Configuring Weighted Fair Queueing

Last Updated: December 5, 2011

Feature History

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS</td>
<td>For information about feature support in Cisco IOS software, use Cisco Feature Navigator.</td>
</tr>
</tbody>
</table>

This module describes the tasks for configuring flow-based weighted fair queueing (WFQ), distributed WFQ (DWFQ), and class-based WFQ (CBWFQ), and distributed class-based WFQ (DCBWFQ) and the related features described in the following section, which provide strict priority queueing (PQ) within WFQ or CBWFQ:

- IP RTP Priority Queueing
- Frame Relay IP RTP Priority Queueing
- Frame Relay PVC Interface Priority Queueing
- Low Latency Queueing
- Distributed Low Latency Queueing
- Low Latency Queueing (LLQ) for Frame Relay
- Burst Size in Low Latency Queueing
- Per-VC Hold Queue Support for ATM Adapters

- Finding Feature Information, page 2
- Flow-Based Weighted Fair Queueing Configuration Task List, page 2
- Distributed Weighted Fair Queueing Configuration Task List, page 4
- Class-Based Weighted Fair Queueing Configuration Task List, page 6
- Distributed Class-Based Weighted Fair Queueing Configuration Task List, page 13
- IP RTP Priority Configuration Task List, page 15
- Frame Relay IP RTP Priority Configuration Task List, page 16
- Frame Relay PVC Interface Priority Configuration Task List, page 17
- Low Latency Queueing Configuration Task List, page 19
- Distributed LLQ Configuration Task List, page 20
- Low Latency Queueing for Frame Relay Configuration Task List, page 23
- Configuring Burst Size in LLQ Configuration Task List, page 27
- Per-VC Hold Queue Support for ATM Adapters Configuration Task List, page 28
Flow-Based Weighted Fair Queueing Configuration Task List

WFQ provides traffic priority management that automatically sorts among individual traffic streams without requiring that you first define access lists. WFQ can also manage duplex data streams such as those between pairs of applications, and simplex data streams such as voice or video. There are two categories of WFQ sessions: high bandwidth and low bandwidth. Low-bandwidth traffic has effective priority over high-bandwidth traffic, and high-bandwidth traffic shares the transmission service proportionally according to assigned weights.

When WFQ is enabled for an interface, new messages for high-bandwidth traffic streams are discarded after the configured or default congestive messages threshold has been met. However, low-bandwidth conversations, which include control message conversations, continue to enqueue data. As a result, the fair queue may occasionally contain more messages than its configured threshold number specifies.

With standard WFQ, packets are classified by flow. Packets with the same source IP address, destination IP address, source TCP or User Datagram Protocol (UDP) port, or destination TCP or UDP port belong to the same flow. WFQ allocates an equal share of the bandwidth to each flow. Flow-based WFQ is also called fair queuing because all flows are equally weighted.

The Cisco IOS software provides two forms of flow-based WFQ:

- Standard WFQ, which is enabled by default on all serial interfaces that run at 2 Mbps or below, and can run on all Cisco serial interfaces.
- Distributed WFQ, which runs only on Cisco 7000 series routers with a Route Switch Processor (RSP)-based RSP7000 interface processor or Cisco 7500 series routers with a Versatile Interface Processor (VIP)-based VIP2-40 or greater interface processor. (A VIP2-50 interface processor is strongly
recommended when the aggregate line rate of the port adapters on the VIP is greater than DS3. A VIP2-50 interface processor is required for OC-3 rates.)

Flow-based WFQ uses a traffic data stream discrimination registry service to determine to which traffic stream a message belongs. Refer to the table accompanying the description of the `fair-queue` (WFQ) command in the Cisco IOS Quality of Service Solutions Command Reference for the attributes of a message that are used to classify traffic into data streams.

Defaults are provided for the congestion threshold after which messages for high-bandwidth conversations are dropped, and for the number of dynamic and reservable queues; however, you can fine-tune your network operation by changing these defaults. Refer to the tables accompanying the description of the `fair-queue` (WFQ) command in the Cisco IOS Quality of Service Solutions Command Reference for the default number of dynamic queues that WFQ and CBWFQ use when they are enabled on an interface or ATM VC. These values do not apply for DWFQ.

**Note**

WFQ is the default queueing mode on interfaces that run at E1 speeds (2.048 Mbps) or below. It is enabled by default for physical interfaces that do not use Link Access Procedure, Balanced (LAPB), X.25, or Synchronous Data Link Control (SDLC) encapsulations. WFQ is not an option for these protocols. WFQ is also enabled by default on interfaces configured for Multilink PPP (MLP). However, if custom queueing (CQ) or priority queueing (PQ) is enabled for a qualifying link, it overrides fair queueing, effectively disabling it. Additionally, WFQ is automatically disabled if you enable autonomous or silicon switching.

If you enable flow-based DWFQ and then enable class-based DWFQ (either QoS-group based or ToS-based), class-based DWFQ will replace flow-based DWFQ.

If you enable class-based DWFQ and then want to switch to flow-based DWFQ, you must disable class-based DWFQ using the `no fair-queue class-based` command before enabling flow-based DWFQ.

If you enable one type of class-based DWFQ and then enable the other type, the second type will replace the first.

DWFQ runs only on Cisco 7000 series routers with an RSP-based RSP7000 interface processor or Cisco 7500 series routers with a VIP-based VIP2-40 or greater interface processor. (A VIP2-50 interface processor is strongly recommended when the aggregate line rate of the port adapters on the VIP is greater than DS3. A VIP2-50 interface processor is required for OC-3 rates.)

DWFQ can be configured on interfaces but not subinterfaces. It is not supported on Fast EtherChannel, tunnel, or other logical or virtual interfaces such as MLP.

For flow-based DWFQ, packets are classified by flow. Packets with the same source IP address, destination IP address, source TCP or UDP port, destination TCP or UDP port, and protocol belong to the same flow.

To configure flow-based WFQ, perform the tasks described in the following sections.

Flow-based WFQ is supported on unavailable bit rate (UBR), variable bit rate (VBR), and available bit rate (ABR) ATM connections.

- [Configuring WFQ](#)
- [Monitoring Fair Queueing](#)
Configuring WFQ

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config-if)# fair-queue {congestive-discord-threshold {dynamic-queues {reservable-queues}}}}</td>
<td>Configures an interface to use WFQ.</td>
</tr>
</tbody>
</table>

Monitoring Fair Queueing

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# show interfaces [interface]</td>
<td>Displays statistical information specific to an interface.</td>
</tr>
<tr>
<td>Router# show queue interface-type interface-number</td>
<td>Displays the contents of packets inside a queue for a particular interface or virtual circuit (VC).</td>
</tr>
<tr>
<td>Router# show queueing fair</td>
<td>Displays status of the fair queueing configuration.</td>
</tr>
</tbody>
</table>

Distributed Weighted Fair Queueing Configuration Task List

To configure DWFQ, perform one of the mutually exclusive tasks described in the following sections:

- Configuring Flow-Based DWFQ, page 4
- Configuring QoS-Group-Based DWFQ, page 5
- Configuring Type of Service-Based DWFQ, page 6
- Monitoring DWFQ, page 6

Configuring Flow-Based DWFQ

**SUMMARY STEPS**

1. Router(config-if)# fair-queue
2. Router(config-if)# fair-queue aggregate-limit aggregate-packet
3. Router(config-if)# fair-queue individual-limit individual-packet

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 Router(config-if)# fair-queue</td>
<td>Enables flow-based DWFQ.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Router(config-if)# <code>fair-queue aggregate-limit aggregate-packet</code></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Router(config-if)# <code>fair-queue individual-limit individual-packet</code></td>
</tr>
</tbody>
</table>

### Configuring QoS-Group-Based DWFQ

**SUMMARY STEPS**

1. Router(config-if)# `fair-queue qos-group`
2. Router(config-if)# `fair-queue qos-group number weight weight`
3. Router(config-if)# `fair-queue aggregate-limit aggregate-packet`
4. Router(config-if)# `fair-queue individual-limit individual-packet`
5. Router(config-if)# `fair-queue qos-group number limit class-packet`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Router(config-if)# <code>fair-queue qos-group</code></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Router(config-if)# <code>fair-queue qos-group number weight weight</code></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Router(config-if)# <code>fair-queue aggregate-limit aggregate-packet</code></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Router(config-if)# <code>fair-queue individual-limit individual-packet</code></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Router(config-if)# <code>fair-queue qos-group number limit class-packet</code></td>
</tr>
</tbody>
</table>
Configuring Type of Service-Based DWFQ

**SUMMARY STEPS**

1. Router(config-if)# fair-queue tos
2. Router(config-if)# fair-queue tos number weight weight
3. Router(config-if)# fair-queue aggregate-limit aggregate-packet
4. Router(config-if)# fair-queue individual-limit individual-packet
5. Router(config-if)# fair-queue tos number limit class-packet

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> Router(config-if)# fair-queue tos</td>
<td>Enables ToS-based DWFQ</td>
</tr>
<tr>
<td><strong>Step 2</strong> Router(config-if)# fair-queue tos number weight weight</td>
<td>(Optional) For each ToS class, specifies the percentage of the bandwidth to be allocated to each class.</td>
</tr>
<tr>
<td><strong>Step 3</strong> Router(config-if)# fair-queue aggregate-limit aggregate-packet</td>
<td>(Optional) Sets the total number of buffered packets before some packets may be dropped. Below this limit, packets will not be dropped. <strong>Note</strong> In general, you should not change the aggregate, individual, or class limit value from the default. Use the fair-queue aggregate-limit, fair-queue individual-limit, and fair-queue limit commands only if you have determined that you would benefit from using different values, based on your particular situation.</td>
</tr>
<tr>
<td><strong>Step 4</strong> Router(config-if)# fair-queue individual-limit individual-packet</td>
<td>(Optional) Sets the maximum queue size for every per-flow queue during periods of congestion.</td>
</tr>
<tr>
<td><strong>Step 5</strong> Router(config-if)# fair-queue tos number limit class-packet</td>
<td>(Optional) Sets the maximum queue size for a specific ToS queue during periods of congestion.</td>
</tr>
</tbody>
</table>

**Monitoring DWFQ**

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# show interfaces [interface]</td>
<td>Displays the statistical information specific to an interface.</td>
</tr>
<tr>
<td>Router# show queueing fair-queue</td>
<td>Displays status of the fair queueing configuration.</td>
</tr>
</tbody>
</table>

**Class-Based Weighted Fair Queueing Configuration Task List**

CBWFQ is supported on VBR and ABR ATM connections. It is not supported on UBR connections.

