Configuring Hierarchical Modular QoS

Hierarchical QoS (H-QoS) is a QoS model that enables you to specify QoS behavior at multiple levels of hierarchy. This chapter provides information about this feature and the different steps involved in configuring it.

This chapter covers the following topics:

- Overview of Hierarchical Modular QoS, on page 1
- Restrictions, Limitations, and Requirements for Configuring H-QoS, on page 2
- Configuring Hierarchical Queuing, on page 3
- Conform Aware Hierarchical Policy Overview, on page 8

Overview of Hierarchical Modular QoS

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

H-QoS is applied on the router interface using nested traffic policies. The first level of traffic policy, the parent traffic policy, is used for controlling the traffic at the main interface or sub-interface level. The second level of traffic policy, the child traffic policy, is used for more control over a specific traffic stream or class. The child traffic policy, is a previously defined traffic policy, that is referenced within the parent traffic policy using the **service-policy** command.

Two-level H-QoS is supported on both ingress and egress directions on all line cards and on physical or bundle main interfaces and sub-interfaces.

Three-level Hierarchical QoS (H-QoS) enables enforcement of class/service, group/ Ethernet Flow Point (EFP), and port level SLAs. You can apply regular two-level egress H-QoS policies on the sub-interfaces to achieve class and EFP SLAs at child and parent levels. In addition, you can apply a port shaper policy on the main interface to achieve an aggregated port level SLA in a 1+2 H-QoS or three-level H-QoS model.

An important point to note is that before Release 6.6.25 (where the three-level H-QoS capability was introduced), when you applied class-default shaper on a main interface, it was enforced *only* on the traffic going through the main interface. With three-level HQoS, a class default shaper that is applied on the main interface is considered as a port shaper and enforced on *all* traffic going out of that physical port. The advantage of three-level H-QoS is that the parent shaper on the sub-interfaces is allowed to oversubscribe, thus enabling best effort sharing of the aggregate port shaper at the third level.

Restrictions, Limitations, and Requirements for Configuring H-QoS

There are some key restrictions, limitations, and requirements for configuring H-QoS on your router to ensure optimal performance and compliance.

Parent Traffic Policy Restrictions

- The parent traffic policy only supports the traffic class of type class-default.
- The parent traffic policy only supports the class-action shape, and no other queuing action can be configured in it.
- For congestion avoidance and management, the traffic shaper in the parent traffic policy calculates the queue limit and drop priority

Child Policy Requirements

- While configuring on the router, it is mandatory that the priority class must have a traffic shaper in the child traffic policy.
- The sum of the bandwidth of the child policies must be less than the parent policy's traffic shaper.

H-QoS Profile Limitations

- H-QoS profile and ingress peering profile don't work simultaneously. Hence, features requiring a peering profile also do not work with the H-QoS profile enabled.
- The PBTS feature does not work when the H-QoS profile is enabled. This is due to TCAM limitations.
- A maximum of 896 bundle subinterfaces are supported in the system, even if there are no QoS policies applied. This is due to an internal LAG_ID resource consumption in H-QoS profile mode for bundle subinterfaces with or without QoS policies being applied.
- A maximum of 4 priority levels are supported in H-QoS profile mode, unlike the default mode where 7-priority levels are supported. The restriction also applies to physical and bundle main interface policies where 7-level priorities were previously used in non-H-QoS profile mode.

Bandwidth Configuration Limitations

- Bandwidth and Bandwidth remaining configurations are not supported simultaneously within the same policy-map. If a class has bandwidth (CIR), other classes must also have only bandwidth configuration. If a class-map has bandwidth remaining percent/ratio (EIR), other classes should also have only the bandwidth remaining configuration. Shaping is applied on any class.
- In H-QoS mode, if multiple queues are configured with BRR and there is high congestion in the LP (low-priority) queues, one BRR queue receives more credits than the others. The deviation is proportional to the congestion in the LP queues.

- The granularity of bandwidth or the bandwidth remaining ratio (BRR) is 1:64, compared to 1:4096 in non-H-QoS mode. Consequently, there may be differences in bandwidth performance accuracy depending on the values used.

Priority Classes Requirements

Priority classes must be rate-limited using a shaping configuration. The effective shaper value is used as the priority bandwidth reservation. The total priority bandwidth reservations across all subinterfaces and main interfaces must not exceed the network interface (NIF) port speed to prevent oversubscription of priority traffic. Non-priority class rates and parent shaping can be oversubscribed.

