Configuring Weighted Random Early Detection

- Avoiding Network Congestion, on page 1
- Tail Drop, on page 1
- Weighted Random Early Detection, on page 1
- Limitations for WRED Configuration, on page 3
- Usage Guidelines for WRED, on page 4
- Configuring WRED, on page 4
- WRED Configuration Example, on page 9
- WRED Support with Hierarchical QoS, on page 10
- Displaying WRED Configuration, on page 11
- Best Practices for WRED Configuration, on page 13
- Feature History for WRED, on page 15

Avoiding Network Congestion

Heterogeneous networks include different protocols used by applications, giving rise to the need to prioritize traffic in order to satisfy time-critical applications while still addressing the needs of less time-dependent applications, such as file transfer. If your network is designed to support different traffic types that share a single data path between devices in a network, implementing congestion avoidance mechanisms ensures fair treatment across the various traffic types and avoids congestion at common network bottlenecks. Congestion avoidance mechanism is achieved through packet dropping.

Random Early Detection (RED) is a commonly used congestion avoidance mechanism in a network.

Tail Drop

Tail drop treats all traffic equally and does not differentiate within a class of service. When the output queue is full and tail drop is in effect, packets are dropped until the congestion is eliminated and the queue is no longer full.

Weighted Random Early Detection

The RED mechanism takes advantage of the congestion control mechanism of TCP. Packets are randomly dropped prior to periods of high congestion. Assuming the packet source uses TCP, it decreases its transmission

rate until all the packets reach their destination, indicating that the congestion is cleared. You can use RED as a way to cause TCP to slow down transmission of packets. TCP not only pauses, but also restarts quickly and adapts its transmission rate to the rate that the network can support.

WRED is the Cisco implementation of RED. It combines the capabilities of RED algorithm with IP Precedence or Differentiated Services Code Point (DSCP) or Class of Service (COS) values. On the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2), WRED combines the capabilities of RED algorithim with discard-class 0 and 1.

How WRED Works

WRED reduces the chances of tail drop by selectively dropping packets when the output interface begins to show signs of congestion. WRED drops some packets early rather than waiting until the queue is full. Thus it avoids dropping large number of packets at once and minimizes the chances of TCP global synchronization.

Approximate Fair Drop (AFD) is an Active Queue Management (AQM) algorithm that determines the packet drop probability. The probability of dropping packets depends upon the arrival rate calculation of a flow at ingress and the current queue length.

AFD based WRED is implemented on wired network ports.

AFD based WRED emulates the preferential dropping behavior of WRED. This preferential dropping behavior is achieved by changing the weights of AFD sub-classes based on their corresponding WRED drop thresholds. Within a physical queue, traffic with larger weight incurs less drop probability than that of smaller weight.

- Each WRED enabled queue has high and low thresholds.
- A sub-class of higher priority has a larger AFD weight.
- The sub-classes are sorted in ascending order, based on lowest of WRED minThreshold.

Note On the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2), AFD is not used. Drop probability for different queue length and discard-class is programmed in the hardware.

WRED Weight Calculation

AFD weight is calculated using low and high threshold values; AFD is an adjsuted index of the average of WRED high and WRED low threshold values.

When a packet arrives at an interface, the following events occur:

1. The drop probability is calculated. The drop probability increases as the AFD weight decreases. That means, if the average of low and high threshold values is less, the drop probability is more.
2. WRED consideres the priority of packet flows and the threshold values before deciding to drop the packet. The CoS, DSCP or IP Precedence values are mapped to the specified thresholds. Once these thresholds are exceeded, packets with the configured values that are mapped to these thresholds are eligible to be dropped. Other packets with CoS, DSCP or IP Precedence values assigned to the higher thresholds are en-queued. This process keeps the higher priority flows intact and minimizes the latency in packet transmission.

Note On the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2), discard-class 0 and 1 are mapped to the specified thresholds.

3. If packets are not dropped using WRED, they are tail-dropped.

Limitations for WRED Configuration

- Weighted Tail Drop (WTD) is enabled by default on all the queues.
- WRED can be enabled / disabled per queue. When WRED is disabled, WTD is adapted on the target queue. Policy-map with WRED profile is configured only on physical ports as output policy.
- WRED is supported only in network port queues and is not supported on internal CPU queues and stack queues.
- Each WRED physical queue can support three different threshold pairs. Each pair is for one QoS tag value.