- Defining Class Maps, page 7
- Configuring Class Policy in the Policy Map, page 7
Defining Class Maps

**SUMMARY STEPS**

1. **Step 1**
   - Router(config)# `class-map class-map-name`

2. **Step 2**
   - Do one of the following:
     - Router(config-cmap)# `match access-group {access-group|name access-group-name}`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> Router(config)# <code>class-map class-map-name</code></td>
<td>Specifies the name of the class map to be created.</td>
</tr>
<tr>
<td><strong>Step 2</strong> Do one of the following:</td>
<td></td>
</tr>
<tr>
<td>• Router(config-cmap)# `match access-group {access-group</td>
<td>name access-group-name}`</td>
</tr>
</tbody>
</table>

**Example:**

- Router(config-cmap)# `match input-interface interface-name`
- Router(config-cmap)# `match protocol protocol`
- Router(config-cmap)# `match mpls experimental number`

**Configuring Class Policy in the Policy Map**

To configure a policy map and create class policies that make up the service policy, use the `policy-map` command to specify the policy map name, then use one or more of the following commands to configure policy for a standard class or the default class:

- `class`
- `bandwidth` (policy-map class)
• fair-queue (for class-default class only)
• queue-limit or random-detect

For each class that you define, you can use one or more of the listed commands to configure class policy. For example, you might specify bandwidth for one class and both bandwidth and queue limit for another class.

The default class of the policy map (commonly known as the class-default class) is the class to which traffic is directed if that traffic does not satisfy the match criteria of other classes whose policy is defined in the policy map.

You can configure class policies for as many classes as are defined on the router, up to the maximum of 64. However, the total amount of bandwidth allocated for all classes included in a policy map must not exceed 75 percent of the available bandwidth on the interface. The other 25 percent is used for control and routing traffic. (To override the 75 percent limitation, use the max-reserved bandwidth command.) If not all of the bandwidth is allocated, the remaining bandwidth is proportionally allocated among the classes, based on their configured bandwidth.

The class-default class is used to classify traffic that does not fall into one of the defined classes. Once a packet is classified, all of the standard mechanisms that can be used to differentiate service among the classes apply. The class-default class was predefined when you created the policy map, but you must configure it. If no default class is configured, then by default the traffic that does not match any of the configured classes is flow classified and given best-effort treatment.

To configure class policies in a policy map, perform the optional tasks described in the following sections. If you do not perform the steps in these sections, the default actions are used.

- Configuring Class Policy Using Tail Drop, page 8
- Configuring Class Policy Using WRED Packet Drop, page 9
- Configuring the Class-Default Class Policy for WFQ, page 10

Configuring Class Policy Using Tail Drop

SUMMARY STEPS

1. Router(config)# policy-map policy-map
2. Router(config-pmap)# class class-name
3. Router(config-pmap-c)# bandwidth {bandwidth-kbps | percent percent}
4. Router(config-pmap-c)# queue-limit number-of-packets

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 Router(config)# policy-map policy-map</td>
<td>Specifies the name of the policy map to be created or modified.</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Steps</strong></td>
</tr>
<tr>
<td><strong>Step 2</strong> Router(config-pmap)# <em>class</em> class-name</td>
</tr>
<tr>
<td><strong>Step 3</strong> Router(config-pmap-c)# <em>bandwidth</em> *(bandwidth-kbps</td>
</tr>
<tr>
<td><strong>Step 4</strong> Router(config-pmap-c)# <em>queue-limit</em> number-of-packets</td>
</tr>
</tbody>
</table>

### Configuring Class Policy Using WRED Packet Drop

**SUMMARY STEPS**

1. Router(config)# *policy-map* policy-map
2. Router(config-pmap)# *class* class-name
3. Router(config-pmap-c)# *bandwidth* *(bandwidth-kbps | percent)* percent
4. Router(config-pmap-c)# *random-detect*
5. Do one of the following:
   - Router(config-pmap-c)# *random-detect exponential-weighting-constant* exponent

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Steps</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Step 1</strong> Router(config)# <em>policy-map</em> policy-map</td>
<td>Specifies the name of the policy map to be created or modified.</td>
</tr>
<tr>
<td><strong>Step 2</strong> Router(config-pmap)# <em>class</em> class-name</td>
<td>Specifies the name of a class to be created and included in the service policy. <strong>Note</strong> To configure policy for more than one class in the same policy map, repeat Configuring Class Policy Using WRED Packet Drop, page 9 through Configuring Class Policy Using WRED Packet Drop, page 9. Note that this set of commands uses WRED packet drop, not tail drop.</td>
</tr>
<tr>
<td><strong>Step 3</strong> Router(config-pmap-c)# <em>bandwidth</em> *(bandwidth-kbps</td>
<td>percent)* percent</td>
</tr>
</tbody>
</table>
### Configuring the Class-Default Class Policy for WFQ

#### SUMMARY STEPS
1. Router(config)# `policy-map policy-map`
2. Router(config-pmap)# `class class-default default-class-name`
3. Do one of the following:
   - Router(config-pmap-c)# `bandwidth {bandwidth-kbps | percent percent}`
4. Router(config-pmap-c)# `queue-limit number-of-packets`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> Router(config)# <code>policy-map policy-map</code></td>
<td>Specifies the name of the policy map to be created or modified.</td>
</tr>
<tr>
<td><strong>Step 2</strong> Router(config-pmap)# <code>class class-default default-class-name</code></td>
<td>Specifies the default class so that you can configure or modify its policy.</td>
</tr>
</tbody>
</table>

### Step 4

**Command or Action:**
Router(config-pmap-c)# `random-detect`

**Purpose:**
Enables WRED. The class policy will drop packets using WRED instead of tail drop.

**Note:** If you configure a class in a policy map to use WRED for packet drop instead of tail drop, you must ensure that WRED is not configured on the interface to which you intend to attach that service policy.

### Step 5

**Command or Action:**
- Router(config-pmap-c)# `random-detect exponential-weighting-constant exponent`
- Example:
  Router(config-pmap-c)# `random-detect precedence precedence min-threshold max-threshold mark-prob-denominator`

**Purpose:**
- Configures the exponential weight factor used in calculating the average queue length.
- Configures WRED parameters for packets with a specific IP precedence. Repeat this command for each precedence.
Attaching the Service Policy and Enabling CBWFQ

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 3</strong> Do one of the following:</td>
<td>Specifies the maximum number of packets that the queue for the default class can accumulate.</td>
</tr>
<tr>
<td>• Router(config-pmap-c)# <strong>bandwidth</strong> {<strong>bandwidth-kbps</strong> (</td>
<td>) <strong>percent</strong> ( ) <strong>percent</strong>}</td>
</tr>
<tr>
<td>Example: Router(config-pmap-c)# <strong>fair-queue</strong> ( [) <strong>number-of-dynamic-queues]</strong></td>
<td></td>
</tr>
</tbody>
</table>

Step 4 Router(config-pmap-c)# **queue-limit** **number-of-packets**

Modifying the Bandwidth for an Existing Policy Map Class

**SUMMARY STEPS**
1. Router(config)# **policy-map** **policy-map**
2. Router(config-pmap)# **class** **class-name**
3. Router(config-pmap-c)# **bandwidth** {**bandwidth-kbps** \( | \) **percent** **percent**}

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> Router(config)# <strong>policy-map</strong> <strong>policy-map</strong></td>
<td>Specifies the name of the policy map containing the class to be modified.</td>
</tr>
</tbody>
</table>

**Attaching the Service Policy and Enabling CBWFQ**

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config-if)# <strong>service-policy</strong> output <strong>policy-map</strong></td>
<td>Enables CBWFQ and attaches the specified service policy map to the output interface. <strong>Note</strong> Configuring CBWFQ on a physical interface is only possible if the interface is in the default queueing mode. Serial interfaces at E1 (2.048 Mbps) and below use WFQ by default--other interfaces use FIFO by default. Enabling CBWFQ on a physical interface overrides the default interface queueing method. Enabling CBWFQ on an ATM permanent virtual circuit (PVC) does not override the default queueing method.</td>
</tr>
</tbody>
</table>
### Modifying the Queue Limit for an Existing Policy Map Class

**SUMMARY STEPS**

1. Router(config)# policy-map policy-map
2. Router(config-pmap)# class class-name
3. Router(config-pmap-c)# queue-limit number-of-packets