Multicast Traffic Restriction

Filtering egress IPv4 and IPv6 multicast traffic is not supported when H-QoS is configured on the router.

Three-Level H-QoS Restrictions

- There is no support for bandwidth action at the EFP parent level. All EFP and subinterface policies get a fair share of the port shaper.
- Three-level H-QoS does not apply to ingress policies or egress marking policies.
- Executing clear qos counters on the main interface clears only the main interface policy statistics. Use the all option to clear all sub-interface statistics, or alternatively, clear the sub-interface policy statistics individually.
- Main interface policy statistics do not reflect the subinterface packet or byte counters, although the port shaper is enforced on all logical ports for a given physical interface. The subinterface policy-map statistics reflect the transmitted and dropped packet or byte count post-port shaper enforcement.

Configuring Hierarchical Queuing

Before you configure H-QoS, you must enable the H-QoS profile on the router. After enabling H-QoS profile, reload the router, as shown in the following configuration.

```
admin
hw-module location all reload
Router# configure
Router(config)# hw-module profile qos hqos-enable
Router(config)# commit
Router# admin
sysadmin-vm:0_RP0# hw-module location all reload
```

The steps that are involved in configuring hierarchical queuing are as follows:

1. Configure a class-map.
2. Configure a child traffic policy using the class-map that was configured in the previous step.
3. Configure a parent traffic policy and add the child traffic policy in it.

The parent traffic policy is the H-QoS traffic policy and it can be applied on physical or bundle main interfaces and sub-interfaces.

Configuration Example

Configuration of a class-map is as follows:

```
Router# configure
Router(config)# class-map match-any tc2
Router(config-cmap)# match traffic-class 1
Router(config-cmap)# end-class-map
Router(config)# commit
```

Configuration of a child traffic policy is as follows:

```
Router# configure
Router(config)# policy-map child
Router(config-pmap)# class tc2
Router(config-pmap-c)# shape average percent 20
Router(config-pmap-c)# exit
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average percent 1
Router(config-pmap)# end-policy-map
Router(config)# commit
```

Configuration of a parent traffic policy is as follows:

```
Router# configure
Router(config)# policy-map parent
Router(config-pmap)# class class-default
Router(config-pmap-c)# service-policy child
Router(config-pmap-c)# shape average percent 50
Router(config-pmap)# end-policy-map
Router(config)# commit
```

Running Configuration

```
/* Configuration of a Class-map */
class-map match-any tc2
 match traffic-class 1
 end-class-map
!
/* Configuration of a Child Traffic Policy */
policy-map child
 class tc2
  shape average percent 20
 !
 class class-default
  shape average percent 1
 !
 end-policy-map
!
/* Configuration of a Parent Traffic Policy */
policy-map parent
 class class-default
  service-policy child
  shape average percent 50
 !
 end-policy-map
!
```

Applying the Parent Traffic Policy on a Main Interface

```
Router# configure
Router(config)# Interface TenGigE 0/0/0/10
Router(config-int)# service-policy output parent
Router(config-int)# commit
```

Applying the Parent Traffic Policy on a Sub-interface

```
Router# configure
Router(config)# Interface TenGigE 0/0/0/10.1
Router(config-int)# service-policy output parent
Router(config-int)# commit
```

Verification

Verify if the H-QoS traffic policy is applied correctly on the interface using the commands **show qos interface** *interface-name* **output**. In the following example, the **Level1 Class** gives information about the class-map that is associated with the parent traffic policy and the **Level2 Class** gives information about the class-maps that are associated with the child traffic policy.

```
RP/0/RP0/CPU0:ios#show qos interface ten0/0/0/10 output

NOTE:- Configured values are displayed within parentheses
Interface TenGigE0/0/0/10 ifh 0x1e0  -- output policy
NPU Id:                        0
Total number of classes:       3
Interface Bandwidth:           10000000 kbps
VOQ Base:                      1136
Accounting Type:               Layer1 (Include Layer 1 encapsulation and above)
------------------------------------------------------------------------------
Level1 Class                              =   class-default
Queue Max. BW.                            =   no max (50 %)
Queue Min. BW.                            =   0 kbps (default)
Inverse Weight / Weight                   =   0 / (BWR not configured)
   Level2 Class                              =   tc2
   Egressq Queue ID                          =   1138 (LP queue)
   Queue Max. BW.                            =   1020015 kbps (20 %)
   Queue Min. BW.                            =   0 kbps (default)
   Inverse Weight / Weight                   =   1 / (BWR not configured)
   Guaranteed service rate                   =   1000000 kbps
   TailDrop Threshold                        =   1253376 bytes / 10 ms (default)
   WRED not configured for this class
   Level2 Class                              =   class-default
   Egressq Queue ID                          =   1136 (Default LP queue)
   Queue Max. BW.                            =   50625 kbps (1 %)
   Queue Min. BW.                            =   0 kbps (default)
   Inverse Weight / Weight                   =   1 / (BWR not configured)
   Guaranteed service rate                   =   50000 kbps
   TailDrop Threshold                        =   62720 bytes / 10 ms (default)
   WRED not configured for this class
```