  On the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2), WRED threshold pair configuration are in percentage of queue-limit for that particular queue. WRED can be configured for seven queues in the queueing policy, and upto 2 WRED threshold pairs per queue are supported. One threshold pair is for discard-class 0 and the other for discard-class 1.
- Ensure that you configure bandwidth or shape in the policy-map along with WRED.

  On the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2), ensure that you configure bandwidth remaining ratio or shape in the policy-map along with WRED.
- Specify all the WRED thresholds only in percentage mode.
- Map the WRED threshold pairs by mapping class-map filter with corresponding match filters.

  We recommend the class-map with match “any” filter.
- WRED for priority traffic is not supported.
- WRED and queue limit are not supported for the same policy.

  WRED and queue limit for the same policy are supported on the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2).
- Wired ports support a maximum of eight physical queues, of which you can configure WRED only on four physical queues, each with three threshold pairs. The remaining queues are configured with WTD. Policies with more than four WRED queues are rejected.

  This restriction is not applicable to the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2).

Usage Guidelines for WRED

To configure AFD based WRED feature, specify the policy map and add the class. Use the **random-detect** command to specify the method (using the dscp-based / cos-based / precedence-based /discard-class-based arguments) that you want WRED to use to calculate the drop probability.

Note You can modify the policy on the fly. The AFD weights are automatically recalculated.

WRED can be configured for any kind of traffic like IPv4/IPv6, Multicast, and so on. WRED is supported on all 8 queueing classes.

On the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2), WRED configuration is based only on discard-class. WRED threshold pair configuration is a percentage of queue-limit for that particular queue. By configuring threshold values for discard-class 0 and 1 separately, WRED for that particular queue is achieved. Similarly, this can be done for all the seven queues in the queueing policy.

Consider the following points when you are configuring WRED with **random-detect** command:

- With dscp-based argument, WRED uses the DSCP value to calculate drop probability.
- With cos-based argument, WRED uses the COS value to calculate drop probability.
- With discard-class-based argument, WRED uses the discard-class value to calculate drop probability.
- By default, WRED uses the IP Precedence value to calculate drop probability. **precedence-based** argument is the default and it is not displayed in the CLI.

Note **show run policy-map** *policy-map* command does not display “precedence” though precedence is configured with **random-detect** command.

- The dscp-based and precedence-based arguments are mutually exclusive.
- Each of the eight physical queues can be configured with different WRED profiles.

Configuring WRED

Configuring WRED based on DSCP Values

Use the following steps to configure WRED profile based on DSCP values in packet mode:

This procedure is not applicable to the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2).

Procedure

	Command or Action	Purpose
Step 1	**class-map** *match-criteria class-name* **Example:** `device(config)# class-map match-any CS`	Configures match criteria for the class map. Recommended match-criteria is match-any
Step 2	**match** *class-map-name* **Example:** `device(config-cmap)#match dscp CS1`	Match a class-map.
Step 3	**policy-map** *name* **Example:** `device(config)#policy-map PWRED`	Specifies the name of the WRED profile policy to be created.
Step 4	**class** *class-name* **Example:** `device(config-pmap)#class CS`	Specifies the name of the Class to be associated with the policy.
Step 5	Use either **bandwidth** {*kbps* | **remaining ratio** | **percent** *percentage*} or **shape** { **average** | **peak** } *cir* **Example:** `device(config-pmap-c)#bandwidth percent 10`	Specify either the bandwidth allocated for a class belonging to a policy map or the traffic shaping.
Step 6	**random-detect** *dscp-based* **Example:** `device(config-pmap-c)#random-detect dscp-based`	Configures WRED to use the DSCP value when it calculates the drop probability for the packet.
Step 7	**random-detect dscp** *dscp-value* **percent** *minThreshold maxThreshold* **Example:** `device(config-pmap-c)#random-detect dscp cs1 percent 10 20`	Specifies the minimum and maximum thresholds, in percentage.
Step 8	**interface** *interface-name* **Example:** `device(config)#interface HundredGigE1/0/2`	Enters the interface configuration mode.
Step 9	**service-policy output** *policy-map* **Example:** `device(config-if)#service-policy output pwred`	Attaches the policy map to an output interface.

