Policy-Based Queueing

- Prerequisites for Policy-Based Queueing, page 61-1
- Restrictions for Policy-Based Queueing, page 61-2
- Information About Policy-Based Queueing, page 61-3
- How to Configure Policy-Based Queueing, page 61-10
- Configuration Examples for Policy-Based Queueing, page 61-16

Note
- Queueing is optional. Use the commands described in this section to configure queueing on ports that serve congested links.
- For complete syntax and usage information for the commands used in this chapter, see these publications:
- Cisco IOS Release 12.2SY supports only Ethernet interfaces. Cisco IOS Release 12.2SY does not support any WAN features or commands.

Tip
For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:
Participate in the Technical Documentation Ideas forum

Prerequisites for Policy-Based Queueing

None.
Restrictions for Policy-Based Queueing

- DSCP-based queueing is supported on 8q4t, 1p7q2t, and 1p7q4t ports. The Supervisor Engine 2T-10GE ports are 8q4t/1p7q4t with the `platform qos 10g-only` global configuration command configured, which disables the Gigabit Ethernet ports on the supervisor engine.

  Note: In releases where CSCts82932 is not resolved, do not use the default DSCP-based queue mapping for 8q4t ingress queues unless you configure supporting bandwidth and queue limits.

- The presence of a `match dscp` command or a `match precedence` command in a class map that is used in a queueing policy attached to a port enables DSCP-based queueing on the port in the direction that the queueing policy is attached.

- CoS-based queueing is always used for non-IP traffic, IP multicast traffic, and IP unknown unicast flood traffic.

- Class maps that are used in queueing policies can contain any combination and number of `match dscp`, `match precedence`, or `match cos` commands.

- You can attach one input and one output queueing policy to an interface, in addition to policies that configure marking or policing.

- To support migration from Cisco IOS Release 12.2SX configurations, Cisco IOS Release 12.2SY supports global configuration mode and interface configuration mode queueing commands.
  - When you attach an ingress or egress queueing policy to a port, all interface configuration mode queueing commands on the port are deleted.
  - An attached ingress or egress queueing policy supercedes the effect of any configured global configuration mode queueing commands.
  - If you attach a queueing policy in only one direction, the queueing configuration of the other direction is either the defaults or is defined by any configured global configuration mode queueing commands.
  - You cannot configure any interface configuration mode queueing commands on a port that has a queueing policy attached.

- Policy-based queueing is supported on all modules that Cisco IOS Release 12.2SY supports with the Supervisor Engine 2T-10GE.

- Queueing policies are specific to particular port types because a queueing policy cannot contain any commands that are not supported by the port to which it will be attached (see the “Configuration Examples for Policy-Based Queueing” section on page 61-16). Commands not supported on a port are not ignored. You cannot successfully apply a policy with unsupported commands to a port.

- For clarity, configure queueing policy names that correspond to the port type it supports; for example `1q2t_1q8t_ingress`.

- Queueing policies can contain multiple class maps; each class map configures a queue.
A policy-map class defined by a `class` command that uses a class map and that contains the policy-map class `priority` command configures the priority queue. (The priority queue is the highest-numbered queue.) The class-map filters the QoS values (CoS or DSCP) it is configured with to the priority queue.

Enabling SRR on a port disables the priority queue.

A policy-map class defined by a `class` command that uses a class map and that does not contain the policy-map class `priority` command configures the highest-numbered nonpriority queue. (If the port has a priority queue, the highest-numbered nonpriority queue is numbered one less than the priority queue. If the port does not have a priority queue, the highest-numbered queue is a nonpriority queue.) Subsequent such commands define the configuration of the remaining nonpriority queues in reverse numerical order. You cannot skip a queue, but configuration of all queues is not required. The class-map filters the QoS values it is configured with to the nonpriority queue being configured.

A `class` command that uses the `class-default` keywords configures queue 1. The `class-default` keywords filter all remaining QoS values to queue 1.

For each nonpriority queue, policy-map class `queue-limit` or `random-detect` commands assign QoS values (CoS or DSCP) to thresholds within a queue. Thresholds are configured in numerical order. You cannot skip a threshold, but configuration of all threshold is not required. QoS values that are to be applied to the thresholds must be from the group of values that the class-map filters to the queue.

- The first `queue-limit` or `random-detect` command assigns QoS values to the first threshold, and configures the percentage value applied to it.
- Subsequent `queue-limit` or `random-detect` commands that are configured with the same percentage value assign additional QoS values to the first threshold.
- The next `queue-limit` or `random-detect` command that is configured with a different percentage value assigns QoS values to the numerically next threshold, and configures the percentage value applied to it.
- Subsequent `queue-limit` or `random-detect` commands that are configured with the same percentage value assign additional QoS values to the numerically next threshold.
- Each `queue-limit` or `random-detect` command with a different percentage defines the next unconfigured threshold and any subsequent commands that repeat a percentage value assign additional QoS values to a configured threshold.
- All unconfigured thresholds are at 100%.
- All unassigned QoS labels are assigned to the highest-numbered threshold.

**Information About Policy-Based Queueing**

- Port-Based Queue Types, page 61-3
- Queueing Policies, page 61-8

**Port-Based Queue Types**

- Ingress and Egress Buffers and Queues, page 61-4
- Ingress Queue Types, page 61-5
Ingress and Egress Buffers and Queues

The Ethernet port ASICs have buffers that are divided into a fixed number of queues. When congestion avoidance is enabled, PFC QoS uses the traffic’s Layer 2 CoS value or, on some port types, the Layer 3 DSCP values, to assign traffic to the queues. The buffers and queues store frames temporarily as they transit the switch. PFC QoS allocates the port ASIC memory as buffers for each queue on each port.

The Ethernet ports support the following types of queues:

- Nonpriority queues
- Priority queues

The Ethernet ports support the following types of scheduling algorithms between queues:

- Shaped round robin (SRR)—SRR allows a queue to use only the allocated bandwidth.
- Deficit weighted round robin (DWRR)—DWRR keeps track of any lower-priority queue under-transmission caused by traffic in a higher-priority queue and compensates in the next round.
- Weighted Round Robin (WRR)—WRR does not explicitly reserve bandwidth for the queues. Instead, the amount of bandwidth assigned to each queue is user configurable. The percentage or weight allocated to a queue defines the amount of bandwidth allocated to the queue.
- Priority queueing—Strict priority queueing allows delay-sensitive data such as voice to be dequeued and sent before packets in other queues are dequeued, giving delay-sensitive data preferential treatment over other traffic. The switch services traffic in the strict-priority transmit queue before servicing the nonpriority queues. After transmitting a packet from a nonpriority queue, the switch checks for traffic in the strict-priority queue. If the switch detects traffic in the strict-priority queue, it suspends its service of the nonpriority queue and completes service of all traffic in the strict-priority queue before returning to the nonpriority queue.

The Ethernet ports provide congestion avoidance with these types of thresholds within a queue:

- Weighted Random Early Detection (WRED)—On ports with WRED drop thresholds, frames with a given QoS label are admitted to the queue based on a random probability designed to avoid buffer congestion. The probability of a frame with a given QoS label being admitted to the queue or discarded depends on the weight and threshold assigned to that QoS label.

  For example, if CoS 2 is assigned to queue 1, threshold 2, and the threshold 2 levels are 40 percent (low) and 80 percent (high), then frames with CoS 2 will not be dropped until queue 1 is at least 40 percent full. As the queue depth approaches 80 percent, frames with CoS 2 have an increasingly higher probability of being discarded rather than being admitted to the queue. Once the queue is over 80 percent full, all CoS 2 frames are dropped until the queue is less than 80 percent full. The frames the switch discards when the queue level is between the low and high thresholds are picked out at random, rather than on a per-flow basis or in a FIFO manner. This method works well with protocols such as TCP that can adjust to periodic packet drops by backing off and adjusting their transmission window size.

- Tail-drop thresholds—On ports with tail-drop thresholds, frames with a given QoS label are admitted to the queue until the drop threshold associated with that QoS label is exceeded; subsequent frames of that QoS label are discarded until the threshold is no longer exceeded. For example, if CoS 1 is assigned to queue 1, threshold 2, and the threshold 2 watermark is 60 percent, then frames with CoS 1 will not be dropped until queue 1 is 60 percent full. All subsequent CoS 1 frames will be dropped until the queue is less than 60 percent full. With some port types, you can configure the nonpriority receive queue to use both a tail-drop and a WRED-drop threshold by
mapping a CoS value to the queue or to the queue and a threshold. The switch uses the tail-drop threshold for traffic carrying CoS values mapped only to the queue. The switch uses WRED-drop thresholds for traffic carrying CoS values mapped to the queue and a threshold. All LAN ports of the same type use the same drop-threshold configuration.

```
Note
• You can enable DSCP-based queues and thresholds on 8q4t, 1p7q2t, and 1p7q4t ports (see the “Module to Queue Type Mappings” section on page 61-6), either in a queuing policy or with legacy interface commands (see the “Legacy Configuration Procedures for DSCP-Based Queue Mapping” section on page 62-14).

• DSCP-based queueing is supported on 8q4t, 1p7q2t, and 1p7q4t ports. The Supervisor Engine 2T-10GE ports are 8q4t/1p7q4t with the platform qos 10g-only global configuration command configured. To configure DSCP-based queue mapping on Supervisor Engine 2T ports, you must enter shutdown interface configuration mode commands for the Supervisor Engine 2T Gigabit Ethernet ports, and then enter the platform qos 10g-only global configuration command, which disables the Gigabit Ethernet ports on the Supervisor Engine 2T.

• In releases where CSCTs82932 is not resolved, do not use the default DSCP-based queue mapping for 8q4t ingress queues unless you configure supporting bandwidth and queue limits.
```

The switch provides congestion avoidance with the combination of multiple queues and the scheduling algorithms associated with each queue.

