



# Configuring Modular QoS Congestion Management

---

This chapter covers the following topics:

- [Congestion Management Overview, on page 1](#)
- [Modified Deficit Round Robin Queueing, on page 1](#)
- [Low-Latency Queueing with Strict Priority Queueing, on page 6](#)
- [Traffic Shaping, on page 9](#)
- [Traffic Policing, on page 12](#)

## 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:

- [Modified Deficit Round Robin Queueing, on page 1](#)
- [Low-Latency Queueing with Strict Priority Queueing, on page 6](#)
- [Traffic Shaping, on page 9](#)
- [Traffic Policing, on page 12](#)

Among the ones listed above, traffic policing is the one used for congestion management on the ingress side. Others are used for congestion management on the egress side.

## Modified Deficit Round Robin Queueing

Modified Deficit Round Robin (MDRR) is a class-based composite scheduling mechanism that allows for queueing of up to eight traffic classes. It operates in the same manner as class-based weighted fair queueing (CBWFQ) and allows definition of traffic classes based on customer match criteria. When MDRR is configured in the queueing strategy, non-empty queues are served one after the other. Each time a queue is served, a fixed amount of data is dequeued. The algorithm then services the next queue.

## Bandwidth Remaining

The MDRR 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 MDRR. 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.

## Configure Bandwidth Remaining

### Guidelines

- It is mandatory to configure all the eight qos-group classes (including class-default) for the egress policies in Cisco NCS 5000 Series Routers.

### 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)# policy-map egress_policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# bandwidth remaining percent 2
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface TenGigE 0/0/0/0
Router(config-if)# service-policy output egress_policy1
Router(config-if)# commit

/* Bandwidth remaining can also be configured as ratio, instead of percentage */
Router(config-pmap-c)# bandwidth remaining ratio 50
```

### Running Configuration

Here bandwidth remaining is configured as percentage:

```
/* Class-map configuration */

class-map match-any class1
  match qos-group 1
end-class-map
!
```

```

class-map match-any class2
  match qos-group 2
end-class-map
!
class-map match-any class3
  match qos-group 3
end-class-map
!
class-map match-any class4
  match qos-group 4
end-class-map
!
class-map match-any class5
  match qos-group 5
end-class-map
!
class-map match-any class6
  match qos-group 6
end-class-map
!
class-map match-any class7
  match qos-group 7
end-class-map

/* Policy-map configuration */

policy-map egress_policy1
  class class1
    bandwidth remaining percent 2
  !
  class class2
    bandwidth remaining percent 3
  !
  class class3
    priority level 1
  !
  class class4
    bandwidth remaining percent 50
    shape average 100 mbps
  !
  class class5
    bandwidth remaining percent 2
  !
  class class6
    bandwidth remaining percent 25
  !
  class class7
    bandwidth remaining percent 6
  !
  class class-default
    bandwidth remaining percent 12
  !
end-policy-map

interface TenGigE 0/0/0/0
  service-policy output egress_policy1
!
```

Here bandwidth remaining is configured as ratio:

```

/* Class-map configuration */

class-map match-any class1
```

```

    match qos-group 1
  end-class-map
!
class-map match-any class2
  match qos-group 2
end-class-map
!
class-map match-any class3
  match qos-group 3
end-class-map
!
class-map match-any class4
  match qos-group 4
end-class-map
!
class-map match-any class5
  match qos-group 5
end-class-map
!
class-map match-any class6
  match qos-group 6
end-class-map
!
class-map match-any class7
  match qos-group 7
end-class-map

/* Policy-map configuration */

policy-map egress_policy2
  class class1
    bandwidth remaining ratio 50
  !
  class class2
    bandwidth remaining ratio 25
  !
  class class3
    bandwidth remaining ratio 12
  !
  class class4
    bandwidth remaining ratio 6
  !
  class class5
    bandwidth remaining ratio 3
  !
  class class6
    bandwidth remaining ratio 2
  !
  class class7
    priority level 1
  !
  class class-default
    bandwidth remaining ratio 2
  !
end-policy-map

```

### Verification

Each class has a classification statistics (that include total transmitted and dropped counts of that class) and a queueing statistics (that include the tail drop counts, the conformed queue statistics and the instantaneous queue length).

Verify if the queue is growing, by checking the values of *Inst-queue-len (cells)*. Also, check the values of *Total Dropped* field to see if there are any queue drops.