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> Router(config)# policy-map policy-map</td>
<td>Specifies the name of the policy map containing the class to be modified.</td>
</tr>
<tr>
<td><strong>Step 2</strong> Router(config-pmap)# class class-name</td>
<td>Specifies the name of a class whose queue limit you want to modify.</td>
</tr>
<tr>
<td><strong>Step 3</strong> Router(config-pmap-c)# queue-limit number-of-packets</td>
<td>Specifies the new maximum number of packets that can be queued for the class to be reconfigured. The default and maximum number of packets is 64.</td>
</tr>
</tbody>
</table>

### Deleting Class Maps From Service Policy Maps

**SUMMARY STEPS**

1. Router(config)# policy-map policy-map
2. Router(config-pmap)# no class class-name
3. Router(config-pmap-c)# no class class-default

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> Router(config)# policy-map policy-map</td>
<td>Specifies the name of the policy map containing the classes to be deleted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> Router(config-pmap)# no class class-name</td>
<td>Specifies the name of the classes to be deleted.</td>
</tr>
<tr>
<td><strong>Step 3</strong> Router(config-pmap-c)# no class class-default</td>
<td>Deletes the default class.</td>
</tr>
</tbody>
</table>
Deleting Policy Maps

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config)# no policy-map policy-map</td>
<td>Specifies the name of the policy map to be deleted.</td>
</tr>
</tbody>
</table>

Verifying Configuration of Policy Maps and Their Classes

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# show policy-map policy-map</td>
<td>Displays the configuration of all classes that make up the specified policy map.</td>
</tr>
<tr>
<td>Router# show policy-map policy-map class class-name</td>
<td>Displays the configuration of the specified class of the specified policy map.</td>
</tr>
<tr>
<td>Router# show policy-map interface interface-name</td>
<td>Displays the configuration of all classes configured for all policy maps on the specified interface.</td>
</tr>
<tr>
<td>Note</td>
<td>The counters displayed after issuing the <code>show policy-map interface</code> command are updated only if congestion is present on the interface.</td>
</tr>
<tr>
<td>Router# show queue interface-type interface-number</td>
<td>Displays queueing configuration and statistics for a particular interface.</td>
</tr>
</tbody>
</table>

Distributed Class-Based Weighted Fair Queueing Configuration Task List

To configure DCBWFQ, perform the tasks described in the following sections. Although all the tasks are listed as optional, you must complete the task in either the first or second section.

DCBWFQ is configured using user-defined traffic classes and service policies. Traffic classes and service policies are configured using the Modular Quality of Service Command-Line Interface (CLI) feature.

- Modifying the Bandwidth for an Existing Traffic Class, page 14
- Modifying the Queue Limit for an Existing Traffic Class, page 14
- Monitoring and Maintaining DCBWFQ, page 15
Modifying the Bandwidth for an Existing Traffic Class

**SUMMARY STEPS**

1. Router(config)# policy-map policy-map
2. Router(config-pmap)# class class-name
3. Router(config-pmap-c)# bandwidth bandwidth-kbps

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Router(config)# policy-map policy-map</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Router(config-pmap)# class class-name</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Router(config-pmap-c)# bandwidth bandwidth-kbps</td>
</tr>
</tbody>
</table>

**Note** After configuring the traffic policy with the `policy-map` command, you must still attach the traffic policy to an interface before it is successfully enabled. For information on attaching a traffic policy to an interface, see the "Applying QoS Features Using the MQC" module.

Modifying the Queue Limit for an Existing Traffic Class

**SUMMARY STEPS**

1. Router(config)# policy-map policy-map
2. Router(config-pmap)# class class-name
3. Router(config-pmap-c)# queue-limit number-of-packets

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Router(config)# policy-map policy-map</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Router(config-pmap)# class class-name</td>
</tr>
</tbody>
</table>
Step 3  Router(config-pmap-c)# queue-limit number-of-packets

Specifies the new maximum number of packets that can be queued for the traffic class to be reconfigured. The default and maximum number of packets is 64.

Note  After configuring the service policy with the policy-map command, you must still attach the traffic policy to an interface before it is successfully enabled. For information on attaching a traffic policy to an interface, see the "Applying QoS Features Using the MQC" module.

---

**Monitoring and Maintaining DCBWFQ**

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# show policy-map</td>
<td>Displays all configured traffic policies.</td>
</tr>
<tr>
<td>Router# show policy-map policy-map-name</td>
<td>Displays the user-specified traffic policy.</td>
</tr>
<tr>
<td>Router# show policy-map interface</td>
<td>Displays statistics and configurations of all input and output policies attached to an interface.</td>
</tr>
<tr>
<td>Router# show policy-map interface interface-spec</td>
<td>Displays configuration and statistics of the input and output policies attached to a particular interface.</td>
</tr>
<tr>
<td>Router# show policy-map interface interface-spec input</td>
<td>Displays configuration and statistics of the input policy attached to an interface.</td>
</tr>
<tr>
<td>Router# show policy-map interface interface-spec output</td>
<td>Displays configuration statistics of the output policy attached to an interface.</td>
</tr>
<tr>
<td>Router# show policy-map interface interface-spec [class class-name]</td>
<td>Displays the configuration and statistics for the class name configured in the policy.</td>
</tr>
</tbody>
</table>

---

**IP RTP Priority Configuration Task List**

Frame Relay Traffic Shaping (FRTS) and Frame Relay Fragmentation (FRF.12 or higher) must be configured before the Frame Relay IP RTP Priority feature is used.
## Configuring IP RTP Priority

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config-if)# ip rtp priority starting-rtp-port-number port-number-range bandwidth</td>
<td>Reserves a strict priority queue for a set of RTP packet flows belonging to a range of UDP destination ports.</td>
</tr>
</tbody>
</table>

**Note**  The `ip rtp reserve` and `ip rtp priority` commands cannot be configured on the same interface.

**Caution** Because the `ip rtp priority` command gives absolute priority over other traffic, it should be used with care. In the event of congestion, if the traffic exceeds the configured bandwidth, then all the excess traffic is dropped.

## Verifying IP RTP Priority

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# show queue interface-type interface-number</td>
<td>Displays queueing configuration and statistics for a particular interface.</td>
</tr>
</tbody>
</table>

## Monitoring and Maintaining IP RTP Priority

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# debug priority</td>
<td>Displays priority queueing output if packets are dropped from the priority queue.</td>
</tr>
<tr>
<td>Router# show queue interface-type interface-number</td>
<td>Displays queueing configuration and statistics for a particular interface.</td>
</tr>
</tbody>
</table>

### Frame Relay IP RTP Priority Configuration Task List

- Configuring Frame Relay IP RTP Priority, page 17
- Verifying Frame Relay IP RTP Priority, page 17
- Monitoring and Maintaining Frame Relay IP RTP Priority, page 17
Configuring Frame Relay IP RTP Priority

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config-map-class)# frame-relay ip rtp priority starting-rtp-port-number port-number-range bandwidth</td>
<td>Reserves a strict priority queue for a set of RTP packet flows belonging to a range of UDP destination ports.</td>
</tr>
</tbody>
</table>

**Note** Because the `frame-relay ip rtp priority` command gives absolute priority over other traffic, it should be used with care. In the event of congestion, if the traffic exceeds the configured bandwidth, then all the excess traffic is dropped.

Verifying Frame Relay IP RTP Priority

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# show frame relay pvc</td>
<td>Displays statistics about PVCs for Frame Relay interfaces.</td>
</tr>
<tr>
<td>Router# show queue interface-type interface-number</td>
<td>Displays fair queueing configuration and statistics for a particular interface.</td>
</tr>
<tr>
<td>Router# show traffic-shape queue</td>
<td>Displays information about the elements queued at a particular time at the VC data-link connection identifier (DLCI) level.</td>
</tr>
</tbody>
</table>

Monitoring and Maintaining Frame Relay IP RTP Priority

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# debug priority</td>
<td>Displays priority queueing output if packets are dropped from the priority queue.</td>
</tr>
</tbody>
</table>

Frame Relay PVC Interface Priority Configuration Task List

- Configuring PVC Priority in a Map Class, page 18
- Enabling Frame Relay PIPQ and Setting Queue Limits, page 18
- Assigning a Map Class to a PVC, page 18
- Verifying Frame Relay PIPQ, page 19
- Monitoring and Maintaining Frame Relay PIPQ, page 19
Configuring PVC Priority in a Map Class

SUMMARY STEPS

1. Router(config)# map-class frame-relay map-class-name
2. Router(config-map-class)# frame-relay interface-queue priority {high | medium | normal | low}

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 Router(config)# map-class frame-relay map-class-name</td>
<td>Specifies a Frame Relay map class.</td>
</tr>
<tr>
<td>Step 2 Router(config-map-class)# frame-relay interface-queue priority {high</td>
<td>medium</td>
</tr>
</tbody>
</table>

Enabling Frame Relay PIPQ and Setting Queue Limits

SUMMARY STEPS

1. Router(config)# interface type number [name-tag]
2. Router(config-if)# encapsulation frame-relay [cisco | ietf]
3. Router(config-if)# frame-relay interface-queue priority [high-limit medium-limit normal-limit low-limit]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 Router(config)# interface type number [name-tag]</td>
<td>Configures an interface type and enters interface configuration mode.</td>
</tr>
<tr>
<td>Step 2 Router(config-if)# encapsulation frame-relay [cisco</td>
<td>ietf]</td>
</tr>
<tr>
<td>Step 3 Router(config-if)# frame-relay interface-queue priority [high-limit medium-limit normal-limit low-limit]</td>
<td>Enables Frame Relay PIPQ and sets the priority queue limits.</td>
</tr>
</tbody>
</table>