**Ingress Queue Types**

To see the queue structure of a LAN port, enter the show queueing interface type slot/port | include type command. The command displays one of the following architectures:

- **1q2t** indicates one nonpriority queue with one configurable tail-drop threshold and one nonconfigurable tail-drop threshold.
- **2q4t** indicates two nonpriority queues, each with four configurable tail-drop thresholds.
- **2q8t** indicates two nonpriority queues, each with eight configurable tail-drop thresholds.
- **8q4t** indicates eight nonpriority queues, each with four thresholds, each configurable as either WRED-drop or tail-drop, with support for DSCP-based queueing.

```
Note
In releases where CSCTs82932 is not resolved, do not use the default DSCP-based queue mapping for 8q4t ingress queues unless you configure supporting bandwidth and queue limits.
```

- **8q8t** indicates eight nonpriority queues, each with eight thresholds, each configurable as either WRED-drop or tail-drop.
- **1p1q4t** indicates:
  - One strict-priority queue
  - One nonpriority queue with four configurable tail-drop thresholds.
Information About Policy-Based Queueing

- **1p7q2t** indicates the following:
  - One strict-priority queue
  - Seven nonpriority queues, each with two thresholds, each threshold configurable as either WRED-drop or tail-drop
  - Supports DSCP-based queuing

Egress Queue Types

To see the queue structure of an egress LAN port, enter the `show queueing interface type slot/port | include type` command. The command displays one of the following architectures:

- **1p3q8t** indicates the following:
  - One strict-priority queue
  - Three nonpriority queues, each with eight thresholds, each threshold configurable as either WRED-drop or tail-drop
- **1p7q4t** indicates the following:
  - One strict-priority queue
  - Seven nonpriority queues, each with four thresholds, each threshold configurable as either WRED-drop or tail-drop
  - Supports DSCP-based queuing
- **1p7q8t** indicates the following:
  - One strict-priority queue
  - Seven nonpriority queues, each with eight thresholds, each threshold configurable as either WRED-drop or tail-drop

Module to Queue Type Mappings

- Table 61-1—Supervisor Engine Module QoS Queue Structures
- Table 61-2—10-Gigabit Ethernet Modules
- Table 61-3—Gigabit and 10/100/1000 Ethernet Modules

### Table 61-1 Supervisor Engine Module QoS Queue Structures

<table>
<thead>
<tr>
<th>Supervisor Engines</th>
<th>Ingress Queue and Drop Thresholds</th>
<th>Ingress Queue Scheduler</th>
<th>Egress Queue and Drop Thresholds</th>
<th>Egress Queue Scheduler</th>
<th>Total Buffer Size</th>
<th>Ingress Buffer Size</th>
<th>Egress Buffer Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>VS-S2T-10G-XL, VS-S2T-10G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Chapter 61   Policy-Based Queueing

Information About Policy-Based Queueing

With Gigabit Ethernet ports enabled

2q4t WRR 1p3q4t DWRR or SRR

Does not support DSCP-based queueing.

With Gigabit Ethernet ports disabled

8q4t WRR 1p7q4t DWRR or SRR

- Supports DSCP-based queueing.
- In releases where CSCts82932 is not resolved, do not use the default DSCP-based queue mapping for 8q4t ingress queues unless you configure supporting bandwidth and queue limits.

Table 61-2 10-Gigabit Ethernet Modules

<table>
<thead>
<tr>
<th>Modules</th>
<th>Ingress Queue and Drop Thresholds</th>
<th>Ingress Queue Scheduler</th>
<th>Egress Queue and Drop Thresholds</th>
<th>Egress Queue Scheduler</th>
<th>Total Buffer Size</th>
<th>Ingress Buffer Size</th>
<th>Egress Buffer Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>WS-X6908-10GE (supports DSCP-based queuing)</td>
<td>8q4t DWRR</td>
<td>1p7q4t DWRR SRR</td>
<td></td>
<td>200 MB</td>
<td>108 MB</td>
<td>90 MB</td>
<td></td>
</tr>
<tr>
<td>Performance mode</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note</td>
<td></td>
<td></td>
<td>In releases where CSCts82932 is not resolved, do not use the default DSCP-based queue mapping for 8q4t ingress queues unless you configure supporting bandwidth and queue limits.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WS-X6816-10T-2T, WS-X6716-10T, WS-X6816-10G-2T, WS-X6716-10GE (supports DSCP-based queuing)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oversubscription mode</td>
<td>1p7q2t DWRR</td>
<td>1p7q4t DWRR SRR</td>
<td></td>
<td>91 MB</td>
<td>90 MB per port</td>
<td>1 MB per port group</td>
<td></td>
</tr>
<tr>
<td>WS-X6704-10GE</td>
<td>8q8t WRR</td>
<td>1p7q8t DWRR</td>
<td></td>
<td>16 MB</td>
<td>2 MB</td>
<td>14 MB</td>
<td></td>
</tr>
<tr>
<td>Note</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>In releases where CSCts82932 is not resolved, do not use the default DSCP-based queue mapping for 8q4t ingress queues unless you configure supporting bandwidth and queue limits.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 61-3 Gigabit and 10/100/1000 Ethernet Modules

<table>
<thead>
<tr>
<th>Modules</th>
<th>Ingress Queue and Drop Thresholds</th>
<th>Ingress Queue Scheduler</th>
<th>Egress Queue and Drop Thresholds</th>
<th>Egress Queue Scheduler</th>
<th>Total Buffer Size</th>
<th>Ingress Buffer Size</th>
<th>Egress Buffer Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>WS-X6848-TX-2T, WS-X6748-GE-TX, WS-X6848-SFP-2T, WS-X6748-SFP, WS-X6824-SFP-2T, WS-X6724-SFP</td>
<td>2q8t WRR</td>
<td>1p3q8t DWRR</td>
<td></td>
<td>1.3 MB</td>
<td>166 KB</td>
<td>1.2 MB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>In releases where CSCts82932 is not resolved, do not use the default DSCP-based queue mapping for 8q4t ingress queues unless you configure supporting bandwidth and queue limits.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Queueing Policies

Queueing policies use class-maps with `match` commands (see Table 61-4) and policy maps with scheduling and congestion management commands (see Table 61-5).

**Table 61-4 Queueing Policy Class Map match Commands and Match Criteria**

<table>
<thead>
<tr>
<th>match Commands</th>
<th>Match Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>match cos cos_list</code></td>
<td>CoS values.</td>
</tr>
<tr>
<td><code>match dscp dscp_list</code></td>
<td>DSCP values.</td>
</tr>
<tr>
<td><code>match precedence precedence_list</code></td>
<td>Precedence values.</td>
</tr>
</tbody>
</table>

**Note**

- The presence of a `match dscp` command or a `match precedence` command in a class map that is used in a queueing policy attached to a port configures DSCP-based queueing in the direction of the policy (ingress or egress) on that port.
- CoS-based queueing is always used for non-IP traffic, IP multicast traffic, and IP unknown unicast flood traffic.
- Class maps that are used in queueing policies can contain both `match dscp` commands and `match cos` commands.
### Table 61-5  Queueing Policy-Map Class Commands

<table>
<thead>
<tr>
<th>Queueing Commands</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bandwidth [remaining] percent percentage</td>
<td>Allocates bandwidth between nonpriority queues. The remaining keyword is required on ports that have a priority queue.</td>
</tr>
<tr>
<td>shape average percent percentage</td>
<td>Enables SRR, which allocates limited bandwidth between nonpriority egress queues (see “SRR” in the “Module to Queue Type Mappings” section on page 61-6).</td>
</tr>
<tr>
<td>priority</td>
<td>Applies a policy-map class to a priority queue.</td>
</tr>
<tr>
<td>queue-limit percent percent_of_total_buffer_size</td>
<td>Sets the queue size.</td>
</tr>
<tr>
<td>queue-limit multiple-type-based</td>
<td>Enables application of CoS, precedence, and DSCP values to a tail-drop threshold.</td>
</tr>
<tr>
<td>queue-limit cos cos_value percent percent_of_qsize</td>
<td>Applies one CoS value to a tail-drop threshold and configures the threshold percentage.</td>
</tr>
<tr>
<td>queue-limit dscp dscp_value percent percent_of_qsize</td>
<td>Applies one DSCP value to a tail-drop threshold and configures the threshold percentage.</td>
</tr>
<tr>
<td>queue-limit precedence precedence_value percent percent_of_qsize</td>
<td>Applies one precedence value to a tail-drop threshold and configures the threshold percentage.</td>
</tr>
<tr>
<td>queue-limit cos values cos_list percent percent_of_qsize</td>
<td>Applies multiple CoS values to a tail-drop threshold and configures the threshold percentage.</td>
</tr>
<tr>
<td>queue-limit dscp values dscp_list percent percent_of_qsize</td>
<td>Applies multiple DSCP values to a tail-drop threshold and configures the threshold percentage.</td>
</tr>
<tr>
<td>queue-limit precedence values precedence_list percent percent_of_qsize</td>
<td>Applies multiple precedence values to a tail-drop threshold and configures the threshold percentage.</td>
</tr>
</tbody>
</table>
### Table 61-5  Queueing Policy-Map Class Commands (continued)

<table>
<thead>
<tr>
<th>Queueing Commands</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>random-detect cos-based [aggregate]</td>
<td>Enables application of CoS values to a WRED-drop threshold. The <code>aggregate</code> keyword enables use of the <code>values</code> keyword.</td>
</tr>
<tr>
<td>random-detect dscp-based [aggregate]</td>
<td>Enables application of DSCP values to a WRED-drop threshold. The <code>aggregate</code> keyword enables use of the <code>values</code> keyword.</td>
</tr>
<tr>
<td>random-detect precedence-based [aggregate]</td>
<td>Enables application of precedence values to a WRED-drop threshold. The <code>aggregate</code> keyword enables use of the <code>values</code> keyword.</td>
</tr>
<tr>
<td>random-detect multiple-type-based [aggregate]</td>
<td>Enables application of CoS, precedence, and DSCP values to a WRED-drop threshold. The <code>aggregate</code> keyword enables use of the <code>values</code> keyword.</td>
</tr>
<tr>
<td>random-detect cos <code>cos_value</code> percent <code>min_percent</code> <code>max_percent</code></td>
<td>Applies one CoS value to a WRED-drop threshold and configures the threshold percentages.</td>
</tr>
<tr>
<td>random-detect dscp <code>dscp_value</code> percent <code>min_percent</code> <code>max_percent</code></td>
<td>Applies one DSCP value to a WRED-drop threshold and configures the threshold percentages.</td>
</tr>
<tr>
<td>random-detect precedence <code>precedence_value</code> percent <code>min_percent</code> <code>max_percent</code></td>
<td>Applies one DSCP value to a WRED-drop threshold and configures the threshold percentages.</td>
</tr>
<tr>
<td>random-detect cos values <code>cos_list</code> percent <code>min_percent</code> <code>max_percent</code></td>
<td>Applies multiple CoS values to a WRED-drop threshold and configures the threshold percentages; requires the <code>aggregate</code> keyword.</td>
</tr>
<tr>
<td>random-detect dscp values <code>dscp_list</code> percent <code>min_percent</code> <code>max_percent</code></td>
<td>Applies multiple DSCP values to a WRED-drop threshold and configures the threshold percentages; requires the <code>aggregate</code> keyword.</td>
</tr>
<tr>
<td>random-detect precedence values <code>precedence_list</code> percent <code>min_percent</code> <code>max_percent</code></td>
<td>Applies multiple precedence values to a WRED-drop threshold and configures the threshold percentages.</td>
</tr>
</tbody>
</table>

