Configuring Hierarchical Modular QoS on Cisco ASR 9000 Series Routers

Hierarchical QoS allows you to specify QoS behavior at multiple policy levels, which provides a high degree of granularity in traffic management.

Line Card, SIP, and SPA Support

<table>
<thead>
<tr>
<th>Feature</th>
<th>ASR 9000 Ethernet Line Cards</th>
<th>SIP 700 for the ASR 9000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enhanced Hierarchical Ingress Policing</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Hierarchical Policing</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Hierarchical QoS</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Three-Parameter Scheduler</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

Feature History for Hierarchical QoS on Cisco ASR 9000 Series Routers

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
</table>
| Release 3.7.1 | The Hierarchical Policing feature was introduced on Cisco ASR 9000 Series Routers on ASR 9000 Ethernet Line Cards.  
|           | The Hierarchical QoS feature was introduced on Cisco ASR 9000 Series Routers on ASR 9000 Ethernet Line Cards.  
|           | The Three-Parameter Scheduler feature was introduced on Cisco ASR 9000 Series Routers on ASR 9000 Ethernet Line Cards.  |
| Release 3.9.0 | The Hierarchical QoS feature was supported on the SIP 700 for the ASR 9000. (two-level policies only)  |
| Release 4.0.0 | The Enhanced Hierarchical Ingress Policing feature was introduced on Cisco ASR 9000 Series Routers on the SIP 700 for the ASR 9000.  
|           | The Hierarchical Policing feature was supported on Cisco ASR 9000 Series Routers on the SIP 700 for the ASR 9000.  
|           | For the Hierarchical QoS feature, support was added for three-level policies on the SIP 700 for the ASR 9000.  
|           | The Three-Parameter Scheduler feature was supported on the SIP 700 for the ASR 9000.  |
Information About Hierarchical QoS

Hierarchical QoS allows you to specify QoS behavior at multiple policy levels, which provides a high degree of granularity in traffic management. A hierarchical policy is a QoS model that enables you to specify QoS behavior at multiple levels of hierarchy. You can use hierarchical policies to:

- Allow a parent class to shape multiple queues in a child policy
- Apply specific policy map actions on the aggregate traffic
- Apply class-specific policy map actions
- Restrict the maximum bandwidth of a virtual circuit (VC) while allowing policing and marking of traffic classes within the VC

The `service-policy` command is used to apply a policy to another policy, and a policy to an interface, subinterface, VC, or VLAN.

For example, in a three-level hierarchical policy, use the `service-policy` command to apply:

- Bottom-level policy to a middle-level policy
- Middle-level policy to a top-level policy
- Top-level policy to an interface, subinterface, VC, or VLAN

Benefits of Hierarchical Policies

Depending on the type of hierarchical QoS policy you configure, you can:

- Shape multiple queues to a single rate
- Divide a single class of traffic into one or more subclasses
- Specify the maximum transmission rate of a set of traffic classes that are queued separately, which is essential for virtual interfaces such as Frame Relay PVCs and IEEE 802.1Q virtual VLANs
- Configure minimum bandwidth queues on virtual circuits
- Shape the aggregate traffic of queues on a physical interface (for example, provide a 10-megabits per second (Mbps) service on a 100-Mbps physical interface)
- Restrict the maximum bandwidth of a VC while allowing policing and marking of classes within the VC
Restrictions for Hierarchical Policies

In Cisco IOS XR Release 4.0.0, the following restrictions apply:

**ASR 9000 Ethernet Line Cards**

In a three-level hierarchy, the configuration is rejected if the top-level class has queueing actions but the middle-level class or bottom-level class do not have queueing actions.

For example, the following configuration was valid in previous releases but is not valid in Cisco IOS XR Release 4.0.0 because the top-level class has a queueing action (shaping) but the middle-level and bottom-level classes do not have queueing actions:

```plaintext
policy-map grand-parent
   class class-default
      shape average 10 mbps
      service-policy parent

policy-map parent
   class p1
      service-policy child
      police rate 10 mbps
   class p2
      service-policy child
      police rate 10 mbps
   class p3
      police rate 10 mbps
      service-policy child
   class class-default

policy-map child
   class c1
      police rate 2 mbps
   class c2
      police rate 5 mbps
   class class-default
      police rate 10 mbps
```

To correct the configuration and retain queueing in the top-level class, you can add a queueing action to the middle-level or bottom-level classes. For example, the bandwidth remaining action is added to the middle-level classes in this example. You could also correct this configuration by removing queueing from the top-level class.

```plaintext
policy-map grand-parent
   class class-default
      shape average 10 mbps
      service-policy parent

policy-map parent
   class p1
      service-policy child
      police rate 10 mbps
      bandwidth remaining ratio 1
   class p2
      service-policy child
      police rate 10 mbps
      bandwidth remaining ratio 1
   class p3
      police rate 10 mbps
      service-policy child
      bandwidth remaining ratio 1
   class class-default
```
Information About Hierarchical QoS

SIP 700 for the ASR 9000

- In a three-level hierarchy, the bottom level does not allow queueing actions.
- On ingress, queueing actions are not allowed.

Sample Scenarios of Hierarchical Policies

Two-Level Hierarchical Policies

Two-level hierarchical policies, also called nested polices, can be illustrated with a parent-level policy for the top level of the hierarchy and a child-level for the bottom level of the hierarchy. A two-level hierarchical policy can have policing-only policies at the parent and child levels or can have queueing and policing at the parent and child levels. Hierarchical policies are configured by attaching a policy directly to a class of traffic.