The statistics for the packets that have matched the different traffic classes of the parent and child traffic policies can be viewed using the command **show policy-map interface** *interface-name* **output**. Also, this command also shows the number of packets that are transmitted or dropped when the specified action is applied on the packets that have matched the respective traffic class.

```
Router# show policy-map interface ten0/0/0/10 output

TenGigE0/0/0/10 output: parent
```

```
Class class-default
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :          2313578823/296138089344          8494665
    Transmitted         :           232805738/29799134464           854465
    Total Dropped       :          2080773085/266338954880          7640200
  Policy child Class tc2
    Classification statistics          (packets/bytes)     (rate - kbps)
      Matched             :          2313578823/296138089344          8494665
      Transmitted         :           232805738/29799134464           854465
      Total Dropped       :          2080773085/266338954880          7640200
    Queueing statistics
      Queue ID                              : 1138

      Taildropped(packets/bytes)            : 2080773085/266338954880
  Policy child Class class-default
    Classification statistics          (packets/bytes)     (rate - kbps)
      Matched             :                  0/0                   0
      Transmitted         :                  0/0                   0
      Total Dropped       :                  0/0                   0
    Queueing statistics
      Queue ID                              : 1136

      Taildropped(packets/bytes)            : 0/0
```

When using hierarchical policers, there is no independent set of hardware counters to store the parent policer statistics. Instead, parent policer statistics are manipulated in the software to be the sum of all child policers under the same policy-map.

This is shown in the following example where two streams of traffic, with CoS value of 1 and 2 are sent at a speed of 3.5 Gbps each.

```
/*Hierarchical Policy Map Configuration*/
=====================================================
Router# show running-config policy-map Hingress
policy-map Hingress
 class class-default
  service-policy ingress
  police rate 5 gbps peak-rate 9 gbps
  !
 !
 end-policy-map
!
 /*Ingress Policy Map Configuration*/
====================================
Router#show running-config policy-map ingress
policy-map ingress
 class cos1
  set traffic-class 1
  police rate 5 gbps
  !
 !
 class cos2
  set traffic-class 2
  police rate 5 gbps
  !
 !
 class class-default
 !
 end-policy-map
!
 /*Policy Map applied at TenGigE0/0/0/6.100 Interface*/
=====================================================
Router#show policy-map interface tenGigE 0/0/0/6.100 input
```

```
TenGigE0/0/0/6.100 input: Hingress

Class class-default
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :           856717937/109659895936         6683676
    Transmitted         :           856717937/109659895936         6683676
    Total Dropped       :                   0/0                    0
  Policing statistics                (packets/bytes)     (rate - kbps)
    Policed(conform)    :           856717937/109659895936         6683674
    Policed(exceed)     :                   0/0                    0
    Policed(violate)    :                   0/0                    0
    Policed and dropped :                   0/0

  Policy ingress Class cos1
    Classification statistics          (packets/bytes)     (rate - kbps)
      Matched             :           437826303/56041766784          3341838
      Transmitted         :           437826303/56041766784          3341838
      Total Dropped       :                   0/0                    0
    Policing statistics                (packets/bytes)     (rate - kbps)
      Policed(conform)    :           437826303/56041766784          3341838
      Policed(exceed)     :                   0/0                    0
      Policed(violate)    :                   0/0                    0
      Policed and dropped :                   0/0
      Policed and dropped(parent policer)  : 0/0

  Policy ingress Class cos2
    Classification statistics          (packets/bytes)     (rate - kbps)
      Matched             :           418891634/53618129152          3341838
      Transmitted         :           418891634/53618129152          3341838
      Total Dropped       :                   0/0                    0
    Policing statistics                (packets/bytes)     (rate - kbps)
      Policed(conform)    :           418891634/53618129152          3341838
      Policed(exceed)     :                   0/0                    0
      Policed(violate)    :                   0/0                    0
      Policed and dropped :                   0/0
      Policed and dropped(parent policer)  : 0/0

  Policy ingress Class class-default
    Classification statistics          (packets/bytes)     (rate - kbps)
      Matched             :                   0/0                    0
      Transmitted         :                   0/0                    0
      Total Dropped       :                   0/0                    0
Policy Bag Stats time: 0
Policy Bag Stats time: 0
```