### How to Configure Policy-Based Queueing

- Configuring a Queueing Policy Class Map, page 61-11
- Verifying a Queueing Policy Class Map, page 61-11
- Configuring Queueing Policy Maps, page 61-11
- Verifying a Queueing Policy Map, page 61-15
- Attaching a Queueing Policy Map to an Interface, page 61-15

**Note**

See the “Configuration Examples for Policy-Based Queueing” section on page 61-16 for detailed information about which queueing commands are supported by each queue type.
Configuring a Queueing Policy Class Map

To configure a queueing policy class map, perform this task:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>Router(config)# class-map type lan-queuing</strong>&lt;br&gt;<strong>match-any class_name</strong>&lt;br&gt;Creates a class map. Create a class map for the thresholds on each type of queue that you are configuring. Give the class map a name that allows you to easily associate it with the queue type and threshold for which you configure it.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>Router(config-cmap)# match cos cos_value1</strong>&lt;br&gt;<strong>[cos_value2 ... [cos_valueN]]</strong>&lt;br&gt;(Optional) Configures the queueing policy class map to filter based on CoS values. You can enter multiple commands.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>Router(config-cmap)# match dscp dscp_value1</strong>&lt;br&gt;<strong>[dscp_value2 ... [dscp_valueN]]</strong>&lt;br&gt;(Optional) Configures the queueing policy class map to filter based on DSCP values and enables DSCP-based queueing on the port in the direction that the queueing policy is attached. You can enter multiple commands.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>Router(config-cmap)# match precedence precedence_value1</strong>&lt;br&gt;<strong>[precedence_value2 ... [precedence_valueN]]</strong>&lt;br&gt;(Optional) Configures the queueing policy class map to filter based on precedence values and enables DSCP-based queueing on the port in the direction that the queueing policy is attached. You can enter multiple commands.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>Router(config-cmap)# end</strong>&lt;br&gt;Exits configuration mode.</td>
</tr>
</tbody>
</table>

This example shows how to create a class map named `cos5` and how to configure filtering to match traffic with CoS 5:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# class-map cos5
Router(config-cmap)# match cos 5
Router(config-cmap)# end
```

Verifying a Queueing Policy Class Map

To verify the queueing policy class map, perform this task:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Router# show class-map class_name</strong></td>
<td>Verifies the configuration.</td>
</tr>
</tbody>
</table>

Configuring Queueing Policy Maps

- Creating a Queueing Policy, page 61-12
- Configuring a Priority Queue, page 61-12
- Configuring Nonpriority Queues, page 61-13
- Configuring Thresholds, page 61-13
### Creating a Queueing Policy

To create a queueing policy, perform this task:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Router(config)# policy-map type lan-queuing &lt;br&gt;\texttt{policy_name} &lt;br&gt;Creates a policy map.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Router(config-pmap)# end &lt;br&gt;(Optional) Exits policy map class configuration mode.</td>
</tr>
</tbody>
</table>

### Configuring a Priority Queue

1p1q4t, 1p1q8t, 1p7q2t, 1p3q8t, 1p7q8t, and 1p7q4t ports have priority queues. To configure the priority queue, perform this task:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Router(config-pmap)# class \texttt{class_map_name} &lt;br&gt;Creates a policy map class.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Router(config-pmap-c)# priority &lt;br&gt;Applies the class map to the priority queue. &lt;br&gt;Note The priority queue is not supported if SRR is enabled.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Router(config-pmap-c)# queue-buffers ratio \texttt{weight} &lt;br&gt;(Optional) Sets queue buffer size.</td>
</tr>
<tr>
<td>Step 4</td>
<td>Router(config-pmap-c)# end &lt;br&gt;(Optional) Exits policy map class configuration mode.</td>
</tr>
</tbody>
</table>
Configuring Nonpriority Queues

The first `class class_map_name` command that you enter that is not followed by the `priority` keyword configures the highest numbered nonpriority queue. Subsequent `class class_map_name` commands configure the remaining nonpriority queues in reverse numerical order (highest-numbered to queue #2). The `class class-default` command configures queue #1. To configure a nonpriority queue, perform this task:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Router(config-pmap)# class {class_map_name</td>
</tr>
<tr>
<td></td>
<td>Creates a policy map class.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>• Enter a class map name to configure nonpriority queues numbered greater than one.</td>
</tr>
<tr>
<td></td>
<td>• Enter the <code>class-default</code> keyword to configure queue #1.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Router(config-pmap-c)# bandwidth [remaining] percent percentage</td>
</tr>
<tr>
<td></td>
<td>Allocates WRR or DWRR bandwidth.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>• Not required on ports that have one queue.</td>
</tr>
<tr>
<td></td>
<td>• The <code>remaining</code> keyword is required on ports that have a priority queue.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Router(config-pmap-c)# shape average percent percentage</td>
</tr>
<tr>
<td></td>
<td>(Egress queues only) Enables SRR, which allocates limited bandwidth between nonpriority egress queues.</td>
</tr>
<tr>
<td></td>
<td>• See “SRR” in the “Module to Queue Type Mappings” section on page 61-6).</td>
</tr>
<tr>
<td></td>
<td>• Configuring SRR disables the priority queue.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Router(config-pmap-c)# queue-limit percent percentage</td>
</tr>
<tr>
<td></td>
<td>(Optional) Sets queue buffer size.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td>• Not required on ports that have one queue.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Router(config-pmap-c)# end</td>
</tr>
<tr>
<td></td>
<td>(Optional) Exits policy map class configuration mode.</td>
</tr>
</tbody>
</table>

Configuring Thresholds

- Threshold Configuration Guidelines and Restrictions, page 61-13
- Configuring a Threshold as Tail-Drop with CoS-Based Queuing, page 61-14
- Configuring a Threshold as WRED-Drop with CoS-Based Queuing, page 61-14
- Configuring a Threshold as Tail-Drop with DSCP-Based Queuing, page 61-14
- Configuring a Threshold as WRED-Drop with DSCP-Based Queuing, page 61-15

Threshold Configuration Guidelines and Restrictions

- To configure tail drop thresholds, enter `queue-limit` commands.
- For ports that support configuration as either tail-drop or WRED drop:
  - Enter `queue-limit` commands to configure a threshold as tail drop.
  - Enter `random-detect` commands to configure a threshold as WRED drop.
The first `queue-limit cos`, `queue-limit dscp`, `random-detect cos`, or `random-detect dscp` command that you enter configures threshold #1.

Subsequent commands with different threshold percentage values configure the remaining thresholds in numerical order (from threshold #2 to the highest-numbered threshold).

Subsequent commands with an already configured threshold percentage value apply additional QoS values to the threshold indicated by the percentage value.

Ports configured for DSCP-based queueing use CoS-based queueing for non-IP traffic, IP multicast traffic, and IP unknown unicast flood traffic. In queueing policies that configure DSCP-based queueing, configure CoS-based queueing to provide specific QoS for non-IP traffic, IP multicast traffic, and IP unknown unicast flood traffic.

### Configuring a Threshold as Tail-Drop with CoS-Based Queueing

To configure a threshold as tail-drop with CoS-based queueing, perform this task:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>`Router(config-pmap-c)# queue-limit cos (one_value</td>
</tr>
<tr>
<td></td>
<td>Applies CoS to a tail-drop threshold and configures the threshold percentage.</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>Router(config-pmap-c)# end</code></td>
</tr>
<tr>
<td></td>
<td>(Optional) Exits policy map class configuration mode.</td>
</tr>
</tbody>
</table>

### Configuring a Threshold as WRED-Drop with CoS-Based Queueing

To configure a threshold as WRED-drop with CoS-based queueing, perform this task:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>Router(config-pmap-c)# random-detect cos-based [aggregate]</code></td>
</tr>
<tr>
<td></td>
<td>Enables application of CoS values to a WRED-drop threshold. Enter the <code>aggregate</code> keyword to configure multiple CoS values on the threshold with the <code>values</code> keyword.</td>
</tr>
<tr>
<td>Step 2</td>
<td>`Router(config-pmap-c)# random-detect cos (one_value</td>
</tr>
<tr>
<td></td>
<td>Applies CoS to a WRED-drop threshold and configures the threshold percentage.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>Router(config-pmap-c)# end</code></td>
</tr>
<tr>
<td></td>
<td>(Optional) Exits policy map class configuration mode.</td>
</tr>
</tbody>
</table>

### Configuring a Threshold as Tail-Drop with DSCP-Based Queueing

To configure a threshold as tail-drop with DSCP-based queueing, perform this task:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>Router(config-pmap-c)# queue-limit multiple-type-based</code></td>
</tr>
<tr>
<td></td>
<td>(Optional) Enables application of CoS, DSCP, or precedence values to a tail-drop threshold.</td>
</tr>
<tr>
<td>Step 2</td>
<td>`Router(config-pmap-c)# queue-limit (cos</td>
</tr>
<tr>
<td></td>
<td>Applies QoS values to a tail-drop threshold and configures the threshold percentage.</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>Router(config-pmap-c)# end</code></td>
</tr>
<tr>
<td></td>
<td>(Optional) Exits policy map class configuration mode.</td>
</tr>
</tbody>
</table>
How to Configure Policy-Based Queueing

Configuring a Threshold as WRED-Drop with DSCP-Based Queueing

To configure a threshold as WRED-drop with DSCP-based queueing, perform this task:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Router(config-pmap-c)# random-detect (dscp-based</td>
</tr>
<tr>
<td></td>
<td>Step 2</td>
</tr>
<tr>
<td></td>
<td>Step 3</td>
</tr>
</tbody>
</table>

Verifying a Queueing Policy Map

Use the `show policy-map policy_name` command to verify the configuration.

Attaching a Queueing Policy Map to an Interface

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Router(config)# interface type slot/port</td>
</tr>
<tr>
<td>Step 2</td>
<td>Router(config-if)# service-policy type lan-queuing [input</td>
</tr>
<tr>
<td>Step 3</td>
<td>Router(config-if)# end</td>
</tr>
</tbody>
</table>

Use the `show policy-map interface` command to verify the configuration.
Configuration Examples for Policy-Based Queueing

- Queueing Policy Sample Configuration, page 61-16
- Queueing Policy Commands Supported by Each Queue Type, page 61-17
- Queueing Policy Commands Sample Configurations for Each Queue Type, page 61-28

Queueing Policy Sample Configuration

Without comments:

```plaintext
policy-map type lan-queuing p1
  class cos5
    priority
    class cos123
      bandwidth remaining percent 25
      queue-limit cos 2 percent 20
      queue-limit cos 3 percent 30
    class class-default
      queue-limit cos 6 percent 60
```

With comments:

```plaintext
policy-map type lan-queuing p1 ! For 1p3q8t
  class cos5 ! Configured to filter CoS 5
    !The filtering configured in the class map selects the values that go to the queue
    !  priority ! Applies the class map to the priority queue (#4)
    !
    !First non-priority class applies to highest-numbered non-priority queue (#3)
      class cos123 ! Configured to filter CoS 1, 2, and 3
        !The filtering configured in the class map selects the values that go to the queue
        !
        !'remaining' keyword required on ports that have a priority queue
        bandwidth remaining percent 25
        !
        !First queue-limit command assigns CoS 2 to threshold #1 and configures it at 20%
        queue-limit cos 2 percent 20
        !Any other queue-limit command with the same percentage
        !applies additional configuration to this threshold
        !
        !Next queue-limit command with different percentage value configures the next threshold
        !Assigns CoS 3 to threshold #2 and configures it at 30%
        queue-limit cos 3 percent 30
        !Any other queue-limit command with the same percentage
        !applies additional configuration to this threshold
        !
        !Thresholds 3-8 are unconfigured
        !All unconfigured thresholds are at 100%
        !No explicit configuration provided for CoS 1: defaults to last threshold
        !
        !End of queue 3 configuration
        !
        !Queue 2 is unconfigured
        !
        !class class-default ! applies to queue #1
        !'class-default' gets all remaining CoS values:
```

Cisco IOS Software Configuration Guide, Release 12.2SY
Threshold 1 is explicitly configured:
  queue-limit cos 6 percent 50

Remaining thresholds (2-8) are not configured by the queueing policy
and cannot be configured by anything else
No explicit configuration provided for CoS 0, 4, and 7:
CoS values not explicitly configured default to the last threshold

Queueing Policy Commands Supported by Each Queue Type

- 1q2t, 1q8t Ingress Queue Supported Commands, page 61-18
- 2q8t Ingress Queue Supported Commands, page 61-19
- 8q4t Ingress Queue Supported Commands, page 61-20
- 8q8t Ingress Queue Supported Commands, page 61-21
- 1p1q4t Ingress Queue Supported Commands, page 61-22
- 1p1q8t Ingress Queue Supported Commands, page 61-23
- 1p7q2t Ingress Queue Supported Commands, page 61-24
- 1p3q8t Egress Queue Supported Commands, page 61-25
- 1p7q8t Egress Queue Supported Commands, page 61-26
- 1p7q4t Ingress or Egress Queue Supported Commands, page 61-27
1q2t, 1q8t Ingress Queue Supported Commands

- Supports CoS-based queueing with tail-drop thresholds.
- Unsupported commands are included in this section as comments.

policy-map type lan-queuing policy_map_name ! For 1q2t, 1q8t

- Nonpriority queue 1 policy commands:
  class class-default ! Receives all CoS values.
  ! bandwidth percent percentage ! WRR or DWRR bandwidth allocation.
  ! queue-limit percent percentage ! Queue buffer size.

- Nonpriority queue threshold configuration (repeat to configure sequential thresholds):
  ! queue-limit multiple-type-based
  ! Enables application of CoS, precedence, and DSCP values
  ! to a tail-drop threshold.

  queue-limit cos {one_value | values value_list} percent percentage

  ! queue-limit dscp {one_value | values value_list} percent percentage
  ! Applies one DSCP value to a tail-drop threshold
  ! and configures the threshold percentage

  ! random-detect cos-based [aggregate]
  ! Enables application of CoS values to a WRED-drop threshold.
  ! The 'aggregate' keyword allows use of the 'values' keyword

  ! random-detect dscp-based [aggregate]
  ! Enables application of DSCP values to a WRED-drop threshold.
  ! The 'aggregate' keyword allows use of the 'values' keyword

  ! random-detect precedence-based [aggregate]
  ! Enables application of precedence values to a WRED-drop threshold.
  ! The 'aggregate' keyword allows use of the 'values' keyword

  ! random-detect multiple-type-based [aggregate]
  ! Enables application of CoS, precedence, and DSCP values
  ! to a WRED-drop threshold.
  ! The 'aggregate' keyword allows use of the 'values' keyword

  ! random-detect cos {one_value | values value_list} percent min_% max_%
  ! Applies CoS to a WRED-drop threshold and configures the threshold percentages.

  ! random-detect dscp {one_value | values value_list} percent min_% max_%
  ! Applies DSCP to a WRED-drop threshold and configures the threshold percentages.

  ! random-detect (precedence one_value | values precedence value_list) percent min_% max_%
  ! Applies DSCP to a WRED-drop threshold and configures the threshold percentages.
2q8t Ingress Queue Supported Commands

- Supports CoS-based queueing with tail-drop thresholds.
- Unsupported commands are included in this section as comments.

```
policy-map type lan-queuing policy_map_name

- Class commands for nonpriority queues numbered higher than 1—Configures queues in reverse numerical order; repeat to configure the next queue:
  class class_map_name
    Receives CoS values filtered by class_map_name.

- Class command for queue #1:
  class class-default
    Receives all remaining CoS values.

- Nonpriority queue configuration commands:
  bandwidth percent percentage
  queue-limit percent percentage

- Nonpriority queue threshold configuration—Repeat to configure sequential thresholds in each queue:
  queue-limit cos {one_value | values value_list} percent percentage
  queue-limit dscp {one_value | values value_list} percent percentage
  random-detect cos-based [aggregate]
  random-detect dscp-based [aggregate]
  random-detect precedence-based [aggregate]
  random-detect multiple-type-based [aggregate]
```

Cisco IOS Software Configuration Guide, Release 12.2SY

61-19


8q4t Ingress Queue Supported Commands

- Supports CoS-based, DSCP-based, and precedence-based queueing with tail-drop and WRED-drop thresholds.
- Unsupported commands are included in this section as comments.
- In releases where CSCts82932 is not resolved, do not use the default DSCP-based queue mapping for 8q4t ingress queues unless you configure supporting bandwidth and queue limits.

```
policy-map type lan-queuing policy_map_name ! For 8q4t

- Class commands for nonpriority queues numbered higher than 1—Configures queues in reverse numerical order; repeat to configure the next queue:
  class class_map_name
  ! Receives QoS values (CoS, DSCP, precedence) values filtered by class_map_name.

- Class command for queue #1:
  class class-default ! Receives all remaining QoS values (CoS, DSCP, precedence).

- Nonpriority queue configuration commands:
  bandwidth percent percentage ! WRR or DWRR bandwidth allocation.
  queue-limit percent percentage ! Queue buffer size.

- Nonpriority queue threshold configuration—Repeat to configure sequential thresholds in each queue:
  queue-limit multiple-type-based
  ! Enables application of CoS, precedence, and DSCP values to a tail-drop threshold.
  queue-limit cos {one_value | values value_list} percent percentage
  ! Applies CoS to a tail-drop threshold and configures the threshold percentage
  queue-limit dscp {one_value | values value_list} percent percentage
  ! Applies one DSCP value to a tail-drop threshold
  ! and configures the threshold percentage
  random-detect cos-based [aggregate]
  ! Enables application of CoS values to a WRED-drop threshold.
  ! The 'aggregate' keyword allows use of the 'values' keyword
  random-detect dscp-based [aggregate]
  ! Enables application of DSCP values to a WRED-drop threshold.
  ! The 'aggregate' keyword allows use of the 'values' keyword
  random-detect precedence-based [aggregate]
  ! Enables application of precedence values to a WRED-drop threshold.
  ! The 'aggregate' keyword allows use of the 'values' keyword
  random-detect multiple-type-based [aggregate]
  ! Enables application of CoS, precedence, and DSCP values to a WRED-drop threshold.
  ! The 'aggregate' keyword allows use of the 'values' keyword
  random-detect cos {one_value | values value_list} percent min_% max_%
  ! Applies CoS to a WRED-drop threshold and configures the threshold percentages.
  random-detect dscp {one_value | values value_list} percent min_% max_%
  ! Applies DSCP to a WRED-drop threshold and configures the threshold percentages.
```
random-detect {precedence one_value | values precedence value_list} percent min_% max_%

Applies DSCP to a WRED-drop threshold and configures the threshold percentages.

### 8q8t Ingress Queue Supported Commands

- Supports CoS-based queuing with tail-drop or WRED-drop thresholds.
- Unsupported commands are included in this section as comments.

---

policy-map type lan-queuing policy_map_name ! For 1p1q8t

- **Priority queue:**
  ```
  class class_map_name ! Receives CoS values filtered by class_map_name.
  priority ! Applies the class map to the priority queue
  ```

- **Nonpriority queue policy commands:**
  ```
  class class-default ! Receives all remaining CoS values.
  ! bandwidth remaining percent percentage ! WRR or DWRR bandwidth allocation.
  ! queue-limit percent percentage ! Queue buffer size.
  ```

- **Nonpriority queue threshold configuration—Repeat to configure sequential thresholds:**
  ```
  ! queue-limit multiple-type-based
  ! Enables application of CoS, precedence, and DSCP values to a tail-drop threshold.
  !
  ! queue-limit cos {one_value | values value_list} percent percentage
  ! Applies CoS to a tail-drop threshold and configures the threshold percentage
  !
  ! queue-limit dscp {one_value | values value_list} percent percentage
  ! Applies one DSCP value to a tail-drop threshold
  ! and configures the threshold percentage
  !
  ! random-detect cos-based [aggregate]
  ! Enables application of CoS values to a WRED-drop threshold.
  ! The 'aggregate' keyword allows use of the 'values' keyword
  !
  ! random-detect dscp-based [aggregate]
  ! Enables application of DSCP values to a WRED-drop threshold.
  ! The 'aggregate' keyword allows use of the 'values' keyword
  !
  ! random-detect precedence-based [aggregate]
  ! Enables application of precedence values to a WRED-drop threshold.
  ! The 'aggregate' keyword allows use of the 'values' keyword
  !
  ! random-detect multiple-type-based [aggregate]
  ! Enables application of CoS, precedence, and DSCP values
  ! to a WRED-drop threshold.
  ! The 'aggregate' keyword allows use of the 'values' keyword
  !
  ! random-detect cos {one_value | values value_list} percent min_% max_%
  ! Applies CoS to a WRED-drop threshold and configures the threshold percentages.
  !
  ! random-detect dscp {one_value | values value_list} percent min_% max_%
  ! Applies DSCP to a WRED-drop threshold and configures the threshold percentages.
  !
  ! random-detect {precedence one_value | values precedence value_list} percent min_% max_%
  ! Applies DSCP to a WRED-drop threshold and configures the threshold percentages.
1p1q4t Ingress Queue Supported Commands