Two-Level Hierarchical Policing Policy

Multi-level traffic policing is generally applied on ingress, and is a good application of two-level policies, as shown in Example 1. Aggregate traffic is policed to 10 Mbps and at the same time FTP traffic is policed to 1 Mbps and HTTP traffic is policed to 3 Mbps. The policy child-policy is attached to the class-default of the policy parent-policy.

```
class-map ftp
match protocol ftp
class-map http
match protocol http

policy-map child-policy
    class ftp
        police rate 1 mbps
    class http
        police rate 3mbps
    class class-default

policy-map parent-policy
    class class-default
        police rate percent 10 mbps
    service-policy child-policy
```
Two-Level Hierarchical Queueing Policy

A hierarchical Queueing Policy is generally applied on egress, and can be configured by modifying the hierarchical policing policy in Example 1.

In Example 2, the class-default of parent-policy is shaped to 10 Mbps. Policy child-queueing-policy is attached to the class-default of parent-policy, and defines three traffic classes:

- voice-ip matches the real-time protocol RTP and is configured as a priority class with 2 Mbps policing
- ftp is configured with 1 Mbps of guaranteed bandwidth
- http is configured with 3 Mbps of guaranteed bandwidth

Example 2  Two-Level Hierarchical Queueing Policy

class-map voice-ip
    match protocol rtp

class-map ftp
    match protocol ftp

class-map http
    match protocol http

policy-map child-queueing-policy
    class voice-ip
        priority level 1
        police rate 2 mbps
    class ftp
        bandwidth 1 mbps
    class http
        bandwidth 3 mbps
    class class-default

policy-map parent-policy
    class class-default
    shape average 10 mbps
    service-policy child-queueing-policy

Three-Level Hierarchical Policies

Example 3 shows a three-level hierarchical policy in a Frame Relay environment in which the goal is to shape a Frame Relay PVC to a maximum rate (effectively creating an interface with a specific bandwidth) and then manage the bandwidth of the PVC.

Note

This example is valid only on the SIP 700 for the ASR 9000.

Example 3  Three-Level Hierarchical Queueing Policy: Frame Relay

policy-map frpvc-policy
    class class-default
        shape average 2 mbps
        service-policy bandwidth-mgmt

policy-map bandwidth-mgmt
    class voice
        priority level 1
        police rate percent 20
class video
    bandwidth percent 50
class class-default
    service-policy set-frde

policy-map set-frde
    class ftp
        police rate 2 mbps
        conform-action transmit
        exceed-action set fr-de 1
    class class-default
        set-frde 1

The policy frpvc-policy is attached to a Frame Relay PVC, which shapes that PVC to 2 Mbps, using the **shape average** command that is configured on class-default in frpvc-policy. The absence of any user-defined classes in frpvc-policy means that all traffic matches class-default and is shaped (the entire PVC is shaped).

Attaching the bandwidth-mgmt policy to class-default in frpvc-policy specifies how the 2 Mbps in the PVC is allocated to a voice class, a video class, and all other traffic on the PVC. In this case, voice traffic is guaranteed low latency for 20 percent of the 2 Mbps, and the video class is guaranteed 50 percent of the 2 Mbps.

The remaining traffic is classified into two traffic classes. The ftp traffic class is policed at 2 Mbps and any FTP traffic above the policing rate has the Frame Relay DE bit set to 1. The remaining traffic has the Frame Relay DE bit set to 1.

This type of configuration is common in service providers that offer QoS-based services.

### Processing Order for Actions in Hierarchical Policies

In hierarchical policies, some actions are applied starting at the top level of the hierarchy and others are applied starting at the bottom level of the hierarchy.

For example, if the **set dscp** command appears at multiple layers in the hierarchy (multiple **set dscp** commands can apply to the same packet), the final DSCP value for the packet is the one in the set command at the bottom level of the hierarchy. The hierarchy is processed starting at the top and traversing to the bottom. Packet classification also happens starting at the top and traversing to the bottom.

For all actions other than those using **set** commands, the hierarchy is traversed from the bottom level of the hierarchy to the top level. For example, all queueing actions and the police action (including when the policer performs marking with the conform, exceed, or violate action at any level of the hierarchy) execute from the bottom level of the hierarchy to the top level of the hierarchy.

### WRED

If WRED is configured in the middle-level policy and a set action is configured for the same filter in the bottom-level policy, WRED is based on the newly remarked value of the filter from the bottom-level policy.

If there is a stand-alone **set** action as part of a policer action, the packet is marked with the set value in the policer, because that is executed last.
Three-Parameter Scheduler

In hierarchical QoS, if a policy is configured with queueing classes at any level in the hierarchy, then the traffic for different classes needs to be scheduled according to certain rules of the scheduler based on the configuration. The Three-Parameter Scheduler is a queueing algorithm that uses the following parameters to control traffic:

- Minimum bandwidth using the `bandwidth` command
- Excess bandwidth using the `bandwidth remaining` command
- Maximum bandwidth using the `shape average` command

The Three-Parameter Scheduler can be implemented in one-level, two-level, or three-level policies. In Example 4, a one-level policy is a sample configuration for a three-parameter scheduler.