Assigning a Map Class to a PVC

SUMMARY STEPS

1. Router(config-if)# frame-relay interface-dlci dlci
2. Router(config-fr-dlci)# class map-class-name
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1&lt;br&gt;Router(config-if)# frame-relay interface-dlci dlci</td>
<td>Specifies a single PVC on a Frame Relay interface.</td>
</tr>
<tr>
<td>Step 2&lt;br&gt;Router(config-fr-dlci)# class map-class-name</td>
<td>Associates a map class with a specified PVC.</td>
</tr>
</tbody>
</table>

## Verifying Frame Relay PIPQ

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# show frame-relay pvc {interface interface}{dlci}</td>
<td>Displays statistics about PVCs for Frame Relay interfaces.</td>
</tr>
<tr>
<td>Router# show interfaces {type number}{first}{last}</td>
<td>Displays the statistical information specific to a serial interface.</td>
</tr>
<tr>
<td>Router# show queueing {custom</td>
<td>fair</td>
</tr>
</tbody>
</table>

## Monitoring and Maintaining Frame Relay PIPQ

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# debug priority</td>
<td>Displays priority queueing output if packets are dropped from the priority queue.</td>
</tr>
<tr>
<td>Router# show frame-relay pvc {interface interface}{dlci}</td>
<td>Displays statistics about PVCs for Frame Relay interfaces.</td>
</tr>
<tr>
<td>Router# show interfaces {type number}{first}{last}</td>
<td>Displays the statistical information specific to a serial interface.</td>
</tr>
<tr>
<td>Router# show queue {interface-name interface-number}{vc}{vpi/vci}{queue-number}</td>
<td>Displays the contents of packets inside a queue for a particular interface or VC.</td>
</tr>
<tr>
<td>Router# show queueing {custom</td>
<td>fair</td>
</tr>
</tbody>
</table>

## Low Latency Queueing Configuration Task List

- Configuring LLQ, page 20
Configuring LLQ

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config-pmap-c)# priority bandwidth</td>
<td>Reserves a strict priority queue for this class of traffic.</td>
</tr>
</tbody>
</table>

Verifying LLQ

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# show queue interface-type interface-number</td>
<td>Displays queueing configuration and statistics for a particular interface.</td>
</tr>
</tbody>
</table>

Monitoring and Maintaining LLQ

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# debug priority</td>
<td>Displays priority queueing output if packets are dropped from the priority queue.</td>
</tr>
<tr>
<td>Router# show queue interface-type interface-number</td>
<td>Displays queueing configuration and statistics for a particular interface.</td>
</tr>
<tr>
<td>Router# show policy-map interface interface-name</td>
<td>Displays the configuration of all classes configured for all traffic policies on the specified interface. Displays if packets and bytes were discarded or dropped for the priority class in the traffic policy attached to the interface.</td>
</tr>
</tbody>
</table>

Distributed LLQ Configuration Task List

- Configuring a Priority Queue for an Amount of Available Bandwidth, page 21
- Configuring a Priority Queue for a Percentage of Available Bandwidth, page 21
- Configuring a Transmission Ring Limit on an ATM PVC, page 22
- Configuring a Transmission Ring Limit on an ATM Subinterface, page 22
- Verifying Distributed LLQ, page 23
- Verifying a Transmission Ring Limit, page 23
- Monitoring and Maintaining Distributed LLQ, page 23
Configuring a Priority Queue for an Amount of Available Bandwidth

SUMMARY STEPS

1. Router(config)# policy-map policy-name
2. Router(config-pmap)# class class-name
3. Router(config-pmap-c)# priority kpbs |bytes

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 Router(config)# policy-map policy-name</td>
<td>Specifies the name of the policy map to configure. Enters policy-map configuration mode.</td>
</tr>
<tr>
<td>Step 2 Router(config-pmap)# class class-name</td>
<td>Specifies the name of a predefined class included in the service policy. Enters policy-map class configuration mode.</td>
</tr>
<tr>
<td>Step 3 Router(config-pmap-c)# priority kpbs</td>
<td>Reserves a priority queue with a specified amount of available bandwidth for CBWFQ traffic.</td>
</tr>
</tbody>
</table>

Note: The traffic policy configured in this section is not yet attached to an interface. For information on attaching a traffic policy to an interface, see the "Applying QoS Features Using the MQC" module.

Configuring a Priority Queue for a Percentage of Available Bandwidth

SUMMARY STEPS

1. Router(config)# policy-map policy-name
2. Router(config-pmap)# class class-name
3. Router(config-pmap-c)# priority percent percent

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 Router(config)# policy-map policy-name</td>
<td>Specifies the name of the traffic policy to configure. Enters policy-map configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Step 2 Router(config-pmap)# class class-name</td>
<td>Specifies the name of a predefined class included in the service policy. Enters policy-map class configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
Configuring a Transmission Ring Limit on an ATM PVC

**SUMMARY STEPS**

1. Router(config)# interface atm interface-name
2. Router(config-if)# atm pvc vcd-number vpi-number vci-number Encapsulation-type tx-ring-limit ring-limit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> Router(config)# interface atm interface-name</td>
<td>Specifies the name of the ATM interface to configure.</td>
</tr>
<tr>
<td><strong>Step 2</strong> Router(config-if)# atm pvc vcd-number vpi-number vci-number Encapsulation-type tx-ring-limit ring-limit</td>
<td>Specifies the ATM PVC to configure, the encapsulation type, and the transmission ring limit value.</td>
</tr>
</tbody>
</table>

Configuring a Transmission Ring Limit on an ATM Subinterface

**SUMMARY STEPS**

1. Router(config)# interface atm subinterface name
2. Router(config-subif)# pvc pvc-name
3. Router(config-if-atm-vc)# tx-ring-limit ring-limit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> Router(config)# interface atm subinterface name</td>
<td>Specifies the name of the subinterface to configure.</td>
</tr>
<tr>
<td><strong>Step 2</strong> Router(config-subif)# pvc pvc-name</td>
<td>Specifies the name of the PVC to configure.</td>
</tr>
<tr>
<td><strong>Step 3</strong> Router(config-if-atm-vc)# tx-ring-limit ring-limit</td>
<td>Specifies the transmission ring limit value.</td>
</tr>
</tbody>
</table>
Verifying Distributed LLQ

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# <code>show interfaces [interface-type interface-number] fair-queue</code></td>
<td>Displays information and statistics about WFQ for a VIP-based interface.</td>
</tr>
<tr>
<td>Router# <code>show policy-map policy-map-name</code></td>
<td>Displays the contents of a policy map, including the priority setting in a specific policy map.</td>
</tr>
</tbody>
</table>

Verifying a Transmission Ring Limit

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# <code>show atm vc vc-name</code></td>
<td>Displays the contents of a VC. The <code>show atm vc</code> command output will indicate the transmission ring limit value if the <code>tx-ring-limit</code> command is successfully enabled.</td>
</tr>
</tbody>
</table>

Monitoring and Maintaining Distributed LLQ

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# <code>show interfaces [interface-type interface-number] fair-queue</code></td>
<td>Displays information and statistics about WFQ for a VIP-based interface.</td>
</tr>
<tr>
<td>Router# <code>show policy-map policy-map-name</code></td>
<td>Displays the contents of a traffic policy, including the priority setting in a specific policy map.</td>
</tr>
<tr>
<td>Router# <code>show policy interface interface-name</code></td>
<td>Displays the configuration of all classes configured for all service policies on the specified interface. Displays if packets and bytes were discarded or dropped for the priority class in the service policy attached to the interface.</td>
</tr>
<tr>
<td>Router# <code>show atm vc vc-name</code></td>
<td>Displays the contents of a VC. The <code>show atm vc</code> command output will indicate the transmission ring limit value if the <code>tx-ring-limit</code> command is successfully enabled.</td>
</tr>
</tbody>
</table>

Low Latency Queueing for Frame Relay Configuration Task List

- Defining Class Maps, page 24
- Configuring Class Policy in the Policy Map, page 24
Defining Class Maps

**SUMMARY STEPS**

1. `Router(config)# class-map class-map-name`
2. Do one of the following:
   - `Router(config-cmap)# match access-group {access-group name access-group-name}`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config)# class-map class-map-name</code></td>
<td>Specifies the name of the class map to be created.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td>Do one of the following:</td>
<td></td>
</tr>
<tr>
<td>- <code>Router(config-cmap)# match access-group {access-group name access-group-name}</code></td>
<td>Specifies the name of the ACL against whose contents packets are checked to determine if they belong to the class.</td>
</tr>
<tr>
<td></td>
<td>Specifies the name of the input interface used as a match criterion against which packets are checked to determine if they belong to the class.</td>
</tr>
<tr>
<td></td>
<td>Specifies the name of the protocol used as a match criterion against which packets are checked to determine if they belong to the class.</td>
</tr>
</tbody>
</table>

**Example:**

- `Router(config-cmap)# match input-interface interface-name`
- `Router(config-cmap)# match protocol protocol`

Configuring Class Policy in the Policy Map

To configure a policy map and create class policies that make up the service policy, begin with the `policy-map` command to specify the policy map name. Then use one or more of the following commands to configure the policy for a standard class or the default class:

- `priority`
- `bandwidth`
- `queue-limit` or `random-detect`
- `fair-queue` (for class-default class only)

For each class that you define, you can use one or more of the commands listed to configure the class policy. For example, you might specify bandwidth for one class and both bandwidth and queue limit for another class.

