

# Configuring Modular QoS Congestion Avoidance

This chapter covers the following topics:


- Modular QoS Congestion Avoidance , on page 1
- Tail Drop and the FIFO Queue, on page 2
- Fair Adaptive Drop Thresholds , on page 4
- Random Early Detection and TCP, on page 8
- Weighted Random Early Detection, on page 10
- Explicit Congestion Notification (ECN), on page 18


## Modular QoS Congestion Avoidance

Congestion avoidance techniques monitor traffic flow to anticipate and avoid congestion at common network bottlenecks. Avoidance techniques are implemented before congestion occurs as compared with congestion management techniques that control congestion after it has occurred.

**Note** From Cisco IOS XR Release 7.3.1 onwards, systems with Cisco NC57 line cards running in compatibility mode support QoS over Layer 2 services for:

- Local switching [xconnect or bridging]
- L2 VPN – VPWS

Starting with Cisco IOS XR Release 7.4.1 systems with Cisco NC57 line cards running in native mode support QoS over Layer 2 services for:

- Local switching [xconnect or bridging]
- L2 VPN – VPWS

**Note** For traffic requiring header decapsulation, the size of the header that is being removed is still included for the egress queuing actions. To offset this header size (required to achieve line rate for small frame sizes), configure an egress user policy with user overhead accounting on the egress interface. This policy can be a dummy policy configuration as well (allowing full traffic rate), if a policy isn't already in use or required on the egress interface.

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.

Congestion avoidance is achieved through packet dropping. The router supports these QoS congestion avoidance techniques:


- Tail Drop and the FIFO Queue, on page 2
- Random Early Detection and TCP, on page 8
- Weighted Random Early Detection, on page 10


# Tail Drop and the FIFO Queue

Tail drop is a congestion avoidance technique that drops packets when an output queue is full until congestion is eliminated. Tail drop treats all traffic flow equally and does not differentiate between classes of service. It manages the packets that are unclassified, placed into a first-in, first-out (FIFO) queue, and forwarded at a rate determined by the available underlying link bandwidth.

# Tail Drop and the FIFO Queue: Tip to Optimize Hardware Resources

Tail drop is a congestion avoidance mechanism that works by discarding packets at the tail end of a queue when the queue length exceeds a certain threshold. These thresholds are critical in managing network congestion and are determined based on the Guaranteed Service Rate (GSR) of a traffic class, typically set at a value representing 10 milliseconds of the GSR.

### Tail Drop Thresholds-Rate Class Profile Association and ASIC Limitations

When configuring tail drop thresholds, each threshold value is associated with a rate class profile, which defines the performance characteristics for that threshold. (A rate class profile typically includes bandwidth limits, priority levels, queueing policies, and drop policies like tail drop thresholds. These profiles manage how different types of traffic are treated as they traverse the network, ensuring that critical services get the bandwidth and latency treatment they require.)

However, there is a limitation: each ASIC can support only up to 64 rate class profiles. This hardware constraint necessitates efficient utilization of these profiles.

### Resource Optimization Tip: Share Rate Class Profiles

Since multiple traffic classes can have the same tail drop threshold value, they can share a rate class profile. This sharing conserves hardware resources, allowing for a more scalable network design. Conversely, if each traffic class requires a unique tail drop threshold, more rate class profiles must be created, which can quickly exhaust the available hardware resources.

Therefore, to optimize network performance and resource utilization, we recommend that you strategically fine-tune tail drop threshold values so that more traffic classes share rate class profiles.

# Configure Tail Drop

Packets satisfying the match criteria for a class accumulate in the queue reserved for the class until they are serviced. The **queue-limit** command is used to define the maximum threshold for a class. When the maximum threshold is reached, the enqueued packets to the class queue result in tail drop (packet drop).

### Restrictions

- When configuring the **queue-limit** command, you must configure one of the following commands: **priority**, **shape average**, **bandwidth** or **bandwidth remaining**, except for the default class.

### Configuration Example

You have to accomplish the following to complete the tail drop configuration:

1. Creating (or modifying) a policy map that can be attached to one or more interfaces to specify a service policy
2. Associating the traffic class with the traffic policy
3. Specifying the maximum limit the queue can hold for a class policy configured in a policy map.
4. Specifying priority to a class of traffic belonging to a policy map.
5. (Optional) Specifying the bandwidth allocated for a class belonging to a policy map or specifying how to allocate leftover bandwidth to various classes.
6. Attaching a policy map to an output interface to be used as the service policy for that interface.

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

Router(config)# policy-map test-qlimit-1
Router(config-pmap)# class qos-1
Router(config-pmap-c)# queue-limit 100 us
Router(config-pmap-c)# priority level 7
Router(config-pmap-c)# exit
Router(config-pmap)# exit

Router(config)# interface HundredGigE 0/6/0/18
Router(config-if)# service-policy output test-qlimit-1
Router(config-if)# commit
```

### Running Configuration

```
class-map qos-1
 match traffic-class 1
commit
```

```
policy-map test-qlimit-1
 class qos-1
  queue-limit 100 us
  priority level 7
 !
 class class-default
 !
 end-policy-map
!
```

### Verification

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

NOTE:- Configured values are displayed within parentheses
Interface HundredGigE0/6/0/18 ifh 0x3000220  -- output policy
NPU Id:                        3
Total number of classes:       2
Interface Bandwidth:           100000000 kbps
VOQ Base:                      11176
VOQ Stats Handle:              0x88550ea0
Accounting Type:               Layer1 (Include Layer 1 encapsulation and above)
------------------------------------------------------------------------------
Level1 Class (HP7)                       =   qos-1
Egressq Queue ID                         =   11177 (HP7 queue)
TailDrop Threshold                       =   1253376 bytes / 100 us (100 us)
WRED not configured for this class

Level1 Class                             =   class-default
Egressq Queue ID                         =   11176 (Default LP queue)
Queue Max. BW.                           =   101803495 kbps (default)
Queue Min. BW.                           =   0 kbps (default)
Inverse Weight / Weight                  =   1 (BWR not configured)
TailDrop Threshold                       =   1253376 bytes / 10 ms (default)
WRED not configured for this class
```

### Related Topics

- Tail Drop and the FIFO Queue, on page 2

### Associated Commands

- queue-limit

# Fair Adaptive Drop Thresholds

Fair adaptive drop threshold is a queue management mechanism that

- dynamically sets the tail drop threshold for packet queues that use shared system buffers, based on current system resource usage
- allocates buffer descriptors fairly across all active virtual output queues (VOQs) by recalculating the maximum queue limit for each queue
- prevents a single queue from monopolizing resources, and

- optimizes burst absorption and enhances network stability.