For the purposes of this example, if `policy_3parameter_scheduler` is applied to a T1 serial interface and the T1 interface link bandwidth is 1536 kbps, then the link bandwidth distribution for each class is as follows:

- Class A is configured explicitly for minimum, maximum, and excess bandwidth. Class A receives a guaranteed minimum bandwidth of 30 percent of 1536 kbps (460.8 kbps) and 80 percent of remaining (excess) bandwidth. When other classes are not using all of their bandwidth share, class A can receive a maximum bandwidth of up to 50 percent of the link bandwidth (768 kbps).
- Class B is configured explicitly for minimum and excess bandwidth. Class B receives a guaranteed minimum bandwidth of 60 percent of 1536 kbps (921.6 kbps) and 10 percent of remaining (excess) bandwidth. If maximum bandwidth is not explicitly configured, the default maximum bandwidth is the link bandwidth (1536 kbps—equal to a configuration of `shape percent 100`). When other classes are not using all of their bandwidth share, class B can receive a maximum bandwidth of up to 100 percent of the link bandwidth.
- Class class-default is not configured explicitly with any queueing parameters. There is no default minimum bandwidth. If maximum bandwidth is not explicitly configured, the default maximum bandwidth is the link bandwidth (1536 kbps). When other classes are not using all of their bandwidth share, class-default can receive a maximum bandwidth of up to 100 percent of the link bandwidth. For excess bandwidth, because class A and class B receive a total of 90 percent of remaining (excess) bandwidth, class-default receives the remaining 10 percent.

Example 4  Three-Parameter Scheduler in a One-Level Policy

```
policy-map policy_3parameter_scheduler
class A
    bandwidth percent 30  <-------------- minimum bandwidth
    bandwidth remaining percent 80 <----------- excess bandwidth
    shape average percent 50 <----------------- maximum bandwidth
class B
    bandwidth percent 60
    bandwidth remaining percent 10
class class-default
```

Support for Three-Parameter Scheduler in Hierarchical Policies

In some cases, a scheduler might not support the three parameters at all levels in hierarchical policies.

ASR 9000 Ethernet Line Cards

Queueing is supported at all levels of hierarchical policies:
Top level—Two parameters only: excess bandwidth and maximum bandwidth
Middle level—Minimum bandwidth, excess bandwidth, and maximum bandwidth
Bottom level—Two parameters only: minimum bandwidth or excess bandwidth; and maximum bandwidth with 128 Mbps upper limit

SIP 700 for the ASR 9000
Queueing is supported at the top and middle levels of hierarchical policies:

- Top level—Two parameters only: excess bandwidth and maximum bandwidth
- Middle level—Minimum bandwidth, excess bandwidth, and maximum bandwidth
- Bottom level—Queueing is not supported

Hierarchical Policing
Hierarchical policing is supported on ingress and egress interfaces. Example 1 shows traffic policing in a two-level policy. Hierarchical policing allows enforcement of service level agreements (SLAs) while applying the classification submodel for different QoS classes on the interface. This allows you to police the interface while applying different classification modes on the interfaces.

Hierarchical Policing on ASR 9000 Ethernet Line Cards
Hierarchical policing is supported with these considerations:

- Ingress and egress interfaces
- Main interfaces and subinterfaces on all encapsulation types
- Two-level hierarchical policies and three-level hierarchical policies
- Police actions
  - In middle and bottom policies
  - Processed from the bottom level of the hierarchy to the top level of the hierarchy
- Policers
  - For bottom-level policers, the top-level policer rate is used as the reference bandwidth
- Policies applied on a bundle interface are replicated to all of the bundle members
- Statistics for all levels in the hierarchy

Hierarchical Policing on the SIP 700 for the ASR 9000
Hierarchical policing is supported with these considerations:

- Ingress and egress interfaces
- Main interfaces and subinterfaces on all encapsulation types
- Two-level hierarchical policies; in three-level policies at middle and bottom levels.
- Police actions
  - In top and bottom policies
  - Processed from the bottom level of the hierarchy to the top level of the hierarchy
• Policers
  – Color-blind policers are supported in both top and bottom policies
  – Policers are not required in bottom-level classes
  – Police rates can be configured as absolute rates or as percentages
  – For bottom-level policers, the top-level policer rate is used as the reference bandwidth
• Statistics for all levels in the hierarchy

Enhanced Hierarchical Ingress Policing

In hierarchical policing, traffic is policed first at the child policer level and then at the parent policer level. It is possible for traffic that conforms to the committed rate specified by the child policer to be dropped by the parent policer.

In Enhanced Hierarchical Ingress Policing, the child-conform-aware command prevents the parent policer from dropping any ingress traffic that conforms to the committed rate specified in the child policer.

How to Configure Hierarchical QoS

When configuring hierarchical QoS, consider the following guidelines:

• When defining polices, start at the bottom level of the hierarchy. For example, for a two-level hierarchical policy, define the bottom-level policy and then the top-level policy. For a three-level hierarchical policy, define the bottom-level policy, the middle-level policy, and then the top-level policy.
• Do not specify the input or output keyword in the service-policy command when configuring a bottom-level policy within a top-level policy.
• Configure bottom-level policies only in middle-level and top-level policies.

This section describes the following tasks:

• Configuring the Three-Parameter Scheduler, page 140
• Policing Traffic at Two Levels of Hierarchy, page 145
• Attaching Hierarchical Policies to Physical and Virtual Links, page 147
• Configuring Enhanced Hierarchical Ingress Policing, page 149
Configuring the Three-Parameter Scheduler

When configuring the Three-Parameter Scheduler, consider the following guidelines:

- To use the three-parameter scheduler, a queueing class must be enabled. To enable a queueing class, you must configure at least one of the three parameters. When at least one parameter is configured, a queue is assigned to the class.
- If you configure only one parameter, the scheduler uses default values for the other two parameters.
- You can configure all 3 parameters in the same class.
- Minimum bandwidth must be less than maximum bandwidth.