The default class of the policy map (commonly known as the class-default class) is the class to which traffic is directed if that traffic does not satisfy the match criteria of the other classes defined in the policy map.
The class-default class is used to classify traffic that does not fall into one of the defined classes. Even though the class-default class is predefined when you create the policy map, you still have to configure it. If a default class is not configured, then traffic that does not match any of the configured classes is given best-effort treatment, which means that the network will deliver the traffic if it can, without any assurance of reliability, delay prevention, or throughput.

You can configure class policies for as many classes as are defined on the router, up to the maximum of 64. However, the total amount of bandwidth allocated for all classes in a policy map must not exceed the minimum committed information rate (CIR) configured for the VC minus any bandwidth reserved by the `frame-relay voice bandwidth` and `frame-relay ip rtp priority` commands. If the minimum CIR is not configured, the bandwidth defaults to one half of the CIR. If all of the bandwidth is not allocated, the remaining bandwidth is allocated proportionally among the classes on the basis of their configured bandwidth.

To configure class policies in a policy map, perform the tasks described in the following sections.

- Configuring Class Policy for a LLQ Priority Queue, page 25
- Configuring Class Policy Using a Specified Bandwidth and WRED Packet Drop, page 25
- Configuring the Class-Default Class Policy, page 26

### Configuring Class Policy for a LLQ Priority Queue

**SUMMARY STEPS**

1. Router(config)# `policy-map` `policy-map`
2. Router(config-pmap)# `class` `class-name`
3. Router(config-pmap-c)# `priority` `bandwidth-kbps`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 Router(config)# <code>policy-map</code> <code>policy-map</code></td>
<td>Specifies the name of the policy map to be created or modified.</td>
</tr>
<tr>
<td>Step 2 Router(config-pmap)# <code>class</code> <code>class-name</code></td>
<td>Specifies the name of a class to be created and included in the service policy.</td>
</tr>
<tr>
<td>Step 3 Router(config-pmap-c)# <code>priority</code> <code>bandwidth-kbps</code></td>
<td>Creates a strict priority class and specifies the amount of bandwidth, in kbps, to be assigned to the class.</td>
</tr>
</tbody>
</table>

### Configuring Class Policy Using a Specified Bandwidth and WRED Packet Drop

**SUMMARY STEPS**

1. Router(config)# `policy-map` `policy-map`
2. Router(config-pmap)# `class` `class-name`
3. Router(config-pmap-c)# `bandwidth` `bandwidth-kbps`
4. Router(config-pmap-c)# `random-detect`
**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> Router(config)# policy-map policy-map</td>
<td>Specifies the name of the policy map to be created or modified.</td>
</tr>
<tr>
<td><strong>Step 2</strong> Router(config-pmap)# class class-name</td>
<td>Specifies the name of a class to be created and included in the service policy.</td>
</tr>
<tr>
<td><strong>Step 3</strong> Router(config-pmap-c)# bandwidth bandwidth-kbps</td>
<td>Specifies the amount of bandwidth to be assigned to the class, in kbps, or as a percentage of the available bandwidth. Bandwidth must be specified in kbps or as a percentage consistently across classes. (Bandwidth of the priority queue must be specified in kbps.)</td>
</tr>
<tr>
<td><strong>Step 4</strong> Router(config-pmap-c)# random-detect</td>
<td>Enables WRED.</td>
</tr>
</tbody>
</table>

**Configuring the Class-Default Class Policy**

**SUMMARY STEPS**

1. Router(config)# policy-map policy-map
2. Router(config-pmap)# class class-default default-class-name
3. Do one of the following:
   - Router(config-pmap-c)# bandwidth bandwidth-kbps
4. Router(config-pmap-c)# queue-limit number-of-packets

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> Router(config)# policy-map policy-map</td>
<td>Specifies the name of the policy map to be created or modified.</td>
</tr>
<tr>
<td><strong>Step 2</strong> Router(config-pmap)# class class-default default-class-name</td>
<td>Specifies the default class so that you can configure or modify its policy.</td>
</tr>
<tr>
<td><strong>Step 3</strong> Do one of the following:</td>
<td>Specifies the amount of bandwidth, in kbps, to be assigned to the class.</td>
</tr>
<tr>
<td>- Router(config-pmap-c)# bandwidth bandwidth-kbps</td>
<td>Specifies the number of dynamic queues to be reserved for use by flow-based WFQ running on the default class. The number of dynamic queues is derived from the bandwidth of the interface.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-pmap-c)# fair-queue {number-of-dynamic-queues}</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> Router(config-pmap-c)# queue-limit number-of-packets</td>
<td>Specifies the maximum number of packets that the queue for the default class can accumulate.</td>
</tr>
</tbody>
</table>
Attaching the Service Policy and Enabling LLQ for Frame Relay

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(config-map-class)# service-policy output policy-map</td>
<td>Attaches the specified service policy map to the output interface and enables LLQ for Frame Relay. <strong>Note</strong> When LLQ is enabled, all classes configured as part of the service policy map are installed in the fair queueing system.</td>
</tr>
</tbody>
</table>

Verifying Configuration of Policy Maps and Their Classes

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# show frame-relay pvc dlci</td>
<td>Displays statistics about the PVC and the configuration of classes for the policy map on the specified DLCI.</td>
</tr>
<tr>
<td>Router# show policy-map interface interface-name</td>
<td>When FRTS is configured, displays the configuration of classes for all Frame Relay VC-level policy maps. When FRTS is not configured, displays the configuration of classes for the interface-level policy.</td>
</tr>
<tr>
<td>Router# show policy-map interface interface-name dlci dlci</td>
<td>When FRTS is configured, displays the configuration of classes for the policy map on the specified DLCI.</td>
</tr>
</tbody>
</table>

Monitoring and Maintaining LLQ for Frame Relay

For a list of commands that can be used to monitor LLQ for Frame Relay, see the previous section "Verifying Configuration of Policy Maps and Their Classes, page 13."

Configuring Burst Size in LLQ Configuration Task List

- Configuring the LLQ Bandwidth, page 28
- Configuring the LLQ Burst Size, page 28
- Verifying the LLQ Burst Size, page 28
Configuring the LLQ Bandwidth

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Router(config)# priority bandwidth</code></td>
<td>Specifies the maximum amount of bandwidth, in kpbs, for the priority traffic.</td>
</tr>
</tbody>
</table>

Configuring the LLQ Burst Size

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Router(config)# priority bandwidth burst</code></td>
<td>Specifies the burst size in bytes. The range is from 32 to 2 million.</td>
</tr>
</tbody>
</table>

Verifying the LLQ Burst Size

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Router# show policy-map</code></td>
<td>Displays the configuration of all classes comprising the specified service policy map or all classes for all existing policy maps.</td>
</tr>
<tr>
<td><code>Router# show policy-map interface</code></td>
<td>Displays the configuration of classes configured for service policies on the specified interface or PVC.</td>
</tr>
</tbody>
</table>

Per-VC Hold Queue Support for ATM Adapters Configuration Task List

- Configuring the per-VC Hold Queue on an ATM Adapter, page 28
- Verifying the Configuration of the per-VC Hold Queue on an ATM Adapter, page 29

Configuring the per-VC Hold Queue on an ATM Adapter

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Router(config)# vc-hold-queue number-of-packets</code></td>
<td>Specifies the number of packets contained in the per-VC hold queue. This can be a number from 5 to 1024.</td>
</tr>
</tbody>
</table>
### Verifying the Configuration of the per-VC Hold Queue on an ATM Adapter

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router# show queueing interface</td>
<td>Displays the queueing statistics of an interface or VC.</td>
</tr>
</tbody>
</table>

### Examples Flow-Based WFQ Configuration

The following example requests a fair queue with a congestive discard threshold of 64 messages, 512 dynamic queues, and 18 RSVP queues:

```plaintext
Router(config)# interface Serial 3/0
Router(config-if)# ip unnumbered Ethernet 0/0
Router(config-if)# fair-queue 64 512 18
```

### Examples DWFQ Configuration

- Example Flow-Based DWFQ, page 29
- Example QoS-Group-Based DWFQ, page 29
- Example ToS-Based DWFQ, page 30

### Example Flow-Based DWFQ

The following example enables DWFQ on the HSSI interface 0/0/0:

```plaintext
Router(config)# interface Hssi0/0/0
Router(config-if)# description 45Mbps to R2
Router(config-if)# ip address 200.200.14.250 255.255.255.252
Router(config-if)# fair-queue
```

The following is sample output from the `show interfaces fair-queue` command for this configuration:

```plaintext
Router# show interfaces hssi 0/0/0 fair-queue
Hssi0/0/0 queue size 0
    packets output 35, drops 0
    WFQ: global queue limit 401, local queue limit 200
```

### Example QoS-Group-Based DWFQ

The following example configures QoS-group-based DWFQ. Committed access rate (CAR) policies are used to assign packets with an IP Precedence value of 2 to QoS group 2, and packets with an IP Precedence value of 6 are assigned to QoS group 6.

```plaintext
Router(config)# interface Hssi0/0/0
Router(config-if)# ip address 188.1.3.70 255.255.255.0
Router(config-if)# rate-limit output access-group rate-limit 2 155000000 2000000 8000000 conform-action set-qos-transmit 6 exceed-action drop
Router(config-if)# rate-limit output access-group rate-limit 6 155000000 2000000 8000000 conform-action set-qos-transmit 2 exceed-action drop
```
Router(config-if)# fair-queue qos-group
Router(config-if)# fair-queue qos-group 2 weight 10
Router(config-if)# fair-queue qos-group 2 limit 27
Router(config-if)# fair-queue qos-group 6 weight 30
Router(config-if)# fair-queue qos-group 6 limit 27

Router(config)# access-list rate-limit 2 2
Router(config)# access-list rate-limit 6 6

The following sample output shows how to view WFQ statistics using the `show interfaces fair-queue` command:

Router# show interfaces fair-queue
Hssi0/0/0 queue size 0
packets output 806232, drops 1
WFQ: aggregate queue limit 54, individual queue limit 27
max available buffers 54
Class 0: weight 60 limit 27 qsize 0 packets output 654 drops 0
Class 2: weight 10 limit 27 qsize 0 packets output 402789 drops 0
Class 6: weight 30 limit 27 qsize 0 packets output 402789 drops 1

**Example ToS-Based DWFQ**

The following example configures type of service (ToS)-based DWFQ using the default parameters:

Router# configure terminal
Router(config)# interface Hssi0/0/0
Router(config-if)# fair-queue tos
Router(config-if)# end

The following is output of the `show running-config` command for the HSSI interface 0/0/0. Notice that the router automatically adds the default weights and limits for the ToS classes to the configuration.

interface Hssi0/0/0
ip address 188.1.3.70 255.255.255.0
fair-queue tos
fair-queue tos 1 weight 20
fair-queue tos 1 limit 27
fair-queue tos 2 weight 30
fair-queue tos 2 limit 27
fair-queue tos 3 weight 40
fair-queue tos 3 limit 27

The following sample output shows how to view DWFQ statistics using the `show interfaces fair-queue` command:

Router# show interfaces fair-queue
Hssi0/0/0 queue size 0
packets output 1417079, drops 2
WFQ: aggregate queue limit 54, individual queue limit 27
max available buffers 54
Class 0: weight 10 limit 27 qsize 0 packets output 1150 drops 0
Class 1: weight 20 limit 27 qsize 0 packets output 0 drops 0
Class 2: weight 30 limit 27 qsize 0 packets output 775482 drops 1
Class 3: weight 40 limit 27 qsize 0 packets output 0 drops 0

**Examples CBWFQ Configuration**

- Example Class Map Configuration, page 31
- Example Policy Creation, page 31
- Example Policy Attachment to Interfaces, page 31
Example Class Map Configuration

In the following example, ACLs 101 and 102 are created. Next, two class maps are created and their match criteria are defined. For the first map class, called class1, the numbered ACL 101 is used as the match criterion. For the second map class, called class2, the numbered ACL 102 is used as the match criterion. Packets are checked against the contents of these ACLs to determine if they belong to the class.

```
Router(config)# access-list 101 permit udp host 10.10.10.10 host 10.10.10.20 range 16384 20000
Router(config)# access-list 102 permit udp host 10.10.10.10 host 10.10.10.20 range 53000 56000
Router(config)# class-map class1
Router(config-cmap)# match access-group 101
Router(config-cmap)# exit
Router(config-cmap)# class-map class2
Router(config-cmap)# match access-group 102
Router(config-cmap)# exit
```

Example Policy Creation

In the following example, a policy map called policy1 is defined to contain policy specification for the two classes, class1 and class2. The match criteria for these classes were defined in the previous "Example Class Map Configuration, page 31" section.

For class1, the policy specifies the bandwidth allocation request and the maximum number of packets that the queue for this class can accumulate. For class2, the policy specifies only the bandwidth allocation request, so the default queue limit of 64 packets is assumed.

```
Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# bandwidth 3000
Router(config-pmap-c)# queue-limit 30
Router(config-pmap-c)# exit
Router(config-pmap)# class class2
Router(config-pmap-c)# bandwidth 2000
Router(config-pmap-c)# exit
```

Example Policy Attachment to Interfaces

The following example shows how to attach an existing policy map. After you define a policy map, you can attach it to one or more interfaces to specify the service policy for those interfaces. Although you can assign the same policy map to multiple interfaces, each interface can have only one policy map attached at the input and one policy map attached at the output.

The policy map in this example was defined in the previous section, "Example Policy Creation, page 31."

```
Router(config)# interface e1/1
Router(config-if)# service output policy1
Router(config-if)# exit
Router(config)# interface fa1/0/0
Router(config-if)# service output policy1
Router(config-if)# exit
```
Example CBWFQ Using WRED Packet Drop

In the following example, the class map called class1 is created and defined to use the input FastEthernet interface 0/1 as a match criterion to determine if packets belong to the class. Next, the policy map policy1 is defined to contain policy specification for class1, which is configured for WRED packet drop.

```
Router(config)# class-map class1
Router(config-cmap)# match input-interface FastEthernet0/1

Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# bandwidth 1000
Router(config-pmap-c)# random-detect

Router(config)# interface serial0/0
Router(config-if)# service-policy output policy1
```

Examples Display Service Policy Map Content

The following examples show how to display the contents of service policy maps. Four methods can be used to display the contents.

- All Classes for a Specified Service Policy Map, page 32
- All Classes for All Service Policy Maps, page 32
- Specified Class for a Service Policy Map, page 33
- All Classes for All Service Policy Maps on a Specified Interface, page 33

All Classes for a Specified Service Policy Map

The following example displays the contents of the service policy map called pol1:

```
Router# show policy-map pol
Policy Map pol
Weighted Fair Queueing
  Class class1
    Bandwidth 937 (kbps) Max thresh 64 (packets)
  Class class2
    Bandwidth 937 (kbps) Max thresh 64 (packets)
  Class class3
    Bandwidth 937 (kbps) Max thresh 64 (packets)
  Class class4
    Bandwidth 937 (kbps) Max thresh 64 (packets)
  Class class5
    Bandwidth 937 (kbps) Max thresh 64 (packets)
  Class class6
    Bandwidth 937 (kbps) Max thresh 64 (packets)
  Class class7
    Bandwidth 937 (kbps) Max thresh 64 (packets)
  Class class8
    Bandwidth 937 (kbps) Max thresh 64 (packets)
```

All Classes for All Service Policy Maps

The following example displays the contents of all policy maps on the router:

```
Router# show policy-map
Policy Map poH1
Weighted Fair Queueing
```
Class class1
  Bandwidth 937 (kbps) Max thresh 64 (packets)
Class class2
  Bandwidth 937 (kbps) Max thresh 64 (packets)
Class class3
  Bandwidth 937 (kbps) Max thresh 64 (packets)
Class class4
  Bandwidth 937 (kbps) Max thresh 64 (packets)
Class class5
  Bandwidth 937 (kbps) Max thresh 64 (packets)
Class class6
  Bandwidth 937 (kbps) Max thresh 64 (packets)
Class class7
  Bandwidth 937 (kbps) Max thresh 64 (packets)
Class class8
  Bandwidth 937 (kbps) Max thresh 64 (packets)

Policy Map policy2
  Weighted Fair Queueing
Class class1
  Bandwidth 300 (kbps) Max thresh 64 (packets)
Class class2
  Bandwidth 300 (kbps) Max thresh 64 (packets)
Class class3
  Bandwidth 300 (kbps) Max thresh 64 (packets)
Class class4
  Bandwidth 300 (kbps) Max thresh 64 (packets)
Class class5
  Bandwidth 300 (kbps) Max thresh 64 (packets)
Class class6
  Bandwidth 300 (kbps) Max thresh 64 (packets)
Class class7
  Bandwidth 300 (kbps) Max thresh 64 (packets)
Class class8
  Bandwidth 300 (kbps) Max thresh 64 (packets)

Specified Class for a Service Policy Map

The following example displays configurations for the class called class7 that belongs to the policy map called pol:

Router# show policy-map pol class class7

Class class7
  Bandwidth 937 (kbps) Max Thresh 64 (packets)

All Classes for All Service Policy Maps on a Specified Interface

The following example displays configurations for classes on the output Ethernet interface 2/0. The numbers shown in parentheses are for use with the Management Information Base (MIB).

Router# show policy-map interface e2/0
Ethernet2/0
  Service-policy output:pl (1057)
    class-map:cl (match-all) (1059/2)
      19 packets, 1140 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match:ip precedence 0 (1063)
      Weighted Fair Queueing
      Output Queue:Conversation 265
      Bandwidth 10 (%) Max Threshold 64 (packets)
      (pkts matched/bytes matched) 0/0
      (depth/total drops/no-buffer drops) 0/0/0
    class-map:cl (match-all) (1067/3)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match:ip precedence 1 (1071)
      Weighted Fair Queueing
      Output Queue:Conversation 266
      Bandwidth 10 (%) Max Threshold 64 (packets)
      (pkts matched/bytes matched) 0/0
Examples Distributed CBWFQ Configuration

Example Traffic Class Configuration

In the following example, two traffic classes are created and their match criteria are defined. For the first traffic class, called class1, the numbered ACL 101 is used as the match criterion. For the second traffic class, called class2, the numbered ACL 102 is used as the match criterion. Packets are checked against the contents of these ACLs to determine if they belong to the traffic class.

Router(config)# class-map class1
Router(config-cmap)# match access-group 101
Router(config-cmap)# exit
Router(config)# class-map class2
Router(config-cmap)# match access-group 102
Router(config-cmap)# exit

Example Traffic Policy Creation

In the following example, a traffic policy called policy1 is defined to associate QoS features with the two traffic classes, class1 and class2. The match criteria for these traffic classes were defined in the previous "Example Class Map Configuration, page 31" section.

For class1, the QoS policies include bandwidth allocation request and maximum packet count limit for the queue reserved for the traffic class. For class2, the policy specifies only a bandwidth allocation request, so the default queue limit of 64 packets is assumed.

Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# bandwidth 3000
Router(config-pmap-c)# queue-limit 30
Router(config-pmap)# exit
Router(config-pmap)# class class2
Router(config-pmap-c)# bandwidth 2000
Router(config-pmap)# exit

Example Traffic Policy Attachment to an Interface

The following example shows how to attach an existing traffic policy to an interface. After you define a traffic policy, you can attach it to one or more interfaces to specify a traffic policy for those interfaces. Although you can assign the same traffic policy to multiple interfaces, each interface can have only one traffic policy attached at the input and one policy map attached at the output at one time.

Router(config)# interface fe1/0/0
Examples IP RTP Priority Configuration

- Example CBWFQ Configuration, page 35
- Example Virtual Template Configuration, page 35
- Example Multilink Bundle Configuration, page 36
- Example Debug, page 37

Example CBWFQ Configuration

The following example first defines a CBWFQ configuration and then reserves a strict priority queue:

! The following commands define a class map:
Router(config)# class-map class1
Router(config-cmap)# match access-group 101
Router(config-cmap)# exit
! The following commands create and attach a policy map:
Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# bandwidth 3000
Router(config-pmap-c)# queue-limit 30
Router(config-pmap-c)# random-detect
Router(config-pmap-c)# random-detect precedence 0 32 256 100
Router(config-pmap-c)# exit
Router(config)# interface Serial1
Router(config-if)# service-policy output policy1
! The following command reserves a strict priority queue:
Router(config-if)# ip rtp priority 16384 16383 40

The queue-limit and random-detect commands are optional commands for CBWFQ configurations. The queue-limit command is used for configuring tail drop limits for a class queue. The random-detect command is used for configuring RED drop limits for a class queue, similar to the random-detect command available on an interface.

Example Virtual Template Configuration

The following example configures a strict priority queue in a virtual template configuration with CBWFQ:

Router(config)# multilink virtual-template 1
Router(config)# interface virtual-template 1
Router(config-if)# ip address 172.16.1.1 255.255.255.0
Router(config-if)# no ip directed-broadcast
Router(config-if)# ip rtp priority 16384 16383 25
Router(config-if)# service-policy output policy1
Router(config-if)# ppp multilink
Router(config-if)# ppp multilink fragment-delay 20
Router(config-if)# ppp multilink interleave
Router(config-if)# end
Router(config)# interface Serial0/1
Router(config-if)# bandwidth 64
Router(config-if)# ip address 1.1.1.2 255.255.255.0
Router(config-if)# no ip directed-broadcast
Router(config-if)# encapsulation ppp
Router(config-if)# ppp multilink
Router(config-if)# end
To make the virtual access interface function properly, the bandwidth policy-map class configuration command should not be configured on the virtual template. It needs to be configured on the actual interface, as shown in the example.

**Example Multilink Bundle Configuration**

The following example configures a strict priority queue in a multilink bundle configuration with WFQ. The advantage to using multilink bundles is that you can specify different ip rtp priority parameters on different interfaces.

The following commands create multilink bundle 1, which is configured for a maximum ip rtp priority bandwidth of 200 kbps:

```
Router(config)# interface multilink 1
Router(config-if)# ip address 172.17.254.161 255.255.255.248
Router(config-if)# no ip directed-broadcast
Router(config-if)# ip rtp priority 16384 16383 200
Router(config-if)# no ip mroute-cache
Router(config-if)# fair-queue 64 256 0
Router(config-if)# ppp multilink
Router(config-if)# ppp multilink fragment-delay 20
Router(config-if)# ppp multilink interleave
```

The following commands create multilink bundle 2, which is configured for a maximum ip rtp priority bandwidth of 100 kbps:

```
Router(config)# interface multilink 2
Router(config-if)# ip address 172.17.254.162 255.255.255.248
Router(config-if)# no ip directed-broadcast
Router(config-if)# ip rtp priority 16384 16383 100
Router(config-if)# no ip mroute-cache
Router(config-if)# fair-queue 64 256 0
Router(config-if)# ppp multilink
Router(config-if)# ppp multilink fragment-delay 20
Router(config-if)# ppp multilink interleave
```

In the next part of the example, the multilink-group command configures serial interface 2/0 to be part of multilink bundle 1:

```
Router(config)# interface serial 2/0
Router(config-if)# bandwidth 256
Router(config-if)# no ip address
Router(config-if)# no ip directed-broadcast
Router(config-if)# encapsulation ppp
Router(config-if)# no ip mroute-cache
Router(config-if)# no fair-queue
Router(config-if)# clockrate 256000
Router(config-if)# ppp multilink
Router(config-if)# multilink-group 1
```

Next, serial interface 2/1 is configured to be part of multilink bundle 2:

```
Router(config)# interface serial 2/1
Router(config-if)# bandwidth 128
Router(config-if)# no ip address
Router(config-if)# no ip directed-broadcast
Router(config-if)# encapsulation ppp
Router(config-if)# no ip mroute-cache
Router(config-if)# no fair-queue
Router(config-if)# clockrate 128000
Router(config-if)# ppp multilink
Router(config-if)# multilink-group 2
```
Example Debug

The following example shows sample output from the `debug priority` command. In this example, 64 indicates the actual priority queue depth at the time the packet was dropped.

```
Router# debug priority
*Feb 28 16:46:05.659:WFQ:dropping a packet from the priority queue 64
*Feb 28 16:46:05.671:WFQ:dropping a packet from the priority queue 64
*Feb 28 16:46:05.679:WFQ:dropping a packet from the priority queue 64
*Feb 28 16:46:05.691:WFQ:dropping a packet from the priority queue 64
*Feb 28 16:46:05.699:WFQ:dropping a packet from the priority queue 64
*Feb 28 16:46:05.711:WFQ:dropping a packet from the priority queue 64
*Feb 28 16:46:05.719:WFQ:dropping a packet from the priority queue 64
```

Examples Frame Relay IP RTP Priority Configuration

- Example Strict Priority Service to Matching RTP Packets, page 37

Example Strict Priority Service to Matching RTP Packets

The following example first configures the Frame Relay map class called voip and then applies the map class to PVC 100 to provide strict priority service to matching RTP packets. In this example, RTP packets on PVC 100 with UDP ports in the range 16384 to 32764 will be matched and given strict priority service.

```
map-class frame-relay voip
  frame-relay cir 256000
  frame-relay bc 2560
  frame-relay be 600
  frame-relay mincir 256000
  no frame-relay adaptive-shaping
  frame-relay fair-queue
  frame-relay fragment 250
  frame-relay ip rtp priority 16384 16380 210
interface Serial5/0
  ip address 10.10.10.10 255.0.0.0
  no ip directed-broadcast
  encapsulation frame-relay
  no ip mroute-cache
  load-interval 30
  clockrate 1007616
  frame-relay traffic-shaping
  frame-relay interface-dlci 100
  class voip
  frame-relay ip rtp header-compression
  frame-relay intf-type dce
```

Examples Frame Relay PVC Interface PQ Configuration

This section provides configuration examples for Frame Relay PIQ.

This example shows the configuration of four PVCs on serial interface 0. DLCI 100 is assigned high priority, DLCI 200 is assigned medium priority, DLCI 300 is assigned normal priority, and DLCI 400 is assigned low priority.

The following commands configure Frame Relay map classes with PVC priority levels:

```
Router(config)# map-class frame-relay HI
```
Router(config-map-class)# frame-relay interface-queue priority high
Router(config-map-class)# exit
Router(config)# map-class frame-relay MED
Router(config-map-class)# frame-relay interface-queue priority medium
Router(config-map-class)# exit
Router(config)# map-class frame-relay NORM
Router(config-map-class)# frame-relay interface-queue priority normal
Router(config-map-class)# exit
Router(config)# map-class frame-relay LOW
Router(config-map-class)# frame-relay interface-queue priority low
Router(config-map-class)# exit

The following commands enable Frame Relay encapsulation and Frame Relay PIPQ on serial interface 0. The sizes of the priority queues are set at a maximum of 20 packets for the high priority queue, 40 for the medium priority queue, 60 for the normal priority queue, and 80 for the low priority queue.

Router(config)# interface Serial0
Router(config-if)# encapsulation frame-relay
Router(config-if)# frame-relay interface-queue priority 20 40 60 80

The following commands assign priority to four PVCs by associating the DLCIs with the configured map classes:

Router(config-if)# frame-relay interface-dlci 100
Router(config-fr-dlci)# class HI
Router(config-fr-dlci)# exit
Router(config-if)# frame-relay interface-dlci 200
Router(config-fr-dlci)# class MED
Router(config-fr-dlci)# exit
Router(config-if)# frame-relay interface-dlci 300
Router(config-fr-dlci)# class NORM
Router(config-fr-dlci)# exit
Router(config-if)# frame-relay interface-dlci 400
Router(config-fr-dlci)# class LOW
Router(config-fr-dlci)# exit

Examples LLQ Configuration

- Example ATM PVC Configuration, page 38
- Example Virtual Template Configuration, page 39
- Example Multilink Bundle Configuration, page 39

Example ATM PVC Configuration

In the following example, a strict priority queue with a guaranteed allowed bandwidth of 50 kbps is reserved for traffic that is sent from the source address 10.10.10.10 to the destination address 10.10.10.20, in the range of ports 16384 through 20000 and 53000 through 56000.

First, the following commands configure access list 102 to match the desired voice traffic:

Router(config)# access-list 102 permit udp host 10.10.10.10 host 10.10.10.20 range 16384 20000
Router(config)# access-list 102 permit udp host 10.10.10.10 host 10.10.10.20 range 53000 56000

Next, the class map voice is defined, and the policy map called policy1 is created; a strict priority queue for the class voice is reserved, a bandwidth of 20 kbps is configured for the class bar, and the default class is configured for WFQ. The service-policy command then attaches the policy map to the PVC interface 0/102 on the subinterface atm1/0.2.

Router(config)# class-map voice

Router(config)# service-policy policy1
Example Virtual Template Configuration

The following example configures a strict priority queue in a virtual template configuration with CBWFQ. Traffic on virtual template 1 that is matched by access list 102 will be directed to the strict priority queue.

First, the class map voice is defined, and the policy map called policy1 is created. A strict priority queue (with a guaranteed allowed bandwidth of 50 kbps) is reserved for the class called voice.

```
Router(config)# class-map voice
Router(config-cmap)# match access-group 102
Router(config)# policy-map policy1
Router(config-pmap)# class voice
Router(config-pmap-c)# priority 50
```

Next, the `service-policy` command attaches the policy map called policy1 to virtual template 1.

```
Router(config)# multilink virtual-template 1
Router(config-if)# interface virtual-template 1
Router(config-if)# ip address 172.16.1.1 255.255.255.0
Router(config-if)# no ip directed-broadcast
Router(config-if)# service-policy output policy1
Router(config-if)# ppp multilink
Router(config-if)# ppp multilink fragment-delay 20
Router(config-if)# ppp multilink interleave
Router(config-if)# end
```

Example Multilink Bundle Configuration

The following example configures a strict priority queue in a multilink bundle configuration with CBWFQ. Traffic on serial interface 2/0 that is matched by access list 102 will be directed to the strict priority queue. The advantage to using multilink bundles is that you can specify different priority parameters on different interfaces. To specify different priority parameters, you would configure two multilink bundles with different parameters.

First, the class map voice is defined, and the policy map called policy1 is created. A strict priority queue (with a guaranteed allowed bandwidth of 50 kbps) is reserved for the class called voice.

```
Router(config)# class-map voice
Router(config-cmap)# match access-group 102
Router(config)# policy-map policy1
Router(config-pmap)# class voice
Router(config-pmap-c)# priority 50
```
The following commands create multilink bundle 1. The policy map called policy1 is attached to the bundle by the `service-policy` command.

```
Router(config)# interface multilink 1
Router(config-if)# ip address 172.17.254.161 255.255.255.248
Router(config-if)# no ip directed-broadcast
Router(config-if)# no ip mroute-cache
Router(config-if)# service-policy output policy1
Router(config-if)# ppp multilink
Router(config-if)# ppp multilink fragment-delay 20
Router(config-if)# ppp multilink interleave
```

In the next part of the example, the `multilink-group` command configures serial interface 2/0 to be part of multilink bundle 1, which effectively directs traffic on serial interface 2/0 that is matched by access list 102 to the strict priority queue:

```
Router(config)# interface serial 2/0
Router(config-if)# bandwidth 256
Router(config-if)# no ip address
Router(config-if)# no ip directed-broadcast
Router(config-if)# encapsulation ppp
Router(config-if)# no fair-queue
Router(config-if)# clockrate 256000
Router(config-if)# ppp multilink
Router(config-if)# multilink-group 1
```

### Examples Distributed LLQ Configuration

- Example Enabling PQ for an Amount of Available Bandwidth on an ATM Subinterface, page 40
- Example Enabling PQ for a Percentage of Available Bandwidth on an ATM Subinterface, page 41
- Example Limiting the Transmission Ring Limit on an ATM Interface, page 41
- Example Limiting the Transmission Ring Limit on an ATM PVC Subinterface, page 41

### Example Enabling PQ for an Amount of Available Bandwidth on an ATM Subinterface

The `priority` command can be enabled on an ATM subinterface, and that subinterface must have only one enabled ATM PVC. This configuration provides a sufficient amount of ATM PVC support.

In the following example, a priority queue with a guaranteed allowed bandwidth of 50 kbps is reserved for traffic that is sent from the source address 10.10.10.10 to the destination address 10.10.10.20, in the range of ports 16384 through 20000 and 53000 through 56000.

First, the following commands configure access list 102 to match the desired voice traffic:

```
Router(config)# access-list 102 permit udp host 10.10.10.10 host 10.10.10.20 range 16384 20000
Router(config)# access-list 102 permit udp host 10.10.10.10 host 10.10.10.20 range 53000 56000
```

Next, the traffic class called voice is defined, and the policy map called policy1 is created; a priority queue for the class voice is reserved with a guaranteed allowed bandwidth of 50 kbps and an allowable burst size of 60 bytes, a bandwidth of 20 kbps is configured for the class called bar, and the default class is configured for flow-based fair queuing. The `service-policy` command then attaches the policy map to the PVC interface 0/102 on the subinterface atm1/0.

```
Router(config)# class-map voice
```
Example Enabling PQ for a Percentage of Available Bandwidth on an ATM Subinterface

The `priority percent` command can be enabled on an ATM subinterface, and that subinterface must have only one enabled ATM PVC. This configuration provides a sufficient amount of ATM PVC support.

In the following example, a priority queue with a guaranteed allowed bandwidth percentage of 15 percent is reserved for traffic that is sent from the source address 10.10.10.10 to the destination address 10.10.10.20, in the range of ports 16384 through 20000 and 53000 through 56000.

First, the following commands configure access list 102 to match the desired voice traffic:

```
Router(config)# access-list 102 permit udp host 10.10.10.10 host 10.10.10.20 range 16384 20000
Router(config)# access-list 102 permit udp host 10.10.10.10 host 10.10.10.20 range 53000 56000
```

Next, the traffic class called voice is defined, and the policy map called policy1 is created; a priority queue for the class voice is reserved with a guaranteed allowed bandwidth percentage of 15 percent, a bandwidth percentage of 20 percent is configured for the class called bar, and the default class is configured for flow-based fair queuing. The `service-policy` command then attaches the policy map to the ATM subinterface 1/0.2.

```
Router(config)# class-map voice
Router(config-cmap)# match access-group 102
Router(config)# policy-map policy1
Router(config-pmap)# class voice
Router(config-pmap-c)# priority percent 15
Router(config)# class bar
Router(config-pmap-c)# bandwidth percent 20
Router(config)# class class-default
Router(config-pmap-c)# fair-queue
Router(config)# interface atm1/0
Router(config-subif)# pvc 0/102
Router(config-subif)# service-policy output policy1
```

Example Limiting the Transmission Ring Limit on an ATM Interface

In the following example, the number of particles on the transmission ring of an ATM interface is limited to seven particles:

```
Router(config)# interface atm 1/0/0
Router(config-if)# atm pvc 32 0 32 tx-ring-limit 7
```

Example Limiting the Transmission Ring Limit on an ATM PVC Subinterface

In the following example, the number of particles on the transmission ring of an ATM PVC subinterface is limited to ten particles:

```
Router(config)#
```
The `tx-ring-limit` command can be applied to several ATM PVC subinterfaces on a single interface. Every individual PVC can configure a transmission ring limit.

### Examples LLQ for Frame Relay Configuration

The following example shows how to configure a PVC shaped to a 64K CIR with fragmentation. The shaping queue is configured with a class for voice, two data classes for IP precedence traffic, and a default class for best-effort traffic. WRED is used as the drop policy on one of the data classes.

The following commands define class maps and the match criteria for the class maps:

```plaintext
! class-map voice
match access-group 101
!
class-map immediate-data
match access-group 102
!
class-map priority-data
match access-group 103
!
access-list 101 permit udp any any range 16384 32767
access-list 102 permit ip any any precedence immediate
access-list 103 permit ip any any precedence priority
```

The following commands create and define a policy map called `mypolicy`:

```plaintext
! policy-map mypolicy
class voice priority 16
class immediate-data bandwidth 32 random-detect
class priority-data bandwidth 16
class class-default fair-queue 64 queue-limit 20
```

The following commands enable Frame Relay fragmentation and attach the policy map to DLCI 100:

```plaintext
! interface Serial1/0.1 point-to-point
   frame-relay interface-dlci 100
   class fragment
!
   map-class frame-relay fragment
   frame-relay cir 64000
   frame-relay mincir 64000
   frame-relay bc 640
   frame-relay fragment 50
   service-policy output mypolicy
```

### Examples Burst Size in LLQ Configuration
The following example configures the burst parameter to 1250 bytes for the class called Voice, which has an assigned bandwidth of 1000 kbps:

```
policy policy1
  class Voice
    priority 1000 1250
```

**Examples Per-VC Hold Queue Support for ATM Adapters**

The following example sets the per-VC hold queue to 55:

```
interface atm2/0.1
  pvc 1/101
    vc-hold-queue 55
```