Buffer descriptors are data structures that store information about the memory location, size, and status of each packet or buffer in the queue. The number of available buffer descriptors limits the number of packets that can be queued.

***Table 1: Feature History Table***

| Feature Name | Release Information | Feature Description |
|---|---|---|
| Configurable fair adaptive drop thresholds | Release 25.3.1 | Introduced in this release on: NCS 5500 line cards<br>You can improve burst absorption and reduce packet drops on both priority and non-priority queues by configuring buffer allocation strategy as per your need. With this feature, you can now configure adaptive limits for buffer descriptor usage per queue at ingress. By configuring different adaptive limits, you control the share of system buffer resources that each queue can consume. This approach optimizes burst handling and helps prevent resource starvation across multiple classes of service.<br>The feature introduces the **hw-module profile qos ingress-fadt-set** command. |

### Fair adaptive threshold implementation on NCS 5500 series routers

Fair adaptive drop thresholds are implemented on Cisco NCS 5500 series line cards and fixed routers. These routers have shallow on-chip buffers and a shared dynamic random-access memory (DRAM) buffer for congestion scenarios. The fair adaptive drop threshold mechanism prevents any single VOQ from exceeding a configurable percentage of the available buffer descriptors and protects system stability during traffic bursts.

Example: If a sudden microburst occurs on a low-priority class, the adaptive drop threshold limits the queue's growth, preventing it from consuming excessive buffer resources and causing instability.

## How fair adaptive drop thresholds work

Unexpected packet loss (VOQ tail drops) can occur on any traffic class on Cisco NCS 5500 Series line cards and fixed routers, even when traffic rates are below the guaranteed service rate and no sustained congestion exists. This typically happens when a single VOQ consumes excessive buffer resources, starving other VOQs. Using fair adaptive drop threshold dynamically adjusts queue limits (tail drop thresholds) based on available resources, preventing any one VOQ from monopolizing buffers. This ensures balanced buffer allocation across all active queues.

### Summary

Key components involved in the process:

- Virtual Output Queue (VOQ): Buffers packets per traffic class.
- Buffer Descriptors: Track packets and buffers in memory.
- Fair adaptive drop thresholds mechanism: Adjusts queue thresholds dynamically to allocate buffer descriptors fairly.
- QoS policy map: Defines queue limits and traffic shaping per class.

The process involves monitoring queue lengths, dynamically adjusting tail drop thresholds, and enforcing limits based on system resource status.

### Workflow

These stages describe how fair adaptive drop thresholds impact burst absorption and packet drops:

1. The router receives traffic and enqueues packets per class as defined by the QoS policy.
2. Each queue consumes buffer descriptors as packets are added.
3. The fair adaptive drop thresholds mechanism monitors the total buffer descriptors usage across all VOQs.
4. When a queue approaches its adaptive limit, determined as a percentage (for example, 20%) of available buffer descriptors, the tail drop threshold is enforced, and additional incoming packets are dropped.
5. If you change the system configuration (for example, by adjusting the fair adaptive drop thresholds percentage using the **hw-module profile qos ingress-fadt-set** command), the new threshold applies after you reload the line card.

| When... | Then... |
|---|---|
| A traffic burst occurs on a priority or non-priority class | Fair adaptive drop threshold applies a dynamic queue limit (tail drop threshold) to a congested queue, which can be lower than the configured queue limit |
| Multiple queues are active on the same core | Fair adaptive drop threshold mechanism dynamically divides buffer descriptor resources among all active queues, reducing risk of system instability. |
| The system is configured with a **high** fair adaptive drop thresholds setting | High percentage of available buffer descriptors allocated to a queue; allows better burst absorption on that queue but may adversely impact other active queues. |
| The system is configured with a **low** fair adaptive drop thresholds setting | Very low percentage of available buffer descriptors allocated to a queue; results in poor burst absorption but is helpful in high-scale scenarios. |
| The system is configured with a **medium** fair adaptive drop thresholds setting | Moderate (default setting) usage of buffer descriptor allocation; an optimal condition for resource usage. |
| Fair adaptive drop thresholds are disabled (not recommended) | A single congested queue may consume all available resources, risking system instability. |

### Result

Fair adaptive drop thresholds reduce the likelihood of system instability and packet loss by ensuring fair resource allocation. However, the queue may not reach the configured maximum limit, because the actual maximum depends on the packet size and the number of active queues.

# Fair adaptive drop threshold settings

### Recommendation: fine-tune queue limits for microburst absorption

Use higher queue limits where possible to improve the system's ability to absorb microbursts.

### Requirement: reload line card

Reload the line card after you configure the **hw-module profile qos ingress-fadt-set** command to ensure configuration changes take effect.

# Fair adaptive drop threshold settings and trade-offs

Use this table to evaluate the fair adaptive drop threshold configuration that best meets your network requirements.

***Table 2: Trade-offs for Each Fair Adaptive Drop Threshold Configuration Option***

| Fair adaptive threshold setting | Burst absorption | Risk of resource starvation |
|---|---|---|
| Low | Low | Very low |
| Medium | Moderate | Moderate |
| High | High | High |
| Disable | Very high | Very high, may have negative impact on operations (not recommended) |

# Configure fair adaptive drop threshold values

To configure fair adaptive drop threshold values, follow these steps.

### Before you begin

- Plan for a line card reload to apply the setting.

### Procedure

**Step 1** Enter hardware module configuration mode.

**Example:**

```
Router(config)#hw-module profile qos ingress-fadt-set
```

**Step 2** Set the fair adaptive drop threshold value.

**Example:**

```
Router(config)#hw-module profile qos ingress-fadt-set medium
```

This sets the threshold to medium for moderate burst absorption.

**Example:**

```
Router(config)#hw-module profile qos ingress-fadt-set medium location 0/0/CPU0
```

This configuration sets the threshold to medium specifically for the location or line card, 0/0/CPU0.

If both global and per-location configurations exist for a line card, the per-location configuration takes effect.

**Step 3** Reload the line card to apply the setting.

**Example:**

```
Router#reload location 0/0/CPU0
```

The router enforces the new fair adaptive drop threshold setting, optimizing burst absorption and resource distribution across all active VOQs.

# Random Early Detection and TCP

The Random Early Detection (RED) congestion avoidance technique takes advantage of the congestion control mechanism of TCP. By randomly dropping packets prior to periods of high congestion, RED tells the packet source to decrease its transmission rate. Assuming the packet source is using TCP, it decreases its transmission rate until all packets reach their destination, indicating that the congestion is cleared. You can use RED as a way to cause TCP to slow transmission of packets. TCP not only pauses, but it also restarts quickly and adapts its transmission rate to the rate that the network can support.

RED distributes losses in time and maintains normally low queue depth while absorbing traffic bursts. When enabled on an interface, RED begins dropping packets when congestion occurs at a rate you select during configuration.

# Configure Random Early Detection

The **random-detect** command with the **default** keyword must be used to enable random early detection (RED). The **random-detect default** command enables RED using system-defined default thresholds. These thresholds are derived dynamically based on router-specific queue characteristics and available buffer resources. Because the values are hardware-dependent, explicit minimum and maximum thresholds are not displayed or configurable when the default option is used.

### Guidelines

If you configure the **random-detect default** command on any class including class-default, you must configure one of the following commands: **shape average**, **bandwidth**, and **bandwidth remaining**.

### Configuration Example

You have to accomplish the following to complete the random early detection configuration:

1. Creating (or modifying) a policy map that can be attached to one or more interfaces to specify a service policy
2. Associating the traffic class with the traffic policy
3. Enabling RED with default minimum and maximum thresholds.
4. (Optional) Specifying the bandwidth allocated for a class belonging to a policy map or specifying how to allocate leftover bandwidth to various classes.
5. (Optional) Shaping traffic to the specified bit rate or a percentage of the available bandwidth.
6. Attaching a policy map to an output interface to be used as the service policy for that interface.

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

Router# configure
Router(config)# policy-map test-wred-2
Router(config-pmap)# class qos-1
Router(config-pmap-c)# random-detect default
Router(config-pmap-c)# shape average percent 10
Router(config-pmap-c)# end-policy-map
Router(config)# commit
Router(config)# interface HundredGigE 0/6/0/18
Router(config-if)# service-policy output test-wred-2
Router(config-if)# commit
```

### Running Configuration

```
class-map qos-1
 match traffic-class 1
commit

policy-map test-wred-2
 class qos-1
  random-detect default
  shape average percent 10
 !
 class class-default
 !
 end-policy-map
!

interface HundredGigE 0/6/0/18
 service-policy output test-wred-2
!
```

### Verification

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

NOTE:- Configured values are displayed within parentheses
Interface HundredGigE0/6/0/18 ifh 0x3000220  -- output policy
NPU Id:                        3
Total number of classes:       2
Interface Bandwidth:           100000000 kbps
VOQ Base:                      11176
VOQ Stats Handle:              0x88550ea0
Accounting Type:               Layer1 (Include Layer 1 encapsulation and above)
------------------------------------------------------------------------------
Level1 Class                             =   qos-1
Egressq Queue ID                         =   11177 (LP queue)
Queue Max. BW.                           =   10082461 kbps (10 %)
Queue Min. BW.                           =   0 kbps (default)
Inverse Weight / Weight                  =   1 (BWR not configured)
Guaranteed service rate                  =   10000000 kbps
TailDrop Threshold                       =   12517376 bytes / 10 ms (default)

Default RED profile
WRED Min. Threshold                      =   12517376 bytes (10 ms)
WRED Max. Threshold                      =   12517376 bytes (10 ms)

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

- Random Early Detection and TCP, on page 8

### Associated Commands

- random-detect

# Weighted Random Early Detection

The Weighted Random Early Detection (WRED) drops packets selectively based on any specified criteria, like discard-class. WRED uses this matching criteria to determine how to treat different types of traffic.

You can configure WRED using the **random-detect** command and different discard-class values. The value can be range or a list of values that are valid for that field. You can also use minimum and maximum queue thresholds to determine the dropping point. Ensure that the WRED maximum threshold value is close to the queue limit. When the maximum threshold value is reached, packets start to get dropped.

You can also configure WRED threshold values per discard class. Such an approach helps differentiate when to drop packets among different discard classes, helping prioritize packets among discard classes.

When a packet arrives, the following actions occur:

- The average queue size is calculated.
- If the average queue size is less than the minimum queue threshold, the arriving packet is queued.
- If the average queue size is between the minimum queue threshold for that type of traffic and the maximum threshold for the interface, the packet is either dropped or queued, depending on the packet drop probability for that type of traffic.
- If the average queue size is greater than the maximum threshold, the packet is dropped.

# Average Queue Size for WRED

The router automatically determines the parameters to use in the WRED calculations. The average queue size is based on the previous average and current size of the queue. The formula is:

average = (old_average * (1-2 -x)) + (current_queue_size * 2 -x)

where *x* is the exponential weight factor.

For high values of *x*, the previous average becomes more important. A large factor smooths out the peaks and lows in queue length. The average queue size is unlikely to change very quickly, avoiding a drastic change in size. The WRED process is slow to start dropping packets, but it may continue dropping packets for a time after the actual queue size has fallen below the minimum threshold. The slow-moving average accommodates temporary bursts in traffic.



**Note**

- The exponential weight factor, *x*, is fixed and is not user configurable.
- If the value of *x* gets too high, WRED does not react to congestion. Packets are sent or dropped as if WRED were not in effect.
- If the value of *x* gets too low, WRED overreacts to temporary traffic bursts and drops traffic unnecessarily.

For low values of *x*, the average queue size closely tracks the current queue size. The resulting average may fluctuate with changes in the traffic levels. In this case, the WRED process responds quickly to long queues. Once the queue falls below the minimum threshold, the process stops dropping packets.

# Configure Weighted Random Early Detection

This configuration task is similar to that used for RED except that the **random-detect** command is not configured in RED.

### Restrictions

- You cannot use the **random-detect** command in a class configured with the **priority** command, because WRED cannot be configured in a class that has been set for priority queueing (PQ).
- When configuring the **random-detect** command, you must configure one of the following commands: **shape average**, **bandwidth**, and **bandwidth remaining**.
- Do not enable Explicit Congestion Notification (ECN) and Weighted Random Early Detection (WRED) at the same time. So, do not use the following commands together:

```
*/ ECN Configuration */
Router(config-pmap-c)# random-detect ecn

*/ WRED Configuration */
Router(config-pmap-c)# random-detect discard-class 0 10 ms 500 ms
```

### Configuration Example

You have to accomplish the following to complete the random early detection configuration:

1. Creating (or modifying) a policy map that can be attached to one or more interfaces to specify a service policy
2. Associating the traffic class with the traffic policy
3. Enabling WRED by specifying the match criteria (discard-class).
4. (Optional) Specifying the bandwidth allocated for a class belonging to a policy map or specifying how to allocate leftover bandwidth to various classes.
5. (Optional) Shaping traffic to the specified bit rate or a percentage of the available bandwidth.
6. (Optional) Changing queue limit to fine-tune the amount of buffers available for each queue.
7. Attaching a policy map to an output interface to be used as the service policy for that interface.

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

Router# configure
Router(config)# policy-map test-wred-1
Router(config-pmap)# class qos-1
Router(config-pmap-c)# random-detect default
Router(config-pmap-c)# random-detect discard-class 0 10 ms 500 ms
Router(config-pmap-c)# shape average percent 10
Router(config-pmap-c)# commit

Router(config)# interface HundredGigE 0/6/0/18
Router(config-if)# service-policy output policy1
Router(config-if)# commit
```

### Running Configuration

```
class-map qos-1
 match traffic-class 1
commit

policy-map test-wred-1
 class qos-1
  random-detect default
  random-detect discard-class 0 10 ms 500 ms
  shape average percent 10
 !
 class class-default
 !
 end-policy-map
!
```

```
interface HundredGigE 0/6/0/18
 service-policy output test-wred-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:       2
Interface Bandwidth:           100000000 kbps
Policy Name:                   test-wred-1
VOQ Base:                      1184
Accounting Type:               Layer1 (Include Layer 1 encapsulation and above)
------------------------------------------------------------------------------
Level1 Class                                =   qos-1
Egressq Queue ID                            =   1185 (LP queue)
Queue Max. BW.                              =   10000152 kbps (10 %)
Queue Min. BW.                              =   0 kbps (default)
Inverse Weight / Weight                     =   1 / (BWR not configured)
Guaranteed service rate                     =   10000000 kbps
Peak burst                                  =   36864 bytes (default)
TailDrop Threshold                          =   1250000896 bytes / 1000 ms (default)

WRED profile for Discard_Class 0
WRED Min. Threshold                         =   12499968 bytes (10 ms)
WRED Max. Threshold                         =   624999936 bytes (500 ms)

Default RED profile
WRED Min. Threshold                         =   7499776 bytes (6 ms)
WRED Max. Threshold                         =   12499968 bytes (10 ms)

WRED ECN                                    =   Disabled

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                     =   50000000 kbps
Peak burst                                  =   36864 bytes (default)
TailDrop Threshold                          =   62499840 bytes / 10 ms (default)
WRED not configured for this class
```

## Related Topics

- Weighted Random Early Detection, on page 10
- Configure Random Early Detection, on page 8

## Associated Commands

- random-detect

# Configure WRED Counters by Class

*Table 3: Feature History Table*

| Feature Name | Release Information | Feature Description |
|---|---|---|
| Configure WRED Counters by Class on Cisco NCS 5700 Series Base (Non-SE) Version Line Cards | Release 7.8.1 | You can now view WRED statistics per class on Cisco NCS 5700 Series base (non-SE) version line cards operating in native mode. This functionality provides a more accurate and granular statistics profile for packet drops. |
| Configure WRED Counters by Class | Release 7.4.1 | This feature enables the display of WRED statistics per class, thus providing a more accurate and granular statistics profile for packet drops. Such insight allows you to monitor, anticipate, and avoid congestion at common bottlenecks on your network.<br>This functionality introduces the hw-module profile qos wred-stats-enable command and modifies the output of the **show policy-map interface** command. |

- **Statistics profile**—Up until now, running the **show policy-map interface** command displayed the drop statistics per class under tail drop. Which meant that no matter what drop mechanism you deployed (Tail Drop and the FIFO Queue, on page 2, Random Early Detection and TCP, on page 8, or Weighted Random Early Detection, on page 10), the statistics profile that displayed was a number aggregated under the tail drop entry. Given that WRED provides a far more granular control over packet drops than the other mechanisms, the generic statistics profile didn't provide you with the validation that you required for your WRED actions.
- **WRED counters by class**—This functionality introduces a new command, **hw-module profile qos wred-stats-enable**. Configuring this command enables the display of WRED and RED statistics per class, thus providing a more accurate and granular statistic for packet drops.
- **Why WRED counters by class**—Prior to this functionality, you did not have visibility into the WRED drop counters and had no way to validate WRED actions. With this improvement, the **show policy-map interface** command provides the much-needed visibility by clearly depicting the drop counters per class for WRED. This visibility gives you the confidence that WRED is indeed in action and is providing the requisite congestion avoidance.
- **Guidelines and limitations**
  - WRED max-drop threshold statistics are unavailable.
  - WRED statistics are collected separately for each drop precedence value.

