	**class** [*class-map-name* | **class-default**] **Example:** `Device(config-cmap-p)# class ipclass1`	Defines a traffic classification, and enters policy-map class configuration mode.
Step 6	**police aggregate** *aggregate-policer-name* **Example:** `Device(configure-cmap-p)# police aggregate transmit1`	Applies an aggregate policer to multiple classes in the same policy map. For *aggregate-policer-name*, enter the name specified in Step 2. To remove the specified aggregate policer from a policy map, use the **no police aggregate** *aggregate-policer-name* policy map configuration command. To delete an aggregate policer and its parameters, use the **no mls qos aggregate-policer** *aggregate-policer-name* global configuration command.
Step 7	**exit** **Example:** `Device(configure-cmap-p)# exit`	Returns to global configuration mode.
Step 8	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 2/0/1`	Specifies the port to attach to the policy map, and enters interface configuration mode. Valid interfaces include physical ports.

	Command or Action	Purpose
Step 9	**service-policy input** *policy-map-name* **Example:** `Device(config-if)# service-policy input aggflow1`	Specifies the policy-map name, and applies it to an ingress port. Only one policy map per ingress port is supported.
Step 10	**end** **Example:** `Device(configure-if)# end`	Returns to privileged EXEC mode.
Step 11	**show mls qos aggregate-policer** [*aggregate-policer-name*] **Example:** `Device# show mls qos aggregate-policer transmit1`	Verifies your entries.
Step 12	**copy running-config startup-config** **Example:** `Device# copy-running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring DSCP Maps

Configuring the CoS-to-DSCP Map

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

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

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**mls qos map cos-dscp** *dscp1...dscp8*	Modifies the CoS-to-DSCP map.

	Command or Action	Purpose
	Example: `Device(config)# mls qos map cos-dscp 10 15 20 25 30 35 40 45`	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. **Note** To return to the default map, use the **no mls qos cos-dscp** global configuration command.
Step 3	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 4	**show mls qos maps cos-dscp** **Example:** `Device# show mls qos maps cos-dscp`	Verifies your entries.
Step 5	**copy running-config startup-config** **Example:** `Device# copy-running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring the IP-Precedence-to-DSCP Map

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

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

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**mls qos map ip-prec-dscp** *dscp1...dscp8* **Example:**	Modifies the IP-precedence-to-DSCP map.

	Command or Action	Purpose
	`Device(config)# mls qos map ip-prec-dscp 10 15 20 25 30 35 40 45`	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. **Note** To return to the default map, use the **no mls qos ip-prec-dscp** global configuration command.
Step 3	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 4	**show mls qos maps ip-prec-dscp** **Example:** `Device# show mls qos maps ip-prec-dscp`	Verifies your entries.
Step 5	**copy running-config startup-config** **Example:** `Device# copy-running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring the Policed-DSCP Map

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

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

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

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**mls qos map policed-dscp** *dscp-list* **to** *mark-down-dscp*	Modifies the policed-DSCP map.

	Command or Action	Purpose
	Example: `Device(config)# mls qos map policed-dscp 50 51 52 53 54 55 56 57 to 0`	• 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. **Note** To return to the default map, use the **no mls qos policed-dscp** global configuration command.
Step 3	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 4	**show mls qos maps policed-dscp** **Example:** `Device(config)# show mls qos maps policed-dscp`	Verifies your entries.
Step 5	**copy running-config startup-config** **Example:** `Device#`	(Optional) Saves your entries in the configuration file.

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 egress queues.

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

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**mls qos map dscp-cos** *dscp-list* **to** *cos* **Example:**	Modifies the DSCP-to-CoS map.

	Command or Action	Purpose
	`Device# mls qos map dscp-cos 0 8 16 24 32 40 48 50 to 0`	• 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. **Note** To return to the default map, use the **no mls qos dscp-cos** global configuration command.
Step 3	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 4	**show mls qos maps dscp-to-cos** **Example:** `Device# show mls qos maps dscp-to-cos`	Verifies your entries.
Step 5	**copy running-config startup-config** **Example:** `Device# copy-running-config startup-config`	(Optional) Saves your entries in the configuration file.