For information about three-parameter scheduler support on specific line cards or SIPs, see “Support for Three-Parameter Scheduler in Hierarchical Policies” section on page 137.

ASR 9000 Ethernet Line Cards

SUMMARY STEPS

1. configure
2. policy-map policy-name
3. class class-name
4. shape average { percent percentage | rate [units] }
5. exit
6. policy-map policy-name
7. class class-default
8. bandwidth { rate [units] | percent percentage-value }
   or
   bandwidth remaining { percent percentage-value | ratio ratio-value }
   or
   shape average { percent percentage | rate [units] }
9. service-policy policy-map-name
10. end
    or
    commit
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> policy-map policy-name</td>
<td>Creates or modifies the bottom-level policy.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# policy-map bottom-child</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> class class-name</td>
<td>Assigns the traffic class that you specify to the policy map. Enters policy map class configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-pmap)# class Bronze</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> shape average (percent percentage</td>
<td>rate units)</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-pmap-c)# shape average 1 mbps</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> exit</td>
<td>Exits policy map class configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-pmap-c)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> policy-map policy-name</td>
<td>Creates or modifies the top-level policy.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-pmap)# policy-map Top-Parent</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> class class-default</td>
<td>Configures or modifies the parent class-default class.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-pmap)# class class-default</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>You can configure only the class-default class in a parent policy. Do not configure any other traffic class.</td>
</tr>
</tbody>
</table>
# How to Configure Hierarchical QoS

<table>
<thead>
<tr>
<th>Step 8</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>bandwidth (rate [units]</td>
<td>percent percentage-value)</td>
</tr>
<tr>
<td></td>
<td>or</td>
<td></td>
</tr>
<tr>
<td></td>
<td>bandwidth remaining (percent percentage-value</td>
<td>ratio ratio-value)</td>
</tr>
<tr>
<td></td>
<td>or</td>
<td></td>
</tr>
<tr>
<td></td>
<td>shape average (percent percentage</td>
<td>rate [units])</td>
</tr>
</tbody>
</table>

**Example:**

```
RP/0/RSP0/CPU0:router(config-pmap-c)# bandwidth percent 30
or
RP/0/RSP0/CPU0:router(config-pmap-c)# bandwidth remaining percent 80
or
RP/0/RSP0/CPU0:router(config-pmap-c)# shape average percent 50
```

| Step 9 | service-policy policy-map-name | Applies a bottom-level policy to the top-level class-default class. |

**Example:**

```
RP/0/RSP0/CPU0:router(config-pmap-c)# service-policy Bottom-Child
```

<table>
<thead>
<tr>
<th>Step 10</th>
<th>end or commit</th>
<th>Saves configuration changes.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>end or commit</td>
<td>- When you issue the end command, the system prompts you to commit changes:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
</tr>
</tbody>
</table>
## SUMMARY STEPS

1. `configure`
2. `policy-map policy-name`
3. `class class-name`
4. `bandwidth {rate [units] | percent percentage-value}`  
   or  
   `bandwidth remaining [percent percentage-value | ratio ratio-value]`  
   or  
   `shape average {percent percentage | rate [units]}`
5. `exit`
6. `policy-map policy-name`
7. `class class-default`
8. `(Optional) shape average {percent percentage | rate [units]}`
9. `service-policy policy-map-name`
10. `end`  
    or  
    `commit`

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**  
`configure` | Enters global configuration mode. |

**Example:**  
RP/0/RSP0/CPU0:router# configure

| **Step 2**  
`policy-map policy-name` | Creates or modifies the bottom-level policy. |

**Example:**  
RP/0/RSP0/CPU0:router(config)# policy-map bottom-child

| **Step 3**  
`class class-name` | Assigns the traffic class that you specify to the policy map. Enters policy map class configuration mode. |

**Example:**  
RP/0/RSP0/CPU0:router(config-pmap)# class Bronze
## Step 4

**Command or Action**

`bandwidth (rate [units] | percent percentage-value)

or

bandwidth remaining [percent percentage-value | ratio ratio-value]

or

shape average (percent percentage | rate [units])`

### Purpose

- **Specifies the minimum bandwidth allocated to a class as a percentage of link bandwidth.**
- **Specifies how to allocate excess bandwidth to a class.**
- **Specifies maximum bandwidth as a percentage of link bandwidth (when other classes are not using all of their bandwidth share).**

### Example:

```
RP/0/RSP0/CPU0:router(config-pmap-c)# bandwidth percent 30
or
RP/0/RSP0/CPU0:router(config-pmap-c)# bandwidth remaining percent 80
or
RP/0/RSP0/CPU0:router(config-pmap-c)# shape average percent 50
```

### Note

- You must configure at least one of the three parameters.

## Step 5

**Command or Action**

`exit`

### Purpose

- **Exits policy map class configuration mode.**

### Example:

```
RP/0/RSP0/CPU0:router(config-pmap-c)# exit
```

## Step 6

**Command or Action**

`policy-map policy-name`

### Purpose

- **Creates or modifies the top-level policy.**

### Example:

```
RP/0/RSP0/CPU0:router(config-pmap-c)# policy-map Top-Parent
```

## Step 7

**Command or Action**

`class class-default`

### Purpose

- **Configures or modifies the parent class-default class.**

### Note

- You can configure only the class-default class in a parent policy. Do not configure any other traffic class.