```
Router# show policy-map interface tenGigE 0/0/0/0
TenGigE0/0/0/0 direction input: Service Policy not installed

TenGigE0/0/0/0 output: egress_policy1

Class class1
  Classification statistics          (packets/bytes)      (rate - kbps)
  Matched                          : 17312125/12799683759    0
  Transmitted                       : 10774292/7953712189     0
  Total Dropped                  : 6537833/4845971570     0
  Queueing statistics
  Queue ID                          : 9
  High watermark                    : N/A
  Inst-queue-len (cells)         : 22981
  Avg-queue-len                     : N/A
  Taildropped (packets/bytes)       : 6537833/4845971570
  Queue (conform)                   : 10774292/7953712189     0
  RED random drops (packets/bytes)  : 0/0

- - -
- - -

Class class-default
  Classification statistics          (packets/bytes)      (rate - kbps)
  Matched                          : 941172/853166230        0
  Transmitted                       : 926830/840112420        0
  Total Dropped                     : 14342/13053810         0
  Queueing statistics
  Queue ID                          : 8
  High watermark                    : N/A
  Inst-queue-len (cells)            : 0
  Avg-queue-len                     : N/A
  Taildropped (packets/bytes)       : 14342/13053810
  Queue (conform)                   : 926830/840112420        0
  RED random drops (packets/bytes)  : 0/0
```

This is the show command output, if bandwidth remaining is configured as ratio:

```
Router# show policy-map interface tenGigE 0/0/0/0
TenGigE0/0/0/0 direction input: Service Policy not installed

TenGigE0/0/0/0 output: egress_policy2

Class class1
  Classification statistics          (packets/bytes)      (rate - kbps)
  Matched                          : 4867412/3598770345      0
  Transmitted                       : 2901271/2142905575      0
  Total Dropped                     : 1966141/1455864770      0
  Queueing statistics
  Queue ID                          : 9
  High watermark                    : N/A
  Inst-queue-len (cells)            : 10709
  Avg-queue-len                     : N/A
  Taildropped (packets/bytes)       : 1966141/1455864770
  Queue (conform)                   : 2901271/2142905575      0
  RED random drops (packets/bytes)  : 0/0

- - -
- - -
```

```

Class class-default
  Classification statistics          (packets/bytes)  (rate - kbps)
    Matched                        : 262482/237940215  0
    Transmitted                    : 97172/88070686   0
    Total Dropped                  : 165310/149869529  0
  Queueing statistics
    Queue ID                       : 8
    High watermark                  : N/A
    Inst-queue-len (cells)         : 10714
    Avg-queue-len                  : N/A
    Taildropped (packets/bytes)    : 165310/149869529
    Queue (conform)                : 97172/88070686   0
    RED random drops (packets/bytes) : 0/0

```

### Related Topics

- [Bandwidth Remaining, on page 2](#)

### Associated Commands

- [bandwidth remaining](#)

## Low-Latency Queuing with Strict Priority Queuing

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 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 is supported.
- Egress policing is not supported. Hence, in the case of strict priority queuing, there are chances that the other queues do not get serviced. Therefore, in order to minimize this, the user can police the traffic at the ingress side itself or design the network in a such a way that the priority traffic doesn't impact the other traffic on the egress port.
- You can configure **shape average** and **queue-limit** commands along with **priority**.
- There can be a minimal traffic disruption when priority level 1 configuration is applied on any of the 8 queues.
- Any one of the eight egress class-maps (queues) can have priority level 1 configuration.

### 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 egress_policy1
Router(config-pmap)# class qos1
Router(config-pmap-c)# priority level 1

Router(config-pmap-c)# exit
Router(config-pmap)# exit

Router(config)# interface TenGigE 0/0/0/0
Router(config-if)# service-policy output egress_policy1
Router(config-if)# commit
```

