CHAPTER 10

Configuring Quality of Service

This chapter explains how to configure Quality of Service (QoS) on the RPM-XF and contains the following sections:

- General QoS Configuration Procedure
- Class Map Commands
- Policy Map Commands
- Service-Policy Command
- Show Commands
- Quality of Service Policy Propagation Example Using Border Gateway Protocol
- Versatile Traffic Management System
- MultiLink PPP/Link Fragmentation Interleaving
- Configuring Internet Protocol Header Compression
- Enabling IP Radio Access Network

Supported Features

Quality of Service (QoS) on the RPM-XF supports the following features:

- Committed access rate (CAR) measures traffic rates and, based on the rates, takes actions (such as dropping packets). RPM-XF QoS supports CAR ("police") on input packets and shaping ("shape") on output packets.

- Random Early Detection (RED) is a congestion avoidance mechanism that takes advantage of TCP's congestion control mechanism. By randomly dropping packets prior to periods of high congestion, RED tells the packet source to decrease its transmission rate. Assuming the packet source is using TCP, it will decrease its transmission rate until all the packets reach their destination, indicating that the congestion is cleared.

- Weighted random early detection (WRED) uses an algorithm to randomly discard packets during congestion. This approach reduces congestion by causing the packet source to slow down. Weighted RED (WRED) generally drops packets selectively based on IP precedence. Packets with a higher IP precedence are less likely to be dropped than packets with a lower precedence. Thus, higher priority traffic is delivered with a higher probability than lower priority traffic.

- Bandwidth reservation, also referred as fair queueing, assigns bandwidth to certain streams of packets.
• Low-latency priority queueing can be assigned for real-time traffic such as voice and video.
• Traffic shaping is used to control traffic by maintaining data flow at a set rate.
• Set specifies an IP precedence/DSCP or MPLS experimental value that can be used by other routers to manage QoS.
• 802.1q support allows PXF switching for ARPA encapsulation.
• DSCP Marking on RPM-XF Management Interface, see DSCP Marking on RPM-XF Management Interface, page 10-19.
• Versatile Traffic Management System (VTMS), see Versatile Traffic Management System, page 10-22.
• MultiLink PPP/Link Fragmentation Interleaving (MLP/LFI), see MultiLink PPP/Link Fragmentation Interleaving, page 10-24.
• Internet Protocol Header Compression (IPHC), see Configuring Internet Protocol Header Compression, page 10-26.
• In addition, the RPM-XF supports QoS policy propagation through the Border Gateway Protocol (QPPB). For a QPPB configuration example, see “Quality of Service Policy Propagation Example Using Border Gateway Protocol” section on page 10-16.

**General QoS Configuration Procedure**

You can configure WRED, CAR, and other qualities of service by performing the following tasks:

1. Create a QoS boilerplate that defines the criteria for prioritizing traffic.
2. Apply the boilerplate to an interface.

**Figure 10-1** shows an overview of the QoS process.
Creating a QoS Boilerplate

This section provides the information you need to create a QoS boilerplate. To create a QoS boilerplate, perform two procedures:

1. Create a class map—The class map tells the RPM-XF how to recognize the packets that are subject to QoS.
2. Create a policy map—The policy map lists QoS services to be applied to packets described by one or more class maps.

Creating a Class Map

The following procedure describes how to create a class map.

**Step 1**
Assign a name to your class map by entering the `class-map name` command. In the following example, a class map named `mink` is created.

```
Router(config)# class-map mink
Router(config-cmap)#
```

As the example shows, after you enter the `class-map name` command, you enter class map configuration mode (config-cmap).
General QoS Configuration Procedure

Note
Some Cisco IOS documents refer to the QoS configuration modes as the modular CLI.

Step 2
Describe the characteristics of the packets that are subject to QoS by entering the match command. In the following example, the packet is described as being associated to access group 10 and having the IP precedence bit set to 1.

Router(config-cmap)# match access-group 10
Router(config-cmap)# match ip precedence 1

Step 3
Exit class map configuration mode.

Router(config-cmap)# exit
Router(config)

As a result of the creation of a class map, the router can recognize packets that are subject to QoS. You must now tell the router the action to take on those packets.

Creating a Policy Map

The following procedure describes how to create a policy map.

Step 1
Assign a name to your policy map by entering the policy-map name command. In the following example, a policy map named lynx is created.

Router(config)# policy-map lynx
Router(config-pmap)#

As the example shows, after you enter the policy-map name command, you enter policy map configuration mode (config-pmap).

Step 2
Associate the policy map with a class map.

Router(config-pmap)# class mink
Router(config-pmap-c)#

As the example shows, after entering the class name command, enter the policy map class configuration mode (config-pmap-c).

Step 3
Describe the QoS actions you want the router to perform when the router encounters a packet that has the characteristics described by the class map.

In this example, the router executes default behavior for the police command. (See the “Specifying a Committed Access Rate” section on page 10-8 for details.)

Router(config-pmap-c)# police 80000

Step 4
Exit policy map configuration mode.

Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)#

You have completed the creation of a QoS boilerplate, which can be assigned to an interface.
Assigning a QoS Boilerplate to an Interface

Use the `service-policy` command to assign a QoS boilerplate to an interface. In the following example, the policy map `lynx` is assigned to traffic that enters the gigabit Ethernet interface of an RPM-XF.

```
Router(config)# interface gigabitethernet 1/0
Router(config-if)# service-policy input lynx
```

Class Map Commands

This section describes commands for creating and modifying class maps.

You can have up to 2048 policy maps. You can have up to 32 class maps per policy map. However, you can only have up to a total of 256 class maps, including the class-default. The same class map can be applied to different policy maps.

Creating a Class Map

You can create a class map and enter class-map configuration mode by entering the `class-map` command.

```
class-map [match-any | match-all] class-map-name
```

The default value is `match-all`.

Use the `no class-map` command to delete a class map.

Cisco IOS software supports a maximum of 255 unique class maps.

In the following example, a class-map named `mink` is created. In the example, the default value of `match-all` is used.

```
Router(config)# class-map mink
Router(config-cmap)#
```

Matching Attributes

Use the `match` command to define the characteristics of the packets that belong to the class map.

```
match match_statement
[no] match match_statement
```

The `match` command `match_statement` is one of the following values:

- `match [not] access-group number`—Specifies that the packet must (or must not) be permitted by the access group whose number is from 1 to 2699.
**Policy Map Commands**

This section describes commands for creating and modifying policy maps.

You can have up to 2048 policy maps. You can have up to 32 class maps per policy map. However, you can only have up to a total of 256 class maps, including the class-default. The same class map can be applied to different policy maps.

**Creating a Policy Map**

You can create a policy map and enter policy-map configuration mode by entering the `policy-map` command from global configuration mode.

```
policy-map policy-map-name
[no] policy-map policy-map-name
```

The `policy-map-name` can be any word or number.
Use the `no` form of the command to remove a policy map.

In the following example, a policy map named `lynx` is created.

```
Router(config)# policy-map lynx
```

```
Router(config-pmap)#
```

## Assigning a Class to a Policy Map

Use the `class class-map-name` command from policy-map configuration mode to assign a class map to a policy map.

```
class class-map-name
[no] class class-map-name
```

The `class-map-name` is the name assigned to the class map.

Use the `no` form of the command to remove a class.

You can use a special class map name called `class-default` on a given interface to assign QoS policies to all packets that are not already described in the policy map by a class of a different name.

After you enter the `class class-map-name` command, you enter policy-map class configuration mode, in which you can enter QoS policies.

**Tip**

A packet is processed by a policy map as soon as a match is found. When you assign class names to a policy map, assign the first name to the class that is most likely to be used. This can improve QoS performance.

In the following example, the class map named `mink` is assigned to the policy map named `lynx`.

```
Router(config)# policy-map lynx
Router(config-pmap)# class mink
Router(config-pmap-c)#
```

In the following example, the default class map is assigned to the policy map named `lynx`.

```
Router(config)# policy-map lynx
Router(config-pmap)# class class-default
Router(config-pmap-c)#
```
Specifying a Committed Access Rate

To specify a committed access rate, enter the `police` command while you are in policy-map class configuration mode. You can use this command to control low-priority traffic, so that an interface has more bandwidth for high-priority traffic or to enforce a specific rate on an interface.

