

Configuring Modular QoS Congestion Management

This chapter covers the following topics:

- Congestion Management Overview, on page 1
- Best practices for minimizing packet drops during sustained maximum line-rate traffic, on page 1
- Class-based Weighted Fair Queueing, on page 2
- Low-Latency Queuing with Strict Priority Queuing, on page 4
- Overhead Accounting, on page 7
- Traffic Shaping, on page 10
- Burst Size for Port-Level Shaper, on page 13
- 4K Pseudowire on Bundle with QoS Enhancement , on page 19
- Traffic Policing, on page 21
- Shared Policer, on page 32
- References for Modular QoS Congestion Management, on page 36

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:

Best practices for minimizing packet drops during sustained maximum line-rate traffic

When operating Cisco NCS 540 routers at or near 100G line rate, packet drops may occur because of congestion in the egress traffic-management layer. Although it is not possible to eliminate packet drops at sustained wire-rate, you can apply these best practices to minimize loss and maintain optimal traffic flow:

- Shape egress traffic below full line rate: Configure traffic shaping slightly below the 100G line rate, for example, at around 98G, to maintain flows within internal thresholds and avoid Virtual Output Queue (VOQ) congestion. This approach aligns with Cisco's No Drop Rate (NDR) testing guidance.
- Tune egress QoS for burst handling: Increase queue limits moderately for Best-Effort (BE) or default traffic classes, within supported limits. Apply strict queue limits to priority traffic classes to protect their performance. Enable Weighted Random Early Detection (WRED) on BE traffic to smooth packet drops and prevent bursty tail-drop events.
- Avoid sustained oversubscription at 100% utilization: Design traffic profiles with a slight margin below full utilization, especially for mixed Maximum Transmission Unit (MTU) workloads or MPLS/L2VPN flows. While some oversubscription is supported, continuous 100% load will inevitably cause packet drops.
- Align certification and benchmarking expectations: Test and certify using the NDR methodology rather than assuming strict 100% line-rate operation. This provides a more realistic and achievable performance target.

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 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 . 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

Supported Platforms: 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.

Configuration Example

You have to accomplish the following to complete the 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 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 Queuing with Strict Priority Queuing

The Low-Latency Queuing (LLQ) feature brings strict priority queuing (PQ) to the CBWFQ scheduling mechanism. Priority queuing (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.

Configuring Low Latency Queuing with Strict Priority queuing

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

Support information

Supported priority levels

- Priority levels 1 to 7 are supported, with 1 being the highest and 7 the lowest.
- The default Class of Service Queue (CoSQ) 0 has the lowest priority among all levels.

Profile-based priority support

- For non-H-QoS profiles, priority levels 1 to 7 are supported.
- For H-QoS profiles, only priority levels 1 to 4 are supported.
- Regardless of the profile type, class-default always maps to CoSQ 0, which has the lowest priority.

Queue assignment for priority Level 1

- Any one of the eight egress class-maps (queues) can have priority level 1 configured.

Guidelines

Queue configuration commands

- You can configure these commands along with the **priority** command:
  - **shape average**
  - **queue-limit**
  - **random-detect**

Priority queue oversubscription

- A PQ can oversubscribe bandwidth when other queues do not fully utilize the port bandwidth.
- Oversubscription is supported only for a single priority level.

Limitations

Egress policing

- Egress policing is not supported. Hence, in strict priority queuing scenarios, lower-priority queues may not be serviced if the priority queue consumes the available bandwidth.

Multiple priority level oversubscription

- Bandwidth over-subscription with multiple priority levels is not a supported configuration.
- Starting with Cisco IOS XR Release 7.0.1, if multiple priority levels are configured for oversubscription:
  - the configuration is accepted, and a warning message is displayed

- fair queuing is applied across all configured priority levels and,
- higher-priority traffic is not guaranteed precedence over lower-priority traffic.

Traffic Disruption Risk

- There can be minimal traffic disruption when priority level 1 configuration is applied to any of the 8 queues.

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. 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

Overhead Accounting

Traffic shapers and policers use packet traffic descriptors to ensure adherence to the service level agreement in QoS. However, when traffic flows from one hop to another in a network, headers added or removed at interim hops affect the packet bytes being accounted for by QoS at each hop. When your end-user network

measures the packet bytes to ensure they receive the payload as agreed, these additional header bytes cause a discrepancy.

QoS overhead accounting provides the flexibility to operators to decide which header bytes can be excluded by the traffic shaper and policer and which can be included, depending on the end user's requirements and device capabilities, to meet the committed payload in units of bytes.

For example, if the QoS commitment includes the additional header bytes, the overhead accounting feature allows your router to account for this overhead and reduces the traffic policing and shaping rates accordingly. This is also called a **positive accounting overhead**.

If however, the committed rate doesn't include the additional bytes, overhead accounting allows your router to adjust the core stream traffic such that the traffic policing and shaping rates are increased. This is also called a **negative accounting overhead**.

To summarize, QoS overhead accounting enables the router to account for packet overhead when shaping and policing traffic to a specific rate. This accounting ensures that the router runs QoS features on the actual bandwidth that the subscriber traffic consumes.

Any interface that supports QoS policies supports overhead accounting.



Note You can enable user overhead accounting using the optional configuration of **accounting user-defined** <***overhead size in bytes***> while attaching the service policy on the egress interface.

Guidelines and Restrictions

- Overhead accounting for ingress shaping is not supported.
- You can't program more than one compensation value per NPU or router, even if they're on different egress ports.
- You can configure the same egress compensation for different egress ports.

The following restrictions apply for routers that have Cisco NC57 line cards installed and operate in native and compatibility modes.

- More than one compensation value can be programmed, provided you configure egress policy maps on different egress ports.
- You must configure a unique compensation value for a main interface and all sub-interfaces belonging to that main interface. You can't program different compensation values on different sub-interfaces sharing a common main interface.
- You can configure different compensation values on different sub-interfaces if they belong to other main interfaces.
- Compensation value programmed on egress queues (but not on VoQs) will remain active until the last egress policy map (with header compensation) is removed from main or sub-interfaces. This may impact traffic flow on main and sub-interfaces even though no compensation is set for them.

