Configuring Class Based Weighted Fair Queueing

This chapter describes how to configure Class Based Weighted Fair Queueing policies on the Cisco Nexus 1000V to prioritize network traffic.

Information About Class Based Weighted Fair Queueing

With this feature the following goals have been addressed:

- Queueing can ensure that any traffic class does not starve other traffic types.
- Respect the bandwidth guarantees for each traffic class.
- Optimize the utilization of the uplink bandwidth.

Class-based weighted fair queueing (CBWFQ) extends the standard weighted fair queueing (WFQ) functionality to provide support for user-defined traffic classes. For CBWFQ, you define traffic classes based on match criteria including protocols, and cos values. Packets satisfying the match criteria for a class constitute the traffic for that class. A queue is reserved for each class, and traffic belonging to a class is directed to the queue for that class.

Once a class has been defined according to its match criteria, you can assign its characteristics. To characterize a class, you assign it bandwidth, and maximum queue limit. The bandwidth assigned to a class is the guaranteed bandwidth delivered to the class during congestion.

To characterize a class, you also specify the queue limit for that class, which is the maximum number of packets allowed to accumulate in the queue for the class. Packets belonging to a class are subject to the bandwidth and queue limits that characterize the class.

After a queue has reached its configured queue limit, enqueueing of additional packets to the class causes tail drop.

The traffic that does not match any of the configured classes is given best-effort treatment. Once a packet is classified, all of the standard mechanisms that can also be used to differentiate service among the classes apply.

For CBWFQ, the weight specified for the class becomes the weight of each packet that meets the match criteria of the class. Packets that arrive at the egress interface are classified according to the match criteria filters you define, then each one is assigned the appropriate weight. The weight for a packet belonging to a specific class is derived from the bandwidth you assigned to the class when you configured it; in this sense the weight for a class is user-configurable.
After the weight for a packet is assigned, the packet is enqueued in the appropriate class queue. CBWFQ uses the weights assigned to the queued packets to ensure that the class queue is serviced fairly.

Configuring a class policy—thus, configuring CBWFQ—entails these three processes:

- Defining traffic classes to specify the classification policy (class maps).
  This process determines how many types of packets are to be differentiated from one another.
- Associating policies—that is, class characteristics—with each traffic class (policy maps).
  This process entails configuration of policies to be applied to packets belonging to one of the classes previously defined through a class map. For this process, you configure a policy map that specifies the policy for each traffic class.
- Attaching policies to interfaces (service policies).

Note

A queueing policy map can only be applied on an uplink in the egress (outbound) direction.

This process requires that you associate an existing policy map, or service policy, with an interface to apply the particular set of policies for the map to that interface.

Policy maps prioritize network traffic by class. You create policy maps to define how to treat each class of traffic so that it is prioritized for the best quality of service.

### Information About Class Maps for Queuing

The `match {cos_id} {protocol_name}` command defines whether you want packets for this class map to match protocol, class of service, or both.

- CoS is specified as a number from 0 to 7.
  This matches traffic based on the class of service (CoS) in the IEEE 802.1Q header, defined in IEEE 802.1p. CoS is encoded in the high order 3 bits of the VLAN ID Tag field and is referred to as user_priority.
- The following are pre-defined protocol matches:
  - n1k_control
  - n1k_mgmt
  - n1k_packet
  - vmw_ft
  - vmw_iscsi
  - vmw_mgmt
  - vmw_nfs
  - vmw_vmotion
- The following are the default COS values associated with the protocols:
  - n1k_control: 6
  - n1k_mgmt: No default value
  - n1k_packet: 6

For VMware protocols such as vmw_ft, vmw_iscsi, vmw_mgmt, vmw_nfs, and vmw_vmotion, Cisco Nexus 1000V does not set any default COS value. See the VMware documentation for more information.
Licensing Requirements for Class Based Weighted Fair Queueing

There are no licensing requirements for this feature.