You can specify the rate commitment as either a bit rate or as a percentage of the bandwidth. When using the IP-RAN feature, always specify a CIR percentage so you can take advantage of the dynamic bandwidth feature. For more information, see the “Enabling IP Radio Access Network” section on page 10-29. The following command summaries show the two command forms.

```
police bps [burst-normal] [burst-max] [conform-action action exceed-action action]

police cir percent percent [bc conform-burst-in-msec] [be peak-burst-in-msec]
[conform-action action exceed-action action]
```

```
no police
```

The RPM-XF does not support the `pir percent` or `violate-action` keywords for the police command.

### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bps</code></td>
<td>Average rate in bits per second. Valid values are 8000 to 200000000.</td>
</tr>
<tr>
<td><code>normal-burst</code></td>
<td>(Optional) Normal burst size in bytes. Valid values are 1000 to 51200000. The default normal burst size is 1500 bytes.</td>
</tr>
<tr>
<td><code>max-burst</code></td>
<td>(Optional) Excess burst size in bytes. Valid values are 1,000 to 51200000.</td>
</tr>
<tr>
<td><code>conform-action</code></td>
<td>Indicates the action that should be taken if the rate or percent is not exceeded. See Table 10-1 for a list of actions.</td>
</tr>
<tr>
<td><code>exceed-action</code></td>
<td>Indicates the action that should be taken if the rate or percent is exceeded. See Table 10-1 for a list of actions.</td>
</tr>
<tr>
<td><code>cir</code></td>
<td>Committed information rate (CIR). Indicates that the CIR will be used for policing traffic.</td>
</tr>
<tr>
<td><code>percent</code></td>
<td>Specifies that percent of bandwidth will be used for calculating the CIR.</td>
</tr>
<tr>
<td><code>percent</code></td>
<td>Specifies the bandwidth percentage. Valid range is a number from 1 to 100.</td>
</tr>
<tr>
<td><code>bc</code></td>
<td>(Optional) Conform burst (bc) size used by the first token bucket for policing traffic.</td>
</tr>
<tr>
<td><code>conform-burst-in-msec</code></td>
<td>(Optional) Specifies the bc value in milliseconds (ms). Valid range is a number from 1 to 2000.</td>
</tr>
</tbody>
</table>
If you enter only `police bps` at the command line, the following default behavior occurs: traffic that conforms to the bps value is transmitted and traffic that exceeds the bps value is dropped.

Use the `no` form of the command to disable policing.

In the following example, CAR is assigned to the class named `mink`.

```plaintext
Router(config)# policiemap lynx
Router(config-pmap)# class mink
Router(config-pmap-c)# police 720000 90000 90000 conform-action transmit exceed-action drop
```

### Enabling Weighted Random Early Detection

Use the `random-detect` command to enable weighted random early detection (WRED), which randomly discards packets during congestion based on IP precedence settings. The `random-detect` command enables a WRED drop policy for a traffic class that includes a bandwidth guarantee.

| Note | The bandwidth must be set before you can enable WRED (see Bandwidth Reservation and Low-Latency Priority Queueing, page 10-10). |

| Note | On the ATM interface, you can only use WRED on a variable bit rate (VBR) PVCs. You cannot use WRED on PVCs configured for an unspecified bit rate (UBR). |

```plaintext
random-detect [ewc value | prec prec-value min-value max-value mark-denom] [no] random-detect [ewc value | prec prec-value min-value max-value mark-denom]
```
Tip

In most cases, the benefits of WRED can be best realized if you use the `random-detect` keyword without arguments.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ewc value</code></td>
<td>Exponential-weighting-constant (ewc) value allows you to modify the default method that random-detect uses to calculate average queue size. Random-detect determines the average queue size based on the current queue length and the last average queue length. You can specify a value from 1 to 16.</td>
</tr>
<tr>
<td></td>
<td>- The higher the value, the more dependent the average is on the historical average, making WRED slow to react to changing traffic conditions that may be only temporary.</td>
</tr>
<tr>
<td></td>
<td>- The lower the value, the less dependent the average is on the historical average, making WRED more sensitive to rapidly changing traffic conditions.</td>
</tr>
<tr>
<td><code>prec</code></td>
<td>Specify the precedence values according to the information in the following table.</td>
</tr>
</tbody>
</table>

Use the `no` form of the command to disable WRED.

The following example shows the implementation of WRED.

```
Router(config)# policy-map lynx
Router(config-pmap)# class class-default
Router(config-pmap-c)# random-detect
```

### Bandwidth Reservation and Low-Latency Priority Queueing

This section explains how to configure bandwidth reservation and low-latency priority. These queueing methods let you offer differentiated service to customers.

The RPM-XF typically uses a single queue for packets from all traffic streams waiting for the link to transmit them in the order of their arrival. This method is simple, efficient, and offers optimal average delay per packet because it always uses the entire link bandwidth. But the single queue method does not distinguish among different traffic streams—the more traffic in a stream, the larger its share of the link bandwidth.
Bandwidth reservation divides the link bandwidth among the different traffic streams into multiple queues, with each queue receiving its fair share of the link bandwidth divided among all non-empty queues. You do not waste bandwidth associated with an empty queue, and by dividing the unused bandwidth to the queues with packets to send, multiple queueing has the same average delay per packet as the single queue scheme, with the advantage of fairness.

Low-latency priority queueing lets you assign a guaranteed minimum bandwidth to one queue to minimize the packet-delay variance for delay-sensitive traffic, such as live voice and video.

**Note**

Bandwidth and low-latency priority cannot be combined in the same class.

### Bandwidth Reservation Queueing

Use the `bandwidth` command to create multiple class queues.

```
bandwidth rate-in-kbps
[no] bandwidth
```

The `rate-in-kbps` parameter is a value in the range from 8 to 2,000,000 representing between 1% to 99% of the link bandwidth.

Use the `no` form of the command to disable bandwidth queueing.

The following sample configuration creates two class queues.

- A 18 kbps queue for packets with IP precedence bit settings of 1, 2, 3, or 4.
- A 54 kbps queue for packets with IP precedence bit settings of 5, 6, or 7.

Assuming that the interface has 128 kbps bandwidth, the two class queues receive 25% and 62% of the interface bandwidth. All other traffic, including IP precedence 0, receives the rest of the bandwidth—8 kbps or 13%.

```
Router# enable
Router# configure terminal
Router(config)# class-map city
Router(config-cmap)# match ip precedence 1 2 3 4
Router(config-cmap)# class-map boston
Router(config-cmap)# match ip precedence 5 6 7
Router(config-cmap)# policy-map precedence-queues
Router(config-pmap)# class city
Router(config-pmap-c)# bandwidth 16
Router(config-pmap-c)# class boston
Router(config-pmap-c)# bandwidth 40
Router(config-pmap-c)# interface switch1/1
Router(config-if)# service-policy output precedence-queues
Router(config-if)# end
```

The actual throughput of a queue may be higher when one or more of the other queues on the link are idle.

### Low-Latency Priority Queueing

Low-latency priority queueing lets you assign a specified share of the link bandwidth to one queue that receives priority over all others. Low-latency priority queueing minimizes the packet-delay variance for delay-sensitive traffic, such as live voice and video.

Use the `priority` command to create a low-latency priority queue.

```
priority rate-in-kbps
```
[no] priority

The rate-in-kbps parameter is a value in the range from 8 to 2,000,000, representing the guaranteed minimum bandwidth.

Use the no form of the command to disable priority queueing.

The following sample configuration creates a priority queue for voice traffic, and applies it to interface switch1.1.

Router# enable
Router# configure terminal
Router(config)# class-map voice
Router(config-cmap)# match ip rtp 2000 2000
Router(config-cmap)# policy-map voice-queue
Router(config-pmap)# class voice
Router(config-pmap-c)# priority 56
Router(config-pmap-c)# interface switch1:1
Router(config-if)# service-policy output voice-queue
Router(config-if)# end

The actual throughput of a priority queue may be higher than the minimum because it allocates the entire link bandwidth to a priority queue if all the other queues on the link are empty.

**Generic Traffic Shaping**

The RPM-XF uses traffic shaping as a mechanism to control or modify the flow of traffic on an interface to meet the requirements of a remote site, or to conform to a service rate that is provided on that interface. Generic Traffic Shaping (GTS) supports traffic shaping on all interfaces regardless of the encapsulation of the interface.