Configuring WRED based on Class of Service Values

Use the following steps to configure WRED profile based on Class of Service (COS) values in packet mode:

This procedure is not applicable to the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2).

Procedure

	Command or Action	Purpose
Step 1	**class-map** *match-criteria class-name* **Example:** `device(config)# class-map match-any CS`	Configures match criteria for the class map. Recommended match-criteria is match-any
Step 2	**match** *class-map-name* **Example:** `device(config-cmap)#match cos 3`	Match a class-map.
Step 3	**policy-map** *name* **Example:** `device(config)#policy-map PWRED`	Specifies the name of the WRED profile policy to be created.
Step 4	**class** *class-name* **Example:** `device(config-pmap)#class CS`	Specifies the name of the Class to be associated with the policy.
Step 5	**bandwidth** {*kbps*| **remaining** *percentage* | **percent** *percentage*} **Example:** `device(config-pmap-c)#bandwidth percent 10`	Specifies the bandwidth allocated for a class belonging to a policy map.
Step 6	**random-detect** *cos-based* **Example:** `device(config-pmap-c)#random-detect cos-based`	Configures WRED to use the CoS value when it calculates the drop probability for the packet.
Step 7	**random-detect cos** *cos-value* **percent** *minThreshold maxThreshold* **Example:** `device(config-pmap-c)#random-detect cos 3 percent 10 20`	Specifies the minimum and maximum thresholds, in percentage.
Step 8	**interface** *interface-name* **Example:** `device(config)# interface HundredGigE1/0/2`	Enters the interface configuration mode.

	Command or Action	Purpose
Step 9	**service-policy output** *policy-map* **Example:** `device(config-if)#service-policy output pwred`	Attaches the policy map to an output interface.

Configuring WRED based on IP Precedence Values

Use the following steps to configure WRED profile based on IP precedence values in packet mode:

This procedure is not applicable to the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2).

Procedure

	Command or Action	Purpose
Step 1	**class-map** *match-criteria class-name* **Example:** `device(config)# class-map match-any CS`	Configures match criteria for the class map. Recommended match-criteria is match-any
Step 2	**match** *class-map-name* **Example:** `device(config-cmap)#match precedence 3`	Match a class-map.
Step 3	**policy-map** *name* **Example:** `device(config)#policy-map pwred`	Specifies the name of the WRED profile policy to be created.
Step 4	**class** *class-name* **Example:** `device(config-pmap)#class CS`	Specifies the name of the Class to be associated with the policy.
Step 5	**bandwidth** {*kbps*| **remaining** *percentage* | **percent** *percentage*} **Example:** `device(config-pmap-c)#bandwidth percent 10`	Specifies the bandwidth allocated for a class belonging to a policy map.
Step 6	**random-detect** *precedence-based* **Example:** `device(config-pmap-c)#random-detect precedence-based`	Configures WRED to use the IP precedence value when it calculates the drop probability for the packet.
Step 7	**random-detect precedence** *precedence-value* **percent** *minThreshold maxThreshold* **Example:**	Specifies the minimum and maximum thresholds, in percentage.

	Command or Action	Purpose
	`device(config-pmap-c)#random-detect precedence 3 percent 10 20`	
Step 8	**interface** *interface-name* **Example:** `device(config)#interface HundredGigE1/0/2`	Enters the interface configuration mode.
Step 9	**service-policy output** *policy-map* **Example:** `device(config-if)#service-policy output pwred`	Attaches the policy map to an output interface.

Configuring WRED based on Discard-class Values

Use the following steps to configure WRED profile based on discard-class values in packet mode:

This configuration procedure is applicable only to the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600-SUP-2).

