Multi-Level Priority Queues

Last Updated: July 10, 2012

The Multi-Level Priority Queue (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.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

Prerequisites for Multi-Level Priority Queues

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

Restrictions for Multi-Level Priority Queues

- You cannot configure both the **priority** command and the **priority level** command for two different classes in the same policy map.
- You cannot specify the same priority level for two different classes in the same policy map.
- You cannot configure the default queue as a priority queue at any level.
- You cannot configure the **bandwidth** command and multi-level priority queueing 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 queueing 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 queueing configured to a hierarchical multi-level priority queueing 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.
- The sum of the police committed information rate (cir), which can be specified as kbps or a percentage, configured at the two priority levels cannot exceed the interface bandwidth. Any policy that has a police cir sum greater than interface bandwidth will be rejected. For example the following policy will be rejected by a 1-Gigabit interface because the sum of the police cir (600 Mb + 700 Mb) is greater than 1 G.

```
policy-map new
 class prec1
   priority level 1
   police cir 600000000
 class prec2
    priority level 2
    police cir 700000000
```

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. Currently, the device supports two priority levels: level 1 (high) and level 2 (low). 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.

If high-priority traffic is not policed appropriately, bandwidth starvation of low-priority traffic can occur. Therefore, though not required, we recommend that you configure a policer for high-priority traffic using the **police** command. If you configure the **police** command for priority queues, the traffic rate is policed to the police rate for each of the priority queues.

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:

- Conditional traffic policing, as illustrated below:

```
policy-map my_policy
  class voice
   priority 400000   <<< Priority queue conditionally policed to 400M
  class gold
   bandwidth 400000 <<<< 400M minimum guaranteed to class gold
```

With conditional traffic policing on the queue, you run the risk of sudden degradation in priority service when an interface becomes congested. You can go from an instance of a priority class using the entire link to suddenly traffic being policed to the configured value. You need to know the available bandwidth and use some form of admission control to ensure that your offered loads do not exceed the available bandwidth.

Note With the conditional policing, traffic policing does not engage unless the interface is congested.

- Unconditional traffic policing, as illustrated below:

```
policy-map my_policy
  class voice
    priority            <<< Indicates priority scheduling
    police 400000000    <<< Traffic policed to 400M
  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.

- Absolute priority queue (no traffic policing)
  If traffic policing is not configured, the priority traffic may consume the entire interface bandwidth.

How to Configure Multi-Level Priority Queues

Configuring Multi-Level Priority Queues in a Policy Map

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. policy-map policy-name
4. class class-name
5. priority level level
6. **police cir** *bps*
7.
8. police cir percent percent
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.
		• Enter your password if prompted.
	Example:	
	`Router> enable`	
Step 2	**configure terminal**	Enters global configuration mode.
	Example:	
	`Router# configure terminal`	
Step 3	policy-map *policy-name*	Creates or modifies a policy map and enters policy-map configuration mode.
		• Enter the name of the policy map.
	Example:	
	`Router(config)# policy-map Premium`	
Step 4	class *class-name*	Specifies a traffic class and enters policy-map class configuration mode.
		• Enter the name of a previously configured traffic class.
	Example:	
	`Router(config-pmap)# class business`	
Step 5	priority level *level*	Assigns priority to a traffic class at the priority level specified.
		• Enter the level of priority assigned to the priority class. Valid values are 1 (high priority) and 2 (low priority). The default is 1.
	Example:	**Note** Do not specify the same priority level for two different classes in the same policy map.
	`Router(config-pmap-c)# priority level 2`	
Step 6	**police cir** *bps*	(Optional) Configures traffic policing based on a bits per second (bps) rate.
		• Enter the **cir** keyword and a value for the *bps* argument. Note the following:
	Example:	◦ cir is the committed information rate and is based on the interface shape rate. This keyword indicates an average rate at which the policer meters traffic.
	`Router(config-pmap-c)# police cir 8000`	◦ bps specifies the average rate in bits per second (bps). Valid values are from 8000 to 2488320000 bps.
	Example:	
Step 7		

	Command or Action	Purpose
Step 8	police cir percent percent	(Optional) Configures traffic policing based on a percentage of bandwidth available on the interface.
	Example: Router(config-pmap-c)# police cir percent 20 **Example:**	• Enter the **cir** keyword, the **percent** keyword, and a value for the *percent* argument. Note the following: ◦ cir is the committed information rate and is based on the interface shape rate. Indicates an average rate at which the policer meters traffic. ◦ percent percent indicates to use the percentage of available bandwidth specified in percent to calculate the CIR. Valid values are from 1 to 100.
Step 9	end	(Optional) Exits policy-map class mode.
	Example: Router(config-pmap-c)# end	

Verifying Multi-Level Priority Queues

SUMMARY STEPS

1. **enable**
2. **show policy-map interface** *type number*
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
	Example: Router> enable	
Step 2	**show policy-map interface** *type number*	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.
	Example: Router# show policy-map interface serial4/0/0	

	Command or Action	Purpose
Step 3	**exit** **Example:** `Router# 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. 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)# police cir 1000
Device(config-pmap-c)# exit
Device(config-pmap)# class Gold
Device(config-pmap-c)# priority level 1
Device(config-pmap-c)# police cir percent 30
Device(config-pmap-c)# end
```

Note Although a policer is not required, configure policing for priority traffic to prevent bandwidth starvation of low-priority traffic. When policing is configured, the traffic rate is policed at the police rate for each of the priority queues.

Example: Unacceptable MPQ Configurations

You cannot specify 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 rate 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
Device(config-pmap-c)# police cir percent 30
Device(config-pmap-c)# exit
Device(config-pmap)# class Gold
Device(config-pmap-c)# priority level 1
Device(config-pmap-c)# police cir 10000
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 serial2/1/0
Serial2/1/0
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

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
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.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Multi-Level Priority Queues

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 1   ***Feature Information for Multi-Level Priority Queues***

Feature Name	Releases	Feature Information
Multi-Level Priority Queue	Cisco IOS Release XE 2.1 Cisco IOS Release XE 3.7S	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. 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. in Cisco IOS XE Release 2.1, this feature was implemented on the Cisco ASR 1000 Series Routers. In Cisco IOS XE Release 3.7S, support was added for the Cisco ASR 903 Router. The following commands were introduced or modified: **priority level** , **show policy-map interface**.