### Example:

```
RP/0/RSP0/CPU0:router(config-pmap-c)# class class-default
```

## Step 8

**Command or Action**

`shape average (percent percentage | rate [units])`

### Purpose

- **(Optional) Shapes traffic to the indicated bit rate.**

### Example:

```
RP/0/RSP0/CPU0:router(config-pmap-c)# shape average 1 mbps
```
Configuring Hierarchical Modular QoS on Cisco ASR 9000 Series Routers

How to Configure Hierarchical QoS

Policing Traffic at Two Levels of Hierarchy

Use this procedure to configure a hierarchical policing policy to police the traffic that enters or exits the router on a specific interface.

SUMMARY STEPS

1. configure
2. policy-map policy-name
3. class class-name
4. (Optional) police rate [value [units] | percent percentage] [burst burst-size [burst-units]] [peak-rate value [units]] [peak-burst peak-burst [burst-units]]
5. exit
6. policy-map policy-name
7. class class-default

Command or Action | Purpose
--- | ---
**Step 9** service-policy policy-map-name | Applies a bottom-level policy to the top-level class-default class.
**Example:**
RP/0/RSP0/CPU0:router(config-pmap-c)# service-policy Bottom-Child
**Step 10** end or commit | Saves configuration changes.
- When you issue the end command, the system prompts you to commit changes:
  Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:
  - Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.
**Example:**
RP/0/RSP0/CPU0:router(config-pmap-c)# end or
RP/0/RSP0/CPU0:router(config-pmap-c)# commit

Policing Traffic at Two Levels of Hierarchy

Use this procedure to configure a hierarchical policing policy to police the traffic that enters or exits the router on a specific interface.

SUMMARY STEPS

1. configure
2. policy-map policy-name
3. class class-name
4. (Optional) police rate [value [units] | percent percentage] [burst burst-size [burst-units]] [peak-rate value [units]] [peak-burst peak-burst [burst-units]]
5. exit
6. policy-map policy-name
7. class class-default

Command or Action | Purpose
--- | ---
**Step 9** service-policy policy-map-name | Applies a bottom-level policy to the top-level class-default class.
**Example:**
RP/0/RSP0/CPU0:router(config-pmap-c)# service-policy Bottom-Child
**Step 10** end or commit | Saves configuration changes.
- When you issue the end command, the system prompts you to commit changes:
  Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:
  - Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.
**Example:**
RP/0/RSP0/CPU0:router(config-pmap-c)# end or
RP/0/RSP0/CPU0:router(config-pmap-c)# commit

Command or Action | Purpose
--- | ---
**Step 9** service-policy policy-map-name | Applies a bottom-level policy to the top-level class-default class.
**Example:**
RP/0/RSP0/CPU0:router(config-pmap-c)# service-policy Bottom-Child
**Step 10** end or commit | Saves configuration changes.
- When you issue the end command, the system prompts you to commit changes:
  Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:
  - Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.
- Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.
**Example:**
RP/0/RSP0/CPU0:router(config-pmap-c)# end or
RP/0/RSP0/CPU0:router(config-pmap-c)# commit
### How to Configure Hierarchical QoS

8. **(Optional)** `police rate [value [units] | percent percentage] [burst burst-size [burst-units]] [peak-rate value [units]] [peak-burst peak-burst [burst-units]]`

9. `service-policy policy-map-name`

10. `end`
    or
    `commit`

---

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>policy-map policy-name</code></td>
<td>Creates or modifies a bottom-level policy.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# policy-map Business</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>class class-name</code></td>
<td>Assigns the traffic class you specify to the policy map. Enters policy map class configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-pmap)# class Gold</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> `police rate (value [units]</td>
<td>percent percentage) [burst burst-size [burst-units]] [peak-rate value [units]] [peak-burst peak-burst [burst-units]]`</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-pmap-c)# police rate percent 10</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>exit</code></td>
<td>Exits policy map class configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-pmap-c)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> <code>policy-map policy-name</code></td>
<td>Creates or modifies a top-level policy.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# policy-map All_Traffic</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> <code>class class-default</code></td>
<td>Configures or modifies the default traffic class.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-pmap)# class class-default</td>
<td></td>
</tr>
</tbody>
</table>
## Attaching Hierarchical Policies to Physical and Virtual Links

To attach hierarchical policies to interfaces, subinterfaces, virtual circuits, and virtual LANs, use the `service-policy {input | output} policy-map-name` command.

### SUMMARY STEPS

1. `configure`
2. `interface type interface-path-id`
3. `service-policy {input | output} policy-map-name`
### How to Configure Hierarchical QoS

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router# configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Specifies the interface to attach the hierarchical policy.</td>
</tr>
<tr>
<td><code>interface type interface-path-id</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config)# interface pos 0/2/0/0</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Attaches the policy map you specify.</td>
</tr>
<tr>
<td>`service-policy {input</td>
<td>output} policy-map-name`</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-if)# service-policy input All_Traffic</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><code>end</code> or <code>commit</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-pmap-c)# end or commit</td>
</tr>
</tbody>
</table>

#### Command or Action

- **end** or **commit**

#### Purpose

- **Enters global configuration mode.**
- **Specifies the interface to attach the hierarchical policy.**
- **Attaches the policy map you specify.**
  - **input**—Apply the QoS policy to inbound packets.
  - **output**—Apply the QoS policy to outbound packets.
  - **policy-map-name**—Name of a previously configured top-level policy map
- **Saves configuration changes.**
  - When you issue the **end** command, the system prompts you to commit changes:
    Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:
    - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
    - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
    - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
  - Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Configuring Enhanced Hierarchical Ingress Policing

The difference between configuring enhanced hierarchical ingress policing and configuring hierarchical ingress policing is the addition of the `child-conform-aware` command.