Procedure

	Command or Action	Purpose
Step 1	**policy-map type queueing** *name* **Example:** `Device(config)# policy-map type queueing wred`	Specifies the name of the WRED queueing profile policy to be created.
Step 2	**class** *class-name* **Example:** `Device(config-pmap)# class tc2`	Specifies the name of the class to be associated with the policy. **Note** The class-name must be traffic-class based.
Step 3	**shape average percent** *percentage* **Example:** `Device(config-pmap-c)# shape average percent 10`	Specifies the traffic shaping percentage bandwidth for committed information rate.
Step 4	**random-detect discard-class-based** **Example:** `Device(config-pmap-c)# random-detect discard-class-based`	Configures WRED to use the discard-class value when it calculates the drop probability for the packet.
Step 5	**random-detect discard-class** *value* **percent** *minThreshold maxThreshold mark-probability-denominator* **Example:**	Configures WRED discard-class 0, and specifies the minimum and maximum thresholds, in percentage.

	Command or Action	Purpose
	`Device(config-pmap-c)# random-detect discard-class 0 percent 20 80 1`	
Step 6	**random-detect discard-class** *value* **percent** *minThreshold maxThreshold mark-probability-denominator* **Example:** `Device(config-pmap-c)# random-detect discard-class 1 percent 15 70 1`	Configures WRED discard-class 1, and specifies the minimum and maximum thresholds, in percentage.
Step 7	**end** **Example:** `Device(config-pmap-c)# end`	Exits WRED class action configuration mode and returns to privileged EXEC mode.

WRED Configuration Example

The following example enables WRED to use DSCP profile for class CS. It configures three sub-classes cs1, cs2, and cs3 with their WRED minimum and maximum thresholds and finally applies the policy to Hundred Gigabit Ethernet interface 8:

```
Device(config)# class-map match-any CS
Device(config-cmap)# match dscp cs1
Device(config-cmap)# match dscp cs2
Device(config-cmap)# match dscp cs3
Device(config-cmap)# policy-map PWRED
Device(config-pmap)# class CS
Device(config-pmap-c)# bandwidth ratio 10
Device(config-pmap-c)# random-detect dscp-based
Device(config-pmap-c)# random-detect dscp cs1 percent 10 20
Device(config-pmap-c)# random-detect dscp cs2 percent 20 30
Device(config-pmap-c)# random-detect dscp cs3 percent 34 44
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# interface HundredGigE1/0/8
Device(config-if)# service-policy output PWRED
```

The following is a WRED configuration example of the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2):

```
Device# configure terminal
Device(config)# policy-map type queueing wred
Device(config-pmap)# class tc7
Device(config-pmap-c)# shape average percent 40
Device(config-pmap-c)# class tc6
Device(config-pmap-c)# shape average percent 20
Device(config-pmap-c)# random-detect discard-class-based
Device(config-pmap-c)# random-detect discard-class 0 percent 20 80 1
Device(config-pmap-c)# random-detect discard-class 1 percent 15 70 1
```

WRED Support with Hierarchical QoS

Hierarchical QoS allows you to specify QoS behavior at multiple policy levels, which provides a high degree of granularity in traffic management.

For HQoS, WRED is allowed only on the child policy and not on the parent policy. You can have the shaping configured on the parent policy and WRED on the child.

The following example configures the parent policy **pwred-parent** with traffic shaped on the basis of 10 percent of the bandwidth, that applies to its child, **pwred-child** configured for DSCP-based WRED.

```
policy-map PWRED-CHILD
 class CWRED
    bandwidth percent 10
    random-detect dscp-based
    random-detect dscp 1 percent 10 20
    random-detect dscp 10 percent 20 30

policy-map PWRED-PARENT
  class class-default
  shape average percent 10
  service-policy PWRED-CHILD
```

The following are WRED and HQoS WRED sample configuration for the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2):

```
Device(config)# policy-map type queueing wred
Device(config-pmap)# class tc7
Device(config-pmap-c)# shape average percent 40
Device(config-pmap-c)# class tc6
Device(config-pmap-c)# shape average percent 20
Device(config-pmap-c)# random-detect discard-class-based
Device(config-pmap-c)# random-detect discard-class 0 percent 20 80 1
Device(config-pmap-c)# random-detect discard-class 1 percent 15 70 1

Device(config)# policy-map type queueing hqos_wred
Device(config-pmap)# class class-default
Device(config-pmap-c)# shape average percent 80
Device(config-pmap-c)# service-policy wred
Device(config-pmap-c)# end
```

The following show commands for the HQoS WRED and WRED configuration is for the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2):

```
Device# show policy-map type queueing hqos_wred

Policy Map type queueing hqos_wred
Class class-default
Average Rate Traffic Shaping
cir 80%
service-policy wred

Device# show policy-map type queueing wred
Policy Map type queueing wred
Class tc7
Average Rate Traffic Shaping
cir 40%
Class tc6
Average Rate Traffic Shaping
cir 20%
percent-based wred, exponential weight 1
```

```
discard-class min-threshold max-threshold mark-probablity
------------------------------------------------------------
0 20 80 1/1
1 15 70 1/1
2 - - 1/1
3 - - 1/1
4 - - 1/1
5 - - 1/1
6 - - 1/1
7 - - 1/1
```

Displaying WRED Configuration

The following example shows how to display the WRED and threshold labels:

```
Device# show policy-map PWRED

  Policy Map PWRED
    Class CS
      bandwidth 10 (%)
       percent-based wred

      dscp    min-threshold    max-threshold
      ------------------------------------
      cs1 (8)      10               20
      cs2 (16)     20               30
      cs3 (24)     34               44
      default (0)   -
```

The following example shows how to display WRED AFD Weights, WRED Enq (in Packets and Bytes), WRED Drops (in Packets and Bytes), Configured DSCP labels against the Threshold pairs:

Note Use this command only after you initiate the traffic. **show policy-map interface** is updated with WRED configuration only after a traffic is sent.

```
Device# show policy-map interface HundredGigE 1/0/2

HundredGigE1/0/2

  Service-policy output: PWRED

    Class-map: CS (match-any)
      0 packets
      Match:  dscp cs1 (8)
      Match:  dscp cs2 (16)
      Match:  dscp cs3 (24)
      Queueing

      (total drops) 27374016
      (bytes output) 33459200081
      bandwidth 10% (1000000 kbps)

AFD WRED STATS BEGIN
Virtual Class   min/max        Transmit                    Random drop                AFD
 Weight
```

```
   0          10 / 20         (Byte)33459183360             27374016                    12
                              (Pkts)522799759                 427716
              dscp : 8


   1          20 / 30           Byte)0                       0                         20
                                 (Pkts)0                      0
              dscp : 16


   2          34 / 44         (Byte)16721                   0                         31
                                (Pkts)59                     0
              dscp : 24

     Total Drops(Bytes)   : 27374016

     Total Drops(Packets) : 427716
AFD WRED STATS END


    Class-map: class-default (match-any)
      0 packets
      Match: any


      (total drops) 0
      (bytes output) 192
```

Displaying WRED Configurations of the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2)

The following is a sample output from the **show policy-map** command:

```
Device# show policy-map WRED

Policy Map type queueing wred
Class tc7
Average Rate Traffic Shaping
cir 10%
percent-based wred, exponential weight 1

discard-class min-threshold max-threshold mark-probablity
------------------------------------------------------------
0 20 80 1/1
1 15 70 1/1
2 - - 1/1
3 - - 1/1
4 - - 1/1
5 - - 1/1
6 - - 1/1
7 - - 1/1
```

The following is a sample output of the **show policy-map type queueing interface** command:

```
Device(config)# interface FiftyGigE6/0/1
Device(config-if)# service-policy type queueing output wred
Device(config-if)# end

Device# show policy-map type queueing interface FiftyGigE6/0/1

FiftyGigE6/0/1

  Service-policy queueing output: wred

    Class-map: tc7 (match-all)
```

```
  0 packets
  Match: traffic-class 7
  Queueing
  queue limit 30000000 bytes
  (total drops) 0
  (bytes output) 0
  shape (average) cir 20000000000, bc 200000000, be 200000000
  target shape rate 20000000000

Class-map: tc6 (match-all)
  0 packets
  Match: traffic-class 6
  Queueing
  queue limit 7500000 bytes
  (total drops) 0
  (bytes output) 0
  shape (average) cir 10000000000, bc 100000000, be 100000000
  target shape rate 10000000000
    Exp-weight-constant: 1 (1/2)
    Mean queue depth: 0
        Minimum Maximum  Mark
                thresh  thresh  prob
    0              20      80  1/1
    1              15      70  1/1
    2               0       0  1/1
    3               0       0  1/1
    4               0       0  1/1
    5               0       0  1/1
    6               0       0  1/1
    7               0       0  1/1

Class-map: class-default (match-any)
  0 packets
  Match: any

  queue limit 75000000 bytes
  (total drops) 0
  (bytes output) 0
```

Best Practices for WRED Configuration

All best practices listed below are not applicable to the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2), except for **Discard-class Based Configuration**.