There are two implementations of traffic shaping in the current Cisco IOS software: GTS and Frame Relay Traffic Shaping (FRTS). This section describes GTS.

---

**Note**

The RPM-XF does not support Frame Relay.

**Note**

Use the traffic shape command in policy map class configuration mode. It is not supported in interface configuration mode.

The traffic **shape** command limits the throughput equal to rate-in-kbps.

```shape rate-in-kbps
[no] shape
```

The rate-in-kbps parameter is a value in the range from 56 to 2,000,000, representing the maximum throughput allowed.

Use the no form of the command to disable traffic shaping.

In the following sample configuration, the traffic shape is set to a throughput of 100.

```Router(config-pmap-c)# shape 100
Router(config-pmap-c)#
```
Specifying a Queue Limit

This section describes how to specify the number of packets held by the queue. Increase the queue limit to reduce the number of packets dropped due to temporary congestion on the assigned interface. Queue limit operates on the default packet drop method of congestion management. You cannot use the `queue limit` command on ATM PVCs configured for unspecified bit rate (UBR).

Note
On the ATM interface, you can only apply queue limits on variable bit rate (VBR) PVCs.

```
queue-limit packets
[no] queue-limit
```

The `packets` parameter is a number of packets from 32 to 16,384 in powers of 2 (for example, 64, 128, 256).

Note
If the number of packets specified is not a power of 2, the number entered is automatically rounded up to a power of 2. For example, if the number of packets is entered as 60, it will be rounded up to 64.

Use the `no` form of the command to return the queue limit to its default value.

Use the `show interface` command to determine the current queue limit. If you set the queue limit to a high value, this may reduce the number of packet buffers available to other interfaces.

In the following example, the queue limit is set to 256 packets:

```
Router(config)# policy-map lynx
Router(config-pmap)# class class-default
Router(config-pmap-c)# queue-limit 256
```

Applying Set Values

The `set` command allows you to mark bit values that can be used by other routers to manage QoS.

```
set {ip {dscp value | precedence value} | qos-group value | atm-clp | mpls experimental value}
[no] set {ip {dscp value | precedence value} | qos-group value | mpls experimental value}
```

Externally visible `values` are the following.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dscp</td>
<td>A value between 0 and 63.</td>
</tr>
<tr>
<td>precedence</td>
<td>A precedence bit setting between 0 and 7. 0 typically represents low-priority</td>
</tr>
<tr>
<td></td>
<td>traffic; 7 represents high-priority traffic.</td>
</tr>
</tbody>
</table>

Internally visible `values` are the following.
Service-Policy Command

To associate a policy map with an interface, use the `service-policy` command.

```
service-policy [input | output] name
{no} service-policy [input | output] name
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>input</td>
<td>Incoming traffic on an interface.</td>
</tr>
<tr>
<td>output</td>
<td>Outgoing traffic on an interface.</td>
</tr>
<tr>
<td>name</td>
<td>Name of a policy map.</td>
</tr>
</tbody>
</table>

**Note** The bandwidth, low-latency priority, random-delete, queue limit, and shape parameters are used with `output` only, and are ignored when using the `input` argument.

Use the `no` form of the command to remove a service policy from an interface.

No more than two service policies can be associated with an interface, one for input and one for output.

On the ATM interface, you can only apply a policy map on a PVC.

In the following example, the policy map `lynx` is applied to the incoming traffic on an interface of the Gigabit Ethernet line card.

```
Router(config)# policy-map lynx
Router(config-pmap)# class mink
Router(config-pmap-c)# set ip precedence 7
Router(config-pmap-c)# set qos-group 8
```

**Note** CEF switching must be on to use the `service-policy` command.
In the following example, a policy map is applied to an ATM PVC.

```
Router(config)# interface switch1.1
Router(config-if)# pvc 0/101
Router(config-if-atm-vc)# service-policy input lynx
```

### Show Commands

This section lists show commands you can use to get information about class maps and policy maps.

#### show policy map

This command displays the configuration of one or all policy maps and lists information about the configurations. For example,

```
Router# show policy-map lynx
Policy Map lynx
 class mink
 set qos-group 8
Policy Map jaguar
 class class-default
 random-detect
 random-detect exponential-weighting-constant 9
 random-detect precedence 0 16 32 10
 random-detect precedence 1 18 32 10
 random-detect precedence 2 20 32 10
 random-detect precedence 3 22 32 10
 random-detect precedence 4 24 32 10
 random-detect precedence 5 26 32 10
 random-detect precedence 6 28 32 10
 random-detect precedence 7 30 32 10
```

#### show policy-map interface

This command displays statistics of a policy map on one or all interfaces. This example shows statistics for a particular serial interface.