- **WRED counters support cheat-sheet**—Here's a quick look at some key support areas and their details for WRED counters by class.

| Support | Details |
|---|---|
| Line card | Supported on Cisco NC57 line cards with external Ternary Content-Addressable Memory (TCAM) operating in native mode.<br>Supported on Cisco NCS 5700 Series base (non-SE) version line cards operating in native mode. |
| Statistics | WRED drop count is available for all the supported discard-values (from 0 through 2). |

- **Configure WRED counters by class**—To configure WRED counters by class you must:
  - Configure the **hw-module profile** for WRED statistics. Manually reload the chassis to activate the profile.
  - Configure a class map.
  - Create a service policy map.
  - Attach the service policy to an interface.

```
/*Configure hw-module profile for wred stats*/
Router#conf
Router(config)#hw-module profile qos wred-stats-enable
Router(config)#commit
Router#reload

/*Configure a class map*/
Router(config)#class-map TC3
Router(config-cmap)#match traffic-class 3
Router(config-cmap)#commit
Router(config-cmap)#end-class-map


/*Create a service policy map*/
Router(config)#policy-map EGRESS_WRED
Router(config-pmap)#class TC3
Router(config-pmap-c)#shape average 1 gbps
Router(config-pmap-c)#random-detect discard-class 0 1 ms 2 ms
Router(config-pmap-c)#random-detect discard-class 1 3 ms 5 ms
Router(config-pmap-c)#random-detect discard-class 2 10 ms 20 ms
Router(config-pmap-c)#commit
Router(config-pmap)end-policy-map


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

- **Running Configuration**

```
class-map match-any TC3
  match traffic-class 3
  end-class-map
!
policy-map EGRESS_WRED
  class TC3
    shape average 1 gbps
    random-detect discard-class 0 1 ms 2 ms
    random-detect discard-class 1 3 ms 5 ms
    random-detect discard-class 2 10 ms 20 ms
!
    class class-default
!
    end-policy-map
!
interface HundredGigE0/7/0/2
 service-policy output EGRESS_WRED
 ipv4 address 192.168.0.1 255.255.255.0
!
```

- **Verification**

```
Router#show qos int hundredGigE 0/7/0/2 output
NOTE:- Configured values are displayed within parentheses
Interface HundredGigE0/7/0/2 ifh 0xe000088  -- output policy
NPU Id:                        0
Total number of classes:       2
Interface Bandwidth:           100000000 kbps
Policy Name:                   EGRESS_WRED
SPI Id:                        0x0
VOQ Base:                      1160
PFC enabled:                   0
Accounting Type:               Layer1 (Include Layer 1 encapsulation and above)
------------------------------------------------------------------------------
Level1 Class                             =   TC3
Egressq Queue ID                         =   1163 (LP queue)
Queue Max. BW.                           =   1000000 kbps (1 gbits/sec)
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                       =   4999168 bytes / 40 ms (default)

WRED profile for Discard_Class 0
WRED Min. Threshold                      =   124928 bytes (1 ms)
WRED Max. Threshold                      =   249856 bytes (2 ms)

WRED profile for Discard_Class 1
WRED Min. Threshold                      =   374784 bytes (3 ms)
WRED Max. Threshold                      =   624896 bytes (5 ms)

WRED profile for Discard_Class 2
WRED Min. Threshold                      =   1249792 bytes (10 ms)
WRED Max. Threshold                      =   2499840 bytes (20 ms)

Default RED profile
WRED Min. Threshold                      =   256 bytes (0 ms)
WRED Max. Threshold                      =   256 bytes (0 ms)

WRED ECN                                 =   Disabled

Level1 Class                             =   class-default
Egressq Queue ID                         =   1160 (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                   =   50000000 kbps
Peak burst                                =   36864 bytes (default)
TailDrop Threshold                        =   62499840 bytes / 10 ms (default)
WRED not configured for this class
```

- **show policy-map interface output for statistics verification**

```
Router#show policy-map int hundredGigE 0/7/0/2 output

HundredGigE0/7/0/2 output: EGRESS_WRED

Class TC3
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :             1151236/1151236000           0
    Transmitted         :              578807/578807000            0
    Total Dropped       :              572429/572429000            0
  Queueing statistics
    Queue ID                             : 1163
    Taildropped(packets/bytes)           : 572429/572429000

    WRED profile for Discard Class 0
    RED Transmitted (packets/bytes)            : 0/0
    RED random drops(packets/bytes)            : 0/0
    RED maxthreshold drops(packets/bytes)      : N/A
    RED ecn marked & transmitted(packets/bytes): N/A
    WRED profile for Discard Class 1
    RED Transmitted (packets/bytes)            : 0/0
    RED random drops(packets/bytes)            : 0/0
    RED maxthreshold drops(packets/bytes)      : N/A
    RED ecn marked & transmitted(packets/bytes): N/A
    WRED profile for Discard Class 2
    RED Transmitted (packets/bytes)            : 578807/578807000
    RED random drops(packets/bytes)            : 572429/572429000
    RED maxthreshold drops(packets/bytes)      : N/A
    RED ecn marked & transmitted(packets/bytes): N/A
    RED profile (default)
    RED Transmitted (packets/bytes)            : 0/0
    RED random drops(packets/bytes)            : 0/0
    RED maxthreshold drops(packets/bytes)      : N/A
    RED ecn marked & transmitted(packets/bytes): N/A
Class class-default
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :                   0/0                   0
    Transmitted         :                   0/0                   0
    Total Dropped       :                   0/0                   0
  Queueing statistics
    Queue ID                             : 1160
    Taildropped(packets/bytes)           : 0/0
```

- **Clear WRED statistics on an interface**

  Run the following command to clear WRED statistics on an interface.

```
Router#clear qos counters int hundredGigE 0/7/0/2 output
```

## Associated Commands

hw-module profile qos wred-stats-enable

# Explicit Congestion Notification (ECN)

Weighted Random Early Detection (WRED) is implemented at the core routers of a network. Edge routers assign IP precedences to packets, as the packets enter the network. With WRED, core routers then use these precedences to determine how to treat different types of traffic. WRED provides separate thresholds and weights for different IP precedences, enabling the network to provide different qualities of service, in regard to packet dropping, for different types of traffic. Standard traffic may be dropped more frequently than premium traffic during periods of congestion.

ECN is an extension to WRED. ECN marks packets instead of dropping them when the average queue length exceeds a specific threshold value. When configured, ECN helps routers and end hosts to understand that the network is congested and slow down sending packets. However, if the number of packets in the queue is above the maximum threshold, packets are dropped based on the drop probability.

WRED starts dropping packets probabilistically before a queue becomes full, which means that it works during enqueuing. ECN marking with WRED (ECN-WRED) takes place as packets are about to be transmitted from the queue (during the dequeue process), when the router decides whether to forward, drop, or mark packets based on the congestion state of the queue. Also, WRED and ECN don't work together.

RFC 3168, *The Addition of Explicit Congestion Notification (ECN) to IP*, states that with the addition of active queue management (for example, WRED) to the Internet infrastructure, routers are no longer limited to packet loss as an indication of congestion.

**Note** You cannot use this feature when you have set qos-group or mpls experimental along with a traffic class in the ingress policy.

### Implementing ECN

Implementing ECN requires an ECN-specific field that has 2 bits—the ECN-capable Transport (ECT) bit and the CE (Congestion Experienced) bit—in the IP header. The ECT bit and the CE bit can be used to make four ECN field combinations of 00 to 11. The first number is the ECT bit and the second number is the CE bit.

***Table 4: ECN Bit Setting***

| ECT Bit | CE Bit | Combination Indicates |
|---|---|---|
| 0 | 0 | Not-ECN-capable. |
| 0 | 1 | Endpoints of the transport protocol are ECN-capable. |
| 1 | 0 | Endpoints of the transport protocol are ECN-capable. |
| 1 | 1 | Congestion experienced. |

The ECN field combination 00 indicates that a packet is not using ECN. The ECN field combinations 01 and 10—Called ECT(1) and ECT(0), respectively—are set by the data sender to indicate that the endpoints of the transport protocol are ECN-capable. Routers treat these two field combinations identically. Data senders can

use either one or both of these two combinations. The ECN field combination 11 indicates congestion to the endpoints. Packets arriving a full queue of a router will be dropped.

### Packet Handling When ECN Is Enabled

When the number of packets in the queue is below the minimum threshold, packets are transmitted.

If the number of packets in the queue is above the maximum threshold:

- For traffic flows that are not ECN-enabled in only WRED-configured queues, packets are tail-dropped after the queue size exceeds the WRED maximum threshold.
- For traffic flows that are ECN-enabled in only WRED-configured queues, packets are tail-dropped when the queue size exceeds the tail-drop threshold.
- When you configure ECN remarking on your router, incoming packets with ECT bit settings 0 or 1 are marked as CE.

For Layer 3-to-Layer 2 traffic flows in BVI 2-pass mode on routers that have the NC57 line card installed and operating in the native mode:

- ECN remarking to 11 is not supported for ECN-capable flows [ECT(0) or ECT(1)].
- ECN-capable flows on a WRED-enabled queue are subject to WRED processing. The queue starts tail-dropping once it reaches the WRED max threshold.



**Note** When the number of packets reaches the queue limit, all packets are dropped. This is the identical treatment that a packet receives when you enable WRED without ECN configured on the router.

Three different scenarios arise if the number of packets in the queue is between the minimum threshold and the maximum threshold:

- If the ECN field on the packet indicates that the endpoints are ECN-capable (that is, the ECT bit is set to 1 and the CE bit is set to 0, or the ECT bit is set to 0 and the CE bit is set to 1)—and the WRED algorithm determines that the packet should have been dropped based on the drop probability—the ECT and CE bits for the packet are changed to 1, and the packet is transmitted. This happens because ECN is enabled and the packet gets marked instead of dropped.
- If the ECN field on the packet indicates that neither endpoint is ECN-capable (that is, the ECT bit is set to 0 and the CE bit is set to 0), packet is dropped once the queue limit is reached.
- If the ECN field on the packet indicates that the network is experiencing congestion (that is, both the ECT bit and the CE bit are set to 1), the packet is transmitted. No further marking is required.

**Note** Applicable until Cisco IOS XR Release 7.11.1: When the incoming IP traffic with ECN bits set to 10 passes through the ingress qos-policy-map that has the class-map definition of `set DSCP/PREC <value>`, then the ECN bits in the IP header gets modified to 01. This is applicable to NC57 routers operating in the Native mode.

Applicable from Cisco IOS XR Release 7.11.2 : NC57 line cards operating in compatibility mode do not support ECN-bit preservation for terminated VXLAN packets when you configure ACL compression for ingress traffic.

### Limitations

- For release 24.1.2 and later, NC57 line cards operating in compatibility mode do not support ECN bit preservation for terminated VXLAN packets when ACL compression is configured for ingress traffic.
- ECN configuration is not supported per discard class but for all packets enqueued to a VOQ.
- **The following limitation is not applicable to NC57 routers:**

  **SRv6 Disposition Node** — For the incoming traffic that has the outer SRv6 IP header having the ECN bits as non-zero value and the inner IP header ECN bits as zero value, then the ECN bits of the inner IP header will be overwritten to 01.

- Do not enable Explicit Congestion Notification (ECN) and Weighted Random Early Detection (WRED) at the same time. So, do not use the following commands together:

```
*/ ECN Configuration */
Router(config-pmap-c)# random-detect ecn

*/ WRED Configuration */
Router(config-pmap-c)# random-detect discard-class 0 10 ms 500 ms
```

- **The following limitation is specific to NC57 routers operating in Native mode.**

**Note** ECN remarking functionality is not supported with SRv6 tunnels. The inner header in the following scenario is IP and the outer header is SRv6.

**SRv6 Imposition Node** — The ECN bits in outer header traffic class is set to 0 when SRv6 encapsulation is set to traffic-class propagate along with Ingress MQC set to any DSCP/PREC value. This scenario does not impact the inner header ECN bits.

**SRv6 Disposition Node** — The ECN bits in the inner packet header is reset to 0 when the Ingress MQC is set to any DSCP/PREC value.

### Configuration Example

```
Router# configure
Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# bandwidth percent 50
Router(config-pmap-c)# random-detect 1000 packets 2000 packets
Router(config-pmap-c)# random-detect ecn
Router(config-pmap-c)# exit
```

```
Router(config-pmap)# exit
Router(config)# commit
```

### Verification

Use the **show policy-map interface** to verify the configuration.

```
Router# show policy-map interface tenGigE 0/0/0/6 output
TenGigE0/0/0/6 output: pm-out-queue

Class cm-tc-1
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :            85528554/87581239296          4830672
    Transmitted         :            16240891/16630672384          966585
    Total Dropped       :            69287663/70950566912          3864087
  Queueing statistics
    Queue ID                              : 1113
    Taildropped(packets/bytes)            : 69287663/70950566912

    WRED profile for
    RED Transmitted (packets/bytes)             : N/A
    RED random drops(packets/bytes)             : N/A
    RED maxthreshold drops(packets/bytes)       : N/A
    RED ecn marked & transmitted(packets/bytes): N/A
Class class-default
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :                  0/0                   0
    Transmitted         :                  0/0                   0
    Total Dropped       :                  0/0                   0
  Queueing statistics
    Queue ID                              : 1112
    Taildropped(packets/bytes)            : 0/0
```

**Note** No ECN-specific statistics are displayed in the show output for this command. ECN is enabled if all rows indicate **N/A**, as highlighted in the example.

From Cisco IOS XR Release 7.5.4, you can enable the display of ECN-marked packets. See Statistics for ECN-Marked Packets, on page 22.

# Statistics for ECN-Marked Packets

*Table 5: Feature History Table*

| Feature Name | Release Information | Feature Description |
|---|---|---|
| Statistics for ECN-Marked Packets | Release 7.5.4 | By enabling the counters for ECN-marked and transmitted packets, we equip you with the ability to determine the exact congestion point in your network and tweak your network design and traffic profiles accordingly.<br><br>Earlier releases didn't support counters for ECN-marked and transmitted packets.<br><br>This functionality introduces the following options to enable viewing the counters:<br><br>• YANG data model: **Cisco-IOS-XR-fia-hw-profile-cfg.yang** (see Github, YANG Data Models Navigator)<br><br>• CLI: **hw-module profile qos ecn-marking-stats** |

In earlier releases, while you could enable ECN to mark packets and manage packet congestion, you had no insights into the number of ECN-marked and transmitted packets. This meant that while you knew that packets were ECN-marked—that is, there was congestion—you couldn't say precisely where in the network it was happening and how much congestion was occurring at each node.

With Cisco IOS XR Release 7.5.4, you can enable the counters for ECN-marked and transmitted packets. With this information, you now know the extent of packet congestion and can determine the exact congestion point in your network and adjust your traffic profiles accordingly.

## Statistics for ECN-Marked Packets: Guidelines and Limitations

- This feature is only supported on the following line cards and fixed chassis PIDs of the NCS5500 Series:
  - NC55-36X100G
  - NC55-18H18F
  - NC55-24X100G-SE
  - NC55-36X100G-S
  - NC55-24H12F-SE
  - NC55-36X100G-A-SE
  - NCS-55A1-36H-SE-S

- NCS-55A1-36H-S
- NCS-55A1-24H
- NCS-55A1-48Q6H

- When you enable this feature, egress ACL actions will not be supported.
- The number of ECN classes you can enable for statistics is restricted to 120.
- ECN statistics are only supported for egress policies applied on the main interfaces of physical ports or link bundles.

## Configure to Enable Statistics for ECN-Marked Packets

To enable the viewing of statistics for ECN-Marked Packets, configure the **hw-module profile qos ecn-marking-stats** command.

```
Router#config
Router(config)#hw-module profile qos ecn-marking-stats
In order to activate this profile, you must manually reload the chassis/all line cards
Router(config)#commit
```



**Note** The router must be reloaded for the hw-module command to be functional.

### Verification

Run the **show policy-map interface** command to view the details about ECN-marked packets.