- **Support for three WRED configuration pairs**

  Each WRED Physical Queue (AFD Queue) can support three WRED configuation pairs, with unique WRED threshold pair configuration.

```
Policy-map P1
    Class CS
        Random-detect dscp-based
        Random-detect dscp CS1 percent 10 20      // WRED pair 1
        Random-detect dscp CS2  percent 20 30      // WRED pair 2
        Random-detect dscp CS3 percent 30 40      // WRED pair 3
Class-map match-any CS
        match cs1
        match cs2
        match cs3
```

- **Appending WRED configuration pairs**

You can add overlapping threshold pairs into the WRED configuration pairs.

```
Policy-map P1
    Class CS
       Random-detect dscp-based
       Random-detect dscp CS1 percent 10 20       // WRED pair 1
       Random-detect dscp CS2 percent 20 30       // WRED pair 2
       Random-detect dscp CS3 percent 30 40       // WRED pair 3
       Random-detect dscp CS4 percent  30 40     ==> belongs to WRED pair 3
       Random-detect dscp CS5 percent 20 30     ==> belongs to WRED pair 2
Class-map match-any CS
       match cs1
       match cs2
       match cs3
       match cs4  >>
       match cs5  >>
```

- **Default WRED pairs**

  If less than three WRED pairs are configured, any class-map filter participating WRED gets assigned to the third default WRED pair with maximum threshold (100, 100).

```
Policy-map P1
    Class CS
        Random-detect dscp-based
        Random-detect dscp CS1 percent 10 20   // WRED pair 1
        Random-detect dscp CS2 percent 20 30   // WRED pair 2
Class-map match-any CS
        match CS1
        match CS2
        match CS3
        match CS4
```

  In this case, classes CS3 and CS4 are mapped to WRED pair 3 with threshold (100, 100).

- **Rejection of Mismatched Configuration**

  If you configure random-detect without matching filters in a class-map, the policy installation is rejected.

```
Class-map match-any CS
       match CS1
       match CS2
       match CS5
Policy-map P1
    Class CS
        Shape average percent 10
        Random-detect dscp-based
        Random-detect dscp CS1 percent 10 20   // WRED pair 1
        Random-detect dscp CS2 percent 20 30   // WRED pair 2
       Random-detect dscp CS3 percent 30 40   // WRED pair 3  ===> Mismatched sub-class.
```

  When this policy is applied to the interface on the egress side, the policy fails during installation as the class-map values are incorrect:

```
device(config)# int Fo1/0/5
device(config-if)# service-policy output P1
device(config-if)#
*Feb 20 17:33:16.964: %IOSXE-5-PLATFORM: Switch 1 R0/0: fed: WRED POLICY INSTALL
FAILURE.Invalid WRED filter mark: 24 in class-map: CS
*Feb 20 17:33:16.965: %FED_QOS_ERRMSG-3-LABEL_2_QUEUE_MAPPING_HW_ERROR: Switch 1 R0/0:
 fed: Failed to detach queue-map for FortyGigE1/0/5: code 2
```

- **Discard-class Based Configuration**

On the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2), the minimum and maximum threshold values of the random-detect discard-class 1 must be less than or equal to that of the random-detect discard-class 0.

```
policy-map type queueing wred
class tc7
shape average percent 10
random-detect discard-class-based
random-detect discard-class 0 percent 20 80 1
random-detect discard-class 1 percent 15 70 1
```

Feature History for WRED

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	Weighted Random Early Detection mechanism	WRED is a mechanism to avoid congestion in networks. WRED reduces the chances of tail drop by selectively dropping packets when the output interface begins to show signs of congestion, thus avoiding large number of packet drops at once. You can configure WRED to act based on any of the following values: • Differentiated Service Code Point • IP Precedence • Class of Service
Cisco IOS XE Cupertino 17.7.1	Weighted Random Early Detection mechanism	This feature was implemented on Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2).

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/.