```
Router# show policy-map interface [pos1/0]
Pos1/0
 service-policy input: lynx
 class-map: mink (match-all)
  0 packets, 0 bytes
  5 minute rate 0 bps
 match: access-group 3
 set:
   qos-group 8
```
show class-map

This command lists the class maps and displays their match statements. For example,

Router# show class-map mink
Class Map match-all mink (id 3)
  Match access-group 3

Class Map match-all pink (id 4)
  Match access-group 23
  Match qos-group 32

Class Map match-any class-default (id 0)
  Match any

Class Map match-all customer_pri (id 2)

show vlans

This command can list up to 1000 virtual LAN subinterfaces. For example,

Router# show vlans
Virtual LAN ID: 1 (IEEE 802.1Q Encapsulation)
VLAN Trunk Interface: GigabitEthernet1/0
Protocols Configured: Address: Received: Transmitted:
  IP       200.1.1.1      18      273894058

Quality of Service Policy Propagation Example Using Border Gateway Protocol

Quality of Service (QoS) Policy Propagation using Border Gateway Protocol (QPPB) allows you to classify packets by IP precedence based on BGP community lists, BGP autonomous system paths, and access lists. After a packet has been classified, you can use other QoS features such as committed access rate (CAR) and weighted random early detection (WRED) to specify and enforce policies to fit your business model.

The example below shows how to do the following.

1. Create route maps to match BGP community lists, access lists, and BGP AS paths.
2. Apply IP precedence to routes learned from neighbors.

In this example, the RPM-XF learns routes from autonomous system (AS) 10 and AS 60. QoS policy is applied to all packets that match the defined route maps. Any packets from the RPM-XF to AS 10 or AS 60 are sent to the appropriate QoS policy (see Figure 10-2).
RPM-XF Configuration

Router(config)# router bgp 30
Router(config)# table-map precedence-map
Router(config-router)# neighbor 20.20.20.1 remote-as 10
Router(config-router)# neighbor 20.20.20.1 send-community
Router(config-router)# neigh 20.20.20.1 route-map precedence-map out
Router(config)# ip bgp-community new-format

Match community 1, set the IP precedence to priority, and set the QoS group to 1.

Router(config)# route-map precedence-map permit 10
Router(config-route-ma)# match community 1
Router(config-route-ma)# set ip precedence priority
Router(config-route-ma)# set ip qos-group 1

Match community 2 and set the IP precedence to immediate.

Router(config)# route-map precedence-map permit 20
Router(config-route-ma)# match community 2
Router(config-route-ma)# set ip precedence immediate

Match community 3 and set the IP precedence to Flash.

Router(config)# route-map precedence-map permit 30
Router(config-route-ma)# match community 3
Router(config-route-ma)# set ip precedence flash

Match community 4 and set the IP precedence to Flash-override.

Router(config)# route-map precedence-map permit 40
Router(config-route-ma)# match community 4
Router(config-route-ma)# set ip precedence flash-override

Match community 5 and set the IP precedence to critical.

Router(config)# route-map precedence-map permit 50
Router(config-route-ma)# match community 5
Router(config-route-ma)# set ip precedence critical

Match community 6 and set the IP precedence to internet.

Router(config)# route-map precedence-map permit 60
Router(config-route-ma)# match community 6
Router(config-route-ma)# set ip precedence internet

Match community 7 and set the IP precedence to network.

Router(config)# route-map precedence-map permit 70
Router(config-route-ma)# match community 7
Chapter 10  Configuring Quality of Service

Quality of Service Policy Propagation Example Using Border Gateway Protocol

Router(config-route-ma)# set ip precedence network

Match ip address access list 69 or match AS path 1, set the IP precedence to critical, and set the QoS group to 9.

Router(config)# route-map precedence-map permit 75
Router(config-route-ma)# match ip address 69
Router(config-route-ma)# match as-path 1
Router(config-route-ma)# set ip precedence critical
Router(config-route-ma)# set ip qos-group 9

For everything else, set the IP precedence to routine.

Router(config)# route-map precedence-map permit 80
Router(config-route-ma)# set ip precedence routine

Define the community lists.

Router(config)# ip community-list 1 permit 60:1
Router(config)# ip community-list 2 permit 60:2
Router(config)# ip community-list 3 permit 60:3
Router(config)# ip community-list 4 permit 60:4
Router(config)# ip community-list 5 permit 60:5
Router(config)# ip community-list 6 permit 60:6
Router(config)# ip community-list 7 permit 60:7

Define the AS path.

Router(config)# ip as-path access-list 1 permit ^10_60

Define the access list.

Router(config)# access-list 69 permit 69.0.0.0

Router B Running Configuration

RouterB(config)# router bgp 10
RouterB(config-router)# neighbor 30.30.30.1 remote-as 30
RouterB(config-router)# neighbor 30.30.30.1 send-community
RouterB(config-router)# neigh 30.30.30.1 route-map send_community out

RouterB(config)# ip bgp-community new-format

Match prefix 10 and set community to 60:1.

RouterB(config)# route-map send_community permit 10
RouterB(config-route-ma)# match ip address 10
RouterB(config-route-ma)# set community 60:1

Match prefix 20 and set community to 60:2.

RouterB(config)# route-map send_community permit 20
RouterB(config-route-ma)# match ip address 20
RouterB(config-route-ma)# set community 60:2

Match prefix 30 and set community to 60:3.

RouterB(config)# route-map send_community permit 30
RouterB(config-route-ma)# match ip address 30
RouterB(config-route-ma)# set community 60:3

Match prefix 40 and set community to 60:4.

RouterB(config)# route-map send_community permit 40
RouterB(config-route-ma)# match ip address 40
RouterB(config-route-ma)# set community 60:4
Match prefix 50 and set community to 60:5.
```
RouterB(config)# route-map send_community permit 50
RouterB(config-route-ma)# match ip address 50
RouterB(config-route-ma)# set community 60:5
```

Match prefix 60 and set community to 60:6.
```
RouterB(config)# route-map send_community permit 60
RouterB(config-route-ma)# match ip address 60
RouterB(config-route-ma)# set community 60:6
```

Match prefix 70 and set community to 60:7.
```
RouterB(config)# route-map send_community permit 70
RouterB(config-route-ma)# match ip address 70
RouterB(config-route-ma)# set community 60:7
```

For all others, set community to 60:8.
```
RouterB(config)# route-map send_community permit 80
RouterB(config-route-ma)# set community 60:8
```

Define the access lists.
```
RouterB(config)# access-list 10 permit 61.0.0.0
RouterB(config)# access-list 20 permit 62.0.0.0
RouterB(config)# access-list 30 permit 63.0.0.0
RouterB(config)# access-list 40 permit 64.0.0.0
RouterB(config)# access-list 50 permit 65.0.0.0
RouterB(config)# access-list 60 permit 66.0.0.0
RouterB(config)# access-list 70 permit 67.0.0.0
```

The following example shows how to configure several interfaces to classify packets based on the IP precedence and QoS group ID.
```
interface switch1.1
ip address 200.28.38.2 255.255.255.0
bgp-policy source ip-prec-map
no ip mroute-cache
no cdp enable
frame-relay interface-dlci 20 IETF

interface switch1.2
ip address 200.28.28.2 255.255.255.0
bgp-policy source qos-group
no ip mroute-cache
no cdp enable
```

**DSCP Marking on RPM-XF Management Interface**

Cisco IOS Release 12.4(15)T1 supports Differentiated Services Code Point (DSCP) or IP Precedence marking for Quality of Service (QoS) configurations on the RPM-XF management back cards. With this enhancement, the RPM-XF supports Layer 3 QoS on all back card interfaces, including the ATM, POS, and Gigabit Ethernet back cards as well as the Fast Ethernet management back card.
Limitations

The following limitations apply to the DSCP marking of management packets on the RPM_XF management back card:

- The RPM-XF does not support DSCP marking for the interface to the MGX switch cell bus.
- The RPM-XF management back card can be used for only management traffic, not data traffic.

DSCP Fields

The DSCP field is the six most significant bits of the Differentiated Services (DiffServ) byte of the IP header. DiffServ is similar to the earlier ToS byte, but clarifies the precedence levels and provides more granularity for control. The RPM-XF classifies packets and marks packets with either the IP Precedence or DSCP value. Other network devices that support Diffserv use the DSCP value in the IP header to select a per-hop behavior for packets and provide the appropriate QoS treatment.

The DiffServ byte has the following bits:

<table>
<thead>
<tr>
<th>DS5</th>
<th>DS4</th>
<th>DS3</th>
<th>DS2</th>
<th>DS1</th>
<th>DS0</th>
<th>ECN</th>
<th>ECN</th>
</tr>
</thead>
</table>

- DSCP—Six bits (DS5 to DS0).
- ECN—Two bits.

The ToS byte has the following bits:

<table>
<thead>
<tr>
<th>P2</th>
<th>P1</th>
<th>P0</th>
<th>T2</th>
<th>T1</th>
<th>T0</th>
<th>CU1</th>
<th>CU0</th>
</tr>
</thead>
</table>

- IP precedence – three bits (P2 to P0).
- Delay, Throughput and Reliability – three bits (T2 to T0).
- CU (Currently Unused) – two bits (CU1 to CU0).

The most significant three bits of the DSCP and ToS fields, converted to decimal, have the following meaning:

<table>
<thead>
<tr>
<th>Precedence Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Stays the same (link layer and routing protocol keep alive)</td>
</tr>
<tr>
<td>6</td>
<td>Stays the same (used for IP routing protocols)</td>
</tr>
<tr>
<td>5</td>
<td>Express Forwarding (EF)</td>
</tr>
<tr>
<td>4</td>
<td>Class 4</td>
</tr>
<tr>
<td>3</td>
<td>Class 3</td>
</tr>
<tr>
<td>2</td>
<td>Class 2</td>
</tr>
<tr>
<td>1</td>
<td>Class 1</td>
</tr>
<tr>
<td>0</td>
<td>Best effort</td>
</tr>
</tbody>
</table>
Chapter 10      Configuring Quality of Service

The RPM-XF prioritizes traffic by these precedence levels, so assign a level that is appropriate for your management traffic.

Configuring DSCP Marking

You configure DSCP marking on the RPM-XF as you do with other Cisco routers:

**Step 1**
Classify traffic that you want to configure for QoS. See Classifying Traffic, page 10-21.

**Step 2**
Create a DSCP marking policy for the traffic class. See Creating DSCP Marking Policy, page 10-21.

**Step 3**
Apply the DSCP marking policy to the interface. See Applying Policy to Interface, page 10-22.

**Step 4**
Verify the policy enforcement. See Verifying Policy Enforcement, page 10-22.

Classifying Traffic

Packet classification requires a traffic descriptor that categorizes a packet within a specific group. This makes the group accessible for QoS handling in the network. Using packet classification, you can partition network traffic into multiple priority levels or a class of service (CoS).

The following example shows how to create a class-map called `host1-match` that matches all traffic from host 10.76.29.200. The traffic from this host is important because it manages the RPM-XF management back card.

```
Router# config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# access-list 1300 permit 10.76.39.200 0.0.0.0
Router(config)# class-map match-all host1-match
Router(config-cmap)# match access-group 1300
```

Creating DSCP Marking Policy

The DSCP can be set to a desired value at the edge of the network in order to make it easy for core devices to classify the packet and provide a suitable level of service.

The following example shows how to create a DSCP marking policy that sets the precedence of packets that match the `host1-match` criteria to DSCP class 3.

```
Router# config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# policy-map policy-mgmt-if
Router(config-pmap)# class host1-match
Router(config-pmap-c)# set ip dscp cs3
Router(config-pmap-c)# end
Router# show running-config policy-map policy-mgmt-if
Building configuration...

