



## Multi-Level Priority Queues

The Multi-Level Priority Queues (MPQ) feature allows you to configure multiple priority queues for multiple traffic classes by specifying a different priority level for each of the traffic classes in a single service policy map. You can configure multiple service policy maps per device. Having multiple priority queues enables the device to place delay-sensitive traffic (for example, voice) on the outbound link before delay-insensitive traffic. As a result, high-priority traffic receives the lowest latency possible on the device.

- [Prerequisites for Multi-Level Priority Queues, on page 1](#)
- [Restrictions for Multi-Level Priority Queues, on page 1](#)
- [Information About Multi-Level Priority Queues, on page 2](#)
- [How to Configure Multi-Level Priority Queues, on page 4](#)
- [Configuration Examples for Multi-Level Priority Queues, on page 6](#)
- [Additional References for Multi-Level Priority Queues, on page 7](#)

## Prerequisites for Multi-Level Priority Queues

You must configure traffic classes using the **class-map** command.

## Restrictions for Multi-Level Priority Queues

- The Bandwidth kbps and percent command cannot co-exist with strict priority or priority level in the same policy-map. So, a check is added to ensure only a policer with drop action, along with priority is allowed as a conditional priority.
- You cannot configure both the **priority** command and the **priority level** command for two different classes in the same policy map. For example, the device does not accept the following configuration:

```
Device> enable
Device# configure terminal
Device(config)# policy-map Map1
Device(config-pmap)# class Bronze
Device(config-pmap-c)# priority level 1
Device(config-pmap-c)# exit
Device(config-pmap)# class Gold
Device(config-pmap-c)# priority 1000
Device(config-pmap-c)# end
```

- You cannot specify the same priority level for two different classes in the same policy map. For example, the device does not accept the following configuration:

```
Device> enable
Device# configure terminal
Device(config)# policy-map Map1
Device(config-pmap)# class Bronze
Device(config-pmap-c)# priority level 1 10000
Device(config-pmap-c)# exit
Device(config-pmap)# class Gold
Device(config-pmap-c)# priority level 1
Device(config-pmap-c)# end
```

- You cannot configure the default queue as a priority queue at any level.
- You cannot configure the **bandwidth** command and multi-level priority queues on the same class. For example, the device rejects the following configuration:

```
policy-map P1
class C1
priority level 1
bandwidth 200
```

- You cannot configure the **shape** command and multi-level priority queues on the same class. For example, the device rejects the following configuration:

```
policy-map P1
class C1
priority level 1
shape average 56000
```

- To convert a one-level (flat) service policy with multiple priority queues configured to a hierarchical multi-level priority queues service policy, you must first detach the flat service policy from the interface using the **no service-policy** command and then add a child policy map to it.
- We recommend not to use MPQ at the logical level, that is, under the class-map containing match for the VLANs .

## Information About Multi-Level Priority Queues

### Benefits of Multi-Level Priority Queues

The MPQ feature allows you to configure multiple priority queues for multiple traffic classes by specifying a different priority level for each of the traffic classes in a single service policy map. You can configure multiple service policy maps per device.

Previously, devices could have only one strict priority queue per policy map for all delay-sensitive traffic—the device associated all priority traffic with this one single priority queue. However, having only one priority queue can cause significant delay in delivering traffic, especially if the device sends high-priority traffic (for example, voice) behind low-priority traffic (for example, video). Using class-based weighted fair queueing (CBWFQ) to reduce delay by heavily weighting one queue can affect the granularity of bandwidth allocations to the other queues. The MPQ feature addresses these issues and improves latency.

## Functionality of Multi-Level Priority Queues

The **priority** command is used to specify that a class of traffic has latency requirements with respect to other classes. For multiple priority queues, you can use the **priority level** command to configure a level of priority service on a class in a policy map. The device places traffic with a high-priority level on the outbound link ahead of traffic with a low-priority level. High-priority packets, therefore, are not delayed behind low-priority packets.

The device services the high-level priority queues until empty before servicing the next-level priority queues and non-priority queues. While the device services a queue, the service rate is as fast as possible and is constrained only by the rate of the underlying link or parent node in a hierarchy. If a rate is configured and the device determines that a traffic stream has exceeded the configured rate, the device drops the exceeding packets during periods of congestion. If the link is currently not congested, the device places the exceeding packets onto the outbound link.

When configuring MPQ on different traffic classes in a policy map, you must specify different priority levels for the traffic classes. For example, configure one traffic class to have priority level 2 and another class to have priority level 1.



---

**Note** In a hierarchical MPQ configuration in which *all* traffic is sent through the level-2 priority queue only, the traffic sent through the level-2 priority queue receives the same treatment as the traffic sent through the level-1 priority queue.

---

You cannot configure the **priority** command and the **priority level** command on different classes in the same policy map.

## Traffic Policing and Multi-Level Priority Queues

Bandwidth guarantees can be given to other classes only if traffic policing is enabled on the priority queue.

Using the **priority** and **police** commands, multi-level priority queues can be configured to police traffic in one of the following ways:

- Unconditional traffic policing, for example:

```
policy-map my_policy
  class voice
    priority <<< Indicates priority scheduling
  class gold
    bandwidth 400000 <<<400M minimum guaranteed to class gold
```

The priority class is configured with an “always on” (unconditional) policer. The priority class is always policed to the configured value regardless of whether the interface is congested. The advantage of an unconditional policer is that you always know how much priority traffic will be offered to the downstream devices, thus making your bandwidth planning much simpler. This is the recommended choice.

# How to Configure Multi-Level Priority Queues

## Configuring Multi-Level Priority Queues in a Policy Map

### Before you begin

The traffic classes, class maps, and policy maps must exist.

---

#### Step 1 **enable**

##### Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

#### Step 2 **configure terminal**

##### Example:

```
Device# configure terminal
```

Enters global configuration mode.

#### Step 3 **policy-map *policy-name***

##### Example:

```
Device(config)# policy-map Premium
```

Creates or modifies a policy map and enters policy-map configuration mode.

- Enter the name of the policy map.

#### Step 4 **class *class-name***

##### Example:

```
Device(config-pmap)# class business
```

Specifies a traffic class and enters policy-map class configuration mode.

- Enter the name of a previously configured traffic class.

#### Step 5 **priority level *level***

##### Example:

```
Device(config-pmap-c)# priority level 1 100000
```

Assigns priority to a traffic class at the priority level specified.

- Enter the level of priority assigned to the priority class.

**Note** Do not specify the same priority level for two different classes in the same policy map.

**Step 6** **end**

**Example:**

```
Device(config-pmap-c)# end
```

(Optional) Exits policy-map class mode.

---

## Verifying Multi-Level Priority Queues

---

**Step 1** **enable**

**Example:**

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

**Step 2** **show policy-map interface** *type number*

**Example:**

```
Device# show policy-map interface gig0/0/1
```

Displays the packet statistics of all classes that are configured for all service policies either on the specified interface or subinterface or on a specific PVC on the interface.

- Enter the interface type and number.

**Step 3** **exit**

**Example:**

```
Device# exit
```

(Optional) Exits privileged EXEC mode.

---

# Configuration Examples for Multi-Level Priority Queues

## Example: Configuring Multi-Level Priority Queues

The following example shows how to configure multiple priority queues with 2 level priority. The policy map named Business has two traffic classes: Bronze and Gold. Bronze traffic has a level 2 (low) priority, whereas Gold traffic has a level 1 (high) priority. To prevent bandwidth starvation of Bronze traffic, the Gold traffic is policed at 30 percent of the interface bandwidth.

```
Device> enable
Device# configure terminal
Device(config)# policy-map Business
Device(config-pmap)# class Bronze
Device(config-pmap-c)# priority level 2
Device(config-pmap-c)# exit
Device(config-pmap)# class Gold
Device(config-pmap-c)# priority level 1
Device(config-pmap-c)# end
```

The following example shows how to configure multiple priority queues with 7 level priority. The policy map named Business has seven traffic classes: Platinum, Gold, Silver, Bronze, Iron, Aluminium, and Steel. Steel traffic has a level 7 (lowest) priority, whereas Platinum traffic has a level 1 (highest) priority. To prevent bandwidth starvation, the Platinum and Gold traffic is policed at 30 percent and 20 percent respectively, of the interface bandwidth.

```
Device> enable
Device# configure terminal
Device(config)# policy-map Business
Device(config-pmap)# class Platinum
Device(config-pmap-c)# priority level 1
Device(config-pmap-c)# end
Device(config-pmap)# class Gold
Device(config-pmap-c)# priority level 2
Device(config-pmap-c)# end
```

## Example: Verifying Multi-Level Priority Queues

The following is partial sample output from the `show policy-map interface` command.

```
Device# show policy-map interface Gigbit/TenGig 2/1/0

Gigbit/TenGig 2/1/0
Service-policy output: P1
Queue statistics for all priority classes:
.
.
.
Class-map: Gold (match-all)
0 packets, 0 bytes /*Updated for each priority level configured.*/
5 minute offered rate 0 bps, drop rate 0 bps
Match: ip precedence 2
Priority: 0 kbps, burst bytes 1500, b/w exceed drops: 0
Priority Level 2:
0 packets, 0 bytes
```

## Additional References for Multi-Level Priority Queues

### Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Cisco IOS Quality of Service Solutions Command Reference</i>
Priority queues, creating classes, class maps, and policy maps	“Applying QoS Features Using the MQC” module

### Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	<a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a>