- Supports CoS-based queuing with tail-drop thresholds.
- Unsupported commands are included in this section as comments.

```plaintext
policy-map type lan-queuing policy_map_name ! For 1p1q4t

- Priority queue:
  class class_map_name ! Receives CoS values filtered by class_map_name.
  priority ! Applies the class map to the priority queue

- Nonpriority queue policy commands:
  class class-default ! Receives all remaining CoS values.
    ! bandwidth remaining percent percentage ! WRR or DWRR bandwidth allocation.
    ! queue-limit percent percentage ! Queue buffer size.

- Nonpriority queue threshold configuration—Repeat to configure sequential thresholds:
  ! queue-limit multiple-type-based
  ! Enables application of CoS, precedence, and DSCP values
  ! to a tail-drop threshold.
  ! queue-limit cos {one_value | values value_list} percent percentage
  ! Applies CoS to a tail-drop threshold and configures the threshold percentage
  ! queue-limit dscp {one_value | values value_list} percent percentage
  ! Applies one DSCP value to a tail-drop threshold
  ! and configures the threshold percentage
  ! random-detect cos-based [aggregate]
  ! Enables application of CoS values to a WRED-drop threshold.
  ! The 'aggregate' keyword allows use of the 'values' keyword
  ! random-detect dscp-based [aggregate]
  ! Enables application of DSCP values to a WRED-drop threshold.
  ! The 'aggregate' keyword allows use of the 'values' keyword
  ! random-detect precedence-based [aggregate]
  ! Enables application of precedence values to a WRED-drop threshold.
  ! The 'aggregate' keyword allows use of the 'values' keyword
  ! random-detect multiple-type-based [aggregate]
  ! Enables application of CoS, precedence, and DSCP values
  ! to a WRED-drop threshold.
  ! The 'aggregate' keyword allows use of the 'values' keyword
  ! random-detect cos {one_value | values value_list} percent min_% max_%
  ! Applies CoS to a WRED-drop threshold and configures the threshold percentages.
  ! random-detect dscp {one_value | values value_list} percent min_% max_%
  ! Applies DSCP to a WRED-drop threshold and configures the threshold percentages.
  ! random-detect {precedence one_value | values precedence value_list} percent min_% max_%
  ! Applies DSCP to a WRED-drop threshold and configures the threshold percentages.
```

Note: Supports CoS-based queueing with tail-drop thresholds. Unsupported commands are included in this section as comments.
## 1p1q8t Ingress Queue Supported Commands

- Supports CoS-based queueing with tail-drop or WRED-drop thresholds.
- Unsupported commands are included in this section as comments.

```text
policy-map type lan-queuing policy_map_name ! For 1p1q8t

- Priority queue:
  ```
  class class_map_name ! Receives CoS values filtered by class_map_name.
  priority ! Applies the class map to the priority queue
  ```

- Nonpriority queue policy commands:
  ```
  class class-default ! Receives all remaining CoS values.
  ! bandwidth remaining percent percentage ! WRR or DWRR bandwidth allocation.
  ! queue-limit percent percentage ! Queue buffer size.
  ```

- Nonpriority queue threshold configuration—Repeat to configure sequential thresholds:
  ```
  ! queue-limit multiple-type-based
  ! Enables application of CoS, precedence, and DSCP values to a tail-drop threshold.
  ! queue-limit cos {one_value | values value_list} percent percentage
  ! Applies CoS to a tail-drop threshold and configures the threshold percentage
  ! queue-limit dscp {one_value | values value_list} percent percentage
  ! Applies one DSCP value to a tail-drop threshold
  ! and configures the threshold percentage
  !
  random-detect cos-based [aggregate]
  ! Enables application of CoS values to a WRED-drop threshold.
  ! The 'aggregate' keyword allows use of the 'values' keyword
  !
  random-detect dscp-based [aggregate]
  ! Enables application of DSCP values to a WRED-drop threshold.
  ! The 'aggregate' keyword allows use of the 'values' keyword
  !
  random-detect precedence-based [aggregate]
  ! Enables application of precedence values to a WRED-drop threshold.
  ! The 'aggregate' keyword allows use of the 'values' keyword
  !
  random-detect multiple-type-based [aggregate]
  ! Enables application of CoS, precedence, and DSCP values
  ! to a WRED-drop threshold.
  !
  random-detect cos {one_value | values value_list} percent min_% max_%
  ! Applies CoS to a WRED-drop threshold and configures the threshold percentages.
  !
  random-detect dscp {one_value | values value_list} percent min_% max_%
  ! Applies DSCP to a WRED-drop threshold and configures the threshold percentages.
  !
  random-detect {precedence one_value | values precedence value_list} percent min_% max_%
  ! Applies DSCP to a WRED-drop threshold and configures the threshold percentages.
```
1p7q2t Ingress Queue Supported Commands

- Supports CoS-based, DSCP-based, and precedence-based queueing with tail-drop and WRED-drop thresholds.
- Unsupported commands are included in this section as comments.

```
policy-map type lan-queuing policy_map_name ! For 1p7q2t

• Priority queue:
  class class_map_name
  ! Receives QoS values (CoS, DSCP, precedence) values filtered by class_map_name.
  priority ! Applies the class map to the priority queue

• Class commands for nonpriority queues numbered higher than 1—Configures queues in reverse numerical order; repeat to configure the next queue:
  class class_map_name
  ! Receives QoS values (CoS, DSCP, precedence) values filtered by class_map_name.

• Class command for queue #1:
  class class-default ! Receives all remaining QoS values (CoS, DSCP, precedence).

• Nonpriority queue configuration commands:
  bandwidth percent percentage ! WRR or DWRR bandwidth allocation.
  !
  queue-limit percent percentage ! Queue buffer size.