Configuration Example for Three-Level H-QoS

To configure three-level H-QoS:

1. Configure the port shaper or EFP group shaper.
2. Configure the two-level H-QoS policy with an EFP parent shaper and class or service level actions
3. Enable the port or EFP group shaper on the main interface to become the root policy.
4. Enable the two-level H-QoS policy on each EFP instance, thus providing a three-level hierarchy of service, EFP and EFP group or port SLAs.

A sample configuration for three-level H-QoS is as follows.

```
policy-map port_shaper
 class class-default
```

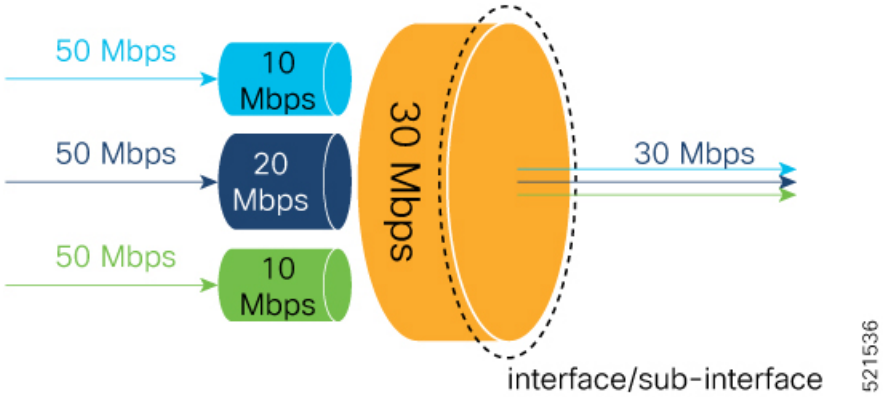
```
  shape average 6 gbps
 !
 end-policy-map
!

policy-map efp_policy
 class class-default
  service-policy efp_policy_child
  shape average 4 gbps
 !
 end-policy-map

!
policy-map efp_policy_child
 class tc1
  shape average 50 mbps
  priority level 1
 !
 class tc2
  bandwidth percent 50
 !
 class tc3
  bandwidth percent 30
 !
 class class-default
 !
 end-policy-map
!

interface TenGigE0/5/0/4
 service-policy output port_shaper
!

interface TenGigE0/5/0/4.1
 service-policy output efp_policy
 encapsulation dot1q 11
!

interface TenGigE0/5/0/4.2
 service-policy output efp_policy
 encapsulation dot1q 12
!
```

Verification

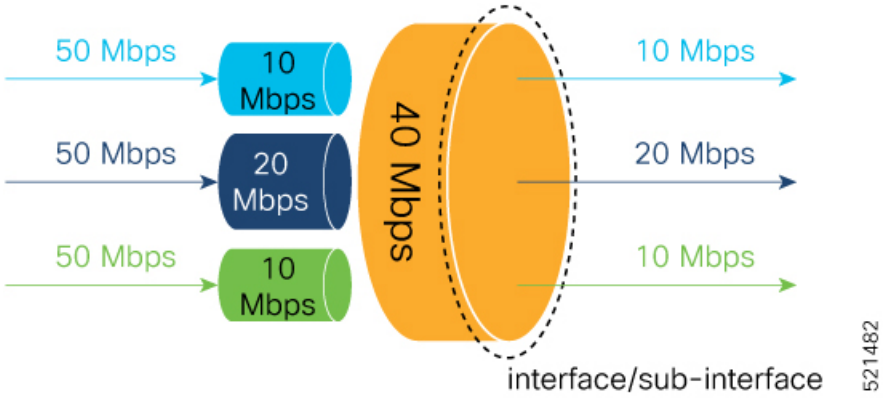
Run the **show policy-map interface** command in XR EXEC mode to display the packet/byte count and rate post port shaper enforcement on each of the sub-interface/EFP policies.