The following restrictions apply for routers that have line cards other than Cisco NC57 line cards.

- You can't program more than one compensation value per NPU or router, even if they're on different egress ports.

- You can configure the same egress compensation for different egress ports.
- NPUs can have different compensation values configured on different line cards in a modular system.
- Compensation value programmed on egress queues (but not on VoQs) will remain active until the last egress policy map (with header compensation) is removed from main or sub-interfaces. This may impact traffic flow on main and sub-interfaces even though no compensation is set for them.

Configuring for Overhead Accounting

To configure overhead accounting, you must:

1. Create a policy map and configure QoS actions for that map.
2. Configure overhead accounting and attach the map to an egress interface.

```
/* create QoS policy */
Router#configure terminal
Router(config)#policy-map policer
Router(config-pmap)#class class-default
Router(config-pmap-c)#police rate percent 10
Router(config-pmap-c-police)#commit

/* configure account overhead value while attaching the QoS policy to an egress interface
*/
Router(config)#int hundredGigE 0/0/0/2
Router(config-if)#service-policy output policer account user-defined 12
Router(config-if)#commit
Router(config-if)#root
Router(config)#end
```

Running Configuration

```
Router#sh run int hundredGigE 0/0/0/2
interface HundredGigE0/0/0/2
service-policy output policer account user-defined 12
!
```

The following example shows how to **configure a negative overhead accounting value**:

```
Router#conf
Router(config)#int hundredGigE 0/0/0/2
Router(config-if)#service-policy output policer account user-defined -12
Router(config-if)#commit
```

To **modify an overhead accounting value,** you must:

1. Remove the existing QoS policy and re-add it.
2. Configure the new overhead accounting value.

```
Router#conf
Router(config)#int hundredGigE 0/0/0/2
Router(config-if)#no service-policy input policer
Router(config-if)#service-policy output policer account user-defined -20
Router(config-if)#commit
Router#sh run int hundredGigE 0/0/0/2
interface HundredGigE0/0/0/2
service-policy output policer account user-defined -20
!
```

Positive Accounting Use Case

If QoS commitment includes Preamble, Frame Delimiter & Interframe Gap and has the following configuration:

```
service-policy output <foo> account user-defined +20
```

For QoS purposes, your router treats this packet as a packet of size = Actual Packet size + 20. Hence, the effective policing and shaping is *reduced* to match the downstream interface.

Negative Accounting Use Case

If QoS commitment to your router does not include VLAN header information, and has the following configuration:

```
service-policy output <foo> account user-defined -4
```

For QoS purposes, your router treats this packet as a packet of size = Actual Packet size – 4. Hence, the effective policing and shaping is *increased* to match the downstream interface.

Associated Commands

service-policy (overhead accounting)

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.

Configure VOQ-Level 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

- 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 and set peak burst size
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 40 percentpercent 50 1000
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
!


class-map c5
 match traffic-class 5
commit

policy-map egress_policy1
 class c5
  shape average percent 50 1000
 !
 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
```