Prerequisites for Class Based Weighted Fair Queueing

Class Based Weighted Fair Queueing has the following prerequisites:

- You are already logged in to the CLI in EXEC mode.

Guidelines and Limitations

Use the following guidelines and limitations when configuring Class Based Weighted Fair Queueing (CBWFQ):

- A queueing policy can only be applied on an uplink interface in the egress (outbound) direction.
- Queueing is supported on ESX/ESXi 4.1.0 and later hosts.
- For port-channel interfaces, queueing bandwidth applies on the member ports. The overall performance will depend on how the vethernets are pinned to member ports and the traffic pattern on the individual ports.
- We recommend that you reserve 10% bandwidth of the uplink for the control traffic.

Default Settings

Class Based Weighted Fair Queueing is disabled by default.

Configuring Class Based Weighted Fair Queueing Policies

This section describes how to create QoS queueing class policies for the Cisco Nexus 1000V:

Creating the policy consists of the following steps:

1. Create a class map
2. Create a policy map
3. Apply the policy map on an interface

Configuring Class Based Weighted Fair Queueing

This feature allows the user to differentiate traffic classes and provide appropriate bandwidth guarantees.

You can use this procedure to configure class-based weighted fair queueing as follows:

- Create a queueing class map with protocol or CoS matching criteria.
- Create a queueing policy map and assign the class map to it.
BEFORE YOU BEGIN

Before beginning this procedure, you must know or do the following:

- You are logged in to the CLI in EXEC mode.
- You know whether you want the queuing class map to match protocol or CoS.
- You know whether you want to designate a minimum guaranteed bandwidth for the traffic class.
- You know whether you want to designate a maximum queue size for the traffic class.

SUMMARY STEPS

1. configure terminal
2. class-map type queuing {match-all | match-any} map-name
3. match {cos id | protocol name}
4. exit
5. policy-map type queuing name
6. class type queuing name
7. bandwidth percent percentage
8. (Optional) show policy-map [{[type qos] [pmap-name-qos]] | {type queuing [pmap-name-que]}]
9. port-profile type ethernet name
10. service-policy type queuing output name
11. show policy-map interface
12. copy running-config startup-config

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td>Example:</td>
<td>n1000v# config t</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>class-map type queuing {match-any</td>
</tr>
<tr>
<td>Example:</td>
<td>n1000v(config)# class-map type queuing match-all class_fin1</td>
</tr>
<tr>
<td>match-any: Use this option to apply this class map to a packet if it matches any of the matching criteria.</td>
<td></td>
</tr>
<tr>
<td>match-all: Use this option to apply this class map to a packet if it matches all of the matching criteria.</td>
<td></td>
</tr>
<tr>
<td>map-name: up to 40 alphanumeric characters in length and can include the hyphen and underscore characters.</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring Class Based Weighted Fair Queueing Policies

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
</tbody>
</table>
| `match (cos id) | (protocol name)` | Defines whether you want packets for this class map to match protocol, class of service, or both.  
  - CoS is specified as a number from 0 to 7.  
  - This matches traffic based on the class of service (CoS) in the IEEE 802.1Q header, defined in IEEE 802.1p. CoS is encoded in the high order 3 bits of the VLAN ID Tag field and is referred to as `user_priority`.  
  - The following are pre-defined protocol matches:  
    - `n1k_control`  
    - `n1k_mgmt`  
    - `n1k_packet`  
    - `vmw_ft`  
    - `vmw_iscsi`  
    - `vmw_mgmt`  
    - `vmw_nfs`  
    - `vmw_vmotion`  
  - The following are the default COS values associated with the protocols:  
    - `n1k_control`: 6  
    - `n1k_mgmt`: No default value  
    - `n1k_packet`: 6  
    - For VMware protocols such as `vmw_ft`, `vmw_iscsi`, `vmw_mgmt`, `vmw_nfs`, and `vmw_vmotion`, Cisco Nexus 1000V does not set any default COS value. See the VMware documentation for more information.  
| Example: | n1000v(config-cmap-que)# match protocol vmw_vmotion  
n1000v(config-cmap-que)# |
| **Step 4** |  |
| `exit` | Exits class-map queueing configuration mode and returns you to global configuration mode.  
| Example: | n1000v(config-cmap-que)# exit  
n1000v(config)# |
| **Step 5** |  |
| `policy-map type queuing name` | Creates a CBWFQ policy map and enters policy map queueing mode for configuring the new policy map.  
| Example: | n1000v(config)# policy-map type queuing Policy-vmotion  
n1000v(config-pmap-que) |
| **Step 6** |  |
| `class type queuing name` | Assigns a CBWFQ class to this policy map and enters policy map class queueing configuration mode.  
| Example: | n1000v(config-pmap-que)# class type queuing class_fin1  
n1000v(config-pmap-c-que)# |
### Configuring Class Based Weighted Fair Queueing Policies