When used in the parent policer, the `child-conform-aware` command prevents the parent policer from dropping any ingress traffic that conforms to the maximum rate specified in the child policer.

Restrictions

Enhanced Hierarchical Ingress Policing has the following limitations:

- Ingress direction only.
- Sum of all child policer rates cannot be greater than the parent policer rate.
- Single-rate two-color policer (color blind) only.
- Configurations that specify burst size in the `police rate` command are supported; configurations that specify peak burst become single-rate three-color policers and are therefore rejected.
- Configure the `child-conform-aware` command only in the parent policer.

SUMMARY STEPS

1. configure
2. policy-map policy-name
3. class class-name
4. service-policy policy-map-name
5. police rate {value [units] | percent percentage} [burst burst-size [burst-units]] [peak-rate value [units]] [peak-burst peak-burst [burst-units]]
6. child-conform-aware
7. conform-action [drop | set options | transmit]
8. exceed-action [drop | set options | transmit]
9. end
or commit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 policy-map policy-name</td>
<td>Enters policy map configuration mode. Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# policy-map parent</td>
<td></td>
</tr>
</tbody>
</table>
### How to Configure Hierarchical QoS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>class class-name</td>
<td>Enters policy map class configuration mode. Specifies the name of the class whose policy you want to create or change.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-pmap)# class class-default</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>service-policy policy-map-name</td>
<td>Applies the bottom-level policy map to the parent class-default class.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> Do not specify an <strong>input</strong> or <strong>output</strong> keyword.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-pmap-c)# service-policy child</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>police rate {value [units]</td>
<td>percent percentage</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-pmap-c)# police rate percent 50</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>child-conform-aware</td>
<td>Prevents the parent policer from dropping any ingress traffic that conforms to the maximum rate specified in a child policer.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-pmap-c-police)# child-conform-aware</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>conform-action [drop</td>
<td>set options</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-pmap-c-police)# conform-action transmit</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>exceed-action [drop</td>
<td>set options</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RP/0/RSP0/CPU0:router(config-pmap-c-police)# exceed-action drop</td>
<td></td>
</tr>
</tbody>
</table>
Configuration Examples for Hierarchical QoS

This section provides the following configuration examples:

- Two-Level Hierarchical Queueing Policy: Example, page 151
- Three-Level Hierarchical Queueing Policy: Examples, page 152
- Three-Parameter Scheduler: Examples, page 156
- Hierarchical Policing: Examples, page 157
- Attaching Service Policies to Physical and Virtual Links: Examples, page 158
- Enhanced Hierarchical Ingress Policing: Example, page 158

Two-Level Hierarchical Queueing Policy: Example

The following example shows a two-level policy applied at the Multilink Frame Relay main interface. The same policy can be applied at Multilink PPP main interface.

```
class-map match-any video
  match precedence 1
end-class-map

class-map match-any premium
  match precedence 2 3
end-class-map

class-map match-any voice-ip
  match precedence 0
end-class-map
```

Step 9

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td></td>
<td>- When you issue the <code>end</code> command, the system prompts you to commit changes:</td>
</tr>
<tr>
<td></td>
<td>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</td>
</tr>
<tr>
<td></td>
<td>- Entering <code>yes</code> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>- Entering <code>no</code> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>- Entering <code>cancel</code> leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>- Use the <code>commit</code> command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
</tr>
</tbody>
</table>

Example:

```
RP/0/RSP0/CPU0:router(config-pmap-c-police)#
end

or

RP/0/RSP0/CPU0:router(config-pmap-c-police)#
commit
```
class-map match-any best-effort
match precedence 4
end-class-map

policy-map parent_shape
class class-default
  service-policy child_policy
  shape average percent 90
!
end-policy-map
!

policy-map child_policy
  class voice-ip
    priority level 1
    police rate percent 20
    !
  class video
    bandwidth percent 40
  class premium
    bandwidth percent 10
    random-detect precedence 2 10 ms 100 ms
    random-detect precedence 3 20 ms 200 ms
    queue-limit 200 ms
    !
  class best-effort
    bandwidth percent 20
    queue-limit 200 ms
    !
  class class-default
  !
end-policy-map
!

interface Multilink0/2/1/0/1
  service-policy output parent_shape
  encapsulation frame-relay
  frame-relay intf-type dce

Three-Level Hierarchical Queueing Policy: Examples

ASR 9000 Ethernet Line Cards

In this example, policy grand-parent is applied to the main Ethernet interface. The grand-parent policy limits all outbound traffic of the interface up to 500 Mbps. The parent policy has class vlan1 and vlan2, and traffic in vlan1 or vlan2 is limited to 40 percent of 500 Mbps. The policy child_policy classifies traffic based on different services and allocates bandwidth for each class accordingly.

class-map match-any video
match precedence 1
end-class-map
!

class-map match-any premium
match precedence 2 3
end-class-map
!

class-map match-any voice-ip
match precedence 0
end-class-map
!
class-map match-any best-effort
  match precedence 4
end-class-map

class-map match-any vlan1
  match vlan 1
end-class-map

class-map match-any vlan2
  match vlan 2
end-class-map

policy-map grand-parent
class class-default
  shape average 500 Mbps
  service-policy parent
!
end-policy-map

policy-map parent
class vlan1
  service-policy child_policy
  shape average percent 40
!
class vlan2
  service-policy child_policy
  shape average percent 40
!
end-policy-map
!