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 applies the new value to the packet. The sends the packet out the port with the new DSCP value.

You can configure multiple DSCP-to-DSCP-mutation maps on an ingress port. The default DSCP-to-DSCP-mutation map is a null map, which maps an incoming 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.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**mls qos map dscp-mutation** *dscp-mutation-name in-dscp* **to** *out-dscp* **Example:** `Device(config)# mls qos map dscp-mutation mutation1 1 2 3 4 5 6 7 to 0`	Modifies 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. **Note** To return to the default map, use the **no mls qos dscp-mutation** *dscp-mutation-name* global configuration command.
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet1/0/1`	Specifies the port to which to attach the map, and enters interface configuration mode. Valid interfaces include physical ports.
Step 4	**mls qos trust dscp** **Example:** `Device(config-if)# mls qos trust dscp`	Configures the ingress port as a DSCP-trusted port. By default, the port is not trusted.
Step 5	**mls qos dscp-mutation** *dscp-mutation-name* **Example:** `Device(config-if)# mls qos dscp-mutation mutation1`	Applies 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** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-if)# end`	
Step 7	**show mls qos maps dscp-mutation** **Example:** `Device# show mls qos maps dscp-mutation`	Verifies your entries.
Step 8	**copy running-config startup-config** **Example:** `Device# copy-running-config startup-config`	(Optional) Saves your entries in the configuration file.

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 modules. You 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?

Configuration Guidelines

Follow these guidelines when the expedite queue is enabled or the egress queues are serviced based on their SRR weights:

- If the egress expedite queue is enabled, it overrides the SRR shaped and shared weights for queue 1.
- If the egress expedite queue is disabled and the SRR shaped and shared weights are configured, the shaped mode overrides the shared mode for queue 1, and SRR services this queue in shaped mode.
- If the egress expedite queue is disabled and the SRR shaped weights are not configured, SRR services this queue in shared mode.

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

You can prioritize traffic by placing packets with particular DSCPs or CoSs into certain queues and 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.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	Use one of the following: • **mls qos srr-queue input dscp-map queue** *queue-id* **threshold** *threshold-id dscp1...dscp8* • **mls qos srr-queue input cos-map queue** *queue-id* **threshold** *threshold-id cos1...cos8* **Example:** `Device(config)# mls qos srr-queue input dscp-map queue 1 threshold 2 20 21 22 23 24 25 26`	Maps 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* **Example:** `Device(config)# mls qos srr-queue input threshold 1 50 70`	Assigns 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.

	Command or Action	Purpose
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show mls qos maps** **Example:** `Device# show mls qos maps`	Verifies 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** **Example:** `Device# copy running-config startup-config`	(Optional) Saves 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

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.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 2	**mls qos srr-queue input buffers** *percentage1 percentage2* **Example:** `Device(config)# mls qos srr-queue input buffers 60 40`	Allocates 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 incoming bursty traffic.
Step 3	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Step 4	Use one of the following:
	• **show mls qos interface buffer**
	• **show mls qos input-queue**
	Example:
	`Device# show mls qos interface buffer`
	or
	`Device# show mls qos input-queue`
		Verifies your entries.

Step 5	**copy running-config startup-config** **Example:** `Device# copy-running-config startup-config`	(Optional) Saves your entries in the configuration file. To return to the default setting, use the **no mls qos srr-queue input buffers** global configuration command.

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. 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.

Note SRR bandwidth limit works in both mls qos enabled and disabled states.

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

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**mls qos srr-queue input bandwidth** *weight1 weight2* **Example:** `Device(config)# mls qos srr-queue input bandwidth 25 75`	Assigns 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.
Step 3	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Step 4	Use one of the following:
	• **show mls qos interface queueing**
	• **show mls qos input-queue**
	Example:
	`Device# show mls qos interface queueing`
	or
	`Device# show mls qos input-queue`
		Verifies your entries.