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 7</strong></td>
<td><strong>Command</strong> bandwidth percent <em>percentage</em>  &lt;br&gt; <strong>Example:</strong> n1000v(config-pmap-c-que)# bandwidth percent 50 n1000v(config-pmap-c-que)#  &lt;br&gt; <strong>Designates the minimum guaranteed bandwidth for this traffic class as a percentage of total available bandwidth.</strong></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>**(Optional) show policy-map [([type qos] [pmap-name-qos])</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td><strong>exit</strong>  &lt;br&gt; <strong>Example:</strong> n1000v(config-pmap-c-que)# exit n1000v(config-pmap-que)#  &lt;br&gt; <strong>Exits policy-map type queueing configuration mode and returns you to policy-map configuration mode.</strong></td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td><strong>exit</strong>  &lt;br&gt; <strong>Example:</strong> n1000v(config-pmap-que)# exit n1000v(config)#  &lt;br&gt; <strong>Exits policy-map configuration mode and returns you to global configuration mode.</strong></td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td><strong>port-profile type ethernet name</strong>  &lt;br&gt; <strong>Example:</strong> n1000v(config)# port-profile type ethernet myppte n1000v(config-port-prof)#  &lt;br&gt; <strong>Creates a port-profile and enters port-profile configuration mode.</strong></td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td><strong>service-policy type queuing output name</strong>  &lt;br&gt; <strong>Example:</strong> n1000v(config-port-prof)# service-policy type queuing output my_pmtq n1000v(config-port-prof)#  &lt;br&gt; <strong>Creates a service-policy of type queueing.</strong></td>
</tr>
</tbody>
</table>
Verifying the Class Based Weighted Fair Queueing Configuration

To verify the QoS queueing policy configuration, perform one of the following tasks:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show policy-map interface</code></td>
<td>Displays the status of the global statistics and the configured policy maps on all interfaces.</td>
</tr>
<tr>
<td><code>copy running-config startup-config</code></td>
<td>(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.</td>
</tr>
</tbody>
</table>

**Step 13**

```
n1000v# show policy-map interface
n1000v(config-port-prof)  
```

**Step 14**

```
n1000v# copy running-config startup-config
n1000v(config-port-prof)# 
```

---

**Command**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show policy-map type queuing name</code></td>
<td>Displays the queueing policy map configuration. See Example 6-1 Display Policy Map Type Queuing, page 6-7</td>
</tr>
<tr>
<td><code>show class-map type queuing name</code></td>
<td>Displays the queueing class map configuration. See Example 6-2 Display Class Map, page 6-7</td>
</tr>
<tr>
<td><code>show policy-map interface</code></td>
<td>Displays the policy map interface configuration. See Example 6-3 Display Policy Map on Interface, page 6-8</td>
</tr>
<tr>
<td><code>show running-config ipqos</code></td>
<td>Displays the QoS running configuration.</td>
</tr>
</tbody>
</table>

**Example 6-1 Display Policy Map Type Queuing**

```
n1000v# show policy-map type queuing Policy-vmotion
```

Type queueing policy-maps

```
policy-map type queuing Policy-vmotion
class type queuing Match-vmotion
bandwidth percent 50
```