• Nonpriority queue threshold configuration—Repeat to configure sequential thresholds in each queue:
  queue-limit multiple-type-based
  ! Enables application of CoS, precedence, and DSCP values to a tail-drop threshold.
  !
  queue-limit cos {one_value | values value_list} percent percentage
  ! Applies CoS to a tail-drop threshold and configures the threshold percentage
  !
  queue-limit dscp {one_value | values value_list} percent percentage
  ! Applies one DSCP value to a tail-drop threshold
  ! and configures the threshold percentage
  !
  random-detect cos-based [aggregate]
  ! Enables application of CoS values to a WRED-drop threshold.
  ! The 'aggregate' keyword allows use of the 'values' keyword
  !
  random-detect dscp-based [aggregate]
  ! Enables application of DSCP values to a WRED-drop threshold.
  ! The 'aggregate' keyword allows use of the 'values' keyword
  !
  random-detect precedence-based [aggregate]
  ! Enables application of precedence values to a WRED-drop threshold.
  ! The 'aggregate' keyword allows use of the 'values' keyword
  !
  random-detect multiple-type-based [aggregate]
  ! Enables application of CoS, precedence, and DSCP values to a WRED-drop threshold.
  ! The 'aggregate' keyword allows use of the 'values' keyword
  !
  random-detect cos {one_value | values value_list} percent min_% max_%
```
! Applies CoS to a WRED-drop threshold and configures the threshold percentages.
! random-detect dscp (one_value | values value_list) percent min_% max_%
! Applies DSCP to a WRED-drop threshold and configures the threshold percentages.
! random-detect (precedence one_value | values precedence value_list) percent min_% max_%
! Applies DSCP to a WRED-drop threshold and configures the threshold percentages.

1p3q8t Egress Queue Supported Commands

- Supports CoS-based queueing with tail-drop and WRED-drop thresholds.
- Unsupported commands are included in this section as comments.

policy-map type lan-queuing policy_map_name ! For 1p3q8t

- Priority queue:
  class class_map_name ! Receives CoS values filtered by class_map_name.
  priority ! Applies the class map to the priority queue

- Class commands for nonpriority queues numbered higher than 1—Configures queues in reverse numerical order; repeat to configure the next queue:
  class class_map_name ! Receives CoS values filtered by class_map_name.

- Class command for queue #1:
  class class-default ! Receives all remaining CoS values.

- Nonpriority queue configuration commands:
  bandwidth percent percentage ! WRR or DWRR bandwidth allocation.
  ! shape average percent percentage ! SRR bandwidth allocation.
  ! queue-limit percent percentage ! Queue buffer size.

- Nonpriority queue threshold configuration—Repeat to configure sequential thresholds:
  ! queue-limit multiple-type-based
  ! Enables application of CoS, precedence, and DSCP values to a tail-drop threshold.
  ! queue-limit cos (one_value | values value_list) percent percentage
  ! Applies CoS to a tail-drop threshold and configures the threshold percentage
  ! queue-limit dscp (one_value | values value_list) percent percentage
  ! Applies one DSCP value to a tail-drop threshold
  ! and configures the threshold percentage
  ! random-detect cos-based [aggregate]
  ! Enables application of CoS values to a WRED-drop threshold.
  ! The ‘aggregate’ keyword allows use of the ‘values’ keyword
  ! random-detect dscp-based [aggregate]
  ! Enables application of DSCP values to a WRED-drop threshold.
  ! The ‘aggregate’ keyword allows use of the ‘values’ keyword
  ! random-detect precedence-based [aggregate]
  ! Enables application of precedence values to a WRED-drop threshold.
Configuration Examples for Policy-Based Queueing

Chapter 61      Policy-Based Queueing

Policy-Based Queueing Examples

! The 'aggregate' keyword allows use of the 'values' keyword
! random-detect multiple-type-based [aggregate]
! Enables application of CoS, precedence, and DSCP values
! to a WRED-drop threshold.
! The 'aggregate' keyword allows use of the 'values' keyword

random-detect cos (one_value | values value_list) percent min_% max_%
! Applies CoS to a WRED-drop threshold and configures the threshold percentages.

random-detect dscp (one_value | values value_list) percent min_% max_%
! Applies DSCP to a WRED-drop threshold and configures the threshold percentages.

random-detect precedence one_value | values precedence value_list)
percent min_% max_%
! Applies DSCP to a WRED-drop threshold and configures the threshold percentages.

1p7q8t Egress Queue Supported Commands

Note

- Supports CoS-based queuing with tail-drop and WRED-drop thresholds.
- Unsupported commands are included in this section as comments.

policy-map type lan-queuing policy_map_name ! For 1p7q8t

- Priority queue:
  class class_map_name ! Receives CoS values filtered by class_map_name.
  priority ! Applies the class map to the priority queue

- Class commands for nonpriority queues numbered higher than 1—Configures queues in reverse numerical order; repeat to configure the next queue:
  class class_map_name ! Receives CoS values filtered by class_map_name.

- Class command for queue #1:
  class class-default ! Receives all remaining CoS values.

- Nonpriority queue configuration commands:
  bandwidth percent percentage ! WRR or DWRR bandwidth allocation.
  !
  ! shape average percent percentage ! SRR bandwidth allocation.
  !
  queue-limit percent percentage ! Queue buffer size.

- Nonpriority queue threshold configuration—Repeat to configure sequential thresholds:
  ! queue-limit multiple-type-based
  ! Enables application of CoS, precedence, and DSCP values to a tail-drop threshold.
  !
  queue-limit cos (one_value | values value_list) percent percentage
  ! Applies CoS to a tail-drop threshold and configures the threshold percentage
  !
  queue-limit dscp (one_value | values value_list) percent percentage
  ! Applies one DSCP value to a tail-drop threshold
  ! and configures the threshold percentage
  !
  random-detect cos-based [aggregate]
  ! Enables application of CoS values to a WRED-drop threshold.
  ! The 'aggregate' keyword allows use of the 'values' keyword
! random-detect dscp-based [aggregate]
! Enables application of DSCP values to a WRED-drop threshold.
! The 'aggregate' keyword allows use of the 'values' keyword
!
! random-detect precedence-based [aggregate]
! Enables application of precedence values to a WRED-drop threshold.
! The 'aggregate' keyword allows use of the 'values' keyword
!
! random-detect multiple-type-based [aggregate]
! Enables application of CoS, precedence, and DSCP values
! to a WRED-drop threshold.
! The 'aggregate' keyword allows use of the 'values' keyword
!
random-detect cos {one_value | values value_list} percent min_% max_
! Applies CoS to a WRED-drop threshold and configures the threshold percentages.
!
random-detect dscp {one_value | values value_list} percent min_% max_
! Applies DSCP to a WRED-drop threshold and configures the threshold percentages.
!
random-detect (precedence one_value | values precedence value_list) percent min_% max_
! Applies DSCP to a WRED-drop threshold and configures the threshold percentages.

1p7q4t Ingress or Egress Queue Supported Commands

- Supports CoS-based, DSCP-based, and precedence-based queueing with tail-drop and WRED-drop thresholds. Supports SRR or DWRR dequeueing.
- Unsupported commands are included in this section as comments.

- Priority queue:
  ```
  class class_map_name
  ! Receives QoS values (CoS, DSCP, precedence) values filtered by class_map_name.
  ! priority ! Applies the class map to the priority queue
  ! Not supported if SRR mode is enabled.
  !
  queue-limit percent percentage ! Queue buffer size.
  ```

- Class commands for nonpriority queues numbered higher than 1— Configures queues in reverse numerical order; repeat to configure the next queue:
  ```
  class class_map_name
  ! Receives QoS values (CoS, DSCP, precedence) values filtered by class_map_name.
  ```

- Class command for queue #1:
  ```
  class class-default ! Receives all remaining QoS values (CoS, DSCP, precedence).
  ```

- Nonpriority queue configuration commands:
  ```
  shape average percent percentage
  ! Enables SRR on nonpriority egress queues.
  bandwidth remaining percent percentage
  ! DWRR bandwidth allocation.
  !
  queue-limit percent percentage ! Queue buffer size.
  ```
• Nonpriority queue threshold configuration—Repeat to configure sequential thresholds in each queue:

   queue-limit multiple-type-based
   ! Enables application of CoS, precedence, and DSCP values to a tail-drop threshold.
   !
   queue-limit cos {one_value | values value_list} percent percentage
   ! Applies CoS to a tail-drop threshold and configures the threshold percentage
   !
   queue-limit dscp {one_value | values value_list} percent percentage
   ! Applies one DSCP value to a tail-drop threshold
   ! and configures the threshold percentage
   !
   random-detect cos-based [aggregate]
   ! Enables application of CoS values to a WRED-drop threshold.
   ! The 'aggregate' keyword allows use of the 'values' keyword
   !
   random-detect dscp-based [aggregate]
   ! Enables application of DSCP values to a WRED-drop threshold.
   ! The 'aggregate' keyword allows use of the 'values' keyword
   !
   random-detect precedence-based [aggregate]
   ! Enables application of precedence values to a WRED-drop threshold.
   ! The 'aggregate' keyword allows use of the 'values' keyword
   !
   random-detect multiple-type-based [aggregate]
   ! Enables application of CoS, precedence, and DSCP values to a WRED-drop threshold.
   ! The 'aggregate' keyword allows use of the 'values' keyword
   !
   random-detect cos {one_value | values value_list} percent min_% max_%
   ! Applies CoS to a WRED-drop threshold and configures the threshold percentages.
   !
   random-detect dscp {one_value | values value_list} percent min_% max_%
   ! Applies DSCP to a WRED-drop threshold and configures the threshold percentages.
   !
   random-detect (precedence one_value | values precedence value_list) percent min_% max_%
   ! Applies DSCP to a WRED-drop threshold and configures the threshold percentages.

Queueing Policy Commands Sample Configurations for Each Queue Type

• 1q2t Ingress Queue Sample Configuration, page 61-29
• 1q8t Ingress Queue Sample Configuration, page 61-29
• 2q8t Ingress Queue Sample Configuration, page 61-29
• 8q4t, 8q8t Ingress Queue Sample Configuration (CoS-Based Queueing), page 61-30
• 8q4t Ingress Queue Sample Configuration (DSCP-Based Queueing), page 61-31
• 1p1q4t Ingress Queue Sample Configuration, page 61-33
• 1p1q8t Ingress Queue Sample Configuration, page 61-34
• 1p3q8t Egress Queue Sample Configuration, page 61-34
• 1p7q8t Egress Queue Sample Configuration, page 61-36
• 1p7q4t Ingress or Egress Queue Sample Configuration (CoS-Based Queueing), page 61-37
• 1p7q4t Ingress or Egress Queue Sample Configuration (DSCP-Based Queueing), page 61-38
Chapter 61  Policy-Based Queueing

Configuration Examples for Policy-Based Queueing

Note
These sample configurations approximate the default queueing enabled by the auto qos default and platform qos queueing-only global configuration commands.

1q2t Ingress Queue Sample Configuration

Supports CoS-based queueing with tail-drop thresholds.

```plaintext
class-map type lan-queuing match-any c_map_cos_5
  match cos 5

policy-map type lan-queuing p_map_2q8t ! For 2q8t
  ! Configures queue #2:
```

Note
Supports CoS-based queueing with tail-drop thresholds.

```plaintext
policy-map type lan-queuing p_map_1q2t
  class class-default ! Receives all CoS values.
    ! Configures threshold #1:
      queue-limit cos values 0 1 3 4 percent 80
      ! Applies CoS values to threshold 1 and configures the threshold percentage
    ! Other thresholds unconfigured; default to 100%
    ! Remaining CoS values are not explicitly configured:
      ! default to threshold 8 at 100%
```

1q8t Ingress Queue Sample Configuration

Supports CoS-based queueing with tail-drop thresholds.

```plaintext
policy-map type lan-queuing p_map_1q8t
  class class-default ! Receives all CoS values.
    ! Configures threshold #1:
      queue-limit cos 0 percent 50
      ! Applies CoS 0 to threshold 1 and configures the threshold percentage
    ! Configures threshold #2:
      queue-limit cos values 1 2 3 4 percent 60
      ! Applies CoS 1, 2, 3, 4 to threshold 2 and configures the threshold percentage
    ! Configures threshold #3:
      queue-limit cos values 6 7 percent 80
      ! Applies CoS 6 and 7 to threshold 3 and configures the threshold percentage
    ! Other thresholds unconfigured; default to 100%
    ! CoS 5 is not explicitly configured; defaults to threshold 8 at 100%
```

2q8t Ingress Queue Sample Configuration

Supports CoS-based queueing with tail-drop thresholds.