Conform Aware Hierarchical Policy Overview

Hierarchical QoS (H-QoS), while allowing for granular and multi-level management of traffic, does not allow for conform traffic from a child-level policy to a parent-level policy to get priority. This means that in case of excess traffic, the parent policer drops conform traffic packets from the child level as well.



The conform-aware hierarchical policy feature enables the parent-level policy to prioritize conform traffic from child-level policy over exceed and violate traffic.



Here is how it works: the child-level policer initially marks its packets as red, yellow, or green. Packets are marked based on the committed information rate (CIR) value and the two associated burst sizes - committed burst size (CBS) and excess burst size (EBS). If a packet does not exceed the CBS, it is marked as conformed packet (green). The packet is marked as exceeded if it exceeds CBS, but not the EBS (yellow). If it exceeds the EBS as well, it is marked as violate packet (red).

When the packets arrive at the parent level policer (which is color aware), the policer prioritizes the packets marked green over the packets marked yellow. After all the conform traffic (green) is transmitted and there are tokens available still, the yellow packets are transmitted next, instead of being marked as violate traffic (red).

To enable the conform-aware hierarchical policy feature run the command.

```
hw-module profile qos conform-aware-policer
```

Configuring Conform Aware Hierarchy Policy

To enable and configure shared policer:

1. Run the **hw-module profile qos conform-aware-policer** command.
2. Reload the affected line card.
3. Configure class maps to be used for matching packets to the class specified.

4. Create a child policy map.
5. Configure traffic policing for the classes in the child policy map.
6. Attach the child policy-map to the parent's class-default class.
7. Configure traffic policing for the parent policy map.

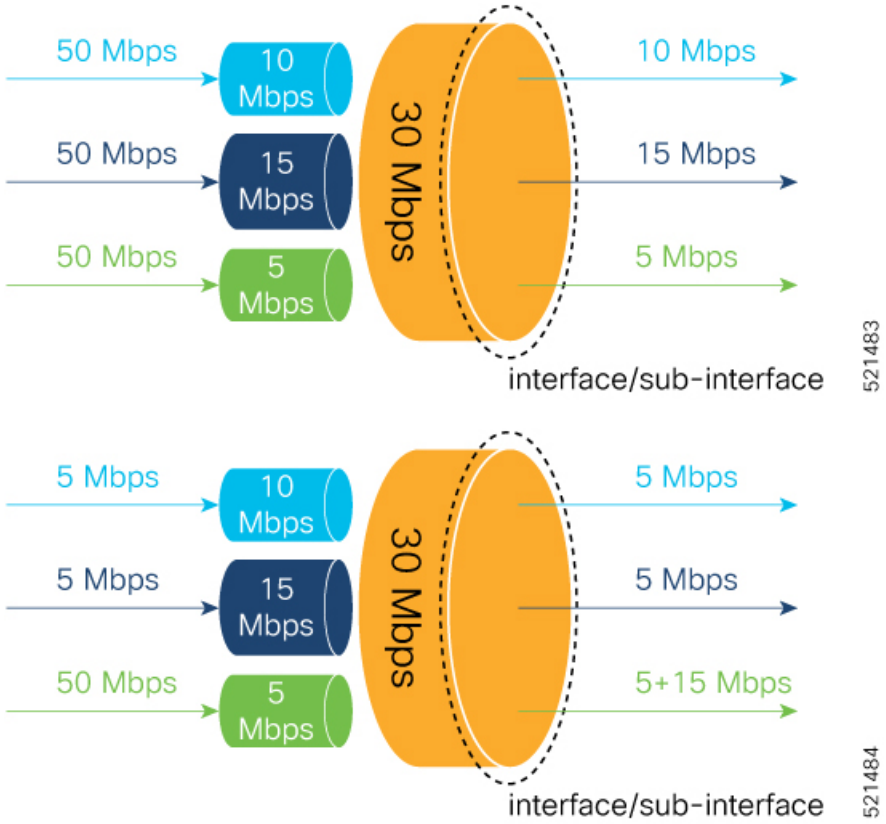
```
RP/0/RP0/CPU0:ios(config)#hw-module profile qos conform-aware-policer
/* To activate this profile, you must manually reload the chassis or all line cards */
RP/0/RP0/CPU0:ios(config)#exit
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
RP/0/RP0/CPU0:router# reload location 0/0/CPU0
RP/0/RSP0/CPU0:ios(config)# policy-map CHILD-INGRESS
RP/0/RSP0/CPU0:ios(config-pmap)# class REAL-TIME
RP/0/RSP0/CPU0:ios(config-pmap-c)# set traffic-class 5
RP/0/RSP0/CPU0:ios(config-pmap-c-police)#police rate 20 mbps
RP/0/RSP0/CPU0:ios(config-pmap)# class NET-CONTROL
RP/0/RSP0/CPU0:ios(config-pmap-c)# set traffic-class 4
RP/0/RSP0/CPU0:ios(config-pmap-c-police)#police rate 5 mbps
RP/0/RSP0/CPU0:ios(config-pmap)# class DATA1
RP/0/RSP0/CPU0:ios(config-pmap-c)# set traffic-class 3
RP/0/RSP0/CPU0:ios(config-pmap-c-police)#police rate 5 mbps
RP/0/RSP0/CPU0:ios(config-cmap)# set qos-group 3
RP/0/RSP0/CPU0:ios(config-cmap)# class DATA2
RP/0/RSP0/CPU0:ios(config-pmap-c)# set traffic-class 2
RP/0/RSP0/CPU0:ios(config-pmap-c-police)#police rate 5 mbps
RP/0/RSP0/CPU0:ios(config-cmap)# set qos-group 3
RP/0/RSP0/CPU0:ios(config-pmap-c)#class class-default
RP/0/RSP0/CPU0:ios(config-pmap-c-police)#police rate 10 mbps
RP/0/RSP0/CPU0:ios(config-pmap)#end-policy-map
RP/0/RSP0/CPU0:ios(config)# policy-map PARENT-INGRESS
RP/0/RSP0/CPU0:ios(config-pmap)#class class-default
RP/0/RSP0/CPU0:ios(config-cmap-c)# service-policy CHILD-INGRESS
RP/0/RSP0/CPU0:ios(config-pmap-c-police)#police rate 100 mbps
RP/0/RSP0/CPU0:ios(config-pmap)#end-policy-map
RP/0/RSP0/CPU0:ios(config-pmap)#
RP/0/RSP0/CPU0:ios(config-pmap)# interface GigabitEthernet0/0/0/22.13
RP/0/RSP0/CPU0:ios(config-pmap)# service-policy in parent-policer
RP/0/RSP0/CPU0:ios (config-subif)#end
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
```

Running Configuration

```
RP/0/RP0/CPU0:ios#show run interface GigabitEthernet0/0/0/22.13
Tue Feb 23 20:45:08.889 GMT
interface GigabitEthernet0/0/0/22.13
 description Testing interface
 service-policy input parent-policer
 vrf customer1
 ipv4 address 172.16.1.1 255.255.255.252

policy-map PARENT-INGRESS
  class class-default
    service-policy CHILD-INGRESS
    police rate 100 mbps
  !
end-policy-map
!
policy-map CHILD-INGRESS
  class REAL-TIME
    set traffic-class 5
    police rate 20 mbps
```

```
  !
  class NET-CONTROL
    set traffic-class 4
    police rate 5 mbps
  !
  class DATA1
    set traffic-class 3
    police rate 5 mbps
    set qos-group 3
  !
  class DATA2
    set traffic-class 2
    police rate 5 mbps
    set qos-group 3
 !
  class class-default
    police rate 10 mbps
!
end-policy-map
```

Figure 1: Parent Conform Aware-Ingress Policing




```
RP/0/RSP0/CPU0:ios(config)#policy-map parent-policer
RP/0/RSP0/CPU0:ios(config-pmap)#class class-default
RP/0/RSP0/CPU0:ios(config-pmap-c)#service-policy child-policer
RP/0/RSP0/CPU0:ios(config-pmap-c)#police rate 30 mbps peak-rate 30 mbps
RP/0/RSP0/CPU0:ios(config-pmap-c-police)#end-policy-map
RP/0/RSP0/CPU0:ios(config)# policy-map child-policer
RP/0/RSP0/CPU0:ios(config-pmap)# class cos3
RP/0/RSP0/CPU0:ios(config-pmap-c-police)#police rate 10 mbps peak-rate 30 mbps
RP/0/RSP0/CPU0:ios(config-pmap)# class cos4
```

```
RP/0/RSP0/CPU0:ios(config-pmap-c-police)#police rate 15 mbps peak-rate 30 mbps
RP/0/RSP0/CPU0:ios(config-pmap-c)#class class-default
RP/0/RSP0/CPU0:ios(config-pmap-c-police)#police rate 5 mbps peak-rate 30 mbps
RP/0/RSP0/CPU0:ios(config-pmap)#end-policy-map
```

Running Configuration

```
RP/0/RP0/CPU0:ios#show run policy-map parent-policer
Tue Feb 23 19:37:57.500 GMT
policy-map parent-policer
 class class-default
  service-policy child-policer
  police rate 30 mbps peak-rate 30 mbps
  !
 end-policy-map

RP/0/RP0/CPU0:ios#show run policy-map child-policer
Tue Feb 23 19:38:35.472 GMT
policy-map child-policer
 class cos3
  police rate 10 mbps peak-rate 30 mbps
  !
 class cos4
  police rate 15 mbps peak-rate 30 mbps
  !
 class class-default
  police rate 5 mbps peak-rate 30 mbps
  !
 end-policy-map
```

Verification

Run the **show policy-map interface** command in XR EXEC mode to confirm that the committed information rate (CIR) is prioritized over the peak information rate (PIR).

In the example below, **Policed (conform)** or CIR from each class is prioritized over **Policed (exceed)** or PIR.

```
RP/0/RP0/CPU0:ios# show policy-map interface GigabitEthernet0/0/0/22.13 input

GigabitEthernet0/0/0/22.13 input: parent-policer

Class class-default
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :           217797200/111512166400         2344847
    Transmitted         :             8314388/4256966656           88089
    Total Dropped       :           209482812/107255199744         2256758

  Policing statistics                (packets/bytes)     (rate - kbps)
    Policed(conform)    :             6602174/3380313088           69926
    Policed(exceed)     :             1712214/876653568            18165
    Policed(violate)    :           209482812/107255199744         2256782
    Policed and dropped :           209482812/107255199744

  Policy child-policer Class cos3
    Classification statistics          (packets/bytes)     (rate - kbps)
      Matched             :            54449300/27878041600          586215
      Transmitted         :             3246813/1662368256           34399
      Total Dropped       :            51202487/26215673344          551816
    Policing statistics                (packets/bytes)     (rate - kbps)
      Policed(conform)    :             2818471/1443057152           29851
      Policed(exceed)     :              428342/219311104            4547
      Policed(violate)    :            51202487/26215673344          551816
      Policed and dropped :            51202487/26215673344
```

```
    Policed and dropped(parent policer)  : 0/0

Policy child-policer Class cos4
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :            54449300/27878041600          586213
    Transmitted         :             2319731/1187702272           24577
   Total Dropped       :            52129569/26690339328          561636
  Policing statistics                (packets/bytes)     (rate - kbps)
    Policed(conform)    :             1891851/968627712            20037
    Policed(exceed)     :              427880/219074560            4540
    Policed(violate)    :            52129569/26690339328          561636
    Policed and dropped :            52129569/26690339328
    Policed and dropped(parent policer)  : 0/0

Policy child-policer Class class-default
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :           108898600/55756083200          1172419
    Transmitted         :             2747844/1406896128           29113
    Total Dropped       :           106150756/54349187072          1143306
  Policing statistics                (packets/bytes)     (rate - kbps)
    Policed(conform)    :             1891852/968628224            20036
    Policed(exceed)     :              855992/438267904            9076
    Policed(violate)    :           106150756/54349187072          1143306
    Policed and dropped :           106150756/54349187072
    Policed and dropped(parent policer)  : 0/0
```

Related Commands hw-module profile qos conform-aware-policer

Conform Aware Hierarchical Policy Restrictions

The parent-policy traffic police rate must be greater than or equal to the sum of child conform rates.

- The Confirm-Aware hierarchical policer function allows the child level policy to color the traffic as either green, yellow, or red for handling at the parent-level policer.
- All classes in the child-level of the QoS policy need to have a policer configured for their packets to be handled correctly by the parent policer.