**Example 6-2 Display Class Map**

```
n1000v# show class-map type queuing Match-vmotion
```

Type queueing class-maps

```
```
Chapter 6  Configuring Class Based Weighted Fair Queueing

Verifying the Class Based Weighted Fair Queueing Configuration

Example 6-3  Display Policy Map on Interface

n1000v# show policy-map interface ethernet 3/3

Global statistics status :  disabled
Ethernet3/3

Service-policy (queuing) output:  Policy-vmotion
policy statistics status:   enabled

Class-map (queuing):  Match-vmotion (match-any)
Match: protocol vmw_vmotion
bandwidth percent 50

Example 6-4  Display Class Map

n1000v# show class-map type queuing Match-Cos

Type queuing class-maps
=================================

class-map type queuing match-all Match-Cos
match cos 5

Example 6-5  Display Service Policy on Interface

n1000v# show policy-map interface ethernet 3/3

Global statistics status :  disabled
Ethernet3/3

Service-policy (queuing) output:  Policy-Cos
policy statistics status:   enabled

Class-map (queuing):  Match-Cos (match-all)
Match: cos 5
bandwidth percent 50

Example 6-6  Display Service Policy on Interface

n1000v# show policy-map interface port-channel 1

Global statistics status :  disabled
port-channel1

Service-policy (queuing) output:  Policy-vmotion
policy statistics status:   enabled
Configuration Examples for Class-Based Weighted Fair Queueing

Example 6-7  Allocating 50% of Bandwidth for vMotion Traffic

conf t
Enter configuration commands, one per line.  End with CNTL/Z.
class-map type queuing match-any Match-vmotion
match protocol vmw_vmotion
exit
policy-map type queuing Policy-vmotion
class type queuing Match-vmotion
bandwidth percent 50
exit
exit
interface ethernet 3/3
service-policy type queuing output Policy-vmotion

Example 6-8  Allocating 50% of Bandwidth for Traffic with CoS Value of 5

conf t
Enter configuration commands, one per line.  End with CNTL/Z.
class-map type queuing match-all Match-Cos
match cos 5
exit
policy-map type queuing Policy-Cos
class type queuing Match-Cos
bandwidth percent 50
exit
exit
interface ethernet 3/3
service-policy type queuing output Policy-Cos

Example 6-9  Policy-Map with Multiple Traffic Classes

conf t
Enter configuration commands, one per line.  End with CNTL/Z.
class-map type queuing match-any class-vmotion
match protocol vmw_vmotion
exit
class-map type queuing match-any class-cos-2
match cos 2
exit
policy-map type queuing policy-priority-vmotion
class type queuing class-vmotion
bandwidth percent 60
class type queuing class-cos-2
bandwidth percent 40
exit
exit
int po1
service-policy type queuing output policy-priority-vmotion
sh policy-map type queuing policy-priority-vmotion
Type queuing policy-maps
========================
policy-map type queuing policy-priority-vmotion
  class type queuing class-vmotion
    bandwidth percent 60
  class type queuing class-cos-2
    bandwidth percent 40

conf t
Enter configuration commands, one per line. End with CNTL/Z.
int po1
  service-policy type queuing output policy-priority-vmotion
end

sh policy-map interface po1

Global statistics status :  disabled
port-channel1

  Service-policy (queuing) output:  policy-priority-vmotion
  policy statistics status:  enabled

  Class-map (queuing):  class-vmotion (match-any)
    Match:  protocol vmw_vmotion
    bandwidth percent 60

  Class-map (queuing):  class-cos-2 (match-any)
    Match:  cos 2
    bandwidth percent 40

Feature History for Class Based Weighted Fair Queueing

This section provides the Class Based Weighted Fair Queueing release history.

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class Based Weighted Fair Queueing</td>
<td>4.2(1)SV1(4)</td>
<td>This feature was introduced.</td>
</tr>
</tbody>
</table>