```
Router# show qos interface tenGigE 0/0/0/0 output

Wed Jul 10 14:18:37.783 UTC
NOTE:- Configured values are displayed within parentheses
Interface TenGigE0/0/0/0 ifh 0x120  -- output policy
NPU Id:                        0
Total number of classes:       1
Interface Bandwidth:           10000000 kbps
Policy Name:                   test
VOQ Base:                      1024
Accounting Type:               Layer1 (Include Layer 1 encapsulation and above)
------------------------------------------------------------------------------
Level1 Class                              =   class-default
Egressq Queue ID                          =   1024 (Default LP queue)
Queue Max. BW.                            =   5031499 kbps (50 %)
Queue Min. BW.                            =   0 kbps (default)
Inverse Weight / Weight                   =   1 / (BWR not configured)
Guaranteed service rate                   =   5000000 kbps
Peak burst                                =   2240 bytes (1000 bytes)
TailDrop Threshold                        =   6258688 bytes / 10 ms (default)
```

Related Topics

- Congestion Management Overview, on page 1

Associated Commands

- shape average

Burst Size for Port-Level Shaper

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
Burst Size for Port-Level Shaper	Release 7.11.1	You can now achieve a predictable and accurate burst size at the link level by configuring port-level shaper burst size, thus ensuring better adherence to traffic SLAs. Also, with the port-level shaper burst size configured in the egress policy maps, the predictability in peak burst ensures that you can configure any next-hop low-capacity device to handle these bursts. Previously, you could configure burst sizes, which impacted traffic flow only at the Virtual Output Queue (VOQ) level but didn't control packet transmission at the link level.

In relation to a port-level shaper, the burst size refers to the maximum amount of data that can be sent through the port within a short period, exceeding the configured shaping rate. When traffic is shaped at the port level, the shaping algorithm smooths the traffic flow by limiting the average rate at which data is transmitted or received. However, there are scenarios where bursts of traffic may occur, such as during periods of high network activity or when multiple packets arrive simultaneously. In such scenarios, the burst size parameter represents the maximum burst of traffic that the port-level shaper allows before it starts to drop or delay packets.

This value influences the packet transmission and peak burst to the wire, irrespective of the shaper bursts configured on different VOQs. Thus, by providing tighter control of packet flow to the wire and next-hop devices, this functionality helps you achieve accurate and predictable peak bursts.

For example, if you configure the burst size to 100 KB, the shaper allows a burst of up to 100 KB of data to be transmitted within a short period, even if it exceeds the configured shaping rate. If the burst size exceeds 100 KB, the shaper takes action to enforce the shaping rate, such as dropping or delaying packets until the traffic falls within the specified limits. As a corollary, the expected port shaping rate may not be achieved if you configure a very low burst value.

The burst size is programmed on the egress port with:

- a default value of 32 KB for all routers

Burst Size for Port-Level Shaper: Why

Because...

Previously, you could configure burst size only for VOQ-level shaping, where the shaper bandwidth and burst sizes are programmed on the Fair Queue Elements (FQEs) in your router hardware. (FQE is a mechanism that allocates bandwidth fairly among different traffic flows or queues, and the FQEs are specific to ingress queues where egress policy maps are applied.)

and..

The VOQ-level shaper burst impacts credit flow for a particular VOQ. Configuring VOQ-level burst is a best-effort technique but may not translate to the intended peak burst at the link level.

Hence...

The burst size for the port level is port or egress queue-specific, and when you configure it, you have control over packet transmission at the link level, which means you achieve an accurate peak burst in the wire. This predictability in the burst ensures that low-end devices can absorb the burst according to their capability, and there's no unexpected drop in traffic.

Burst Size for Port-Level Shaper: Guidelines and Limitations

- You must configure the port burst along with the port shaper to activate this functionality.
- The burst size is configured on the port and egress queue level.
- The default burst values are:
  - 32 KB for all routers.
- **For Hierarchical QoS (H-QoS) and egress traffic management (ETM) models:**
  - Port shaper and burst on a particular physical interface are programmed when an egress policy map with only a **class-default** configuration and a configured shaper value is applied on that interface.
  - The shaper rate on the default class is calculated as the port shaper and the burst as the port burst.
  - For NCS 5700 line cards [Mode: Compatibility; Native] and NCS 5700 fixed port routers:
    - the port shaper is not supported if no port burst is configured. This limitation does not apply to other platforms.
    - Three-level H-QoS isn't supported. This means that you can't apply two-level egress H-QoS policies on the sub-interfaces and 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. See Overview of Hierarchical Modular QoS for details about H-QoS.
- **For non-H-QoS and non-ETM models:**
  - Port level burst on a particular interface is programmed when you configure it along with port shaper.
  - You must attach a two-level egress QoS policy map to the main interface. Here, the parent shaper is taken as the port shaper.

- The shaper rate on the parent policy configured for the default class is considered as port shaper and the burst as port burst.
- For NCS 5700 line cards [Mode: Compatibility; Native] and NCS 5700 fixed port routers, the minimum port shaper is 3 Gbps by default if no port burst is configured. For other platforms, the default minimum port shaper value is 1 Gbps. If the port burst is configured, there is no limitation on the minimum port shaper value.

- The port burst value configured on the egress interface doesn't restrict the burst sizes on the VOQs or child classes.
- The parent shaper burst size doesn't restrict the child shaper burst size. Unlike the child shaper bandwidth, a child shaper burst can be greater than the parent shaper burst.
- The actual burst size programmed on the ASIC could vary from the configured value due to a hardware approximation.

Configure Burst Size for Port-Level Shaper

Scenario 1: For Cisco NCS 5700 Series Routers, NCS 5700 line cards, and NCS 5500 Series Routers, in non-H-QoS and non-ETM mode: Assign Port Burst Value

In this scenario, you specifically assign a burst value (say, 2000 bytes) along with the shaper rate (say, 2 Gbps) on a parent policy configured to the default class. These parent shaper and burst values are taken as the port shaper and burst.

1. Create a two-level egress QoS policy map by configuring the parent policy with name, for example, **port-shaper-non-hqos**.
2. Configure the **port-shaper-non-hqos** policy for the default class. There should be no other class other than the default class in the policy map.
3. To the parent policy, apply a child policy named, for example, **egress-child**, using the **service-policy** command.
4. Configure a burst value of 2000 bytes and shaper rate of 2 Gbps for the parent (**port-shaper-non-hqos**) policy.
5. Using the **class** command, specify a traffic class (say **traffic-class-1**) for the child policy (**egress-child**).
6. Using the **shape average** command, configure the shape average for **traffic-class-1** to say, 25%.
7. Attach the parent policy map to the output interface to be used as the service policy for that interface.

This configuration applies the parent shaper and burst values that are taken as the port-level shaper and burst values.

```
Router(config)# policy-map port-shaper-non-hqos
Router(config-pmap)# class class-default
Router(config-pmap-c)# service-policy egress-child
Router(config-pmap-c)# shape average 2 gbps 2000 bytes
Router(config-pmap-c)# exit
Router(config)#policy-map egress-child
Router(config-pmap)# class traffic-class-1
Router(config-pmap-c)# shape average percent 25
Router(config-pmap-c)# exit
```

```
Router(config)# interface tenGigE 0/0/0/0
Router(config-if)# service-policy output port-shaper-non-hqos
Router(config-if)# commit
```

Running Configuration

```
policy-map port-shaper-non-hqos
  class class-default
   service-policy egress-child
   shape average 2 gbps 2000 bytes
 !

 !
policy-map egress-child
 class traffic-class-1
  shape average percent 25
 !
 !

interface tenGigE 0/0/0/0
 service-policy output port-shaper-non-hqos
!
```

Verification

Run the **show qos interface** command to confirm the Peak burst value (2000 bytes) you configured for the default class.

```
Router#show qos interface tenGigE 0/0/0/0 output
NOTE:- Configured values are displayed within parentheses
Interface TenGigE0/0/0/0 ifh 0x120  -- output policy
NPU Id:                        0
Total number of classes:       3
Interface Bandwidth:           10000000 kbps
Policy Name:                   port-shaper-non-hqos
SPI Id:                        0x0
VOQ Base:                      1024
PFC enabled:                   0
Accounting Type:               Layer1 (Include Layer 1 encapsulation and above)
------------------------------------------------------------------------------
Level1 Class                              =   class-default
Queue Max. BW.                            =   2020026 kbps (2 gbits/sec)
Queue Min. BW.                            =   2020026 kbps (default)
Inverse Weight / Weight                   =   1 / (BWR not configured)
Peak burst                              =   2000 bytes (2000 bytes)

   Level2 Class                           =   traffic-class-1
   Egressq Queue ID                       =   1025 (LP queue)
   Queue Max. BW.                         =   505417 kbps (25 %)
   Queue Min. BW.                         =   0 kbps (default)
   Inverse Weight / Weight                =   1 / (BWR not configured)
   Guaranteed service rate                =   500000 kbps
   Peak burst                             =   33600 bytes (default)
   TailDrop Threshold                     =   626688 bytes / 10 ms (default)
   WRED not configured for this class

   Level2 Class                           =   class-default
   Egressq Queue ID                       =   1024 (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                =   1000000 kbps
   Peak burst                             =   33600 bytes (default)
   TailDrop Threshold                     =   1253376 bytes / 10 ms (default)
   WRED not configured for this class
```

Scenario 2: For Cisco NCS 5700 Series Routers, NCS 5700 line cards, and NCS 5500 Series Routers, in non-H-QoS mode: Don't Assign Port Burst Value

In this scenario, you assign only the port shaper value (say, 2 Gbps) to the default class. In such cases, the burst size for port level shaper isn't activated, and the parent class has no port burst value.

1. Create a parent policy map named, for example, **port-shaper-non-hqos**.
2. Configure the **port-shaper-non-hqos** policy for the default class.
3. Apply a child policy-map named, for example, **egress-child** to the default class.
4. Shape the traffic in the default class to an average rate of 2 Gbps, limiting the egress traffic to a specific bandwidth.
5. Attach the parent policy map to the output interface to be used as the service policy for that interface.

```
Router(config)#policy-map port-shaper-non-hqos
Router(config-pmap)#class class-default
Router(config-pmap-c)#service-policy egress-child
Router(config-pmap-c)#shape average 2 gbps
Router(config-pmap-c)#exit
Router(config-pmap)#exit
Router(config)# interface tenGigE 0/0/0/0
Router(config-if)# service-policy output port-shaper-non-hqos
Router(config-if)#commit
```

Running Configuration

```
policy-map port-shaper-non-hqos
 class class-default
  service-policy egress-child
  shape average 2 gbps
 !
!
interface tenGigE 0/0/0/0
 service-policy output port-shaper-non-hqos
!
```

Verification

When you run the **show qos interface** command, you see no port burst value assigned to the parent class.

```
Router#show qos interface tenGigE 0/0/0/0 output
Sun Sep 10 20:17:16.053 UTC
NOTE:- Configured values are displayed within parentheses
Interface TenGigE0/0/0/0 ifh 0xa0  -- output policy
NPU Id:                        0
Total number of classes:       3
Interface Bandwidth:           10000000 kbps
Policy Name:                   port-shaper-non-hqos
SPI Id:                        0x0
VOQ Base:                      1024
PFC enabled:                   0
Accounting Type:               Layer1 (Include Layer 1 encapsulation and above)
------------------------------------------------------------------------------
Level1 Class                             =   class-default
Queue Max. BW.                           =   2100096 kbps (2 gbits/sec)
Queue Min. BW.                           =   2100096 kbps (default)
Inverse Weight / Weight                  =   1 / (BWR not configured)

   Level2 Class                             =   traffic-class-1
```

```
  Egressq Queue ID                          =   1025 (LP queue)
  Queue Max. BW.                            =   505861 kbps (25 %)
  Queue Min. BW.                            =   0 kbps (default)
  Inverse Weight / Weight                   =   1 / (BWR not configured)
  Guaranteed service rate                   =   500000 kbps
  Peak burst                                =   33600 bytes (default)
  TailDrop Threshold                        =   626688 bytes / 10 ms (default)
  WRED not configured for this class

  Level2 Class                              =   class-default
  Egressq Queue ID                          =   1024 (Default LP queue)
  Queue Max. BW.                            =   1011732 kbps (50 %)
  Queue Min. BW.                            =   0 kbps (default)
  Inverse Weight / Weight                   =   1 / (BWR not configured)
  Guaranteed service rate                   =   1000000 kbps
  Peak burst                                =   33600 bytes (default)
  TailDrop Threshold                        =   1253376 bytes / 10 ms (default)
  WRED not configured for this class
```

Scenario 3: For NCS 5700 Series Routers and NCS 5700 line cards, in H-QoS and ETM-enabled mode: Assign Port Burst Value

In this scenario, you specifically assign a burst value (say, 2000 bytes) along with the shaper rate (say, 2 Gbps) on the default class.) See Burst Size for Port-Level Shaper: Guidelines and Limitations, on page 14 for limitations that apply to this scenario.

1. Create a policy map named, for example, **port-shaper-hqos**.
2. Configure the **port-shaper-hqos** policy for the default class. The **class-default** class matches all traffic that does not match any other specific class. There should be no other class other than the default class in the policy map.
3. Configure the traffic shaping rate and the burst value for the **class-default** class.
4. Apply the policy-map named **port-shaper-hqos** as the output service policy for the interface.

```
Router (config)# policy-map port-shaper-hqos
Router (config-pmap)# class class-default
Router (config-pmap-c)# shape average 2 gbps 2000 bytes
Router (config-pmap-c)#exit
Router (config-pmap)#exit
Router(config)# interface tenGigE 0/0/0/0
Router(config-if)# service-policy output port-shaper-hqos
Router(config-if)#commit
```

Running Configuration

```
policy-map port-shaper-hqos
 class class-default
  shape average 2 gbps 2000 bytes
 !
!
interface tenGigE 0/0/0/0
 service-policy output port-shaper-hqos
!
```

Verification

Run the **show qos interface** to confirm the peak burst value (2000 bytes) you configured for the default class. Because there's no three-level H-QoS shaping support on NCS 5700 line cards [Mode: Compatibility; Native] and NCS 5700 fixed port routers, the output displays only the default class details (and not other child levels) for which you configured the shaper and burst values.

```
Router#show qos interface tenGigE 0/0/0/0 output
NOTE:- Configured values are displayed within parentheses
Interface TenGigE0/0/0/0 ifh 0xa0  -- output policy
NPU Id:                        0
Total number of classes:       1
Interface Bandwidth:           10000000 kbps
Policy Name:                   port-shaper-hqos
SPI Id:                        0x0
VOQ Base:                      1024
PFC enabled:                   0
Accounting Type:               Layer1 (Include Layer 1 encapsulation and above)
------------------------------------------------------------------------------
Level1 Class                             =   class-default
Egressq Queue ID                         =   1024 (Default LP queue)
Queue Max. BW.                           =   2024355 kbps (2 gbits/sec)
Queue Min. BW.                           =   0 kbps (default)
Inverse Weight / Weight                  =   60 / (BWR not configured)
Guaranteed service rate                  =   2000000 kbps
Peak burst                               =   2000 bytes (2000 bytes)
TailDrop Threshold                       =   2506752 bytes / 10 ms (default)
WRED not configured for this class
```

4K Pseudowire on Bundle with QoS Enhancement

Table 2: Feature History Table

Feature Name	Release Information	Feature Description
4K Pseudowire on Bundle with QoS Enhancement	Release 7.3.1	With this feature you can configure a desired traffic policy, to which your network complies, by using the bandwidth management technique of two-level traffic shaping. You can also increase the Link Aggregation Group (LAG) sub-interface scale or pseudowires up to 4K. This increased scale value enables you to increase the number of devices connected to your router, resulting in benefits such as increased bandwidth and cost-effective operations.

Your router supports two modes of egress traffic shaping:

- The **default non-Hierarchical QoS (H-QoS) mode**, where the egress traffic shaping is configured only on main interfaces and no hierarchical policies are supported. In other words, egress traffic shaping on subinterfaces isn't supported in this mode.
- The **H-QoS mode**, where egress traffic shaping is also supported on subinterfaces. This mode also supports hierarchical policies on the main and subinterfaces.

Prior to Cisco IOS XR Release 7.3.1, the H-QoS mode restricted the scale of subinterfaces you could configure. For example, the maximum Link Aggregation Group (LAG) subinterface scale or pseudowires is 1K in the H-QoS mode.

The enhancement described in this section, available from Cisco IOS XR Release 7.3.1 is applicable for the default non-H-QoS mode for egress QoS. It involves configuring two-level traffic shaper policy on the main interface, while enabling you to increase the LAG subinterface scale or pseudowires to as much as 4K. What this means is that you can also scale up the number of access devices you want to connect through LAG to your router, thus enabling:

- Increased reliability and availability
- Better use of physical resources
- Increased bandwidth
- Cost-effective operations

Restrictions and Guidelines

The following restrictions and guidelines apply while configuring two-level traffic shaper policy on the main interface in (default) non-H-QoS mode:

- The hierarchical egress policy support is only for main interfaces.
- Subinterface behavior remains the same in non-H-QoS mode. No egress QoS support is available for subinterfaces.
- This enhancement is applicable only for egress QoS, and there are no changes in ingress QoS behavior.
- There’s no change to the current non-H-QoS flat policy behavior.
- The minimum shaper rate varies between different ASICs.

Configure Two-Level Shaper Policy on Main Interface

To configure two-level shaper policy on main interface, you must:

1. Enter global configuration mode and create a two-level policy map.
2. Attach this policy map to the main interface.

```
/* Enter the global configuration mode and create the two-level policy map */
Router#configure
Router(config)# policy-map  two-level-pm
Router(config-pmap)#class class-default
Router(config-pmap-c)#shape average percent 20
Router(config-pmap-c)#service-policy child
Router(config-pmap-c)#root
Router(config)#policy-map child
Router(config-pmap)#class class-default
Router (config-pmap-c)#shape average percent 5
Router(config-pmap-c)#commit

/* Apply policy-map under interface */
Router(config)#interface hundredGigE 0/0/0/3
```

```
Router(config-intf)#service-policy output policy
Router(config-intf)#commit
```

Running Configuration

```
Router#show running-config policy-map two-level-pm
policy-map two-level-pm
 class class-default
  service-policy child
  shape average percent 20
 !
 end-policy-map
!

Router#show running-config interface hundredGigE 0/0/0/3
interface HundredGigE0/0/0/3
service-policy output two-level-pm
!
```

Verification

Verify that the maximum bandwidth (shaping rate) for the parent policy (Level 1) is greater than the rate for the child policy at Level 2.

```
Router#show qos interface hundredGigE 0/0/0/3 output
NOTE:- Configured values are displayed within parentheses
Interface HundredGigE0/0/0/3 ifh 0x68  -- output policy
NPU Id:                        0
Total number of classes:       2
Interface Bandwidth:           100000000 kbps
Policy Name:                  two-level-pm
SPI Id:                        0x0
VOQ Base:                      1048
Accounting Type:               Layer1 (Include Layer 1 encapsulation and above)
------------------------------------------------------------------------------
Level1 Class                             =   class-default
Queue Max. BW.                           =   20004211 kbps (20 %)
Queue Min. BW.                           =   20000305 kbps (default)
Inverse Weight / Weight                  =   1 / (BWR not configured)

   Level2 Class                             =   class-default
   Egressq Queue ID                         =   1048 (Default LP queue)
   Queue Max. BW.                           =   1000000 kbps (5 %)
   Queue Min. BW.                           =   0 kbps (default)
   Inverse Weight / Weight                  =   1 / (BWR not configured)
   Guaranteed service rate                  =   1000000 kbps
   Peak burst                               =   36864 bytes (default)
   TailDrop Threshold                       =   1249792 bytes / 10 ms (default)
   WRED not configured for this class
```

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 , on page 22.

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

- Single-Rate Two-Color (SR2C) in color-blind mode. See Single-Rate Policer, on page 23.
- Single-Rate Three-Color (SR3C) in color-blind mode.
- Two-Rate Three-Color (2R3C) in color-blind mode. See Two-Rate Policer, on page 26 .



Note In Cisco NCS 560, QoS enhanced stats is enabled by default.

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.
- Ensure that you don't configure a policer and match criteria for **discard-class** in the same class. Even though the configuration is allowed, the policer doesn't work and allows all traffic without dropping packets.
- 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.

Committed Bursts

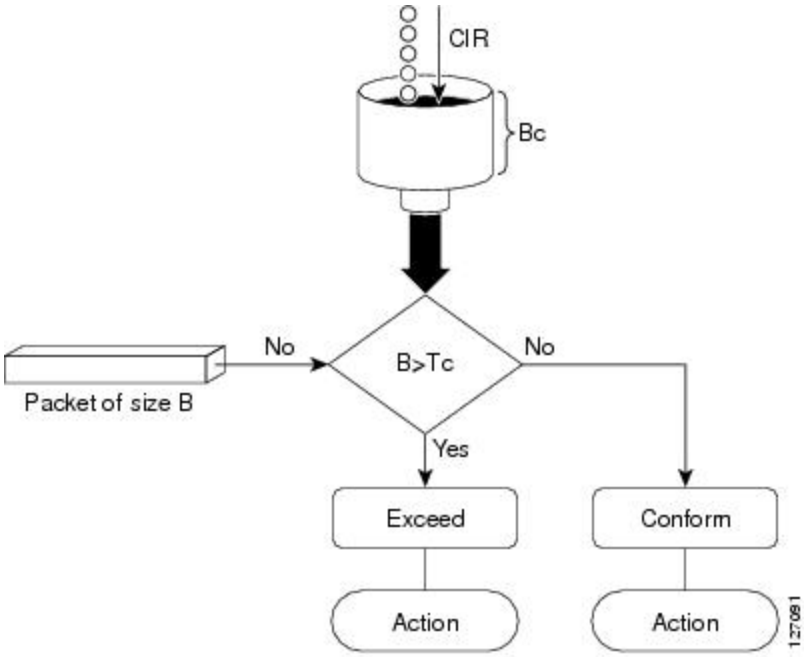
Unlike a traffic shaper, a traffic policer does not buffer excess packets and transmit them later. Instead, the policer executes a "send or do not send" policy without buffering. Policing uses normal or committed burst (bc) values to ensure that the router reaches the configured committed information 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 burst parameter enables the policer to drop packets less aggressively.

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 action is performed. If a packet of size B is less than the Tc token bucket, then the packet conforms and a different 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-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 21

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 21

Associated Commands

- police rate

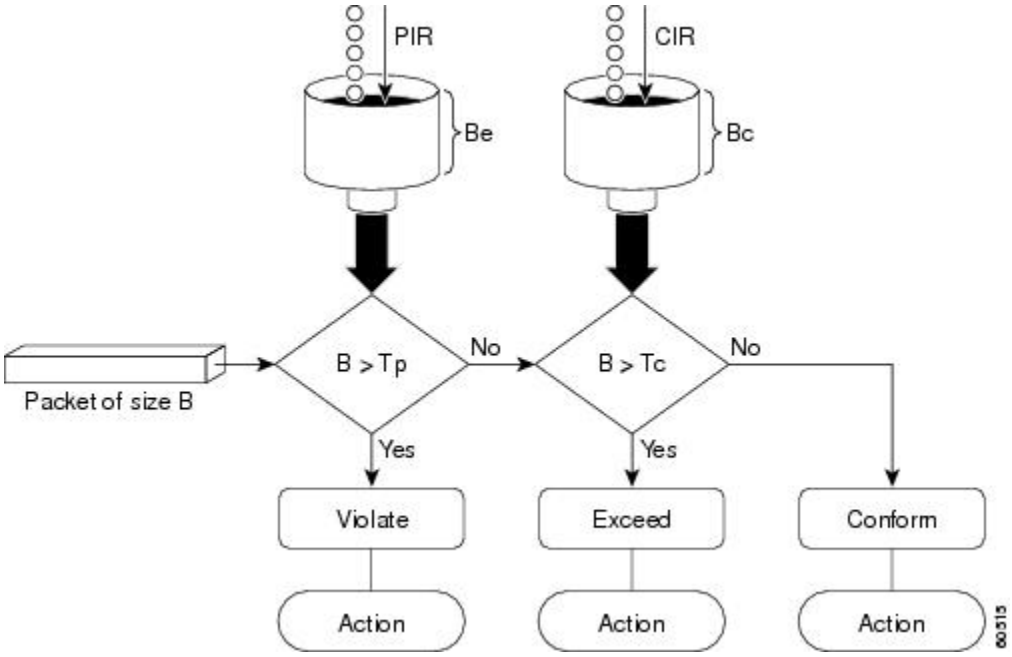
Two-Rate Policer

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.

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 37.

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

- Two-Rate Policer, on page 26

Associated Commands

- police rate

Packets-Per-Second-Based Policer

Table 3: Feature History Table

Feature Name	Release Information	Feature Description
Packets-Per-Second-Based Policer	Release 7.4.1	Prior to this functionality, when configuring policers, the only available option for policer rates was bit-rate measured in units of bits per second (bps). With this release, you can configure policer rates in units of packets per second (pps) as well. pps-based policer is critical in fending off malicious attacks—when attackers target your specific resources with a vast amount of traffic that contain higher number of packets, but move at a slower packet rate. Protection from such attacks is possible because pps-based policers ensure that regardless of the packet size and rate, the policer only accepts a fixed number of packets per second. This functionality modifies the **police rate** command.

- **Policer rates so far**—You used the **police rate** command to configure policers based on two parameters:
  - bit-rates (default unit: bits per second or bps)
  - Burst size (default unit: bytes)
- **packets-per-second (pps)-based policer**—With this additional functionality, you can use the **police rate** command to configure policers in units of packets per second (pps). The pps configuration option is available as **police rate <*pps-value*> pps**. When you configure the pps option, ensure that you configure burst size in packets. (See **Restrictions and guidelines**.) Thus, the parameters for pps-based policer are:

- packets per second (pps)
- burst size (packets)

- **Why pps-based-policer**—Networks face newer types of attacks, and these days malicious operators don't necessarily employ aggressive tactics that involve overwhelming your bandwidth with large amount of traffic to cause distributed denial of service (DDoS). Now, some attackers go the 'softer' route, where they send smaller packet sizes at slower traffic rates. During such malicious network activity, a bandwidth-based policer can still aggregate up to many packets to be processed if the packet size is small. Attackers tend to use this behavior to bypass bandwidth-based policers to exploit vulnerabilities or try to hit performance limitations by increasing the packet rates.

  Packets-per-second-based policers ensure that regardless of the packet size and traffic rate, the policer only accepts a fixed number of packets per second.

  pps-based-policer support cheat-sheet—Here's a quick look at some key support areas and their details for pps-based policer.

Support	Details
Classification and marking support	Same as that for bps-based-policer
Units	Equivalent kbps values display for QoS programming and statistics.
H-QoS	Support for parent and child policers
Bursts	Support for confirm burst (bc) and exceed burst (be) values in units of packets. The default value is in multiple of 128 bytes equivalent to 10 milliseconds.
Minimum pps value	For better granularity, recommended minimum value is 100 pps.

- **Restriction and guidelines**
  - This functionality is **applicable only for ingress.**
  - When using a pps-based policer, **ensure that you configure the burst-size value in number of packets as well.** This is because a policer burst rate determines whether a specific number of packets out of contract would be subject to the next action (that is, exceed or violate).
  - **Within a QoS policy, configure the parent and child policies policers to either bps or pps.** Else, the configuration displays an error when you try attaching the policy to an interface.
  - **For single-level policy maps:** under the same policy map, you can configure one class map with bps-based policer and the other class map with a pps-based policer.
  - **For two-level hierarchical policy maps:**
    - The parent and child-level policy maps must use the same unit-based policer. That is, both must have either pps-based or bps-based policers.
    - If you configure the child-level policy map with pps-based policer, ensure that the parent policy-map class default has a pps-based policer.

- **Configure pps-based policer**—To configure pps-based policer, you must:
  1. Configure a class map.
  2. Create a service policy for the map and configure the pps values.
  3. Attach the service policy to an interface.

```
/*Configure a class map*/

Router(config)#class-map prec1
Router(config-cmap)#match precedence 1
Router(config-cmap)# exit
Router(config)# commit

/*Create a service policy map*/

Router(config)# policy-map policy1
Router(config-pmap)# class prec1
Router(config-pmap-c)#police rate 1000 pps burst 300 packets
Router(config-pmap-c-police)#exit
Router(config-pmap-c)#exit
Router(config-pmap)#exit
Router(config)# commit

/*Attach the service policy to an interface*/
Router#int hundredGigE 0/7/0/2
Router(config-if)service-policy input policy1
Router(config-if)#exit
Router(config)#commit
```

Running Configuration

```
class-map match-any prec1
match precedence 1
 end-class-map
!
policy-map policy1
class prec1
  police rate 1000 pps burst 300 packets
  !
 !
 class class-default
!
 end-policy-map
!
```

Verification

```
Router#show qos int hundredGigE 0/7/0/2 input
NOTE:- Configured values are displayed within parentheses
Interface HundredGigE0/7/0/2 ifh 0xe000088  -- input policy
NPU Id:                        0
Total number of classes:       2
Interface Bandwidth:           100000000 kbps
Policy Name:                   policy1
SPI Id:                        0x0
Accounting Type:               Layer2 (Include Layer 2 encapsulation and above)
------------------------------------------------------------------------------
Level1 Class                              =   prec1

Policer Bucket ID                         =   0x9
```

```
Policer Stats Handle                     =   0x0
Policer committed rate                   =   998 kbps (1000 packets/sec)
Policer conform burst                    =   37632 bytes (300 packets)

Level1 Class                             =   class-default

Default Policer Bucket ID                =   0x8
Default Policer Stats Handle             =   0x0
Policer not configured for this class
```

Associated Commands

police rate

Shared Policer

The classification of the incoming packet occurs only once. However, based on the different classification criteria, the shared policer feature allows sharing of the policer bucket amongst two or more classes in a QoS policy map . That is, the same token bucket is used for a traffic flow matching against any of the classes sharing the policer.

For example, let us say a policer of 10 Mbps is shared among two classes C1 and C2. This feature ensures that both C1 and C2 get traffic flow assigned on First Come First Serve (FCFS) basis. Also that, if C2 does not have any traffic, C1 uses all of the 10 Mbps for transmission.

This feature includes two components:

- Policer Bucket Shared
- Policer Bucket Referred

Policer Bucket Shared

The policer bucket shared feature defines and shares a policer bucket instance among multiple classes.

Here is a sample configuration that defines and shares policer bucket instance sp1 :

```
policy-map parent
        class long-distance
          police bucket shared sp1 rate 1 mbps
```

In this configuration, a policy-map for class long-distance traffic type is created to police at 1Mbps rate and the policer bucket is shared.

Policer Bucket Referred

The policer bucket referred feature refers a defined policer bucket instance. Shared policer is not supported across policy levels. This means for example, that parent and child policy cannot share a common bucket.

Here is a sample configuration that refers shared policer bucket instance sp1 :

```
policy-map voip-child
        class long-distance-voip
         police bucket referred sp1
```

In this configuration, a policy-map for class long-distance-voip traffic type is created and the shared policer bucket sp1 is referred.

Shared Policer Statistics

Currently, individual class statistics are not available as a default option for shared policer. You can access statistics in the following modes.

Aggregate Mode

In this mode the policer bucket is shared among two or more classes. However, statistics are not available for every individual class. You can view the aggregate statistics that combine the numbers for all the classes sharing the policer bucket.

Per-Class Mode

In this mode the policer bucket is shared among two or more classes, and you can also view individual class statistics. However, when this mode is active, the Policy-Based Tunnel Selection (PBTS) mechanism is disabled. To enable the per-class mode, you must configure the **hw-module profile qos shared-policer-per-class-stats** command.

Restrictions and Guidelines

The following restrictions and guidelines apply while configuring the shared policer feature.

- When shared policer is enabled in per-class mode, Policy-Based Tunnel Selection (PBTS) mechanism is disabled. In other words, shared policer-per-class-mode and PBTS are mutually exclusive features.
- Shared policer is not supported across policy levels. This means, for example, that parent and child policies cannot share a common policer bucket.
- Shared policer is not supported in ingress peering mode.
- Shared policer is supported within classes of the same policy. However, cross-policy bucket sharing is not supported.
- There are no limitations on the number of classes that can share policer.
- There are no changes in policer scale numbers in the aggregate and per-class modes.
- All the existing policer types (1R2C, 1R3C and 2R3C) are supported.
- You must reload the affected line card to enable the per-class-stats mode.

Configuring Shared Policer

To configure shared policer, you must:

1. Create a class map to be used for matching packets to the specified class.
2. Create a policy map to be used for matching packets to the specified class.
3. Specify a class name.

4. Define and share a policer bucket.
5. Specify a class name.
6. Refer a shared policer bucket.

```
RP/0/RSP0/CPU0:ios(config)#class-map match-any c1
RP/0/RSP0/CPU0:ios(config-cmap)#match precedence 1
RP/0/RSP0/CPU0:ios(config-cmap)#end-class-map
RP/0/RSP0/CPU0:ios(config)#class-map match-any c2
RP/0/RSP0/CPU0:ios(config-cmap)#match precedence 2
RP/0/RSP0/CPU0:ios(config-cmap)#end-class-map
RP/0/RSP0/CPU0:ios(config)#policy-map s-pol
RP/0/RSP0/CPU0:ios(config-pmap)#class c1
RP/0/RSP0/CPU0:ios(config-pmap-c)#police bucket shared 1 rate 10 mbps
RP/0/RSP0/CPU0:ios(config-pmap-c-police)#exit
RP/0/RSP0/CPU0:ios(config-pmap-c)#exit
RP/0/RSP0/CPU0:ios(config-pmap)#class c2
RP/0/RSP0/CPU0:ios(config-pmap-c)#police bucket referred 1
RP/0/RSP0/CPU0:ios(config-pmap-c-police)#class class-default
RP/0/RSP0/CPU0:ios(config-pmap-c)#exit
RP/0/RSP0/CPU0:ios(config-pmap)#exit
RP/0/RSP0/CPU0:ios(config)#interface HundredGigE 0/6/0/18
RP/0/RSP0/CPU0:ios(config-if)#service-policy input s-pol
RP/0/RSP0/CPU0:ios(config-if)#commit
```

Running Configuration

```
class-map match-any c1
 match precedence 1
 end-class-map

class-map match-any c2
 match precedence 2
 end-class-map

policy-map s-pol
 class c1
  police bucket shared 1 rate 10 mbps
  !
 !
 class c2
  police bucket referred 1
  !
 !
 class class-default
 !
 end-policy-map
!

interface HundredGigE0/6/0/18
 service-policy input s-pol
!
```

Verification

In Aggregate Mode

```
RP/0/RP0/CPU0:ios#sh policy-map interface tenGigE 0/0/0/0 input
Fri Nov 15 12:55:56.817 UTC

TenGigE0/0/0/0 input: s-pol
```

```
Class c1
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :          1784530245/228419871360          8640780
    Transmitted         :             2067504/264640512             10011
    Total Dropped       :          1782462741/228155230848          8630769
  Policing statistics                (packets/bytes)     (rate - kbps)
    Policed(conform)    :             2067504/264640512             10011
    Policed(exceed)     :          1782462741/228155230848          8630769
    Policed(violate)    :                  0/0                      0
    Policed and dropped :          1782462741/228155230848
Class c2
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :                  0/0                      0
    Transmitted         :                  0/0                      0
    Total Dropped       :                  0/0                      0
  Policing statistics                (packets/bytes)     (rate - kbps)

Policed(conform)    :                   0/0                     0
    Policed(exceed)     :                  0/0                      0
    Policed(violate)    :                  0/0                      0
    Policed and dropped :                  0/0
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: 1573822531986  [Local Time: 11/15/19 12:55:31.986]
```

In Per-Class Mode

```
RP/0/RP0/CPU0:ios#sh policy-map interface tenGigE 0/0/0/0 input
Fri Nov 15 15:18:18.319 UTC

TenGigE0/0/0/0 input: s-pol

Class c1
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :          1005369276/128687267328          4320337
    Transmitted         :             1163300/148902400             5013
    Total Dropped       :          1004205976/128538364928          4315324
  Policing statistics                (packets/bytes)     (rate - kbps)
    Policed(conform)    :             1163300/148902400             5013
    Policed(exceed)     :          1004205976/128538364928          4315324
    Policed(violate)    :                  0/0                      0
    Policed and dropped :          1004205976/128538364928
Class c2
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :          1005341342/128683691776          4320335
    Transmitted         :             1166269/149282432             4997
    Total Dropped       :          1004175073/128534409344          4315338
  Policing statistics                (packets/bytes)     (rate - kbps)
    Policed(conform)    :             1166269/149282432             4997
    Policed(exceed)     :          1004175073/128534409344          4315338
    Policed(violate)    :                  0/0                      0
    Policed and dropped :          1004175073/128534409344
Class class-default
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :             49159/6292352                 0
    Transmitted         :             49159/6292352                 0
    Total Dropped       :                  0/0                      0
Policy Bag Stats time: 1573831087338  [Local Time: 11/15/19 15:18:07.338]
```

Related Commands hw-module profile qos shared-policer-per-class-stats

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 Two-Rate Policer, on page 26.