policy-map child_policy
class voice-ip
  priority level 1
  police rate percent 20
!
!
class video
  bandwidth percent 40
!
class premium
  bandwidth percent 10
  random-detect precedence 2 10 ms 100 ms
  random-detect precedence 3 20 ms 200 ms
  queue-limit 200 ms
!
class best-effort
  bandwidth percent 20
  queue-limit 200 ms
!
class class-default
!
end-policy-map
!

interface GigabitEthernet0/0/0/9
service-policy output grand-parent
In this example, the policy parent_policy is applied to the Multilink Frame Relay main interface. The policy parent_policy has two classes, which match on Frame Relay DLCIs. The Multilink Frame Relay main interface has two Frame Relay PVCs configured (DLCI 16, DLCI 17).

```plaintext
interface Multilink0/2/1/0/1
mtu 1504
service-policy output parent_policy
global policy
encapsulation frame-relay
frame-relay intf-type dce
!
policy-map parent_policy
  class parentQ_1
    service-policy child_queuing_policy
    shape average 64 kbps
  !
class parentQ_2
    service-policy child_queuing_policy
    shape average 1 mbps
  !
class class-default
! end-policymap
!
class-map match-any parentQ_1 <----- class map parent class dlci=16
  match frame-relay dlci 16
end-class-map
!
class-map match-any parentQ_2 <----- class map parent class dlci=17
  match frame-relay dlci 17
end-class-map
!
interface Multilink0/2/1/0/1.16 point-to-point <------ dlci 16 pvc config
ipv4 address 192.1.1.1 255.255.255.0
pvc 16
  encap cisco
! 
interface Multilink0/2/1/0/1.17 point-to-point <------ dlci 17 pvc config
ipv4 address 192.1.2.1 255.255.255.0
pvc 17
  encap cisco
! 
policy-map child_queuing_policy <----- child policy map
  class voice-ip
    priority level 1
    police rate percent 20
  !
class video
    bandwidth percent 40
! 
class premium
  service-policy gchild_policy
  bandwidth percent 10
  random-detect discard-class 2 10 ms 100 ms
  random-detect discard-class 3 20 ms 200 ms
```
Configuring Hierarchical Modular QoS on Cisco ASR 9000 Series Routers

Configuration Examples for Hierarchical QoS

```plaintext
queue-limit 200 ms
!
class best-effort
  bandwidth percent 20
  queue-limit 200 ms
!
class class-default
!
end-policy-map
!
policy-map gchild_policy <-------- grandchild policy map
  class premium_g1
    police rate percent 10
    !
    set discard-class 2
  
  class premium_g2
    police rate percent 50
    !
    set discard-class 3
  
  class class-default
  !
end-policy-map
!
show run class-map <-------- shows all class-map configs
Mon Aug 2 11:35:19.479 UTC
class-map match-any video
  match precedence 1
  end-class-map
!
class-map match-any premium
  match precedence 2 3
  end-class-map
!
class-map match-any voice-ip
  match precedence 0
  end-class-map
!
class-map match-any parentQ_1
  match frame-relay dlci 16
  end-class-map
!
class-map match-any parentQ_2
  match frame-relay dlci 17
  end-class-map
!
class-map match-any premium_g1
  match precedence 2
  end-class-map
!
class-map match-any premium_g2
  match precedence 3
  end-class-map
!
class-map match-any best-effort
  match precedence 4
  end-class-map
```

Cisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Configuration Guide

OL-26077-01

QC-155
Three-Parameter Scheduler: Examples

ASR 9000 Ethernet Line Cards

This example shows how to configure a three-parameter scheduler in a two-level hierarchical policy.

```
policy-map Bottom-ChildA
  class A1
    shape average 400 kbps
  class A2
    shape average 400 kbps

policy-map Bottom-ChildB
  class B1
    shape average 250 kbps
  class B2
    shape average 450 kbps

policy-map Top-Parent
  class parentA
    shape average 500 kbps
    bandwidth percent 30
    bandwidth remaining percent 80
    service-policy Bottom-ChildA
  class parentB
    shape average 500 kbps
    bandwidth percent 60
    bandwidth remaining percent 10
    service-policy Bottom-ChildB
```

SIP 700 for the ASR 9000

This example shows how to configure a three-parameter scheduler in a two-level hierarchical policy.

```
policy-map Bottom-Child
  class A
    bandwidth percent 30
    bandwidth remaining percent 80
    shape average percent 50
  class B
    bandwidth percent 60
    bandwidth remaining percent 10
  class class-default
  exit

policy-map Top-Parent
  class-default
    shape average 1 mbps
  service-policy Bottom-Child
```
Hierarchical Policing: Examples

ASR 9000 Ethernet Line Cards

This example shows a two-level policy with police actions at each level. There are two classes in the top level, one for each customer. Aggregated traffic from each customer is subject to a rate limit as specified by the **police rate** command in the top level. Traffic in different classes in the bottom level is limited by an additional set of police actions to control different types of traffic for each customer.

```plaintext
class-map match-any customera
  match vlan 10-14
class-map match-any customerb
  match vlan 15-19
class-map match-any prec1
  match precedence 1
class-map match-any prec3
  match precedence 3

policy-map parent
  class customera
    service-policy childa
    bandwidth remaining ratio 10
    police rate percent 50
    conform-action transmit
    exceed-action drop
  class customerb
    service-policy childb
    bandwidth remaining ratio 100
    police rate percent 70
    conform-action transmit
    exceed-action drop

policy-map childa
  class prec1
    police rate percent 25
    conform-action transmit
    exceed-action drop
  class prec3
    police rate percent 25
    conform-action transmit
    exceed-action drop

policy-map childb
  class prec1
    police rate percent 30
    conform-action transmit
    exceed-action drop
  class prec3
    police rate percent 30
    conform-action transmit
    exceed-action drop
```