Current configuration : 73 bytes

policy-map policy-mgmt-if
  class host1-match
    set ip dscp cs3

```
Applying Policy to Interface

For a policy to take affect, you must apply it to an interface. For the RPM-XF management back card, this is a FastEthernet interface.

The following example shows how to apply the `policy-mgmt-if policy-map` to FastEthernet 2/0:

```
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet2/0
Router(config-if)#service-policy output policy-mgmt-if
Router(config-if)#ip address 10.10.20.200 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#end
```

```
Router#show running-config interface FastEthernet 2/0
Building configuration...

Current configuration : 135 bytes
!
interface FastEthernet2/0
  ip address 10.10.20.200 255.255.255.0
duplex auto
  speed auto
  service-policy output policy-mgmt-if
end
```

Verifying Policy Enforcement

The following example shows statistics for the policy-map on FastEthernet 2/0:

```
Router#show policy-map interface FastEthernet 2/0 output
FastEthernet2/0

  Service-policy output: policy-mgmt-if

  Class-map: host1-match (match-all)
    0 packets, 0 bytes
    5 minute offered rate 0 bps, drop rate 0 bps
    Match: access-group 1300
    QoS Set
dscp cs3
    Packets marked 0

  Class-map: class-default (match-any)
    0 packets, 0 bytes
    5 minute offered rate 0 bps, drop rate 0 bps
    Match: any
```

Versatile Traffic Management System

Versatile Traffic Management System (VTMS) on the RPM-XF allows bandwidth sharing between VCs (virtual channels). When a VC is idle, its bandwidth can be used by other VCs. It allows all VCs to share the same VTMS link and supports ATM and either POS (Packet Over SONET) or GigE links.

VTMS on the RPM-XF uses a bandwidth divisor of 65535, and uses dummy full queues to handle traffic congestion. It allows packet dropping, including UBR (undefined bit rate) packet dropping.

VTMS uses the flow bits in the packet header to suppress packet dequeuing.
There are two kinds of flow bit controls:
- software flow bits
- hardware queue statuses

A packet is enqueued if both the software flow bits and the hardware queue statuses indicate ready and is dequeued if both the software flow bits and the hardware queue statuses indicate congestion.

When a packet is enqueued it adds a flow bit to a flow bit table. The flow bit table is used to determine whether a line card is congested. When a line card is congested, VTMS creates a dummy full queue, which forces the packet to be dropped or dequeued.

---

**Note**
VTMS uses dummy full queues for UBR also; however, since RPM-XF drops headers in UBR packets, UBR packets are dropped if there is traffic congestion on the interface.

---

**Note**
VTMS on the RPM-XF is enabled by default.

### VTMS Buffer Management

In VTMS, it is important to recognize the difference between buffers and queues:
- Buffers are memory areas that only store the packets. RPM-XF has 128 MB of buffer memory.
- Queues are data structures that point to packets in the buffers in a specific order.

Packets are grouped by class and are queued in a first-in-first-out order (fifo). Packets are then directed to one of three possible path:
- fast path
- punt path
- drop path

The memory buffers can be configured by the administrator and allocated during initialization. The administrator can configure up to eight memory buffer pools, designated as pool 0 through pool 7. An example memory buffer pool allocation is as follows:
- pool 0: 9216 bytes–total 100
- pool 1: 4672 bytes–total 500
- pool 2: 1600 bytes–total 30000
- pool 3: 640 bytes–total 67671
- pool 4: 256 bytes–total 98173
- pool 5: 64 bytes–total 131000

**Buffer Management CLI Commands**
```
show pxf cpu buffers
show pxf cpu buffers leaked <pool no>
```
VTMS Queuing

Queues are data structures that point to packets in the buffer in a specific order based on the VTMS configuration. VTMS assign packets to one of the following two classes of queues:

- Work queues
- Packet queues

Queuing techniques depend on QoS features required, throughput, latency, and packet sizes. Queuing on the RMP-XF is optimized for low and high speed interfaces that experience small packets, low latency, and advanced QoS.

The VTMS scheduler determines how packets are directed. After packets are assigned to a class, VTMS scheduler directs them to one the following different types of queues based on the VTMS configuration.

- First-In-First-Out (FIFO)
- Fair Queuing
- Weighted Fair Queuing (WFQ)
- Class Based WFQ (CBWFQ)
- Low Latency Priority Queuing (LLQ)
- Custom Queuing

First-In-First-Out (FIFO) queuing is the highest priority type of queuing and is used for control traffic such as routing updates.

Fair Queuing services packets based on flow and packet sizes, so that smaller packets do not get stuck behind larger packets.

Weighted Fair Queuing (WFQ) services packets based on weight. The weight is assigned to each work queue based on the IP precedence value of the packets in that queue.

Class Based WFQ (CBWFQ) services classified traffic. Classified traffic is configured by the user. The weights are configured by VTMS based on bandwidth for that queue.

Low Latency Priority Queuing (LLQ) is an additional queue created on demand after it has been configured to do so. LLQ services classified packets that are sent to it and is used for small packets and voice.

Custom Queuing services packets in a round robin manner, based on specific user configuration information. Custom Queuing can service up to 16 queues.

Queuing CLI Commands

- `show pxf cpu queue <interface>` - summarized info
- `show pxf cpu queue <qid>` - detailed info including CIR, MIR, EIR, stats, etc.
- `show pxf cpu statistics qos <interface>`
- `show pxf cpu police <policy map>`

MultiLink PPP/Link Fragmentation Interleaving

MultiLink PPP/Link Fragmentation Interleaving (MLP/LFI) allows a large packet to be divided into smaller fragments so that excessive head of line blocking can be avoided for smaller packets such as VoIP packets.
On slow speed interfaces (slower than T1), a packet with maximum MTU (maximum transmission unit) can cause excessive head of line blocking in LLQs (low latency priority queues) especially in VoIP (Voice over IP) applications. The solution is to implement MLP/LFI on these interfaces.

The RPM-XF supports MLP/LFI on MLPPP interfaces and supports up to 200 MLP/LFI-enabled interfaces. MLP/LFI and PPP interfaces use the MLPPP (Multilink Point-to-Point Protocol) long sequence number fragment format headers.

MLP/LFI over multiple links in an MLPPP bundle is not supported. Receiving and reassembling out of sequence fragments is also not supported.

If a packet is dequeued, MLP/LFI will reschedule and retransmit each fragment separately. MLP/LFI will then reassemble the packet at the far end only after all the fragments have been received.

**MLP/LFI Configuration**

The following configuration commands can be used to configure MLP/LFI:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ppp multilink</td>
<td>Enable multilink on the interface.</td>
</tr>
<tr>
<td>ppp multilink framing</td>
<td>Enable multilink framing.</td>
</tr>
<tr>
<td>ppp multilink fragment-delay</td>
<td>Set the maximum delay (in milliseconds) between fragments. For example, you can configure a voice stream with a maximum delay of 20 milliseconds. The MLPPP will choose a fragment size based on this value.</td>
</tr>
<tr>
<td>ppp multilink interleave</td>
<td>Enable real-time packet interleaving on bundled transmissions.</td>
</tr>
<tr>
<td>match ip rtp</td>
<td>Configure which traffic will be prioritized for interleaving based on the starting-port-number or range-of-ports. These assignments can be used to map packets to a specific class.</td>
</tr>
</tbody>
</table>

Using the **match ip rtp** command is just one way of classifying traffic for interleaving. You can also use an access list. The **policy-map** command associates a class of traffic to a priority queue, and the **priority** command sets the priority of that class within the policy-map. The **service policy** then attaches that classification and action to an interface. For example:

```
class-map match-all VOIP
match ip rtp 16384 16383

class-map LESS_CRITICAL
match access-group 101
policy-map VOIP_PRI
class VOIP
priority 50
class LESS_CRITICAL
set ip precedence 5
interface sw1.100 point-to-point
pvc toostr01 0/58
vbr-nrt 406 406
protocol ppp Virtual-Template15
interface Virtual-Template15
bandwidth 320
ip address 10.16.0.105 255.255.255.252
ip tcp header-compression iphc-format
```
Configuring Internet Protocol Header Compression

Internet Protocol Header Compression (IPHC) increases the bandwidth utilization of PPP links when IP headers are extremely large or when the header and payload sizes are similar. PPP link utilization is important because it directly affects the number of calls that an aggregation node can handle. IPHC supports compressed Real Time Protocol (cRTP), compressed User Datagram Protocol (cUDP), and compressed Transport Control Protocol (cTCP).

The RPM-XF compresses IP datagrams on multi-link PPP or PPPoATM PVCs towards the endpoint router, and each PVC supports multiple IPHC data flows. Each flow represents a unique combination of IP/UDP headers. The RPM-XF supports 1000 flows per PVC and 200 PVCs per card. Inactive flows are released after a configurable timeout.

**IPHC Configuration**

An IHC-enabled flow sends the first packet with a full header, which is a special form of the normal IP+UDP and IPHC header. Subsequent packets are compressed using the cUDP or cRTP protocol, which replaces full headers with deltas for the IP/UDP/RTP header fields that differ from the full header, such as IPID, RTP Sequence, and RTP timestamp. Each of the deltas can be zero or non-zero, where zero indicates no header change. Packets with zero deltas contain just the essential fields and are of the smallest possible size. Packets with non-zero deltas vary in size depending on the number of deltas and the delta values themselves. The decompressor maintains a copy of the original full header and reconstructs packet headers. You can configure IPHC as follows:

- **Compressed packet with IPID delta (normal compression with all deltas)**
  
  When the decompressor receives a compressed packet with IPID delta, it reconstructs the packet header by adding the deltas to the respective fields of the saved uncompressed header. In the typical case, IPHC compresses the header to 2-5 bytes if 8-bit compression is used (add 1 additional byte if 16 bit compression is used and another 2 bytes if UDP checksum is present) for cUDP and 2-8 bytes if 8-bit compression is used (add 1 additional byte if 16 bit compression is used and another 2 bytes if UDP checksum is present) for cRTP.

  Use the `ip rtp header-compression iphc-format` command to enable this feature.

- **Compressed packet without IPID delta (cUDP without IPID delta)**

  When the decompressor receives a compressed packet, it reconstructs the packet using stored header information in the same way as described above, except that because the compressor does not encode the IPID delta, the delta is assumed to be 0 and the integrity of the IPID field is not guaranteed.

  Use the `ip rtp header-compression iphc-format` and `hw-module rpm ipran` commands to enable this feature. For more information, see Configuring the RPM-XF for IP-RAN, page 10-32.

**Note**     
This feature is an IPHC enhancement that is enabled only when IP-RAN is enabled. If IP-RAN is disabled, IPHC uses normal compression.
RPM-XF supports TCP decompression, but not TCP compression; TCP packets are always transmitted un-compressed. The decompression of compressed TCP packets is done by punting them to the route processor. A separate queue between the PXF and the route processor is used for compressed TCP traffic. TCP packets are dropped at speeds above 2.4 Mbps. Avoid carrying TCP traffic on an IPHC enabled interface, especially if the customer edge router can NOT stop TCP compression selectively.

**Compression Configuration**

By default, the cRTP protocol compresses all UDP and RTP packets. For those networks where compression of UDP packets apart from RTP is undesirable, you can selectively disable UDP compression. Use the `hw-module rpm udp-comp` command to enable or disable UDP compression when cRTP is enabled.

**Note** You cannot disable UDP compression when IP-RAN is enabled; IP-RAN only compresses cUDP packets.

**IPHC Command Summary**

The following CLI commands support IPHC:

<table>
<thead>
<tr>
<th>Config Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>clear ip rtp header-compression</code></td>
<td>Reset cRTP/cUDP statistics for the interface to zero.</td>
</tr>
<tr>
<td><code>clear ip tcp header-compression</code></td>
<td>Reset TCP decompression statistics for the interface.</td>
</tr>
<tr>
<td><code>hw-module rpm udp-comp</code></td>
<td>Enable UDP header compression. The <code>no</code> version of this command disables UDP compression. By default, cUDP is enabled when cRTP is enabled.</td>
</tr>
<tr>
<td><code>ip rtp compression-connections &lt;number&gt;</code></td>
<td>Specifies the total number of cRTP/cUDP header compression connections supported on the interface. The default is 16. The maximum is 1000. The <code>no</code> version of this command restores the default—16.</td>
</tr>
<tr>
<td><code>ip rtp header-compression iphc-format</code></td>
<td>Enable cRTP/cUDP header compression using iphc-format on an interface. The <code>no</code> version of this command disables header compression.</td>
</tr>
<tr>
<td><code>ppp iphc max-time</code></td>
<td>Set the timeout value for IPHC flows. The default time is 5 seconds. The <code>no</code> version of this command restores the default timeout.</td>
</tr>
<tr>
<td><code>show pxf cpu queue &lt;qid&gt;</code></td>
<td>Use the dedicated Queue ID (qid) for cTCP queue.</td>
</tr>
<tr>
<td><code>show pxf cpu queue RP</code></td>
<td>Show the cTCP packets punted to the route processor.</td>
</tr>
<tr>
<td><code>show pxf cpu statistics crtp [interface]</code></td>
<td>Show all PXF IPHC statistics for an interface. The statistics of any processing done by the route processor will not be reflected in this information.</td>
</tr>
</tbody>
</table>
### Table 10-2 Configuration Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show int &lt;interface&gt; rpmxf-iphc-db</code></td>
<td>Show interface IPHC database for debugging.</td>
</tr>
<tr>
<td><code>show ip rtp header-compression [interface]</code></td>
<td>Show all statistics for the interface.</td>
</tr>
<tr>
<td><code>show ip tcp header-compression [interface]</code></td>
<td>Show TCP decompression statistics for the interface.</td>
</tr>
</tbody>
</table>

The `show ip rtp header-compression` and `show ip tcp header-compression` commands display the following IPHC statistics:

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rcvd:</strong></td>
<td></td>
</tr>
<tr>
<td>total</td>
<td>Total packets processed by the decompressor.</td>
</tr>
<tr>
<td>compressed</td>
<td>Compressed cRTP/cUDP packets received.</td>
</tr>
<tr>
<td>status msgs</td>
<td>Context status messages received. This is sent by the decompressor when the compressed packet sequence number contained in its header is different from the one expected by the de-compressor. Also sent when the flow has timed out on the de-compressor and the compressor has not sent a full-header in response to the timeout.</td>
</tr>
<tr>
<td>dropped</td>
<td>Indicates the number of compressed packets that were dropped because of errors.</td>
</tr>
<tr>
<td><strong>Sent:</strong></td>
<td></td>
</tr>
<tr>
<td>total</td>
<td>Total packets processed by the compressor.</td>
</tr>
<tr>
<td>compressed</td>
<td>Compressed cRTP/cUDP packets sent</td>
</tr>
<tr>
<td>status msgs</td>
<td>Context status messages sent. This is sent by the decompressor when it finds issue with the sequence number of the compressed packet received.</td>
</tr>
<tr>
<td><strong>Connect:</strong></td>
<td></td>
</tr>
<tr>
<td>collisions</td>
<td>Number of uncompressed packets sent when a free connection ID could not be found after retries.</td>
</tr>
<tr>
<td>rx slots, tx slots</td>
<td>Indicates the number of cRTP/cUDP connections on the virtual-access interface. This number represents the final negotiated value for either PPPoATM or Multi-link PPP.</td>
</tr>
</tbody>
</table>
IPHC Examples

The following examples display IPHC information.

**show pxf cpu statistics crtp [interface]**

Interface Virtual-Access3:
- Rcvd: compressed : 0 pkts / 0 bytes
- fullheader : 0 pkts / 0 bytes
- dropped : 0 pkts
- cs (status) : 0 pkts
- Sent: compressed : 0 pkts / 0 bytes
- fullheader : 0 pkts / 0 bytes
- uncompressed: 0 pkts / 0 bytes
- cs (status) : 0 pkts
- Collisions : 0 pkts
- Punted to RP : 0 pkts (IP Options/RTP ext/CSRC)
- Compressed TCP in : 0 pkts
- Max CID : 1000
- Cids in use : 0
- Timeout (compr) : 9
- Timeout (decompr) : 8

**show int <interface> rpmxf-iphc-db**

Interface : Virtual-Access3
- IPHC enabled: yes IPHC id: 1 vcci: 15 states: 0 hashMask: 0x3E8

Tx stats in shadow memory:
- compressedout :pkts = 0 , bytes = 0
- uncompressedout :pkts = 0 , bytes = 0
- fullheaderout :pkts = 0 , bytes = 0
- cs_packet_rcvd 0 num_cid_collisions 0

Rx stats in shadow memory:
- compressedin :pkts = 0 , bytes = 0
- fullheaderin :pkts = 0 , bytes = 0
- compressed Tcp_in :pkts = 0
- cs_packet_sent 0 punted(IP options/RTP ext/CSRC list) 0
- tossed packets(bad CRC) 0
- IPHC enabled on PXF(read from PXF): yes

Enabling IP Radio Access Network

IP Radio Access Network (IP-RAN) is a collection of features that optimize IP communications for the radio access network. IP-RAN has the following features:

- Disable sending IP ID field delta in IPHC packet flows
- Suppress CS packet for cUDP sequence number mismatch
- Flow expiration timer
- cUDP compression only
- Dynamic bandwidth negotiation

These features primarily optimize bandwidth utilization for the radio access network. This is important for voice traffic over WAN links, such as T1/E1 lines, where cost is a significant factor.
IPPRAN Command Summary

The following CLI commands support IP-RAN:

**Table 10-4** IP-RAN Configuration Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hw-module rpm ipran</td>
<td>Enable IP-RAN features. The no version of this command disables IP-RAN.</td>
</tr>
<tr>
<td>ppp iphc max-time</td>
<td>Set the timeout value for IPHC flows. The no version of this command restores the default timeout. This is a generic feature that applies to all IPHC flows, not just IP-RAN.</td>
</tr>
<tr>
<td>show rpm ipran</td>
<td>Display IP-RAN operational status, which is either enabled or disabled.</td>
</tr>
<tr>
<td>show ip rtp header-compression virtual-access</td>
<td>Display header compression statistics.</td>
</tr>
</tbody>
</table>

**hw-module rpm ipran**

To enable IP Radio Access Network (IP-RAN) features on a Route Processor Module (RPM-XF), use the **hw-module rpm ipran** command in global configuration mode. To disable IP-RAN, use the no form of this command.