```plaintext
class-map type lan-queuing match-any c_map_cos_5
  match cos 5

! Configures queue #2:
```
class c_map_cos_5
! Receives QoS values (CoS, DSCP, precedence) values filtered by c_map_cos_5.
! bandwidth percent 10 ! WRR bandwidth allocation.
  queue-limit percent 20 ! Queue buffer size.
! queue-limit cos 5 percent 100
! Applies CoS 5 to threshold 1 and configures the threshold percentage
!
! Configures queue #1:
class class-default ! Receives all remaining CoS values.
! bandwidth percent 90 ! WRR bandwidth allocation.
! queue-limit percent 20 ! Queue buffer size.
!
! Configures threshold #1:
  queue-limit cos values 0 1 percent 70
! Applies CoS 0 1 to threshold 1 and configures the threshold percentage
!
! Configures threshold #2:
  queue-limit cos values 2 3 percent 80
! Applies CoS 2 3 to threshold 2 and configures the threshold percentage
!
! Configures threshold #3:
  queue-limit cos 4 percent 90
! Applies CoS 4 to threshold 3 and configures the threshold percentage
!
! CoS 6 and 7 default to threshold 4 at 100%

8q4t, 8q8t Ingress Queue Sample Configuration (CoS-Based Queueing)

- Supports CoS-based, DSCP-based, and precedence-based queueing with tail-drop and WRED-drop thresholds. This sample configures CoS-based queueing.
- In releases where CSCts82932 is not resolved, do not use the default DSCP-based queue mapping for 8q4t ingress queues unless you configure supporting bandwidth and queue limits.

class-map type lan-queuing match-any c_map_cos_5
  match cos 5
!
policy-map type lan-queuing p_map_8q4t_cos_8q8t ! For 8q4t CoS-based queueing and 8q8t
!
  ! Configures queue #8:
class c_map_cos_5 ! Receives CoS values values filtered by c_map_cos_5.
  ! bandwidth percent 90 ! WRR bandwidth allocation.
    queue-limit percent 20 ! Queue buffer size.
    ! ! CoS 5 defaults to threshold 4 at 100%
  !
  ! Configures queue #1:
class class-default
  ! Receives all remaining CoS values.
  ! bandwidth percent 10 ! WRR bandwidth allocation.
    queue-limit percent 80 ! Queue buffer size.
! Configures threshold #1:
random-detect cos-based aggregate
! The 'aggregate' keyword allows use of the 'values' keyword
! random-detect cos values 0 1 percent 40 70
! Configures threshold #2:
random-detect cos-based aggregate
! The 'aggregate' keyword allows use of the 'values' keyword
! random-detect cos values 2 3 percent 40 80
! Configures threshold #3:
random-detect cos-based
! random-detect cos value 4 percent 50 90
! Configures threshold #4:
random-detect cos-based aggregate
! The 'aggregate' keyword allows use of the 'values' keyword
! random-detect cos values 6 7 percent 50 100

8q4t Ingress Queue Sample Configuration (DSCP-Based Queueing)

- Supports CoS-based, DSCP-based, and precedence-based queueing with tail-drop and WRED-drop thresholds. This sample configures DSCP-based queueing.
- In releases where CSCts82932 is not resolved, do not use the default DSCP-based queue mapping for 8q4t ingress queues unless you configure supporting bandwidth and queue limits.

```
class-map type lan-queuing match-any c_map_cos_5_dscp_40_46
 match cos 5
   ! match dscp 40 46
   ! NOTE: Enables DSCP-based queueing on the port in the direction of the queueing policy.
 !
class-map type lan-queuing match-any c_map_dscp_48-63
 match dscp 48 49 50 51 52 53 54 55
 match dscp 56 57 58 59 60 61 62 63
   ! NOTE: Enables DSCP-based queueing on the port in the direction of the queueing policy.
 !
class-map type lan-queuing match-any c_map_dscp_32_34-38
 match dscp 32 34 35 36 37 38
   ! NOTE: Enables DSCP-based queueing on the port in the direction of the queueing policy.
 !
class-map type lan-queuing match-any c_map_dscp_24_26_28_30
 match dscp 24 26 28 30
   ! NOTE: Enables DSCP-based queueing on the port in the direction of the queueing policy.
 !
class-map type lan-queuing match-any c_map_dscp_18_20_22
 match dscp 18 20 22
   ! NOTE: Enables DSCP-based queueing on the port in the direction of the queueing policy.
 !
class-map type lan-queuing match-any c_map_dscp_10_12_14
 match dscp 10 12 14
   ! NOTE: Enables DSCP-based queueing on the port in the direction of the queueing policy.
```
! policy-map type lan-queuing p_map_8q4t_dscp ! For 8q4t DSCP-based queueing

! Configures queue #8:
class c_map_cos_5_dscp_40_46
! Receives QoS values (CoS, DSCP, precedence) values filtered by c_map_cos_5.
! bandwidth percent 90 ! WRR bandwidth allocation.
  queue-limit percent 20 ! Queue buffer size.
! CoS 5 and DSCP 40, 46 default to threshold 4 at 100%
!
! Configures queue #7:
class c_map_dscp_48-63
! Receives QoS values (CoS, DSCP, precedence) values filtered by c_map_dscp_48-63.
! bandwidth remaining percent 10 ! WRR bandwidth allocation.
  queue-limit percent 10 ! Queue buffer size.
! DSCP 48-63 default to threshold 4 at 100%
!
! Configures queue #6:
class c_map_dscp_32_34-38
! Receives QoS values (CoS, DSCP, precedence) values filtered by c_map_dscp_32_34-38.
! bandwidth remaining percent 10 ! WRR bandwidth allocation.
  queue-limit percent 10 ! Queue buffer size.
! DSCP 32, 34-38 default to threshold 4 at 100%
!
! Configures queue #5:
class c_map_dscp_24_26_28_30
! Receives QoS values (CoS, DSCP, precedence) values filtered by c_map_dscp_24_26_28_30.
! bandwidth remaining percent 10 ! WRR bandwidth allocation.
  queue-limit percent 10 ! Queue buffer size.
! DSCP 24, 26, 28, 30 default to threshold 4 at 100%
!
! Configures queue #4:
class c_map_dscp_18_20_22
! Receives QoS values (CoS, DSCP, precedence) values filtered by c_map_dscp_18_20_22.
! bandwidth remaining percent 10 ! WRR bandwidth allocation.
  queue-limit percent 10 ! Queue buffer size.
! Configures threshold #1:
random-detect dscp-based
random-detect dscp 20 percent 70 100
random-detect dscp 22 percent 70 100
random-detect dscp 18 percent 70 100
!
! Configures queue #3:
class c_map_dscp_10_12_14
! Receives QoS values (CoS, DSCP, precedence) values filtered by c_map_dscp_10_12_14.
! bandwidth remaining percent 10 ! WRR bandwidth allocation.
  queue-limit percent 10 ! Queue buffer size.
Chapter 61  Policy-Based Queueing

Configuration Examples for Policy-Based Queueing

1p1q4t Ingress Queue Sample Configuration

Note: Supports CoS-based queueing with tail-drop thresholds.

class-map type lan-queuing match-any c_map_cos_5
match cos 5
!
policy-map type lan-queuing p_map_1p1q4t ! For 1p1q4t
!
! Configures the priority queue:
class c_map_cos_5 ! Receives CoS values filtered by c_map_cos_5.
    priority ! Applies the class map to the priority queue
!
! Configures queue #1:
class class-default ! Receives all remaining CoS values.
!
! Configures threshold #1
queue-limit cos values 0 1 percent 70
! Applies CoS 0 1 to threshold 1 and configures the threshold percentage
!
! Configures threshold #2
queue-limit cos values 2 3 percent 80
! Applies CoS 2 3 to threshold 2 and configures the threshold percentage
!
! Configures threshold #3
queue-limit cos 4 percent 90
! Applies CoS 4 to threshold 3 and configures the threshold percentage
!
! CoS 6 and 7 default to threshold 4 at 100%

1p1q8t Ingress Queue Sample Configuration

Note
Supports CoS-based queueing with tail-drop or WRED-drop thresholds.

```plaintext
class-map type lan-queuing match-any c_map_cos_5
    match cos 5
!
policy-map type lan-queuing p_map_1p1q8t ! For 1p1q8t
!
! Configures the priority queue:
   class c_map_cos_5 ! Receives CoS values filtered by c_map_cos_5.
       priority ! Applies the class map to the priority queue
!
   class class-default ! Receives all remaining CoS values.
       !
       random-detect cos-based
       ! Enables application of CoS values to a WRED-drop threshold.
!
! Configures threshold #1
   random-detect cos 0 percent 40 70
       ! Applies CoS to WRED-drop threshold 1 and configures the threshold percentages.
!
! Configures threshold #2
   random-detect cos 1 percent 40 70
       ! Applies CoS to WRED-drop threshold 2 and configures the threshold percentages.
!
! Configures threshold #3
   random-detect cos 2 percent 50 80
       ! Applies CoS to WRED-drop threshold 3 and configures the threshold percentages.
!
! Configures threshold #4
   random-detect cos 3 percent 50 80
       ! Applies CoS to WRED-drop threshold 4 and configures the threshold percentages.
!
! Configures threshold #5
   random-detect cos 4 percent 60 90
       ! Applies CoS to WRED-drop threshold 5 and configures the threshold percentages.
!
! Configures threshold #6
   random-detect cos 6 percent 60 90
       ! Applies CoS to WRED-drop threshold 6 and configures the threshold percentages.
!
! Configures threshold #7
   random-detect cos 7 percent 70 100
       ! Applies CoS to WRED-drop threshold 7 and configures the threshold percentages.
```

1p3q8t Egress Queue Sample Configuration

Note
Supports CoS-based queueing with tail-drop and WRED-drop thresholds.