	Command or Action	Purpose
Step 5	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file. To return to the default setting, use the **no mls qos srr-queue input bandwidth** global configuration command.

Configuring Egress Queue Characteristics

Depending on the complexity of your network and your QoS solution, you might need to perform all of the tasks in the following modules. You 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 queues be serviced and which technique (shaped, shared, or both) should be used?

Configuration Guidelines

Follow these guidelines when the expedite queue is enabled or the egress queues are serviced based on their SRR weights:

- If the egress expedite queue is enabled, it overrides the SRR shaped and shared weights for queue 1.
- If the egress expedite queue is disabled and the SRR shaped and shared weights are configured, the shaped mode overrides the shared mode for queue 1, and SRR services this queue in shaped mode.
- If the egress expedite queue is disabled and the SRR shaped weights are not configured, SRR services this queue in shared mode.

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 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 buffers, 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 switch supports 4 egress queues by default, although there is an option to enable a total of 8 egress queues. Use the **mls qos srr-queue output queues 8** global configuration command to enable all 8 egress queues. Once 8 egress queues are enabled, you are able to configure thresholds, buffers, bandwidth share weights, and bandwidth shape weights for all 8 queues. The 8 egress queue configuration is only supported on a standalone switch.

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

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

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**mls qos srr-queue output queues 8** **Example:** `Device(config)# mls qos srr-queue output queues 8`	(Optional) The switch supports 4 egress queues by default, although you can enable a total of 8 egress queues. Use the optional **mls qos srr-queue output queues 8** command to enable the additional 4 egress queues. Once 8 queue support is enabled, you can then proceed to configure the additional 4 queues. Any existing egress queue configuration commands are then modified to support the additional queue parameters. **Note** The option to enable 8 queues is only available on a standalone switch.
Step 3	**mls qos queue-set output** *qset-id* **buffers** *allocation1 ... allocation8* **Example:** `Device(config)# mls qos queue-set output 2 buffers 40 20 20 20 10 10 10 10`	Allocates 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. When eight egress queues are configured, then by default 30 percent of the total buffer space is allocated to queue 2 and 10 percent (each) to queues 1,3,4,5,6,7, and 8.

	Command or Action	Purpose
		If you enabled 8 egress queues as described in Step 2 above, then the following applies: • 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 ... allocation8*, specify eight percentages, one for each queue in the queue set. For *allocation1*, *allocation3*, and *allocation4* to *allocation8*, the range is 0 to 99. For *allocation2*, the range is 1 to 100 (including the CPU buffer). Allocate buffers according to the importance of the traffic; for example, give a large percentage of the buffer to the queue with the highest-priority traffic. **Note** To return to the default setting, use the **no mls qos queue-set output** *qset-id* **buffers** global configuration command.
Step 4	**mls qos queue-set output** *qset-id* **threshold** *queue-id drop-threshold1 drop-threshold2 reserved-threshold maximum-threshold* **Example:** `Device(config)# mls qos queue-set output 2 threshold 2 40 60 100 200`	Configures 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 200 percent. The reserved thresholds for queues 1, 2, 3, and 4 are set to 50 percent. The maximum thresholds for all queues are set to 400 percent by default. If you enabled 8 egress queues as described in Step 2 above, then the following applies: • 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 queue-id range is 1-4 by default and 1-8 when 8 queues are enabled. • 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 3200 percent.

	Command or Action	Purpose
		• 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 queue in the full condition 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 3200 percent. **Note** To return to the default WTD threshold percentages, use the **no mls qos queue-set output** *qset-id* **threshold** [*queue-id*] global configuration command.
Step 5	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet1/0/1`	Specifies the port of the outbound traffic, and enter interface configuration mode.
Step 6	**queue-set** *qset-id* **Example:** `Device(config-id)# queue-set 2`	Maps 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 7	**end** **Example:** `Device(config-id)# end`	Returns to privileged EXEC mode.
Step 8	**show mls qos interface** [*interface-id*] **buffers** **Example:** `Device# show mls qos interface buffers`	Verifies your entries.
Step 9	**copy running-config startup-config** **Example:** `Device# copy-running-config`	(Optional) Saves your entries in the configuration file. To return to the default setting, use the **no mls qos queue-set output** *qset-id* **buffers** global

	Command or Action	Purpose
	`startup-config`	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.

