



Configuring Class of Service Control

- [Overview of Class of Service Control, on page 1](#)
- [Configure Class of Service Control, on page 2](#)
- [Examples: Configuring Class of Service Control, on page 5](#)

Overview of Class of Service Control

Class of Service (CoS) helps resolve the problem of network congestion by giving certain types of traffic priority over others. Network congestion occurs when multiple messages are competing for network resources at the same time.

Common queue scheduling algorithms such as First Come First Serve (FCFS), Strict-Priority Queue scheduling, Weighted Round Robin (WRR) scheduling, and Strict-Priority Queue + WRR scheduling help decongest a network.

First Come First Serve

The FCFS algorithm does not classify a message. It simply follows a first-in-first-out method. When a message arrives at an interface faster than the interface can send it, the FCFS algorithm forwards the message to the queue in the order of receiving the message. It sends out messages in the same order as receiving them.

Strict-Priority Queuing

Strict-Priority Queuing is designed for critical business applications, wherein services are prioritized in order to reduce the latency of response when a congestion occurs. A priority queue classifies all messages into eight classes—7, 6, 5, 4, 3, 2, 1, and 0, in the order of priority. The group of critical services is put into the higher-priority queue, and noncritical business group is put into the lower-priority queue. The higher-priority queue is emptied before the messages in the lower-priority queue are sent. Messages in the group of noncritical business are transmitted in the idle gap of handling critical business data.

The disadvantage of Strict-Priority Queuing is that the messages in the lower-priority group are not sent if the higher-priority queue is not emptied.

Weighted Round Robin Scheduling

Weighted Round Robin (WRR) queue scheduling divides each port into eight output queues—7, 6, 5, 4, 3, 2, 1, and 0, in that order of priority, with 7 being the highest priority. All the queues are scheduled by turns and each queue gets a certain service time. Each queue of WRR can be configured with weighted values of w7, w6, w5, w4, w3, w2, w1, or w0. The weighted value represents the weight of the resource. For example, on a 100 Mb port, if you configure the WRR for 80, 70, 60, 50, 50, 40, 30, and 20, in that order of priority, the WRR of 20 is assured of at least 5 Mbps bandwidth.

An advantage of WRR queuing is that although multiple queues are scheduled by polling, each queue is not assigned a fixed time slot. If a queue is empty, it immediately switches to the next queue schedule so that the bandwidth and resources of that queue can be fully utilized.

Strict-Priority Queuing and Weighted Round Robin Scheduling

Strict-Priority queuing and WRR scheduling combine their algorithms. If the weight of a queue is set to 0, the queue follows the Strict-Priority queuing algorithm to send messages. A non-0 value of the weight switches the queue to the WRR scheduling mechanism.

Weighted Fair Queuing

Weighted Fair Queuing (WFQ) is flow-based queuing that schedules interactive traffic to the front of the queue to reduce response time. WFQ shares the remaining bandwidth between high-bandwidth flows.

Configure Class of Service Control

The following sections provide information about configuring class of service control.

Configure Class of Service



Note By default, Strict-Priority Queuing is configured.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 2	queue-scheduler strict-priority Example: Device(config)# <code>queue-scheduler strict-priority</code>	(Optional) Configures Strict-Priority Queuing mode.

	Command or Action	Purpose
Step 3	queue-scheduler wrr <i>w1 w2 w3 w4 w5 w6 w7 w8</i> Example: Device(config)# queue-scheduler wrr 1 2 3 4 5 6 7 8	(Optional) Configures WRR scheduling mode.
Step 4	queue-scheduler sp-wrr <i>w1 w2 w3 w4 w5 w6 w7 w8</i> Example: Device(config)# queue-scheduler sp-wrr 1 2 3 4 5 6 7 8	(Optional) Configures Strict-Priority and WRR scheduling mode.
Step 5	no queue-scheduler Example: Device(config)# no queue-scheduler	(Optional) Restores the default scheduler, which is Strict-Priority Queuing.
Step 6	show queue-scheduler Example: Device(config)# show queue-scheduler	(Optional) Displays information about the queue scheduler.

Configure DSCP to 802.1p Mapping

Differentiated Services Code Point (DSCP) marking operates in Layer 3 and determines traffic classification for network data. 802.1p marking is a Layer 2 Class of Service. You can define what level of service you want to allocate to specific markings. DSCP allows 64 priority values, whereas 802.1p (hardware queue) allows only eight priority values. By default, the following is the mapping between DSCP and 802.1p:

Table 1: DSCP-802.1p (Hardware Priority Queue) Mapping

DSCP	802.1p
0-7	0
8-15	1
16-23	2
24-31	3
32-39	4
40-47	5
48-55	6
56-63	7

**Note**

- DSCP mapping is disabled by default. Enable DSCP mapping before you perform related configurations.
- You can change the mapping relation between DSCP precedence and output queues by changing the mapping between DSCP priorities and 802.1p priorities according to the actual network requirements.

To configure DSCP to 802.1p mapping, perform the following procedure:

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 2	[no] queue-scheduler dscp-map Example: Device(config)# <code>queue-scheduler dscp-map</code>	(Optional) Enables DSCP mapping function. Use the no form of the command to disable DSCP mapping.
Step 3	queue-scheduler dscp-map dscp-v priority-v Example: Device(config)# <code>queue-scheduler dscp-map 1 4</code>	(Optional) Maps DSCP values to 802.1p values (hardware priority queue).
Step 4	show queue-scheduler dscp-map Example: Device(config)# <code>show queue-scheduler dscp-map</code>	(Optional) Displays the mapping between DSCP and 802.1p values.

Configure 802.1p and Hardware Queue Mapping

802.1p is used to classify the outgoing traffic at the egress port based on the 802.1p priority. For each message that enters the switch, the system maps the specific hardware queue priority according to the 802.1p priority of the message.

The default mapping relation between 802.1p and hardware priority is shown in the following table:

Table 2: 802.1p and Hardware Queue Mapping

802.1p	Hardware Priority Queue
0	0
1	1
2	2

802.1p	Hardware Priority Queue
3	3
4	4
5	5
6	6
7	7

**Note**

- Changing the mapping relation between 802.1p priority and hardware queues changes the mapping relation between 802.1p priorities and output queues.
- If two 802.1p priorities are mapped to the same hardware priority queue, messages of the two 802.1p priorities cannot be forwarded with 1:1 forwarding.

To configure 802.1p to hardware queue mapping, perform the following procedure:

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 2	queue-scheduler cos-map <i>queue-v</i> <i>priority-v</i> Example: Device(config)# <code>queue-scheduler cos-map</code> 1 10	(Optional) Configures 802.1p and the hardware queue map.
Step 3	show queue-scheduler cos-map Example: Device(config)# <code>show queue-scheduler</code> <code>cos-map</code>	(Optional) Displays 802.1p and the hardware queue mapping information.

Examples: Configuring Class of Service Control

The following example shows how to view the default queue scheduling mode:

```
Device(config)# show queue-scheduler

Queue scheduler status : enable
Queue scheduler mode   : SP (Strict Priority)
```

The following example displays the priority-mapping relationship between 802.1p and hardware queues.

```
Device(config)# show queue-scheduler cos-map
```

```
Information about map of cos:
802.1P Priority Queue of class
-----
0             0
1             1
2             2
3             3
4             4
5             5
6             6
7             7
```

The following example shows how to modify the priority-mapping relationship between 802.1p and hardware queue by mapping the packets with priority 0 to queue 1:

```
Device(config)# queue-scheduler cos-map 1 10
Configured successfully.
```

```
Device(config)# show queue-scheduler cos-map
```

```
Information about map of cos:
802.1P Priority Queue of class
-----
0             1
1             1
2             2
3             3
4             4
5             5
6             6
7             7
```

The following example shows how to configure WRR queue scheduling:

```
Device(config)# queue-scheduler wrr 1 2 3 4 5 6 7 8
Configured queue scheduler successfully
```

```
Device(config)# show queue-scheduler
```

```
Queue scheduler status : enable
Queue scheduler mode   : WRR (Weighted Round Robin)
Queue0 weight is 1
Queue1 weight is 2
Queue2 weight is 3
Queue3 weight is 4
Queue4 weight is 5
Queue5 weight is 6
Queue6 weight is 7
Queue7 weight is 8
```

The following example shows how to restore the default queue schedule:

```
Device(config)# no queue-scheduler
```

```
Recovered queue scheduler to default value(strict-priority) successfully.
```

```
Device(config)# show queue-scheduler
```

```
Queue scheduler status : enable
Queue scheduler mode   : SP (Strict Priority)
```