```
hw-module rpm ipran
no hw-module rpm ipran
```

This command enables IP-RAN features on an RPM-XF card. These features optimize multi-link PPP connections through the MPSM to help fully utilize radio access networks.

**ppp iphc max-time**

To specify the maximum amount of time to wait before expiring an IPHC flow, use the **ppp iphc max-time** command in interface configuration mode. To return to the default value, use the no form of this command.

```
ppp iphc max-time length-of-time
no ppp iphc max-time
```

*length-of-time* Specifies the number of seconds to wait before expiring an IPCH flow. The amount of time can be in the range of 0 to 255 seconds. The default value is 5 seconds.
The `ppp iphc max-time` command can improve IPHC performance by expiring the flow context ID for flows that have become idle. If there has been no activity on a flow for `length-of-time`, the flow context ID can be reused by a new flow. This command applies to all IPHC flows, not just the IP-RAN solution. This command implements the same functionality as the `ip header-compression max-time` command.

**show rpm ipran**

To display the operational status of the IP Radio Access Network (IP-RAN) feature, use the `show rpm ipran` command in privileged EXEC mode.

```
show rpm ipran
```

This command displays the operational status of IP-RAN, which is either enabled or disabled.

**show ip rtp header-compression**

To display Compressed Real-Time Transport Protocol (CRTP) statistics, use the `show ip rtp header-compression` command in privileged EXEC mode.

```
show ip rtp header-compression [detail] [interface-type interface-number]
```

<table>
<thead>
<tr>
<th><code>detail</code></th>
<th>(Optional) Displays details of each connection.</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>interface-type</code></td>
<td>(Optional) The interface type and number. For the RPM-XF, enter</td>
</tr>
<tr>
<td><code>interface-number</code></td>
<td><code>virtual-access</code> and the interface number.</td>
</tr>
</tbody>
</table>

Enter this command to retrieve information regarding RTP header compression on a specific interface. When you specify the `detail` keyword, the following information is displayed.

**Table 10-5 Virtual-Access Statistics**

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Header</td>
<td>The maximum size of the full header that can be sent on the link. A full header is the IP packet sent out uncompressed. It contains additional information, like Context ID used for the flow. Not supported on the RPM-XF.</td>
</tr>
<tr>
<td>Max Time</td>
<td>The maximum time between sending of full headers for a particular flow. After this time a full-header must be sent out for the flow. Not supported on the RPM-XF.</td>
</tr>
<tr>
<td>Max Period</td>
<td>The maximum number of compressed packets after which a full header must be sent for a particular flow. Not supported on the RPM-XF.</td>
</tr>
<tr>
<td>Feedback</td>
<td>Enable sending and receiving of CS packets - CS packets are used as a feedback mechanism by the decompressor to inform the compressor about packet loss.</td>
</tr>
<tr>
<td>Tx Context</td>
<td>The number of flows currently in use for the Tx direction.</td>
</tr>
<tr>
<td>Rx Context</td>
<td>The number of flows currently in use for the Rx direction.</td>
</tr>
</tbody>
</table>
Configuring IP-RAN

The IP-RAN solution utilizes the RPM-XF and MPSM cards. The MPSM card connects the RPM-XF router to access routers over MLPPP (Multilink PPP) links (see Figure 10-3 IP-RAN Solution).

Figure 10-3 IP-RAN Solution

The connection between the MPSM and RPM-XF is an ATM PVC that uses PPPoATM encapsulation. To create the slave connection at the RPM-XF, you need bandwidth information for the MLPPP bundle. To create the master connection at the MPSM, you need the NSAP address and VPI/VCI used at the RPM.

This section explains the RPM-XF configuration procedure in detail, and summarizes the MPSM configuration procedure.

Configuring the RPM-XF for IP-RAN

IP-RAN configuration includes the following tasks:

- Define QoS Service Policy
- Enable and Configure IP-RAN
- Configure PVC

Define QoS Service Policy

To define the QoS service policy, perform the following steps:

Step 1 Define QoS classes for the data types in your network. The following example shows typical definitions for voice and data classes.

```
Router(config)# class-map <data>
Router(config-cmap)# match ip precedence <0>
Router(config-cmap)# class-map <voice>
Router(config-cmap)# match ip precedence <1>
```
Step 2 Configure a policy map. Specify class bandwidths as a percentage to fully utilize the dynamic bandwidth feature. The following example shows a typical policy.

```
Router(config)# policy-map <ipran-policy-name>
Router(config-pmap)# class <voice>
Router(config-pmap-c)# priority
Router(config-pmap-c)# police cir percent <1-100>
Router(config-pmap)# class <data>
Router(config-pmap-c)# bandwidth percent <1-100>
```

Enable and Configure IP-RAN

To enable and configure IP-RAN, perform the following steps:

**Step 1** Enable IP-RAN feature set.

```
Router(config)# hw-module rpm ipran
```

**Step 2** Configure virtual templates for PVC endpoints. These templates enable and configure IPHC.

```
Router(config)# interface VirtualTemplate <1>
Router(config-if)# ip address <address> <mask>
Router(config-if)# ip tcp header-compression iphc-format
Router(config-if)# ppp iphc max-time <1-255>
Router(config-if)# ip rtp header-compression iphc-format
Router(config-if)# ip rtp compression-connections <1-1000>
```

**Step 3** Configure Switch1 for SAR-based QoS. The dynamic bandwidth feature requires this mode.

```
Router(config)# interface Switch1
Router(config-if)# atm sar-based-cbwfq
```

Configure PVC

For the IP-RAN solution, you connect the RPM-XF to the MPSM-16-T1E1 with a PVC. When IP-RAN is enabled, the following PVC restrictions apply:

- Cisco PPP over AAL5 encapsulation is compatible with the MPSM-16-T1E1 card only. To use Cisco PPP over AAL5 encapsulation with FRSM or MPSM(ASAP) cards, disable IP-RAN.
- The PVCs cannot be configured for MLP-LFI.
- The dynamic bandwidth feature applies to all PVC slave endpoints on the RPM-XF.
- The service policy that is attached to the PVC must have bandwidths configured as a percent, rather than an absolute value.

To create a PVC between the RPM-XF and MPSM-16-T1E1 for the IP-RAN solution, perform the following steps:

**Step 1** Create a point-to-point subinterface on Switch1. The RPM-XF routes traffic to the endpoint routers through this interface.

