Congestion Management

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:

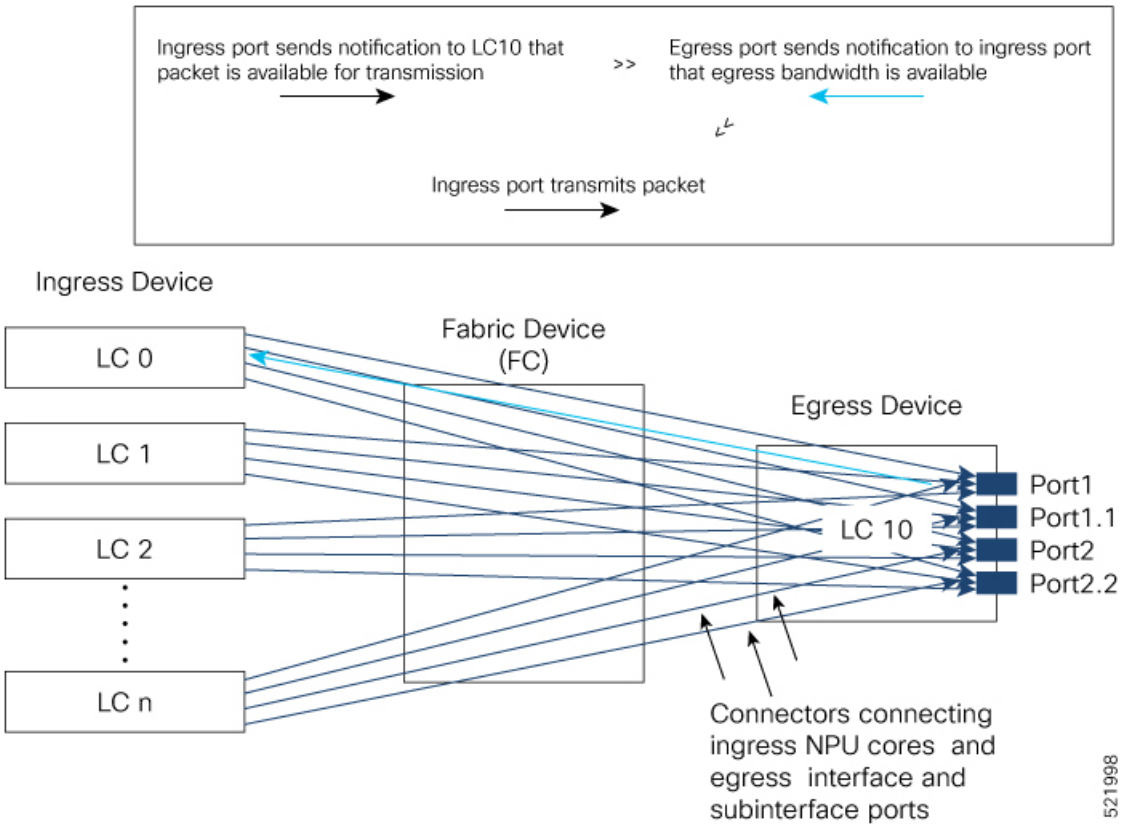
- Low Latency Queueing with Strict Priority Queueing, on page 3
- Traffic Shaping, on page 5
- Traffic Policing, on page 7

Ingress Traffic Management Model

The ingress traffic management model relies on packet queueing on the egress interface using Virtual Output Queueing (VOQ) on the ingress. In this model, buffering takes place at ingress. Here's how the VOQ process works.

Your routers support up to eight output queues per main interface or physical port. For every egress output queue, the VOQ model earmarks buffer space on every ingress pipeline. This buffer space is in the form of dedicated VOQs. These queues are called virtual because the queues physically exist on the ingress interface only when the line card actually has packets enqueued to it. To support the modular model of packet distribution, each network processing unit (NPU) core at the ingress needs connectors to every egress main interface and subinterface. The ingress traffic management model thus requires a mesh of connectors to connect the ingress NPU cores to the egress interfaces, as shown in **The Ingress Traffic Management Model**.

Figure 1: The Ingress Traffic Management Model



In the figure, every ingress interface (LC 0 through LC n) port has eight VOQs for the single egress line card LC 10.

Here's how packet transmission takes place:

1. When a packet arrives at ingress port (say on LC 0), the forwarding lookup on ingress line card points to the egress interface. Based on the egress interface (say it is on LC10), the packet is enqueued to the VOQ of LC 10. The egress interface is always mapped to a physical port.
2. Once egress bandwidth is available, the LC 10 ports ready to receive the packets (based on the packet marking and distribution model) send grants to the ingress ports via the connectors. (The figure shows a separate line for the grant for the sake of visual representation. In reality, the same connector is used for requests, grants, and transmission between an NPU core at the ingress and the egress port on LC 10.)
3. The ingress ports respond to this permission by transmitting the packets via FC to the LC 10 ports. (The time it takes for the ingress ports to request for egress port access, the egress port to grant access, and the packet to travel across FC is the round-trip time.)

The VOQ model thus operates on the principle of storing excess packets in buffers at ingress until bandwidth becomes available. Based on the congestion that builds up and the configured threshold values, packets begin to drop at the ingress itself, instead of having to travel all the way to the egress interface and then getting dropped.

Hardware Limitation:

In a scale scenario where 1000+ VoQs (created using egress QoS policies) store packets due to active traffic flows and may consume all the available on-chip buffer (OCB), unexpected traffic drops will be seen even though the traffic rate at the VoQ level is less than that of the VoQ shaper.

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. Strict PQ allows delay-sensitive data, such as voice, to be de-queued and sent before packets in other queues are de-queued.

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.
- 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**.
- You can configure an optional shaper to cap the maximum traffic rate. This is to ensure the lower priority traffic classes are not starved of bandwidth.

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)# policy-map test-priority-1
Router(config-pmap)# class qos1
Router(config-pmap-c)# priority level 7
Router(config-pmap-c# exit
Router(config-pmap)# exit
Router(config)# interface HundredGigE 0/6/0/18
Router(config-if)# service-policy output test-priority-1
Router(config-if)# no shutdown
```

```
Router(config-if)# commit
```

Running Configuration

```
policy-map strict-priority
class tc7
  priority level 1
  queue-limit 75 mbytes
!
 class tc6
  priority level 2
  queue-limit 75 mbytes
!
 class tc5
  priority level 3
  queue-limit 75 mbytes
 !
 class tc4
  priority level 4
  queue-limit 75 mbytes
!
 class tc3
  priority level 5
  queue-limit 75 mbytes
!
 class tc2
  priority level 6
  queue-limit 75 mbytes
 !
 class tc1
  priority level 7
  queue-limit 75 mbytes
 !
 class class-default
  queue-limit 75 mbytes
 !
 end-policy-map
!



class-map match-any tc1
match traffic-class 1
 end-class-map
!
class-map match-any tc2
match traffic-class 2
 end-class-map
!
class-map match-any tc3
match traffic-class 3
 end-class-map
!
class-map match-any tc4
match traffic-class 4
 end-class-map
!
class-map match-any tc5
match traffic-class 5
 end-class-map
!
class-map match-any tc6
match traffic-class 6
```

```
 end-class-map
!
class-map match-any tc7
match traffic-class 7
 end-class-map
!

interface HundredGigE0/6/0/18
 service-policy input 100g-s1-1
 service-policy output test-priority-1
!
```

Verification

```
Router#  show qos int 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 (HP7)                       =   qos-1
Egressq Queue ID                         =   11177 (HP7 queue)
TailDrop Threshold                       =   125304832 bytes / 10 ms (default)
WRED not configured for this class

Level1 Class (HP6)                       =   qos-2
Egressq Queue ID                         =   11178 (HP6 queue)
TailDrop Threshold                       =   125304832 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)
TailDrop Threshold                       =   1253376 bytes / 10 ms (default)
WRED not configured for this class
```

Related Topics

- Congestion Management Overview, on page 1
- Traffic Shaping, on page 5

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, thereby 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.
- It is recommended that you configure all the eight <traffic-class classes> (including **class-default**) for the egress policies. A limited number of < traffic-class class> combinations are supported, but unicast and multicast traffic-class may not be handled consistently. Hence, such configurations are recommended, when the port egress has only unicast traffic.

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)# 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

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

interface HundredGigE0/6/0/18
```

```
 service-policy input 100g-s1-1
 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
```

Related Topics

- Congestion Management Overview, on page 1

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.

The router supports the following traffic policing mode:

- Single-Rate Two-Color (SR2C) in color-blind mode. See Single-Rate Policer, on page 8.

Restrictions

- 1R3C policers are not supported.
- Up to eight policers are supported per ingress policy.
- Policers are allocated in multiples of 2, so any request for allocating odd number of policers is internally rounded up by a factor of one.
- Only one conditional marking action is supported. You can set discard class to 0/1, that is used to set either virtual output queueing (VOQ) limits or Random Early Detection (RED) profiles.
- If you configure discard-class explicitly at the class level and not under policer, then the explicit mark action is applied to all the transmitted policer traffic.
- Egress policing is not supported.

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 13 and Excess Bursts, on page 14.

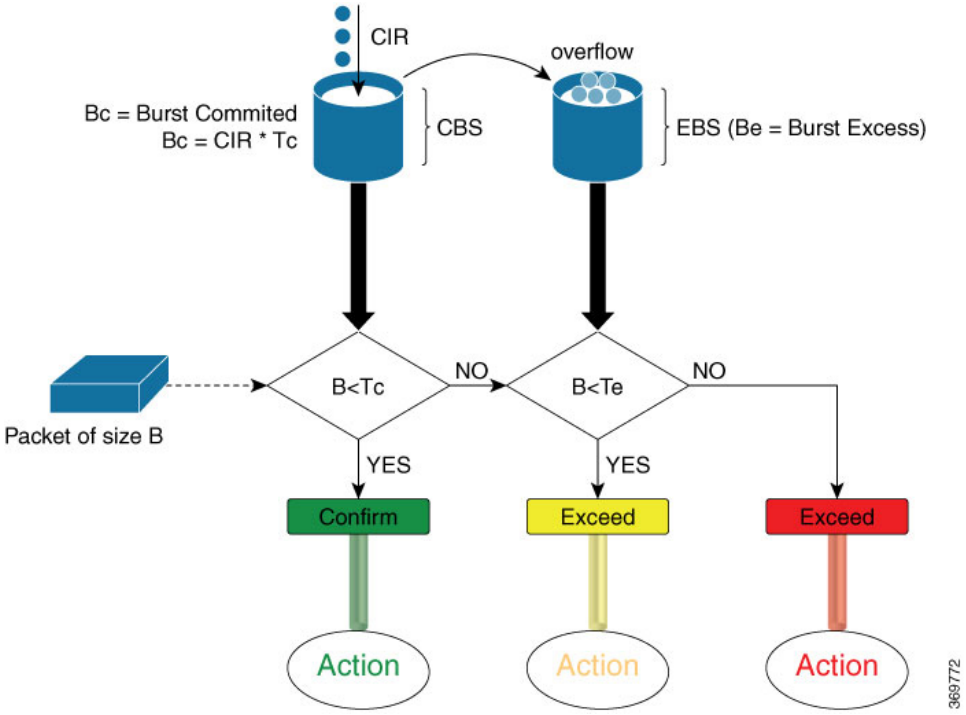
Single-Rate Policer

This section explains the concept of the single-rate two-color policer.

Single-Rate Two-Color Policer

A single-rate two-color (1R2C) policer provides one token bucket with two actions for each packet: a conform action and an exceed action.

Figure 2: Workflow of Single-Rate Two-Color Policey



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.

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-1R2C
Router(config-pmap)# class dscp1
```

```
Router(config-pmap-c)# police rate 10 gbps
Router(config-pmap-c-police)# exit
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface hundredGigE 0/0/0/18
Router(config-if)# service-policy input test-police-1R2C
Router(config-if)# commit
```

Running Configuration

```
class-map match-any dscp1
 match dscp ipv4 1
 end-class-map
!
!
policy-map test-police-1R2C
 class dscp1
  police rate 10 gbps
  !
 !
 class class-default
  !
 !
 end-policy-map
!
!
interface HundredGigE0/0/0/8
service-policy input test-police-1R2C
!
```

Verification

```
Router# show qos interface hundredGigE 0/0/0/8 input
NOTE:- Configured values are displayed within parentheses
Interface HundredGigE0/0/0/8 ifh 0xf0001e8  -- input policy
NPU Id:                        0
Total number of classes:       2
Interface Bandwidth:           100000000 kbps
Policy Name:                   test-police-1R2C
Accounting Type:               Layer1 (Include Layer 1 encapsulation and above)
------------------------------------------------------------------------------
Level1 Class                             =   dscp1
Policer committed rate                   =   10000000 kbps (10 gbits/sec)
Policer conform burst                    =   1024000 bytes (default)
Policer conform action                   =   Just TX
Policer exceed action                    =   DROP PKT

Level1 Class                             =   class-default
Policer not configured for this class
```

Related Topics

- Traffic Policing, on page 7

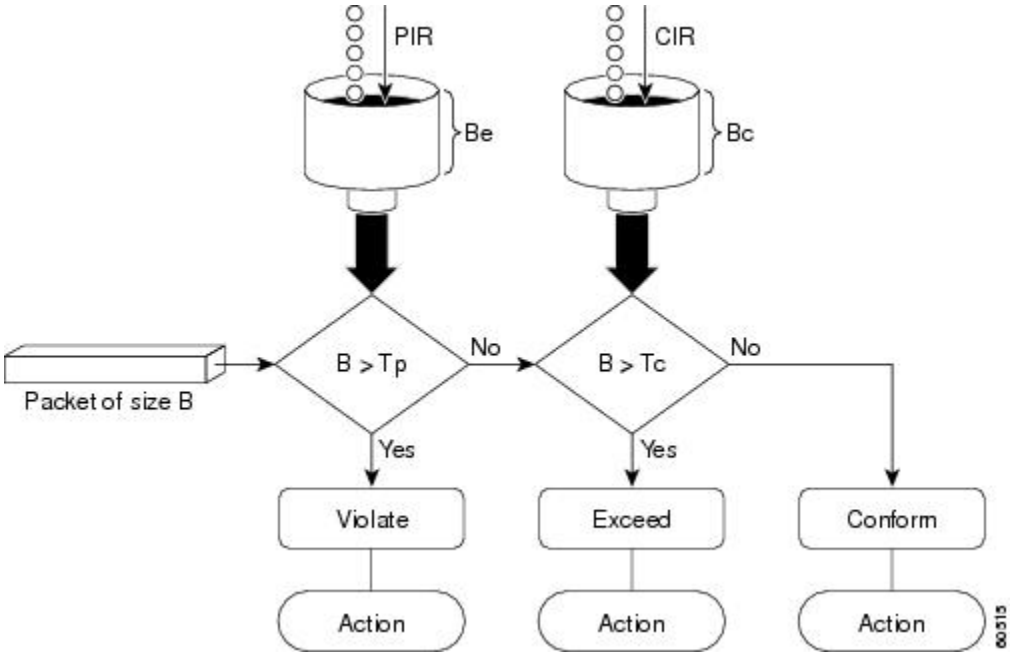
Two-Rate Policer

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 3: Marking Packets and Assigning Actions—Two-Rate Policer



Also, see Two-Rate Policer Details, on page 14.

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 test-police-2R3C
Router(config-pmap)# class dscp1
Router(config-pmap-c)# police rate 10 gbps peak-rate 20 gbps
Router(config-pmap-c-police)# exit
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# iinterface hundredGigE 0/0/0/8
Router(config-if)# service-policy input test-police-2R3C
Router(config-if)# commit
```

Running Configuration

```
class-map match-any dscp1
 match dscp ipv4 1
 end-class-map
!
!
policy-map test-police-2R3C
 class dscp1
  police rate 10 gbps peak-rate 20 gbps
  !
 !
 class class-default
  !
 !
 end-policy-map
!
!
interface HundredGigE0/0/0/8
service-policy input test-police-2R3C
!
```

Verification

```
Router# show qos interface hundredGigE 0/0/0/8 input

NOTE:- Configured values are displayed within parentheses
Interface HundredGigE0/0/0/8 ifh 0xf0001e8  -- input policy
NPU Id:                        0
Total number of classes:       2
Interface Bandwidth:           100000000 kbps
Policy Name:                   test-police-2R3C
Accounting Type:               Layer1 (Include Layer 1 encapsulation and above)
------------------------------------------------------------------------------
Level1 Class                             =   dscp1
Policer committed rate                   =   10000000 kbps (10 gbits/sec)
Policer peak rate                        =   20000000 kbps (20 gbits/sec)
Policer conform burst                    =   1024000 bytes (default)
Policer exceed burst                     =   2048000 bytes (default)
Policer conform action                   =   Just TX
Policer exceed action                    =   DROP PKT
Policer violate action                   =   DROP PKT
```

```
Level1 Class                              =   class-default
Policer not configured for this class


Router# policy-map interface hundredGigE 0/0/0/8 input

HundredGigE0/0/0/8 input: test-police-2R3C

Class dscp1
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :           289228439/289228439000          27734775
    Transmitted         :            56422213/56422213000           5410359
    Total Dropped       :           232806226/232806226000          22324416
  Policing statistics                (packets/bytes)     (rate - kbps)
    Policed(conform)    :            56422213/56422213000           5410359
    Policed(exceed)     :            56422215/56422215000           5410358
    Policed(violate)    :           176384011/176384011000          16914058
    Policed and dropped :           232806226/232806226000
Class class-default
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :            61136620/61136620000           0
    Transmitted         :            61136620/61136620000           0
    Total Dropped       :                   0/0                     0
Policy Bag Stats time: 1570155764000  [Local Time: 10/04/19 02:22:44.000]
```

Related Topics

- Two-Rate Policer, on page 11

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).

Important Points

- User configurable burst values— committed burst size (CBS) and excess burst size (EBS)—are not supported. Instead, they are derived using user configured rates—committed information rates (CIR) and peak information rates (PIR).
- The router supports two burst values: low (10 kilobytes) and high (1 megabyte). For a lower range of configured values (1.6 MBps to less than 1 GBps), the burst value is 10 KB. For a higher range of configured values (1 GBps to 400 GBps), the burst value is 1 MB.

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 Two-Rate Policer, on page 11.

References for Modular QoS Management

Read this section for more information on committed bursts, excess bursts, and two-rate policer.