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

You can prioritize traffic by placing packets with particular DSCPs or costs of service into certain queues and adjusting the queue thresholds so that packets with lower priorities are dropped.

Note The egress queue default settings are suitable for most situations. You should change them only when you have a thorough understanding of egress queues and 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.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	Use one of the following: • **mls qos srr-queue output dscp-map queue** *queue-id* **threshold** *threshold-id dscp1...dscp8* • **mls qos srr-queue output cos-map queue** *queue-id* **threshold** *threshold-id cos1...cos8* **Example:** `Device(config)# mls qos srr-queue output dscp-map queue 1 threshold 2 10 11`	Maps DSCP or CoS values to an egress queue and to a threshold ID. 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.

	Command or Action	Purpose
		Note If you enabled 8 egress queues using the **mls qos srr-queue output queues 8** global configuration command, then the *queue-id* range would be from 1 to 8. • 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. **Note** 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.
Step 3	**mls qos srr-queue output cos-map queue** *queue-id* **threshold** *threshold-id cos1...cos8* **Example:** `Device(config)# mls qos srr-queue output cos-map queue 3 threshold 1 2 3`	Maps CoS values to an egress queue and to a threshold ID. 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 *cos1...cos8*, enter up to eight values, and separate each value with a space. The range is 0 to 7.

	Command or Action	Purpose
		Note To return to the default CoS output queue threshold map, use the **no mls qos srr-queue output cos-map** global configuration command.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show mls qos maps** **Example:** `Device# show mls qos maps`	Verifies 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 6	**copy running-config startup-config** **Example:** `Device# copy-running-config startup-config`	(Optional) Saves 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.

Configuring SRR Shaped Weights on Egress Queues

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

You can configure the egress queues for shaped or shared weights, or both. Use shaping to smooth bursty traffic or to provide a smoother output over time.

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

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet2/0/1`	Specifies the port of the outbound traffic, and enters interface configuration mode.
Step 3	**srr-queue bandwidth shape** *weight1 weight2 weight3 weight4* **Example:** `Device(config-if)# srr-queue bandwidth shape 8 0 0 0`	Assigns SRR weights to the egress queues. 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. To return to the default setting, use the **no srr-queue bandwidth shape** interface configuration command. **Note** If you enabled 8 egress queues using the **mls qos srr-queue output queues 8** global configuration command, then you would be able to assign SRR weights to a total of 8 queues.

	Command or Action	Purpose
Step 4	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 5	**show mls qos interface** *interface-id* **queueing** **Example:** `Device# show mls qos interface interface-id queuing`	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file. To return to the default setting, use the **no srr-queue bandwidth shape** interface configuration command.

Configuring SRR Shared Weights on Egress 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 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 default settings are suitable for most situations. You should change them only when you have a thorough understanding of the egress queues and 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 the four egress queues mapped to a port. This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**interface** *interface-id* **Example:** `Device(config)# interface`	Specifies the port of the outbound traffic, and enters interface configuration mode.

	Command or Action	Purpose
	`gigabitethernet2/0/1`	
Step 3	**srr-queue bandwidth share** *weight1 weight2 weight3 weight4* **Example:** `Device(config-id)# srr-queue bandwidth share 1 2 3 4`	Assigns 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. To return to the default setting, use the **no srr-queue bandwidth share** interface configuration command. **Note** If you enabled 8 egress queues using the **mls qos srr-queue output queues 8** global configuration command, then you would be able to assign SRR weights to a total of 8 queues.
Step 4	**end** **Example:** `Device(config-id)# end`	Returns to privileged EXEC mode.
Step 5	**show mls qos interface** *interface-id* **queueing** **Example:** `Device# show mls qos interface interface_id queuing`	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** `Device# copy-running-config startup-config`	(Optional) Saves your entries in the configuration file. To return to the default setting, use the **no srr-queue bandwidth share** interface configuration command.

Configuring the Egress Expedite Queue

You can ensure that certain packets have priority over all others by queuing them in the egress expedite queue. SRR services this queue until it is empty before servicing the other queues.

Beginning in privileged EXEC mode, follow these steps to enable the egress expedite queue. This procedure is optional.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**mls qos** **Example:** `Device(config)# mls qos`	Enables QoS on a switch.
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet1/0/1`	Specifies the egress port, and enters interface configuration mode.
Step 4	**priority-queue out** **Example:** `Device(config-if)# priority-queue out`	Enables the egress expedite queue, which is disabled by default. When you configure this command, the SRR weight and queue size ratios are affected because there is one fewer queue participating in SRR. This means that *weight1* in the **srr-queue bandwidth shape** or the **srr-queue bandwidth share** command is ignored (not used in the ratio calculation). **Note** To disable the egress expedite queue, use the **no priority-queue out** interface configuration command.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.

	Command or Action	Purpose
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file. To disable the egress expedite queue, use the **no priority-queue out** interface configuration command.

Limiting the Bandwidth on an Egress Interface

You can limit the bandwidth on an egress port. 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 default settings are suitable for most situations. You should change them only when you have a thorough understanding of the egress queues and if these settings do not meet your QoS solution.

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

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet2/0/1`	Specifies the port to be rate-limited, and enters interface configuration mode.
Step 3	**srr-queue bandwidth limit** *weight1* **Example:** `Device(config-if)# srr-queue bandwidth limit 80`	Specifies the percentage of the port speed to which the port should be limited. The range is 10 to 90. By default, the port is not rate-limited and is set to 100 percent. **Note** To return to the default setting, use the **no srr-queue bandwidth limit** interface configuration command.

	Command or Action	Purpose
Step 4	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 5	**show mls qos interface** [*interface-id*] **queueing** **Example:** `Device# show mls qos interface interface_id queueing`	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** `Device# copy-running-config startup-config`	(Optional) Saves your entries in the configuration file. To return to the default setting, use the **no srr-queue bandwidth limit** interface configuration command.

Monitoring Standard QoS

Table 17: Commands for Monitoring Standard QoS on the Switch

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

Command	Description
show policy-map [*policy-map-name* [**class** *class-map-name*]]	Displays QoS policy maps, which define classificatio incoming traffic. Do not use the **show policy-map interface** privileged command to display classification information for inc The **control-plane** and **interface** keywords are not su the statistics shown in the display should be ignored.
show running-config | include rewrite	Displays the DSCP transparency setting.

Configuration Examples for QoS

Example: Configuring Port to the DSCP-Trusted State and Modifying the DSCP-to-DSCP-Mutation Map

This example shows how to configure a port to the DSCP-trusted state and to modify the DSCP-to-DSCP-mutation map (named *gi1/0/2-mutation*) so that incoming DSCP values 10 to 13 are mapped to DSCP 30:

```
Device(config)# mls qos map dscp-mutation gigabitethernet1/0/2-mutation
10 11 12 13 to 30
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# mls qos trust dscp
Device(config-if)# mls qos dscp-mutation gigabitethernet1/0/2-mutation
Device(config-if)# end
```

Examples: Classifying Traffic by Using ACLs

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.

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

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:

```
Device(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:

```
Device(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:

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

This example shows how to create an ACL that permits IPv6 traffic from any source to any destination that has the DSCP value set to 32:

```
Device(config)# ipv6 access-list 100 permit ip any any dscp 32
```

This example shows how to create an ACL that permits IPv6 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:

```
Device(config)# ipv6 access-list ipv6_Name_ACL permit ip host 10::1 host 10.1.1.2
precedence 5
```

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.

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

Examples: Classifying Traffic by Using Class Maps

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.

```
Device(config)# access-list 103 permit ip any any dscp 10
Device(config)# class-map class1
Device(config-cmap)# match access-group 103
Device(config-cmap)# end
Device#
```

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

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

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

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

This example shows how to configure a class map to match IP DSCP and IPv6:

```
Device(config)# Class-map cm-1
Device(config-cmap)# match ip dscp 10
Device(config-cmap)# match protocol ipv6
Device(config-cmap)# exit
Device(config)# Class-map cm-2
Device(config-cmap)# match ip dscp 20
Device(config-cmap)# match protocol ip
Device(config-cmap)# exit
Device(config)# Policy-map pm1
Device(config-pmap)# class cm-1
Device(config-pmap-c)# set dscp 4
Device(config-pmap-c)# exit
Device(config-pmap)# class cm-2
Device(config-pmap-c)# set dscp 6
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# interface G1/0/1
Device(config-if)# service-policy input pm1
```

This example shows how to configure a class map that applies to both IPv4 and IPv6 traffic:

```
Device(config)# ip access-list 101 permit ip any any
Device(config)# ipv6 access-list ipv6-any permit ip any any
Device(config)# Class-map cm-1
Device(config-cmap)# match access-group 101
Device(config-cmap)# exit
Device(config)# class-map cm-2
Device(config-cmap)# match access-group name ipv6-any
Device(config-cmap)# exit
Device(config)# Policy-map pm1
Device(config-pmap)# class cm-1
Device(config-pmap-c)# set dscp 4
Device(config-pmap-c)# exit
Device(config-pmap)# class cm-2
Device(config-pmap-c)# set dscp 6
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# interface G0/1
Device(config-if)# switch mode access
Device(config-if)# service-policy input pm1
```

Examples: Classifying, Policing, and Marking Traffic on Physical Ports Using Policy Maps

This example shows how to create a policy map and attach it to an ingress 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 incoming packet is trusted. If the matched traffic exceeds an average traffic rate of 48000 b/s and a normal burst size of 8000 bytes, its DSCP is marked down (based on the policed-DSCP map) and sent:

```
Device(config)# access-list 1 permit 10.1.0.0 0.0.255.255
Device(config)# class-map ipclass1
Device(config-cmap)# match access-group 1
Device(config-cmap)# exit
Device(config)# policy-map flow1t
Device(config-pmap)# class ipclass1
Device(config-pmap-c)# trust dscp
Device(config-pmap-c)# police 1000000 8000 exceed-action policed-dscp-transmit
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# interface gigabitethernet2/0/1
Device(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 an ingress 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.

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

This example shows how to create a class map that applies to both IPv4 and IPv6 traffic with the default class applied to unclassified traffic:

```
Device(config)# ip access-list 101 permit ip any any
Device(config)# ipv6 access-list ipv6-any permit ip any any
Device(config)# class-map cm-1
Device(config-cmap)# match access-group 101
Device(config-cmap)# exit
Device(config)# class-map cm-2
Device(config-cmap)# match access-group name ipv6-any
Device(config-cmap)# exit
Device(config)# policy-map pm1
```

```
Device(config-pmap)# class cm-1
Device(config-pmap-c)# set dscp 4
Device(config-pmap-c)# exit
Device(config-pmap)# class cm-2
Device(config-pmap-c)# set dscp 6
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap-c)# set dscp 10
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# interface G0/1
Device(config-if)# switch mode access
Device(config-if)# service-policy input pm1
```

Examples: Classifying, Policing, and Marking Traffic by Using Aggregate Policers

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 incoming 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 b/s and a normal burst size of 8000 bytes, its DSCP is marked down (based on the policed-DSCP map) and sent. The policy map is attached to an ingress port.

```
Device(config)# access-list 1 permit 10.1.0.0 0.0.255.255
Device(config)# access-list 2 permit 11.3.1.1
Device(config)# mls qos aggregate-police transmit1 48000 8000 exceed-action
policed-dscp-transmit
Device(config)# class-map ipclass1
Device(config-cmap)# match access-group 1
Device(config-cmap)# exit
Device(config)# class-map ipclass2
Device(config-cmap)# match access-group 2
Device(config-cmap)# exit
Device(config)# policy-map aggflow1
Device(config-pmap)# class ipclass1
Device(config-pmap-c)# trust dscp
Device(config-pmap-c)# police aggregate transmit1
Device(config-pmap-c)# exit
Device(config-pmap)# class ipclass2
Device(config-pmap-c)# set dscp 56
Device(config-pmap-c)# police aggregate transmit1
Device(config-pmap-c)# exit
Device(config-pmap)# class class-default
Device(config-pmap-c)# set dscp 10
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# interface gigabitethernet2/0/1
Device(config-if)# service-policy input aggflow1
Device(config-if)# exit
```

Examples: Configuring DSCP Maps

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

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

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

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

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

```
Device(config)# mls qos map policed-dscp 50 51 52 53 54 55 56 57 to 0
Device(config)# end
Device# 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.

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:

```
Device(config)# mls qos map dscp-cos 0 8 16 24 32 40 48 50 to 0
Device(config)# end
Device# 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.

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):

```
Device(config)# mls qos map dscp-mutation mutation1 1 2 3 4 5 6 7 to 0
Device(config)# mls qos map dscp-mutation mutation1 8 9 10 11 12 13 to 10
Device(config)# mls qos map dscp-mutation mutation1 20 21 22 to 20
Device(config)# mls qos map dscp-mutation mutation1 30 31 32 33 34 to 30
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# mls qos trust dscp
Device(config-if)# mls qos dscp-mutation mutation1
Device(config-if)# end
Device# 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.

Examples: Configuring Ingress Queue Characteristics

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:

```
Device(config)# mls qos srr-queue input dscp-map queue 1 threshold 1 0 1 2 3 4 5 6
```

```
Device(config)# mls qos srr-queue input dscp-map queue 1 threshold 2 20 21 22 23 24
25 26
Device(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.

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:

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

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):

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

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:

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

Examples: Configuring Egress Queue Characteristics

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

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

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:

```
Device(config)# interface gigabitethernet2/0/1
Device(config-if)# srr-queue bandwidth shape 8 0 0 0
```

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.

```
Device(config)# interface gigabitethernet2/0/1
Device(config-if)# srr-queue bandwidth share 1 2 3 4
```

This example shows how to enable the egress expedite queue when the SRR weights are configured. The egress expedite queue overrides the configured SRR weights.

```
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# srr-queue bandwidth shape 25 0 0 0
Device(config-if)# srr-queue bandwidth share 30 20 25 25
Device(config-if)# priority-queue out
Device(config-if)# end
```

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

```
Device(config)# interface gigabitethernet2/0/1
Device(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 Mb/s. These values are not exact because the hardware adjusts the line rate in increments of six.

Where to Go Next

Review the auto-QoS documentation to see if you can use these automated capabilities for your QoS configuration.

Additional References

Related Documents

Related Topic	Document Title
List of Cisco network devices supporting Cisco EnergyWise	Cisco IOS Release Notes for Cisco EnergyWise, EnergyWise Version 2.8
EnergyWise Commands	
IP-Enabled Energy Management	IP-Enabled Energy Management: A Proven Strategy for Administering Energy as a Service
Cisco EnergyWise partner documentation	Go to the Cisco Developer Network. • *Cisco EnergyWise Documentation Roadmap* • *Cisco EnergyWise Partner Development Guide* • *Cisco EnergyWise Programmer Reference Guide for the Endpoint SDK* • *Cisco EnergyWise Programmer Reference Guide for the Management API*

MIBs

MIB	MIBs Link
Cisco EnergyWise domain members support the CISCO-ENERGYWISE-MIB.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco IOS MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature History and Information for QoS

Release	Modification
Cisco IOS Release 15.0(2)EX	This feature was introduced.