**SIP 700 for the ASR 9000**

In this example, policers are specified in the policy child in class Prec1 and class Prec3, and also in the class-default in the policy parent. The policers in the child policy, police traffic in class Prec1 at 30 percent (of 50 percent), police traffic in class Prec3 at 60 percent (of 50 percent) and police any other traffic at 10 percent (of 50 percent). Cumulatively, all traffic on the interface is policed at 50 percent of the interface rate by the policer in the parent policy.
Configuration Examples for Hierarchical QoS

class-map match-any prec1
  match precedence 1

class-map match-any prec3
  match precedence 3

policy-map parent
  class class-default
    service-policy child
      police rate percent 50
      conform-action transmit
      exceed-action drop

policy-map child
  class prec1
    police rate percent 30
    conform-action transmit
    exceed-action drop
  class prec3
    police rate percent 60
    conform-action transmit
    exceed-action drop
  class class-default
    police rate percent 10
    conform-action transmit
    exceed-action drop

Attaching Service Policies to Physical and Virtual Links: Examples

Physical Link: Example

In this example, the p1 policy is applied to a Gigabit Ethernet interface:

```
interface gigabitethernet 0/2/0/0
service-policy input p1
```

Virtual Link: Example

In this example, the p2 policy is applied to the private virtual circuit (PVC) under a multilink Frame Relay subinterface. A QoS policy can be applied only to a PVC under a Frame Relay subinterface; it cannot be applied directly to a Frame Relay subinterface.

```
interface Multilink0/2/1/0/1.16 point-to-point
encapsulation frame-relay
ipv4 address 192.1.1.1 255.255.255.0
pvc 16
  service-policy output p2
  encap cisco
```

Enhanced Hierarchical Ingress Policing: Example

This example shows parent and child policies in which two classes are defined in the child policy. In class AF1, the exceed action is set to an action other than to drop traffic.

If the child-conform-aware command were not configured in the parent policy, the parent policer would drop traffic that matches the conform rate of the child policer but exceeds the conform rate of the parent policer.
When used in the parent policer, the `child-conform-aware` command prevents the parent policer from dropping any ingress traffic that conforms to the committed rate specified in the child policer.

In this example, class EF in the child policy is configured with a committed rate of 1 Mbps, a conform action and an exceed action. The traffic that is below 1 Mbps is presented to the parent policer with the MPLS EXP bit set to 4, and traffic that exceeds 1 Mbps is dropped.

Class AF1 in the child policy is configured with a committed rate of 1 Mbps, a conform action and an exceed action. The traffic that is below 1 Mbps is presented to the parent policer with the MPLS EXP bit set to 3, and traffic that exceeds 1 Mbps is presented to the parent policer with the MPLS EXP bit set to 2.

With this child policy configuration, the parent policer sees traffic from the child classes as exceeding its committed rate of 2 Mbps. Without the `child-conform-aware` command in the parent policer, the parent polices to 2 Mbps, which can result into dropping some conformed traffic from class EF in the child policy. When the `child-conform-aware` command is configured in the parent policer, the parent policer does not drop any traffic that conforms under the child policy.

```
policy-map parent
  class class-default
    service-policy child
    police rate 2 mbps
    child-conform-aware
    conform-action transmit
    exceed-action drop

policy-map child
  class EF
    police rate 1 mbps
    conform-action set mpls experimental imposition 4
    exceed-action drop
  class AF1
    police rate 1 mbps
    conform-action set mpls experimental imposition 3
    exceed-action set mpls experimental imposition 2
```

### Verifying the Configuration of Hierarchical Policies

To verify hierarchical policies, enter any of the following commands in privileged EXEC mode:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show policy-map interface</code></td>
<td>Displays policy configuration information for all classes configured for all service policies on the specified interface.</td>
</tr>
<tr>
<td><code>show qos interface</code></td>
<td>Displays QoS information for all classes in the service policy that is attached to the specified interface.</td>
</tr>
<tr>
<td><code>show running-config class-map</code></td>
<td>Displays the configuration of all class maps configured on the router.</td>
</tr>
<tr>
<td><code>show running-config policy-map</code></td>
<td>Displays the configuration of all policy maps configured on the router.</td>
</tr>
<tr>
<td><code>show running-config policy-map policy-map-name</code></td>
<td>Displays the configuration of all classes contained in the policy map you specify.</td>
</tr>
</tbody>
</table>
Additional References

The following sections provide references related to implementing Hierarchical QoS.

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial system bootup and configuration</td>
<td>Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide</td>
</tr>
<tr>
<td>Master command reference</td>
<td>Cisco ASR 9000 Series Aggregation Services Router Master Command Listing</td>
</tr>
<tr>
<td>QoS commands</td>
<td>Cisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Command Reference</td>
</tr>
<tr>
<td>User groups and task IDs</td>
<td>“Configuring AAA Services on Cisco ASR 9000 Series Router” module of Cisco Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide</td>
</tr>
</tbody>
</table>

Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>

MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: <a href="http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a></td>
</tr>
</tbody>
</table>

RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.</td>
<td>—</td>
</tr>
</tbody>
</table>
## Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>