```
Router(config)# interface Switch1.1 point-to-point
```

**Step 2** Add and configure a PVC on this subinterface. Configure the service type as either vbr-rt or vbr-nrt, and specify bandwidth (PCR and SCR) and burst size (MBS) of the corresponding MLPPP bundle. Apply Cisco PPP over AAL5 encapsulation and the appropriate virtual template. Finally, apply the policy-map created in the “Define QoS Service Policy” section on page 10-32.
Router(config-subif)# pvc <vpi/vci>
Router(config-if-atm-vc)# vbr-nrt <pcr> <scr> <mbs>
Router(config-if-atm-vc)# encapsulation aal5cisco802.1q Virtual-Template
Router(config-if-atm-vc)# service-policy output <ipran-policy-name>

Step 3 Create a slave connection endpoint and display its information.

Router(config-subif)# switch connection vcc <vpi> <vci> master remote

Router(config-if-atm-vc)# show switch connection vcc <vpi> <vci>
-----------------------------------------------------------------------------------------------
  Alarm state           : No alarm
  Local Sub-Interface   : 1
  Local VPI             :
  Local VCI             : 101
  Remote NSAP address   : default
  Local NSAP address    :
    47.009181000000164444b61.000001011802.00
  Remote VPI            : 0
  Remote VCI            : 0
  Routing Priority      : N/A
  Max Cost              : N/A
  Preferred Route Id    : N/A
  Directed Route        : N/A
  Percent Util          : 100
  Remote PCR            : 34400
  Remote SCR            : 34400
  Remote MBS            : 1024
  Local PCR             : 34400
  Local SCR             : 34400
  Remote Percent Util   : 100
  Connection Master     : Remote
  Slave type            : N/A
  Synch Status          : inSynch
  Auto Synch            : OFF
  Admin Status          : UP
  Conn-Id               : 0
  Update Count          : 140840001

Step 4 Record the local NSAP address, VPI, VCI, remote PCR and remote SCR values (shown in bold); you will need these to add an endpoint at the MPSM.

Step 5 Add the MPSM connection for this PVC. For more information, see “Configuring the MPSM Card for IP-RAN” section on page 10-35.

Step 6 Open a management session to the PXM card and verify the connection.

   MGX.PXM> dspcons
   Local Port  Vpi.Vci  Remote Port  Vpi.Vci  State  Owner  Pri   Persistency
   ------------------------------------------+-----------+---------+---+-----------+
   1:1.2:2  0 101    27.65535  8 1000 OK   SLAVE - Persistent
   Local Addr: 47.009181000000164444b61.000001011802.00
   Remote Addr: 47.009181000000164444b61.0000011bffff.00
   Preferred Route ID:-  Cast Type: P2P

Viewing Status

To view IP-RAN status, use the show rpm ipran command and to view IP-RAN statistics use the show ip rtp header-compression virtual-access command, specifying the virtual-access interface for the IP-RAN connection.
Chapter 10 Configuring Quality of Service

Enabling IP Radio Access Network

Configuring the MPSM Card for IP-RAN

This section contains a Quickstart for configuring lines, bundles, and connections on the MPSM-16T1E1 card. For complete configuration information, see the “Adding a Connection to an MP Bundle for the IP-RAN Solution” section in the Cisco ATM and Frame Relay Services (MPSM-T3E3-155 and MPSM-16T1E1) Configuration Guide and Command Reference for MGX Switches, Release 5.1 book.

<table>
<thead>
<tr>
<th>Command</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Establish a configuration session with the MPSM card. Use a user name with GROUP 1 privileges or higher.</td>
</tr>
<tr>
<td>Step 2</td>
<td>addmpbundle</td>
</tr>
<tr>
<td>Step 3</td>
<td>addppplink</td>
</tr>
<tr>
<td>Step 4</td>
<td>addpppmux</td>
</tr>
<tr>
<td>Step 5</td>
<td>addcon</td>
</tr>
<tr>
<td>Step 6</td>
<td>dspcon</td>
</tr>
</tbody>
</table>

Table 10-6 describes the addcon command arguments for an adding a master connection at the MPSM. The remaining addcon arguments are unused for MLPPP connections.

**Table 10-6 addcon Command Parameters for the IP-RAN Solution**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ifnum</td>
<td>Identifies the logical interface on the local end of the connection you want to configure, range 1-16.</td>
</tr>
<tr>
<td>dlci</td>
<td>Identifies the Data Link Connection Identifier (DLCI) value. Use a value of 1000 for all IP-RAN connections.</td>
</tr>
<tr>
<td>chanType</td>
<td>Use a value of 5 to specify a frame-forwarding channel type.</td>
</tr>
<tr>
<td>serviceType</td>
<td>Identifies the ATM service type of the connection. For IP-RAN connections, only rtVBR and nrtVBR is supported. Select one of the following only:</td>
</tr>
<tr>
<td>mastership</td>
<td>Mastership role of the connection. When adding IP-RAN connections, the master side of the connection is provisioned on the MPSM card and the slave side of the connection is provisioned on the RPM-XF side. Use a value of 1 to select master.</td>
</tr>
<tr>
<td>cir</td>
<td>Committed Information Rate (in bits per second). Range: 0–1984001. Set this to the bundle bandwidth on the MPSM.</td>
</tr>
<tr>
<td>-slave</td>
<td>Slave-end connection identifier of the RPM-XF endpoint of the connection in the format nsap_address.vpi.vci. To find the NSAP address use the RPM-XF show switch connection vcc &lt;vpi&gt; &lt;vci&gt; command. Remove all the decimal points from the NSAP address and then append the vpi and vci of the RPM-XF connection endpoint, separated by decimal points.</td>
</tr>
</tbody>
</table>
IP-RAN Examples

The following examples show a typical IP-RAN configuration.

Configuration Example

The following example enables IP-RAN and configures associated QoS and switch parameters.

Router> enable
Router> configure terminal

Router(config)# hw-module rpm ipran

Configure Layer 3 QoS, specifying bandwidths as a percentage only. For more information see the “Class Map Commands” section on page 10-5 and the “Policy Map Commands” section on page 10-6.

Router(config)# class-map data
Router(config-cmap)# match ip precedence 0
Router(config-cmap)# class-map voice
Router(config-cmap)# match ip precedence 1

Router(config)# policy-map foo
Router(config-pmap)# class voice
Router(config-pmap-c)# policy cir percent 50
Router(config-pmap-c-police)# conform-action transmit
Router(config-pmap-c-police)# exceed-action drop
Router(config-pmap-c-police)# class data
Router(config-pmap-c)# bandwidth percent 35
Router(config-pmap-c)# class-default
Router(config-pmap-c)# bandwidth percent 15

Configure a virtual template that defines IPHC settings.

Router(config)# interface Virtual-Template1
Router(config-if)# ip address 192.168.1.1 255.255.255.0
Router(config-if)# ip tcp header-compression iphc-format
Router(config-if)# ip ospf hello-interval 1
Router(config-if)# ip ospf dead-interval 3
Router(config-if)# keepalive 1
Router(config-if)# ppp timeout retry 1
Router(config-if)# ppp iphc max-time 8
Router(config-if)# ip rtp header-compression iphc-format
Router(config-if)# ip rtp compression-connections 1000

Configure switch for SAR-based QoS.

Router(config-if)# interface Switch1
Router(config-if)# atm sar-based-cbwfq
Configure sub interface and PVC. The bundle bandwidth in this example is 6,144.000 bps.

Router(config-if)# interface Switch1.1 point-to-point
Router(config-subif)# pvc 0/101
Router(config-if-atm-vc)# encapsulation aal5cisco ppp Virtual-Template1
Router(config-if-atm-vc)# vbr-nrt 6144 6144 1024
Router(config-if-atm-vc)# service-policy output foo
Router(config-if-atm-vc)# switch connection vcc 0 101 master remote

Display Example

Router# show ip rtp header-compression virtual-access 1 detail
RTP/UDP/IP header compression statistics:
Configured:
Max Header 168 Bytes, Max Time 5 Secs, Max Period 256 Packets, Feedback On
Negotiated:
Max Header 168 Bytes, Max Time 5 Secs, Max Period 256 Packets, Feedback On
TX contexts:
RX contexts: