Configuring Modular QoS Congestion Management

This chapter covers the following topics:

Congestion Management Overview

Congestion management features allow you to control congestion by determining the order in which a traffic flow (or packets) is sent out an interface based on priorities assigned to packets. Congestion management entails the creation of queues, assignment of packets to those queues based on the classification of the packet, and scheduling of the packets in a queue for transmission.

The types of traffic regulation mechanisms supported are:

Class-based Weighted Fair Queueing

Class-based Weighted Fair Queueing (CBWFQ) allows definition of traffic classes based on customer match criteria. With CBWFQ you can define traffic classes and assign guaranteed amount of minimum bandwidth to them. CBWFQ also allows for a strict priority queue for delay-sensitive traffic.

Bandwidth Remaining

The CBWFQ algorithm derives the weight for each class from the bandwidth remaining value allocated to the class. The **bandwidth remaining** option specifies a weight for the class to the CBWFQ . After the priority-queue is serviced, the leftover bandwidth is distributed as per bandwidth remaining ratio (BWRR) or percentage. If you do not configure this command for any class, the default value of the BWRR is considered as 1 (one). In the case of **bandwidth remaining percent**, the remaining bandwidth is equally distributed among other classes, to make it 100 percentage (100%).

Restrictions

• The **bandwidth remaining** command is supported only for egress policies.

Configuring Bandwidth Remaining – Instance 2

Supported Platforms: Cisco NCS 5500, Cisco NCS 540, and Cisco NCS 560 Series Routers

This procedure configures the minimum bandwidth and bandwidth remaining on the router

Note The **bandwidth**, **bandwidth remaining**, **shaping**, **queue-limit** and wred commands may be configured together in the same class. But, **priority** cannot be configured along with **bandwidth**, **bandwidth remaining** and wred commands.

You can configure shape average command along with priority command.

Configuration Example

You have to accomplish the following to complete the minimum bandwidth and bandwidth remaining configuration:

1. Creating or modifying a policy-map that can be attached to one or more interfaces

2. Specifying the traffic class whose policy has to be created or changed

3. Allocating the minimum bandwidth and leftover bandwidth for the class

4. Attaching the policy-map to an output interface

```
Router# configure
Router(config)#class-map qos-6
Router(config-cmap)#match traffic-class 4
Router(config-cmap)#exit
Router(config-cmap)#commit

Router(config)#class-map qos-5
Router(config-cmap)#match traffic-class 5
Router(config-cmap)#commit

Router(config)# policy-map test-bw-bw-rem
Router(config-pmap)# class qos-6
Router(config-pmap-c)# bandwidth percent 60
Router(config-pmap-c)# bandwidth remaining percent 60
```

```
Router(config-pmap)#class qos-5
Router(config-pmap-c)#bandwidth percent 20
Router(config-pmap-c)#bandwidth remaining percent 40
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface HundredGigE 0/6/0/18
Router(config-if)# service-policy output test-bw-bw-rem
Router(config-if)# commit
```

Running Configuration

```
policy-map test-bw-bw-rem
 class qos-6
  bandwidth percent 60
  bandwidth remaining percent 60
 !
 class qos-5
  bandwidth percent 20
  bandwidth remaining percent 40
 !
 class class-default
 !
 end-policy-map
!

interface HundredGigE0/6/0/18
 service-policy output test-bw-bw-rem
!
```

Verification

```
Router# show qos interface HundredGigE 0/6/0/18 output

NOTE:- Configured values are displayed within parentheses
Interface HundredGigE0/6/0/18 ifh 0x3000220  -- output policy
NPU Id:                       3
Total number of classes:      3
Interface Bandwidth:          100000000 kbps
VOQ Base:                     11176
VOQ Stats Handle:             0x88550ea0
Accounting Type:              Layer1 (Include Layer 1 encapsulation and above)
------------------------------------------------------------------------------
Level1 Class                           =   qos-6
Egressq Queue ID                       =   11182 (LP queue)
Queue Max. BW.                         =   100824615 kbps (default)
Queue Min. BW.                         =   60494769 kbps (60 %)
Inverse Weight / Weight                =   2 (60%)
Guaranteed service rate                =   71881188 kbps
TailDrop Threshold                     =   90177536 bytes / 10 ms (default)
WRED not configured for this class

Level1 Class                           =   qos-5
Egressq Queue ID                       =   11181 (LP queue)
Queue Max. BW.                         =   100824615 kbps (default)
Queue Min. BW.                         =   20164923 kbps (20 %)
Inverse Weight / Weight                =   3 (40%)
Guaranteed service rate                =   27920792 kbps
TailDrop Threshold                     =   35127296 bytes / 10 ms (default)
WRED not configured for this class
```

```
Level1 Class                          =   class-default
Egressq Queue ID                      =   11176 (Default LP queue)
Queue Max. BW.                        =   101803495 kbps (default)
Queue Min. BW.                        =   0 kbps (default)
Inverse Weight / Weight               =   120 (BWR not configured)
Guaranteed service rate               =   198019 kbps
TailDrop Threshold                    =   247808 bytes / 10 ms (default)
WRED not configured for this class
```

Related Topics

- Bandwidth Remaining, on page 2

Associated Commands

- bandwidth remaining

Low-Latency Queueing with Strict Priority Queueing

The Low-Latency Queueing (LLQ) feature brings strict priority queuing (PQ) to the CBWFQ scheduling mechanism. Priority Queueing (PQ) in strict priority mode ensures that one type of traffic is sent, possibly at the expense of all others. For PQ, a low-priority queue can be detrimentally affected, and, in the worst case, never allowed to send its packets if a limited amount of bandwidth is available or the transmission rate of critical traffic is high.

Configure Low Latency Queueing with Strict Priority Queueing

Configuring low latency queueing (LLQ) with strict priority queuing (PQ) allows delay-sensitive data such as voice to be de-queued and sent before the packets in other queues are de-queued.

Guidelines

- Only priority level 1 to 7 is supported, with 1 being the highest priority and 7 being the lowest. However, the default CoSQ 0 has the lowest priority among all.

- Priority level 1 to 7 is supported for non-H-QoS profiles, with 1 being the highest priority and 7 being the lowest. For H-QoS profiles, priority level 1 to 4 is supported. For all profiles, however, the class default is CoSQ 0 and has the lowest priority among all.

- Egress policing is not supported. Hence, in the case of strict priority queuing, there are chances that the other queues do not get serviced.

- You can configure **shape average** and **queue-limit** commands along with **priority**.

Configuration Example

You have to accomplish the following to complete the LLQ with strict priority queuing:

1. Creating or modifying a policy-map that can be attached to one or more interfaces

2. Specifying the traffic class whose policy has to be created or changed

3. Specifying priority to the traffic class

4. (Optional) Shaping the traffic to a specific bit rate

5. Attaching the policy-map to an output interface

```
Router# configure
Router(config)#class-map qos-1
Router(config-cmap)#match traffic-class 1
Router(config-cmap)#commit

Router(config)#class-map qos-2
Router(config-cmap)#match traffic-class 2
Router(config-cmap)#commit

Router(config)# policy-map test-priority-1
Router(config-pmap)# class qos1
Router(config-pmap-c)# priority level 7
Router(config-pmap-c)# shape average percent 2
Router(config-pmap-c)# class qos-2
Router(config-pmap-c)# priority level 6
Router(config-pmap-c)# shape average percent 1
Router(config-pmap-c)# commit
Router(config-pmap-c# exit
Router(config-pmap)# exit

Router(config)# interface HundredGigE 0/0/0/20
Router(config-if)# service-policy output test-priority-1
Router(config-if)# commit
```

Running Configuration

```
policy-map test-priority-1
 class qos-1
  priority level 7
  shape average percent 2
 !
 class qos-2
  priority level 6
  shape average percent 1
 !
 class class-default
 !
 end-policy-map
!

interface HundredGigE0/0/0/20
 service-policy output test-priority-1
!
```

Verification

```
Router#  show qos int hundredGigE 0/0/0/20 output

NOTE:- Configured values are displayed within parentheses
Interface HundredGigE0/0/0/20 ifh 0x38  -- output policy
NPU Id:                        0
```

```
Total number of classes:       3
Interface Bandwidth:           100000000 kbps
Policy Name:                   test-priority-1
VOQ Base:                      1184
Accounting Type:               Layer1 (Include Layer 1 encapsulation and above)
-------------------------------------------------------------------------
Level1 Class (HP7)                      =    qos-1
Egressq Queue ID                        =    1185 (HP7 queue)
Queue Max. BW.                          =    2000000 kbps (2 %)
Guaranteed service rate                 =    2000000 kbps
Peak burst                              =    36864 bytes (default)
TailDrop Threshold                      =    2499840 bytes / 10 ms (default)
WRED not configured for this class

Level1 Class (HP6)                      =    qos-2
Egressq Queue ID                        =    1186 (HP6 queue)
Queue Max. BW.                          =    1000000 kbps (1 %)
Guaranteed service rate                 =    1000000 kbps
Peak burst                              =    36864 bytes (default)
TailDrop Threshold                      =    1249792 bytes / 10 ms (default)
WRED not configured for this class

Level1 Class                            =    class-default
Egressq Queue ID                        =    1184 (Default LP queue)
Queue Max. BW.                          =    no max (default)
Queue Min. BW.                          =    0 kbps (default)
Inverse Weight / Weight                 =    1 / (BWR not configured)
Guaranteed service rate                 =    97000000 kbps
Peak burst                              =    36864 bytes (default)
TailDrop Threshold                      =    121249792 bytes / 10 ms (default)
WRED not configured for this class
```

Associated Commands

- priority

Traffic Shaping

Traffic shaping allows you to control the traffic flow exiting an interface to match its transmission to the speed of the remote target interface and ensure that the traffic conforms to policies contracted for it. Traffic adhering to a particular profile can be shaped to meet downstream requirements, hence eliminating bottlenecks in topologies with data-rate mismatches.

Note Traffic shaping is supported only in egress direction.

Configure Traffic Shaping

The traffic shaping performed on outgoing interfaces is done at the Layer 1 level and includes the Layer 1 header in the rate calculation.

Guidelines

- Only egress traffic shaping is supported.

- It is mandatory to configure all the eight qos-group classes (including class-default) for the egress policies.

- You can configure **shape average** command along with **priority** command.

Configuration Example

You have to accomplish the following to complete the traffic shaping configuration:

1. Creating or modifying a policy-map that can be attached to one or more interfaces

2. Specifying the traffic class whose policy has to be created or changed

3. Shaping the traffic to a specific bit rate

4. Attaching the policy-map to an output interface

```
Router# configure
Router(config)#class-map c5
Router(config-cmap)#match traffic-class 5
Router(config-cmap)#commit

Router(config)# policy-map egress_policy1
Router(config-pmap)# class c5
Router(config-pmap-c)# shape average percent 40
Router(config-pmap-c# exit
Router(config-pmap)# exit
Router(config)# interface HundredGigE 0/1/0/0
Router(config-if)# service-policy output egress_policy1
Router(config-if)# commit
```

Running Configuration

```
class-map c5
 match traffic-class 5
commit

policy-map egress_policy1
 class c5
  shape average percent 40
 !
 class class-default
 !
 end-policy-map
!

interface HundredGigE0/6/0/18
 service-policy output egress_policy1
!
```

Verification

```
Router#  show qos interface hundredGigE 0/6/0/18 output

NOTE:- Configured values are displayed within parentheses
Interface HundredGigE0/6/0/18 ifh 0x3000220  -- output policy
NPU Id:                       3
```

```
Total number of classes:        2
Interface Bandwidth:            100000000 kbps
VOQ Base:                       11176
VOQ Stats Handle:               0x88550ea0
Accounting Type:                Layer1 (Include Layer 1 encapsulation and above)
----------------------------------------------------------------------------
Level1 Class                                =   c5
Egressq Queue ID                            =   11177 (LP queue)
Queue Max. BW.                              =   40329846 kbps (40 %)
Queue Min. BW.                              =   0 kbps (default)
Inverse Weight / Weight                     =   1 (BWR not configured)
Guaranteed service rate                     =   40000000 kbps
TailDrop Threshold                          =   50069504 bytes / 10 ms (default)
WRED not configured for this class

Level1 Class                                =   class-default
Egressq Queue ID                            =   11176 (Default LP queue)
Queue Max. BW.                              =   101803495 kbps (default)
Queue Min. BW.                              =   0 kbps (default)
Inverse Weight / Weight                     =   1 (BWR not configured)
Guaranteed service rate                     =   50000000 kbps
TailDrop Threshold                          =   62652416 bytes / 10 ms (default)
WRED not configured for this class
```

Important Notes

Related Topics

- Congestion Management Overview, on page 1

Associated Commands

- shape average

Traffic Policing

Traffic policing allows you to control the maximum rate of traffic sent or received on an interface and to partition a network into multiple priority levels or class of service (CoS).Traffic policing manages the maximum rate of traffic through a token bucket algorithm. The token bucket algorithm uses user-configured values to determine the maximum rate of traffic allowed on an interface at a given moment in time. The token bucket algorithm is affected by all traffic entering or leaving the interface (depending on where the traffic policy with traffic policing is configured) and is useful in managing network bandwidth in cases where several large packets are sent in the same traffic stream. By default, the configured bandwidth value takes into account the Layer 2 encapsulation that is applied to traffic leaving the interface.

Traffic policing also provides a certain amount of bandwidth management by allowing you to set the burst size (Bc) for the committed information rate (CIR). See, Committed Bursts and Excess Bursts, on page 9.

The router supports the following traffic policing mode(s):

- Single-Rate Two-Color (SR2C) in color-blind mode. See Single-Rate Policer, on page 9.

- Single-Rate Three-Color (SR3C) in color-blind mode.

- Two-Rate Three-Color (2R3C) in color-blind mode. See Two-Rate Policer, on page 13 .

Restrictions

- Traffic policing is supported only in ingress direction, and only color-blind mode is supported.

- The policing rate accuracy may vary up to +/-2% from the configured policer value.

- Policer marking is not supported.

- Policers are configured in the interface at the core level and "show qos int <>" value is displayed at the NPU level.

  For policers configured in a bundle interface where bundle members are from the same NPU but different cores (NPU cores), each member sends the traffic up to the core level policer configuration, but "show qos int <>" displays the NPU level policer output.

- Example:

  For bundle interface with two 10GE members (same NPU, but one interface from core0, one interface from core1) 2R3C policer applied on bundle interface (1G confirm rate , 1G exceed rate – total 2G policer rate) will be shown on the "show qos int <>" output):

  Interface in core0 – 500 Mbps confirm rate, 500 Mbps exceed rate

  Interface in core1 – 500 Mbps confirm rate, 500 Mbps exceed rate

  For traffic in one out of two interfaces, the policed rate will be 1Gbps. For traffic on two interfaces, policed rate will be 2Gbps.

Committed Bursts and Excess Bursts

Unlike a traffic shaper, a traffic policer does not buffer excess packets and transmit them later. Instead, the policer executes a "send or do not send" policy without buffering. Policing uses normal or committed burst (bc) values and excess burst values (be) to ensure that the router reaches the configured committed information rate (CIR). Policing decides if a packet conforms or exceeds the CIR based on the burst values you configure. Burst parameters are based on a generic buffering rule for routers, which recommends that you configure buffering to be equal to the round-trip time bit-rate to accommodate the outstanding TCP windows of all connections in times of congestion. During periods of congestion, proper configuration of the excess burst parameter enables the policer to drop packets less aggressively.
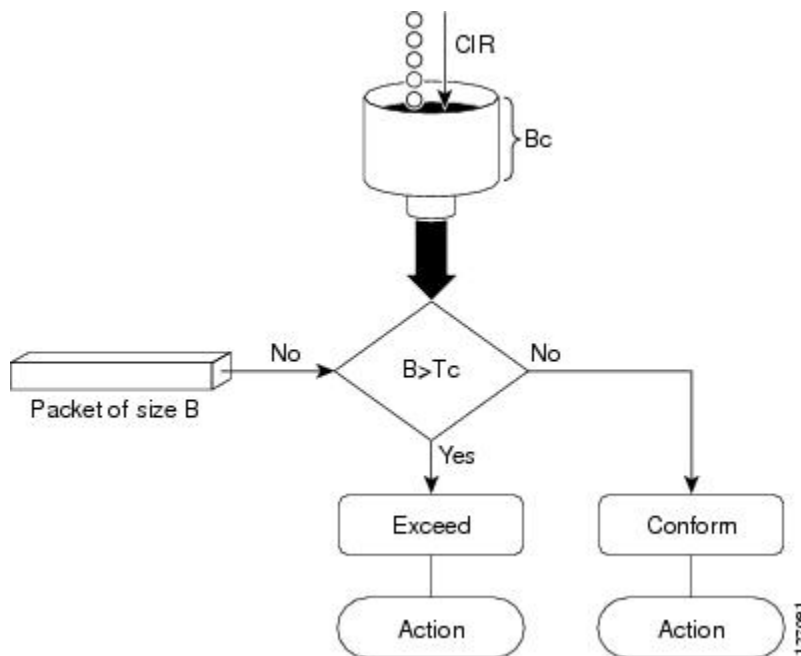
For more details, see Committed Bursts, on page 16 and Excess Bursts, on page 17.

Single-Rate Policer

Single-Rate Two-Color Policer

A single-rate two-color (SR2C) policer provides one token bucket with two actions for each packet: a conform action and an exceed action.

Figure 1: Workflow of Single-Rate Two-Color Policer

Based on the committed information rate (CIR) value, the token bucket is updated at every refresh time interval. The Tc token bucket can contain up to the Bc value, which can be a certain number of bytes or a period of time. If a packet of size B is greater than the Tc token bucket, then the packet exceeds the CIR value and a default action is performed. If a packet of size B is less than the Tc token bucket, then the packet conforms and a different default action is performed.

Single-Rate Three-Color Policer

A single-rate three-color (SR3C) policer provides one token bucket with three actions for each packet: a conform action, an exceed action and a violate action. The packet is marked based on the CIR value and the two associated burst size - committed burst size (CBS) and excess burst size (EBS). If a packet does not exceed the CBS, it is marked as conformed packet. The packet is marked as exceeded if it exceeds CBS, but not the EBS. If it exceeds the EBS as well, it is marked as violate packet.

Configure Traffic Policing (Single-Rate Two-Color)

Traffic policing is often configured on interfaces at the edge of a network to limit the rate of traffic entering or leaving the network. The default conform action for single-rate two color policer is to transmit the packet and the default exceed action is to drop the packet. Users cannot modify these default actions.

Configuration Example

You have to accomplish the following to complete the traffic policing configuration:

1. Creating or modifying a policy-map that can be attached to one or more interfaces

2. Specifying the traffic class whose policy has to be created or changed

3. (Optional) Specifying the marking action

4. Specifying the policy rate for the traffic

5. Attaching the policy-map to an input interface

```
Router# configure
Router(config)# policy-map test-police-1
Router(config-pmap)# class  ipv6-6
Router(config-pmap-c)# set dscp cs2 (optional)
Router(config-pmap-c)# set qos-group 7 (optional)
Router(config-pmap-c)# police rate percent 20 burst 10000 bytes
Router(config-pmap-c-police)# exit
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface HundredGigE 0/6/0/18
Router(config-if)# service-policy input test-police-1
Router(config-if)# commit
```

Running Configuration

```
class-map match-any ipv6-6
 match precedence 3
 end-class-map
!

policy-map test-police-1
 class ipv6-6
  set dscp cs2
  set qos-group 7
  police rate percent 20 burst 10000 bytes
  !
 !
 class class-default
 !
 end-policy-map
!

interface HundredGigE0/6/0/18
 service-policy input test-police-1
 service-policy output test-priority-1
!
```

Verification

```
Router# show qos interface hundredGigE 0/6/0/18 input

NOTE:- Configured values are displayed within parentheses
Interface HundredGigE0/6/0/18 ifh 0x3000220  -- input policy
NPU Id:                      3
Total number of classes:     2
Interface Bandwidth:         100000000 kbps
Accounting Type:             Layer1 (Include Layer 1 encapsulation and above)
----------------------------------------------------------------------------
Level1 Class                         =   ipv6-6
New dscp                             =   16
New qos group                        =   7

Policer Bucket ID                    =   0x102a0
Policer Stats Handle                 =   0x8a8090c0
Policer committed rate               =   19980000 kbps (20 %)
```

```
Policer conform burst                   =   9856 bytes (10000 bytes)

Level1 Class                            =   class-default

Default Policer Bucket ID               =   0x102a1
Default Policer Stats Handle            =   0x8a808e78
Policer not configured for this class
```

Related Topics

- Traffic Policing, on page 8

Associated Commands

- police rate

Configure Traffic Policing (Single-Rate Three-Color)

The default conform action and exceed actions for single-rate three-color policer are to transmit the packet and the default violate action is to drop the packet. User cannot modify these default actions.

Configuration Example

You have to accomplish the following to complete the traffic policing configuration:

1. Creating or modifying a policy-map that can be attached to one or more interfaces

2. Specifying the traffic class whose policy has to be created or changed

3. (Optional) Specifying the marking action

4. Configuring the policy rate for the traffic along with the peak-burst values

5. Attaching the policy-map to an input interface

```
Router# configure
Router(config)# policy-map test-police-1R3C
Router(config-pmap)# class ipv4-5
Router(config-pmap-c)# set qos-group 2 (optional)
Router(config-pmap-c)# police rate percent 20 burst 100000 bytes peak-burst 190000 bytes
Router(config-pmap-c-police)# exit
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface HundredGigE 0/6/0/18
Router(config-if)# service-policy input test-police-1R3C
Router(config-if)# commit
```

Running Configuration

```
class-map match-any ipv4-5
 match precedence 3
 end-class-map
!

policy-map test-police-1R3C
```

```
 class ipv4-5
  set qos-group 7
  police rate percent 20 burst 100000 bytes peak-burst 190000 bytes
  !
 !
 class class-default
 !
 end-policy-map
!

interface HundredGigE0/6/0/18
 service-policy input test-police-1R3C
 service-policy output test-priority-1
!
```

Verification

```
Router# show qos interface hundredGigE 0/6/0/18 input

NOTE:- Configured values are displayed within parentheses
Interface HundredGigE0/6/0/18 ifh 0x3000220  -- input policy
NPU Id:                        3
Total number of classes:       2
Interface Bandwidth:           100000000 kbps
Accounting Type:               Layer1 (Include Layer 1 encapsulation and above)
-------------------------------------------------------------------------
Level1 Class                           =   ipv4-5
New qos group                          =   2

Policer Bucket ID                      =   0x102a1
Policer Stats Handle                   =   0x8a8090c0
Policer committed rate                 =   19980000 kbps (20 %)
Policer conform burst                  =   99584 bytes (100000 bytes)
Policer exceed burst                   =   188672 bytes (190000 bytes)

Level1 Class                           =   class-default

Default Policer Bucket ID              =   0x102a1
Default Policer Stats Handle           =   0x8a808e78
Policer not configured for this class
```

Related Topics

- Traffic Policing, on page 8

### Associated Commands

- police rate

Two-Rate Policer

The two-rate policer manages the maximum rate of traffic by using two token buckets: the committed token bucket and the peak token bucket. The dual-token bucket algorithm uses user-configured values to determine the maximum rate of traffic allowed on a queue at a given moment. In this way, the two-rate policer can meter traffic at two independent rates: the committed information rate (CIR) and the peak information rate (PIR).

The dual-token bucket algorithm provides users with three actions for each packet—a conform action, an exceed action, and an optional violate action. Traffic entering a queue with the two-rate policer configured is placed into one of these categories. The actions are pre-determined for each category. The default conform and exceed actions are to transmit the packet, and the default violate action is to drop the packet.

This figure shows how the two-rate policer marks a packet and assigns a corresponding action to the packet.

Figure 2: Marking Packets and Assigning Actions—Two-Rate Policer
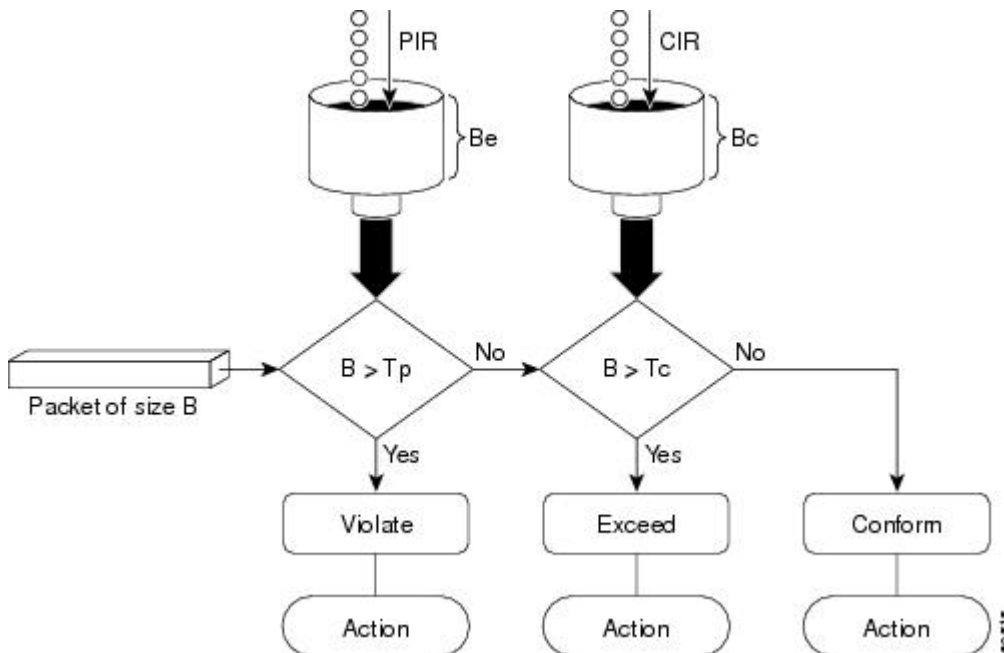
Also, see Two-Rate Policer Details, on page 17.

The router supports Two-Rate Three-Color (2R3C) policer.

Configure Traffic Policing (Two-Rate Three-Color)

The default conform and exceed actions for two-rate three-color (2R3C) policer are to transmit the packet and the default violate action is to drop the packet. Users cannot modify these default actions.

Configuration Example

You have to accomplish the following to complete the two-rate three-color traffic policing configuration:

1. Creating or modifying a policy-map that can be attached to one or more interfaces

2. Specifying the traffic class whose policy has to be created or changed

3. Specifying the packet marking

4. Configuring two rate traffic policing

5. Attaching the policy-map to an input interface

```
Router# configure
Router(config)# policy-map policy1
```

```
Router(config-pmap)# class ipv4-7
Router(config-pmap-c)# set qos-group 4
Router(config-pmap-c)# police rate percent 20 burst 100000 bytes peak-rate percent 50
peak-burst 200000 bytes
Router(config-pmap-c-police)# exit
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface HundredGigE 0/6/0/18
Router(config-if)# service-policy input policy1
Router(config-if)# commit
```

Running Configuration

```
policy-map policy1
 class ipv4-7
  set qos-group 4
  police rate percent 20 burst 100000 bytes peak-rate percent 50 peak-burst 200000 bytes
  !
 !

interface HundredGigE 0/6/0/18
 service-policy input policy1
!
```

Verification

```
Router# show policy-map interface HundredGigE 0/6/0/18

NOTE:- Configured values are displayed within parentheses
Interface HundredGigE0/6/0/18 ifh 0x3000220  -- input policy
NPU Id:                    3
Total number of classes:   8
Interface Bandwidth:       100000000 kbps
Accounting Type:           Layer1 (Include Layer 1 encapsulation and above)
----------------------------------------------------------------------------
Level1 Class                        =   ipv4-4
- - -
- - -
Level1 Class                        =   ipv4-7
New qos group                       =   4

Policer Bucket ID                   =   0x102a3
Policer Stats Handle                =   0x8a8089e8
Policer committed rate              =   19980000 kbps (20 %)
Policer peak rate                   =   49860000 kbps (50 %)
Policer conform burst               =   99584 bytes (100000 bytes)
Policer exceed burst                =   199168 bytes (200000 bytes)

Level1 Class                        =   class-default

Policer Bucket ID                   =   0x102a7
Policer Stats Handle                =   0x8a7c8510
Policer committed rate              =   29880000 kbps (30 %)
Policer conform burst               =   4194304 bytes (default)
```

Important Notes

- A policer is programmed per NPU core on a bundle interface. So, all members on a bundle interface from the same core share the policer.

Related Topics

-

Associated Commands

- police rate

References for Modular QoS Congestion Management

Committed Bursts

The committed burst (bc) parameter of the police command implements the first, conforming (green) token bucket that the router uses to meter traffic. The bc parameter sets the size of this token bucket. Initially, the token bucket is full and the token count is equal to the committed burst size (CBS). Thereafter, the meter updates the token counts the number of times per second indicated by the committed information rate (CIR).

The following describes how the meter uses the conforming token bucket to send packets:

- If sufficient tokens are in the conforming token bucket when a packet arrives, the meter marks the packet green and decrements the conforming token count by the number of bytes of the packet.

- If there are insufficient tokens available in the conforming token bucket, the meter allows the traffic flow to borrow the tokens needed to send the packet. The meter checks the exceeding token bucket for the number of bytes of the packet. If the exceeding token bucket has a sufficient number of tokens available, the meter marks the packet.

  Green and decrements the conforming token count down to the minimum value of 0.

  Yellow, borrows the remaining tokens needed from the exceeding token bucket, and decrements the exceeding token count by the number of tokens borrowed down to the minimum value of 0.

- If an insufficient number of tokens is available, the meter marks the packet red and does not decrement either of the conforming or exceeding token counts.

Note When the meter marks a packet with a specific color, there must be a sufficient number of tokens of that color to accommodate the entire packet. Therefore, the volume of green packets is never smaller than the committed information rate (CIR) and committed burst size (CBS). Tokens of a given color are always used on packets of that color.

Excess Bursts

The excess burst (be) parameter of the police command implements the second, exceeding (yellow) token bucket that the router uses to meter traffic. The exceeding token bucket is initially full and the token count is equal to the excess burst size (EBS). Thereafter, the meter updates the token counts the number of times per second indicated by the committed information rate (CIR).

The following describes how the meter uses the exceeding token bucket to send packets:

- When the first token bucket (the conforming bucket) meets the committed burst size (CBS), the meter allows the traffic flow to borrow the tokens needed from the exceeding token bucket. The meter marks the packet yellow and then decrements the exceeding token bucket by the number of bytes of the packet.

- If the exceeding token bucket does not have the required tokens to borrow, the meter marks the packet red and does not decrement the conforming or the exceeding token bucket. Instead, the meter performs the exceed-action configured in the police command (for example, the policer drops the packets).

Two-Rate Policer Details

The committed token bucket can hold bytes up to the size of the committed burst (bc) before overflowing. This token bucket holds the tokens that determine whether a packet conforms to or exceeds the CIR as the following describes:

- A traffic stream is conforming when the average number of bytes over time does not cause the committed token bucket to overflow. When this occurs, the token bucket algorithm marks the traffic stream green.

- A traffic stream is exceeding when it causes the committed token bucket to overflow into the peak token bucket. When this occurs, the token bucket algorithm marks the traffic stream yellow. The peak token bucket is filled as long as the traffic exceeds the police rate.

The peak token bucket can hold bytes up to the size of the peak burst (be) before overflowing. This token bucket holds the tokens that determine whether a packet violates the PIR. A traffic stream is violating when it causes the peak token bucket to overflow. When this occurs, the token bucket algorithm marks the traffic stream red.

For example, if a data stream with a rate of 250 kbps arrives at the two-rate policer, and the CIR is 100 kbps and the PIR is 200 kbps, the policer marks the packet in the following way:

- 100 kbps conforms to the rate

- 100 kbps exceeds the rate

- 50 kbps violates the rate

The router updates the tokens for both the committed and peak token buckets in the following way:

- The router updates the committed token bucket at the CIR value each time a packet arrives at the interface. The committed token bucket can contain up to the committed burst (bc) value.

- The router updates the peak token bucket at the PIR value each time a packet arrives at the interface. The peak token bucket can contain up to the peak burst (be) value.

- When an arriving packet conforms to the CIR, the router takes the conform action on the packet and decrements both the committed and peak token buckets by the number of bytes of the packet.

- When an arriving packet exceeds the CIR, the router takes the exceed action on the packet, decrements the committed token bucket by the number of bytes of the packet, and decrements the peak token bucket by the number of overflow bytes of the packet.

- When an arriving packet exceeds the PIR, the router takes the violate action on the packet, but does not decrement the peak token bucket.

See .