```
Router#show policy-map interface tenGigE 0/0/0/6 output
TenGigE0/0/0/6 output: pm-out-queue
Class tc2
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :             8102473/1037116544           0
    Transmitted         :             5795588/741835264            0
    Total Dropped       :             2306885/295281280            0
  Queueing statistics
    Queue ID                              : 1026
    Taildropped(packets/bytes)            : 2306885/295281280
    WRED profile for Discard Class 0
    RED Transmitted (packets/bytes)             : N/A
    RED random drops(packets/bytes)             : N/A
    RED maxthreshold drops(packets/bytes)       : N/A
    RED ecn marked & transmitted(packets/bytes): 6193754/836156790
    RED profile (default)
    RED Transmitted (packets/bytes)             : N/A
    RED random drops(packets/bytes)             : N/A
    RED maxthreshold drops(packets/bytes)       : N/A
```

# Configure ECN Maximum Marking Probability

*Table 6: Feature History Table*

| Feature Name | Release Information | Feature Description |
|---|---|---|
| Configure ECN Maximum Marking Probability | Release 7.3.3 | This feature allows you to configure percentages for ECN maximum marking probability after considering your network congestion and end-to-end application needs. With this flexibility, you can fine-tune the network's reactivity to congestion according to the tolerance levels of your end-to-end applications.<br><br>By controlling the probability of packets being ECN marked at the ECN maximum threshold, you can optimize the congestion notification from the traffic destination to the traffic source. This optimization avoids aggressive throttling at the source (from too high ECN maximum marking probability) or traffic drops in transit due to significantly less throttling at the source (from too low ECN maximum marking probability).<br><br>This functionality modifies the existing **random-detect** command. |

- **First, something about packet drop probability and Explicit Congestion Notification (ECN)**—The probability that a packet will be ECN-marked is based on the minimum threshold, maximum threshold, and maximum mark probability percent.

  When the average queue size is above the minimum threshold, ECN starts marking packets to indicate congestion. When an ECN-marked packet reaches the end host, the host transmits confirmation of receipt of this packet to the transmitting host. The confirmation indicates to the transmitting host that congestion has begun and that the transmission or packet rate must be reduced.

  Maximum mark probability is the percentage of packets marked when the average queue size is at the maximum threshold. The ECN mark probability increases linearly from 0 when queue length is less than minimum threshold to 10% when the queue length equals or exceeds the maximum threshold. For example, when the average queue size is halfway between minimum-threshold and maximum-threshold the mark probability is 5%, and half the packets transmitted will be ECN marked.

- **Configurable ECN Maximum Marking Probability**—So far, the ECN maximum marking probability was not user-configurable and was fixed at 10%. The arrangement of preset marking probabilities meant that the router couldn't adapt the congestion notification to network characteristics and application tolerance. With the flexibility to configure maximum ECN marking probability, you can configure the

maximum marking probability to a higher or lower percent, depending on the requirements of congestion handling in the network for a given traffic flow and application type. This also allows you to choose a consistent ECN maximum marking probability across all network elements instead of operating them at the factory default values. The following table maps the linear increase in mark probability percentage as the queue length conditions change from 0 to the tail drop threshold.

| Queue length conditions | Mark probability increases linearly... |
|---|---|
| Queue length is less than the minimum threshold value | from 0 |
| Average queue length is between minimum-threshold and maximum-threshold | corresponding to the average queue length:<br>(maximum mark probability) * (average queue length –minimum threshold value)/(maximum threshold value – minimum threshold value) |
| Queue length equals the maximum threshold | to the configured maximum probability percentage |
| maximum threshold value < queue length < tail drop threshold | to 100% |

- **Some FAQs**
  - **What if I set the ECN marking probability percentage to 10%?**

    In such a case, there's no change in traffic behavior from what it was for previously ECN-marked packets. Tail drop comes into play when the queue length exceeds the tail drop threshold.

  - **How does traffic behave beyond my configured value of maximum ECN marking probability?**

    Let's say you've set the maximum ECN marking probability to 5%. Any more increase in the average queue length beyond maximum ECN threshold shifts the marking probability to 100%, and once the queue length exceeds the tail drop threshold, Tail Drop and the FIFO Queue, on page 2 comes into effect.

  - **How do I enable this functionality?**

    Create a policy with **random-detect ecn** and **random-detect** <*min-threshold*> <*max-threshold*> **probability percent** <*value*> configured on an interface.

  - **How do I disable this functionality?**

    To disable this functionality, do not configure the **probability percent** value.

  - **Are the user-configured values programmed precisely in the hardware?**

    The user-configured values are rounded off to the nearest granularity during programming in the hardware. The round-off is minimal when the 'Pmax/(ECNmax-ECNmin)' ratio is a power of 2, where Pmax is the user-configured maximum marking probability in percent and ECNmin and ECNmax are the user-configured minimum and maximum ECN thresholds.

    The **show qos interface** command output displays the actual value programmed in hardware and the user-configured value for reference.

- **Important Guidelines**

- **Line Cards that support this functionality**:
  - NC55-36X100G
  - NC55-18H18F
  - NC55-24X100G-SE
  - NC55-36X100G-S
  - NC55-24H12F-SE
  - NC55-36X100G-A-SE
  - NCS-55A1-36H-SE-S
  - NCS-55A1-36H-S
  - NCS-55A1-24H
  - NCS-55A1-48Q6H
- **The following interface types** support this functionality:
  - Physical interfaces
  - Bundle interfaces
  - Subinterfaces
  - Bundle subinterfaces
- This functionality is **supported for all interface speeds**.
- If you have a **policy map with one or multiple classes with maximum ECN marking probability enabled**, you can:
  - Apply the map to any of the supported interface types.
  - Remove the map from any of the supported interface types.
  - Modify the map while you're attaching it to multiple interfaces.
- The probability percentage option is supported only with **random-detect ecn** configured in the same class. Else, the policy is rejected when applied on an interface

- **Configuring ECN Maximum Marking Probability**—After you have configured Explicit Congestion Notification (ECN), on page 18, the **random-detect** command displays an option to add the probability in percent.

### Configuration Example

```
Router#configure
Router(config)#policy-map policy1
Router(config-pmap)#class class1
Router(config-pmap-c)#bandwidth percent 50
Router(config-pmap-c)#random-detect 1 mbytes 2 mbytes probability percent 5
Router(config-pmap-c)#random-detect ecn
Router(config-pmap-c)#commit
```

### Verification

Run the **show qos interface** command to view the configured ECN Maximum Marking Probability value (displayed as **ECN Mark Probability**).

```
Router#sh qos interface FourHundredGigE 0/1/0/4 output


NOTE:- Configured values are displayed within parentheses
Interface FourHundredGigE0/1/0/4 ifh 0x800150  -- output policy
------------------------------------------------------------------------------
.
.
 Level2 Class                              =   LOSSLESSTCP_Egress
   Egressq Queue ID                          =   5475 (LP queue)
   Queue Max. BW.                            =   no max (default)
   Inverse Weight / Weight                   =   3 / (20)
   Guaranteed service rate                   =   78329670 kbps
   TailDrop Threshold                        =   390070272 bytes / 40 ms (default)

   Default RED profile
   WRED Min. Threshold                       =   9787392 bytes (1 ms)
   WRED Max. Threshold                       =   19580928 bytes (2 ms)
   ECN Mark Probability                      =   4.99 (5)

   WRED ECN                                  =   Enabled
```