```plaintext
class-map type lan-queuing match-any c_map_cos_2_3_4
    match cos 2 3 4
!
```
class-map type lan-queuing match-any c_map_cos_5
  match cos 5
!
class-map type lan-queuing match-any c_map_cos_6_7
  match cos 6 7
!
policy-map type lan-queuing p_map_1p3q8t
!
! Configures the priority queue:
  class c_map_cos_5 ! Receives CoS values filtered by c_map_cos_5.
    priority ! Applies the class map to the priority queue
  !
! Configures queue #3 for 1p3q8t:
  class c_map_cos_6_7
    ! Receives CoS values values filtered by c_map_cos_6_7.
    bandwidth remaining percent 40 ! WRR bandwidth allocation.
    queue-limit percent 15 ! Queue buffer size.
    !
    ! Configures threshold #1:
    random-detect cos-based aggregate
    ! The 'aggregate' keyword allows use of the 'values' keyword
    !
    random-detect cos values 6 7 percent 70 100
  !
! Configures queue #2 for 1p3q8t:
  class c_map_cos_2_3_4
    ! Receives CoS values values filtered by c_map_cos_2_3_4.
    bandwidth remaining percent 30 ! WRR bandwidth allocation.
    queue-limit percent 20 ! Queue buffer size.
    !
    ! Configures threshold #1:
    random-detect cos-based
    !
    random-detect cos 2 percent 40 70
    ! Applies CoS to WRED-drop threshold 1 and configures the threshold percentages.
    !
    ! Configures threshold #2:
    random-detect cos-based aggregate
    ! The 'aggregate' keyword allows use of the 'values' keyword
    !
    random-detect cos values 3 4 percent 70 100
    ! Applies CoS to WRED-drop threshold 2 and configures the threshold percentages.
  !
! Configures queue #1:
  class class-default
  ! Receives all remaining CoS values.
  bandwidth remaining percent 25 ! WRR bandwidth allocation.
  queue-limit percent 20 ! Queue buffer size.
  !
  ! Configures threshold #1:
  random-detect cos-based
  !
  random-detect cos 0 percent 40 70
  ! Applies CoS to WRED-drop threshold 1 and configures the threshold percentages.
  !
  ! Configures threshold #2:
  random-detect cos-based
  !
  random-detect cos 1 percent 70 100
  ! Applies CoS to WRED-drop threshold 2 and configures the threshold percentages.
1p7q8t Egress Queue Sample Configuration

Supports CoS-based queueing with tail-drop and WRED-drop thresholds.

class-map type lan-queuing match-any c_map_cos_2_3_4
  match cos 2 3 4
!
class-map type lan-queuing match-any c_map_cos_5
  match cos 5
!
class-map type lan-queuing match-any c_map_cos_6_7
  match cos 6 7
!
policy-map type lan-queuing p_map_1p7q8t
!
! Configures the priority queue:
  class c_map_cos_5 ! Receives CoS values filtered by c_map_cos_5.
    priority ! Applies the class map to the priority queue
  !
  ! Configures queue #7:
  class c_map_cos_6_7
  ! Receives CoS values values filtered by c_map_cos_6_7.
  !
    bandwidth remaining percent 40 ! WRR bandwidth allocation.
    queue-limit percent 15 ! Queue buffer size.
  !
  ! Configures threshold #1:
  random-detect cos-based aggregate
  ! The 'aggregate' keyword allows use of the 'values' keyword
  !
  random-detect cos values 6 7 percent 70 100
  !
  ! Configures queue #6 for 1p7q8t:
  class c_map_cos_2_3_4
  ! Receives CoS values values filtered by c_map_cos_2_3_4.
  !
    bandwidth remaining percent 30 ! WRR bandwidth allocation.
    queue-limit percent 20 ! Queue buffer size.
  !
  ! Configures threshold #1:
  random-detect cos-based
  !
  random-detect cos 2 percent 40 70
  ! Applies CoS to WRED-drop threshold 1 and configures the threshold percentages.
  !
  ! Configures threshold #2:
  random-detect cos-based aggregate
  ! The 'aggregate' keyword allows use of the 'values' keyword
  !
  random-detect cos values 3 4 percent 70 100
  ! Applies CoS to WRED-drop threshold 2 and configures the threshold percentages.
  !
  ! Configures queue #1:
  class class-default
  ! Receives all remaining CoS values.
  !
    bandwidth remaining percent 25 ! WRR bandwidth allocation.
    queue-limit percent 50 ! Queue buffer size.
  !
    ! Configures threshold #1:
    random-detect cos-based
61-37

Cisco IOS Software Configur ation Guide, Release 12.2SY

Chapter 61      Policy-Based Queueing

Configuration Examples for Policy-Based Queueing

! random-detect cos 0 percent 40 70
! Applies CoS to WRED-drop threshold 1 and configures the threshold percentages.
!
! Configures threshold #2:
random-detect cos-based
!
random-detect cos 1 percent 70 100
! Applies CoS to WRED-drop threshold 2 and configures the threshold percentages.

1p7q4t Ingress or Egress Queue Sample Configuration (CoS-Based Queueing)

Note Supports CoS-based, DSCP-based, and precedence-based queueing with tail-drop and WRED-drop thresholds. Supports SRR or DWRR dequeueing. This sample configures CoS-based queueing.

class-map type lan-queuing match-any c_map_cos_5
   match cos 5
!
policy-map type lan-queuing p_map_1p7q4t_cos

   ! Configures the priority queue:
   class c_map_cos_5 ! Receives CoS values filtered by c_map_cos_5.
      priority ! Applies the class map to the priority queue
   !
   ! Configures queue #1:
   class class-default
   ! Receives all remaining CoS values.
   !
   bandwidth remaining percent 85 ! WRR bandwidth allocation.
   queue-limit percent 10 ! Queue buffer size.
   !
   ! Configures threshold #1:
   random-detect cos-based aggregate
   ! The 'aggregate' keyword allows use of the 'values' keyword
   !
   random-detect cos values 0 1 percent 40 70
   !
   ! Configures threshold #2:
   random-detect cos-based aggregate
   ! The 'aggregate' keyword allows use of the 'values' keyword
   !
   random-detect cos values 2 3 percent 40 80
   !
   ! Configures threshold #3:
   random-detect cos-based
   !
   random-detect cos value 4 percent 50 90
   !
   ! Configures threshold #4:
   random-detect cos-based aggregate
   ! The 'aggregate' keyword allows use of the 'values' keyword
   !
   random-detect cos values 6 7 percent 50 100
Supports CoS-based, DSCP-based, and precedence-based queueing with tail-drop and WRED-drop thresholds. Supports SRR or DWRR dequeueing. This sample configures DSCP-based queueing.

```
class-map type lan-queuing match-any c_map_cos_5_dscp_40_46
  match cos 5
    match dscp 40 46
    ! NOTE: Enables DSCP-based queueing on the port in the direction of the queueing policy.
!
class-map type lan-queuing match-any c_map_dscp_48-63
  match dscp 48 49 50 51 52 53 54 55
  match dscp 56 57 58 59 60 61 62 63
  ! NOTE: Enables DSCP-based queueing on the port in the direction of the queueing policy.
!
class-map type lan-queuing match-any c_map_dscp_32_34-38
  match dscp 32 34 35 36 37 38
  ! NOTE: Enables DSCP-based queueing on the port in the direction of the queueing policy.
!
class-map type lan-queuing match-any c_map_dscp_24_26_28_30
  match dscp 24 26 28 30
  ! NOTE: Enables DSCP-based queueing on the port in the direction of the queueing policy.
!
class-map type lan-queuing match-any c_map_dscp_18_20_22
  match dscp 18 20 22
  ! NOTE: Enables DSCP-based queueing on the port in the direction of the queueing policy.
!
class-map type lan-queuing match-any c_map_dscp_10_12_14
  match dscp 10 12 14
  ! NOTE: Enables DSCP-based queueing on the port in the direction of the queueing policy.
!
policy-map type lan-queuing p_map_1p7q4t_dscp
  ! Configures the priority queue:
  ! Receives CoS values filtered by c_map_cos_5_dscp_40_46.
  priority ! Applies the class map to the priority queue
  ! Receives queue #7:
  class c_map_dscp_48-63
  ! Receives QoS values (CoS, DSCP, precedence) values filtered by c_map_dscp_48-63.
  ! bandwidth remaining percent 10 ! WRR bandwidth allocation.
  ! queue-limit percent 10 ! Queue buffer size.
  ! DSCP 48-63 default to threshold 4 at 100%
  ! Receives queue #6:
  class c_map_dscp_32_34-38
  ! Receives QoS values (CoS, DSCP, precedence) values filtered by c_map_dscp_32_34-38.
  ! bandwidth remaining percent 10 ! WRR bandwidth allocation.
  ! queue-limit percent 10 ! Queue buffer size.
  ! DSCP 32, 34-38 default to threshold 4 at 100%
```
! Configures queue #5:
class c_map_dscp_24_26_28_30
! Receives QoS values (CoS, DSCP, precedence) values filtered by c_map_dscp_24_26_28_30.
  ! bandwidth remaining percent 10 ! WRR bandwidth allocation.
  queue-limit percent 10 ! Queue buffer size.
  ! Configures threshold #1
  queue-limit dscp values 24 30 percent 100
  ! Configures threshold #2
  queue-limit dscp 28 percent 100
  ! Configures threshold #3
  queue-limit dscp 26 percent 100

! Configures queue #4:
class c_map_dscp_18_20_22
! Receives QoS values (CoS, DSCP, precedence) values filtered by c_map_dscp_18_20_22.
  ! bandwidth remaining percent 10 ! WRR bandwidth allocation.
  queue-limit percent 10 ! Queue buffer size.
  ! Configures threshold #1:
random-detect dscp-based
random-detect dscp 20 percent 70 100
  ! Configures threshold #2:
random-detect dscp-based
random-detect dscp 22 percent 70 100
  ! Configures threshold #3:
random-detect dscp-based
random-detect dscp 18 percent 70 100

! Configures queue #3:
class c_map_dscp_10_12_14
! Receives QoS values (CoS, DSCP, precedence) values filtered by c_map_dscp_10_12_14.
  ! bandwidth remaining percent 10 ! WRR bandwidth allocation.
  queue-limit percent 10 ! Queue buffer size.
  ! Configures threshold #1:
random-detect dscp-based
random-detect dscp 14 percent 70 100
random-detect dscp 12 percent 70 100
random-detect dscp 10 percent 70 100

! Configures queue #1:
class class-default
! Receives all remaining QoS values (CoS, DSCP, precedence).
  ! bandwidth remaining percent 25 ! WRR bandwidth allocation.
  queue-limit percent 10 ! Queue buffer size.
  ! Configures threshold #1:
random-detect cos-based aggregate
random-detect cos values 0 1 percent 40 70
  ! Configures threshold #2:
random-detect cos-based aggregate
random-detect cos values 2 3 percent 40 80
! Configures threshold #3:
random-detect cos-based
random-detect cos value 4 percent 50 90
!
! Configures threshold #4:
! Remaining DSCP values default to this threshold
! 0-9, 11, 13, 15-17, 19, 21, 23, 25, 27, 29, 31, 33, 39, 41-45, 47
!
random-detect cos-based aggregate
random-detect cos values 6 7 percent 50 100

Tip
For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:
Participate in the Technical Documentation Ideas forum