### Running Configuration

```
/* Class-map configuration */

class-map match-any qos-1
  match qos-group 1
end-class-map
!
- - -
- - -
- - -
!
class-map match-any qos-7
  match qos-group 7
end-class-map

/* Policy-map configuration */

policy-map egress_policy2
  class qos-1
    priority level 1
  !
  class qos-2
    bandwidth remaining ratio 1
  !
  class qos-3
    bandwidth remaining ratio 1
  !
```

```

class qos-4
  bandwidth remaining ratio 1
!
class qos-5
  bandwidth remaining ratio 1
!
class qos-6
  bandwidth remaining ratio 1
!
class qos-7
  bandwidth remaining ratio 1
!
class class-default
  bandwidth remaining ratio 2
!
end-policy-map

```

### Verification

Verify if the queue is growing, by checking the values of *Inst-queue-len (cells)*. This should ideally be 0 (zero). Also, check the values of *Total Dropped* field and ensure that there are no queue drops.

```

Router# show policy-map interface tenGigE 0/0/0/0
TenGigE0/0/0/0 direction input: Service Policy not installed

TenGigE0/0/0/0 output: egress_policy2

Class qos-1
Classification statistics          (packets/bytes)    (rate - kbps)
  Matched                        : 17312125/12799683759    0
  Transmitted                     : 17312125/12799683759    0
  Total Dropped                : 0/0                      0
Queueing statistics
  Queue ID                        : 9
  High watermark                  : N/A
  Inst-queue-len (cells)        : 0
  Avg-queue-len                   : N/A
  Taildropped(packets/bytes)      : 6537833/4845971570
  Queue(conform)                  : 10774292/7953712189    0
  RED random drops(packets/bytes) : 0/0

- - -
- - -

Class class-default
Classification statistics          (packets/bytes)    (rate - kbps)
  Matched                        : 941172/853166230        0
  Transmitted                     : 926830/840112420      0
  Total Dropped                   : 14342/13053810       0
Queueing statistics
  Queue ID                        : 8
  High watermark                  : N/A
  Inst-queue-len (cells)          : 0
  Avg-queue-len                   : N/A
  Taildropped(packets/bytes)      : 14342/13053810
  Queue(conform)                  : 926830/840112420      0
  RED random drops(packets/bytes) : 0/0

```



### Related Topics

- [Congestion Management Overview, on page 1](#)
- [Configure VOQ-Level Traffic Shaping, on page 9](#)
- [Bandwidth Remaining, on page 2](#)

### Associated Commands

- [priority](#)

## Traffic Shaping

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

## 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

- The **priority** and **shape average** commands must not be configured together in the same class.

### 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(config)# policy-map egress_policy1
Router(config-pmap)# class c5
Router(config-pmap-c)# shape average 40 percent100 mbps
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface TenGigE 0/0/0/0
Router(config-if)# service-policy output egress_policy1
Router(config-if)# commit
```

## Running Configuration

```

policy-map egress_policy1
  class c1
    bandwidth remaining percent 10
  !
  class c3
    bandwidth remaining percent 10
  !
  class c5
    bandwidth remaining percent 20
    shape average 2 gbps ==>
  !
  class c4
    bandwidth remaining percent 5
  !
  class c2
    priority level 1
  !
  class c7
    bandwidth remaining percent 20
  !
  class c6
    bandwidth remaining percent 15
  !
  class class-default
  !
end-policy-map

class-map c5
  match traffic-class 5
commit

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

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

```

## Verification

```

Router# show qos interface TenGigE 0/0/0/9 output

Mon Nov 16 15:41:15.738 UTC
Interface: TenGigE0_0_0_9 output
Bandwidth configured: 10000000 kbps Bandwidth programed: 10000000 kbps
ANCP user configured: 0 kbps ANCP programed in HW: 0 kbps
Port Shaper programed in HW: 0 kbps
Policy: egress Total number of classes: 8
-----
Level: 0 Policy: egress_policy1 Class: c1
QueueID: 81 (Priority Normal)

```

```

Committed Weight: 0 Excess Weight: 10
Bandwidth: 0 kbps, BW sum for Level 0: 0 kbps, Excess Ratio: 10
-----
Level: 0 Policy: egress_policy1 Class: c3
QueueID: 83 (Priority Normal)
Committed Weight: 0 Excess Weight: 10
Bandwidth: 0 kbps, BW sum for Level 0: 0 kbps, Excess Ratio: 10
-----
Level: 0 Policy: egress_policy1 Class: c5<<<<<<<<<<<<
QueueID: 85 (Priority Normal)
PIR: 2000128 kbps PBS: 25001875 bytes
Committed Weight: 0 Excess Weight: 20
Bandwidth: 0 kbps, BW sum for Level 0: 0 kbps, Excess Ratio: 20
-----
Level: 0 Policy: egress_policy1 Class: c4
QueueID: 84 (Priority Normal)
Committed Weight: 0 Excess Weight: 5
Bandwidth: 0 kbps, BW sum for Level 0: 0 kbps, Excess Ratio: 5
-----
Level: 0 Policy: egress_policy1 Class: c2
QueueID: 82 (Priority 1)
-----
Level: 0 Policy: egress_policy1 Class: c7
QueueID: 87 (Priority Normal)
Committed Weight: 0 Excess Weight: 20
Bandwidth: 0 kbps, BW sum for Level 0: 0 kbps, Excess Ratio: 20
-----
Level: 0 Policy: egress_policy1 Class: c6
QueueID: 86 (Priority Normal)
Committed Weight: 0 Excess Weight: 15
Bandwidth: 0 kbps, BW sum for Level 0: 0 kbps, Excess Ratio: 15
-----
Level: 0 Policy: egress_policy1 Class: class-default
QueueID: 80 (Priority Normal)
Committed Weight: 0 Excess Weight: 20
Bandwidth: 0 kbps, BW sum for Level 0: 0 kbps, Excess Ratio: 20
-----

```

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](#)

### Associated Commands

- [shape average](#)

## Traffic Policing

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

Traffic policing also provides a certain amount of bandwidth management by allowing you to set the burst size (Bc) for the committed information rate (CIR). See, [Committed Bursts, on page 12](#).

In addition to rate-limiting, traffic policing also allows you to independently mark (or classify) the packets. See [Policer Marking, on page 13](#).

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

- Single-Rate Two-Color (SR2C) in color-blind mode. See [Single-Rate Policer, on page 15](#).

### 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.
- Policing credits are lost for traffic dropped by other features.

## 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.

## Committed Burst Calculation

To calculate committed burst, use the following formula:

$$bc \text{ (in bytes)} = \text{CIR bps} * (1 \text{ byte} / 8 \text{ bits}) * 0.1 \text{ seconds}$$

The standard time to be used in this calculation is 100 milliseconds (0.1 seconds).

For example, if the committed information rate is 20,00,000 bps, then using the committed burst formula, the committed burst is 25,000 bytes.

$$bc = 2000000 * (1/8) * (100/1000)$$

$$bc = 25,000 \text{ bytes}$$

It is important that you set the burst values high enough to ensure good throughput. If your router drops packets and reports an exceeded rate even though the conformed rate is less than the configured CIR, use the **show interface** command to monitor the current burst, determine whether the displayed value is consistently close to the committed burst (bc) value, and if the actual rates (the committed rate) are close to the configured committed rate. If not, the burst values might be too low. Try reconfiguring the burst rates using the suggested calculations.

## Policer Marking

In addition to rate-limiting, traffic policing allows you to independently mark (or classify) the packet according to whether the packet conforms or violates a specified rate. Packet marking also allows you to partition your network into multiple priority levels or CoS.

Policer marking is also referred as conditional marking, as the marking is done based on the policer state (conform or exceed). Policer marking is done by setting the IP precedence value or IP DSCP value. Use the traffic policer to set this value for the packets that enter the network. The networking devices within your network can then use this setting to determine how the traffic should be treated.

If you want to mark traffic but do not want to use traffic policing, you can use class-based, unconditional packet marking. See, [Class-based Unconditional Packet Marking](#).



---

**Note** Egress packet marking is not supported.

---

## Multiple Action Set

The Multiple Action Set feature allows you to mark packets with multiple action sets (conditional and unconditional) through a class map.

These are the supported action sets:

- set-qos-group
- set-dscp
- set-cos

At least two set of actions for each policer action can be configured by using the **conform-action** command and the **exceed-action** command, within a class map for IP, MPLS, or Layer 2 data paths.



**Note** The ingress policy action **set qos-group** affects the marking of the packet.

## Configure Conditional Policer Marking

The Cisco NCS 5000 Series Router support conditional policer marking in ingress direction.

### Configuration Example

You have to accomplish the following to complete the conditional policer marking 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 policy rate for the traffic
4. Specifying the action(s) to be take on the packets that conform the rate limit
5. Specifying the action(s) to be take on the packets that exceed the rate limit
6. Attaching the policy-map to an input interface

```
Router# configure
Router(config)# policy-map ingress_policy1
Router(config-pmap)# class icl
Router(config-pmap-c)# set qos-group 5
Router(config-pmap-c)# police rate 5 gbps
Router(config-pmap-c-police)# conform-action set qos-group 4
Router(config-pmap-c-police)# conform-action set dscp 3
Router(config-pmap-c-police)# exceed-action set qos-group 6
Router(config-pmap-c-police)# exit
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface TenGigE 0/0/0/5
Router(config-if)# service-policy input ingress_policy1
Router(config-if)# commit
```

### Running Configuration

```
policy-map ingress
class icl
  set qos-group 5
  police rate 5 gbps
  conform-action set qos-group 4
  conform-action set dscp 3
  exceed-action set qos-group 6
!
!
class class-default
!
end-policy-map
```

```
interface TenGigE 0/0/0/5
  service-policy input ingress_policy1
!
```

## Verification

```
Router# show qos interface TenGigE 0/0/0/5 input
```

```
Interface: TenGigE0_0_0_5 input
Bandwidth configured: 10000000 kbps Bandwidth programed: 10000000 kbps
ANCP user configured: 0 kbps ANCP programed in HW: 0 kbps
Port Shaper programed in HW: 0 kbps
Policy: ingress Total number of classes: 2
-----
Level: 0 Policy: ingress_policy1 Class: icl
QueueID: 0 (Port Default)
Policer Profile: 112 (Single)
Conform: 5000000 kbps (5 gbps) Burst: 62500000 bytes (0 Default)
Child Policer Conform: set qos-grp 4 set dscp 3
Child Policer Exceed: set qos-grp 6
-----
Level: 0 Policy: ingress_policy1 Class: class-default
QueueID: 0 (Port Default)
```

## Related Topics

- [Policer Marking, on page 13](#)

## Associated Commands

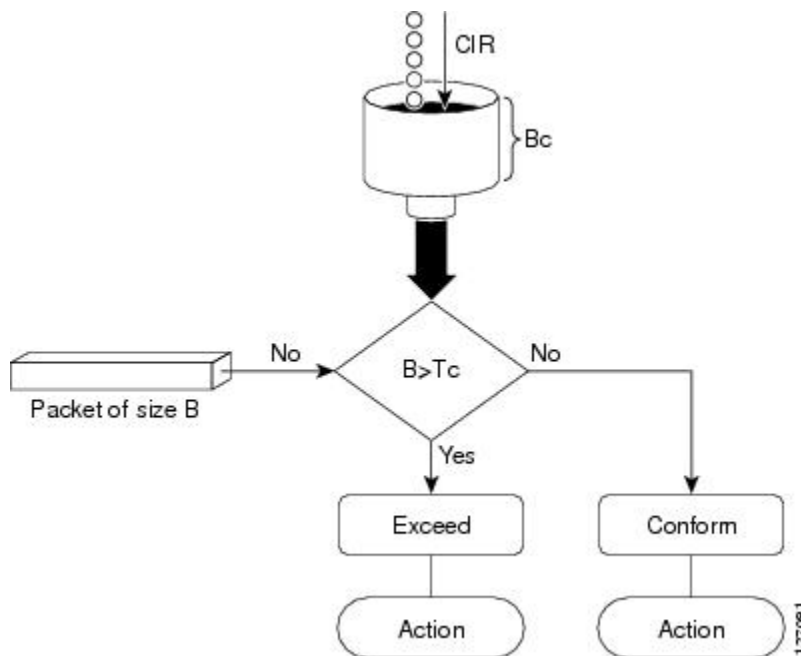
- [conform-action](#)
- [exceed-action](#)
- [police rate](#)

# 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  $T_c$  token bucket can contain up to the  $B_c$  value, which can be a certain number of bytes or a period of time. If a packet of size  $B$  is greater than the  $T_c$  token bucket, then the packet exceeds the CIR value and a configured action is performed. If a packet of size  $B$  is less than the  $T_c$  token bucket, then the packet conforms and a different configured 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. In the most common traffic policing configurations, traffic that conforms to the CIR is sent and traffic that exceeds is sent with a decreased priority or is dropped. Users can change these configuration options to suit their network needs.

### Configuration Example

You have to accomplish the following to complete the Single-Rate Two-Color (SR2C) 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 policy rate for the traffic
4. Specifying the action to be take on the packets that conform the rate limit
5. Specifying the action to be take on the packets that exceed the rate limit
6. Attaching the policy-map to an input interface



```

Router# configure
Router(config)# policy-map ingress_policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# police rate 100 mbps burst 6000 bytes
Router(config-pmap-c-police)# conform-action set qos-group 2
Router(config-pmap-c-police)# conform-action set cos 2
Router(config-pmap-c-police)# exceed-action set qos-group 3
Router(config-pmap-c-police)# exceed-action set cos 3
Router(config-pmap-c-police)# exit
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface 0/0/0/14
Router(config-if)# service-policy input policy1
Router(config-if)# no shutdown
Router(config-if)# commit

```

### Running Configuration

```

/* class-map configuration */
class-map match-any class1
  match dscp 10
end-class-map

/* Traffic policing configuration */
policy-map ingress_policy1
  class class1
    police rate 100 mbps burst 6000 bytes
      conform-action set qos-group 2
      conform-action set cos 2
      exceed-action set qos-group 3
      exceed-action set cos 3
  !
  !
  class class-default
  !
end-policy-map

interface TenGigE 0/0/0/14
  service-policy input ingress_policy1
!

```

### Verification

```

Router# show policy-map interface TenGigE 0/0/0/14 input
TenGigE0/0/0/14 input: ingress_policy1

Class class1
  Classification statistics          (packets/bytes)  (rate - kbps)
  Matched                          :          410861/308109455      0
  Transmitted                       : N/A
  Total Dropped                     :              0/0              0
  Policing statistics               (packets/bytes)  (rate - kbps)
  Policed(conform)                  :          27262/20543870      0
  Policed(exceed)                   :          383599/287565585      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	:	N/A	
Total Dropped	:	N/A	

### Related Topics

- [Single-Rate Policer](#), on page 15

### Associated Commands

- [conform-action](#)
- [exceed-action](#)
- [police rate](#)
- [policy-map](#)
- [service-policy](#)