Byte-Based Weighted Random Early Detection

This module explains how to enable byte-based Weighted Random Early Detection (WRED), and set byte-based queue limits and WRED thresholds.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for Byte-Based Weighted Random Early Detection

- WRED is only useful when the bulk of the traffic is TCP/IP traffic. With TCP, dropped packets indicate congestion, so the packet source will reduce its transmission rate. With other protocols, packet sources may not respond or may resend dropped packets at the same rate. Thus, dropping packets does not decrease congestion.
- You cannot configure byte-based WRED on a class in which the queue-limit is configured in milliseconds or packets.

Information About Byte-Based Weighted Random Early Detection

Changes in functionality of WRED

This feature extends the functionality of WRED. In previous releases, you specified the WRED actions based on the number of packets. With the byte-based WRED, you can specify WRED actions based on the number of bytes.

Changes in Queue Limit and WRED Thresholds

In Cisco IOS XE Release 2.4, the Cisco ASR 1000 Series Aggregation Services Routers support the addition of bytes as a unit of configuration for both queue limits and WRED thresholds. Therefore, as of this release, packet-based and byte-based limits are configurable, with some restrictions.

How to Configure Byte-Based Weighted Random Early Detection

Configuring Byte-Based WRED

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** *class-map-name*
4. **match ip precedence** ip-precedence-value
5. exit
6. **policy-map** *policy-name*
7. **class** *class-name*
8. **random-detect**
9. **random-detect precedence** *precedence min-threshold* **bytes** *max-threshold* **bytes** *mark-prob-denominator*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map** *class-map-name* **Example:** `Router(config)# class-map c1`	Specifies the user-defined name of the traffic class.
Step 4	**match ip precedence** ip-precedence-value **Example:** `Router(config-cmap)# match ip precedence 1`	Specifies up to eight IP Precedence values used as match criteria.
Step 5	exit **Example:** `Router(config-cmap)# exit`	Exits from class-map configuration mode.
Step 6	**policy-map** *policy-name* **Example:** `Router(config)# policy-map p1`	Specifies the name of the traffic policy to configure.
Step 7	**class** *class-name* **Example:** `Router(config-pmap)# class c1`	Specifies the name of a predefined traffic class, which was configured with the **class-map** command, used to classify traffic to the traffic policy.
Step 8	**random-detect** **Example:** `Router(config-pmap-c)# random-detect`	Enables WRED.
Step 9	**random-detect precedence** *precedence min-threshold* **bytes** *max-threshold* **bytes** *mark-prob-denominator* **Example:** **Example:** `Router(config-pmap-c)# random-detect precedence 1 2000 bytes 3000 bytes 200`	Configures the parameters for bytes with a specific IP precedence.

Configuring the Queue Depth and WRED Thresholds

Before you begin

Be sure that your configuration satisfies the following conditions when configuring the queue depth and WRED thresholds:

- When configuring byte-based mode, the queue limit must be configured prior to the WRED threshold and before the service policy is applied.
- When setting the queue depth and WRED thresholds in an enhanced QoS policies aggregation configuration, the limits are supported only for the default class at a subinterface policy map and for any classes at the main interface policy map.

Note Consider the following restrictions when you configure the queue depth and WRED thresholds:

- Do not configure the queue limit unit before you configure a queueing feature for a traffic class.
- If you do not configure a queue limit, then the default mode is packets.
- When you configure WRED thresholds, the following restrictions apply:
  - The WRED threshold must use the same unit as the queue limit. For example, if the queue limit is in packets, then the WRED thresholds also must be in packets.
  - If you do not configure a queue limit in bytes, then the default mode is packets and you must also configure the WRED threshold in packets.
  - The queue limit size must be greater than the WRED threshold.
- The unit modes for either the queue limit or WRED thresholds cannot be changed dynamically after a service policy is applied.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** *class-name*
5. *qos-queueing-feature*
6. **queue-limit** *queue-limit-size* [**bytes** | **packets**]
7. **random-detect** [**dscp-based** | **prec-based**]
8. Do one of the following:
   - **random-detect dscp** dscp-value {*min-threshold max-threshold* | *min-threshold* **bytes** *max-threshold* **bytes**} [*max-probability-denominator*]
   -
   -
   - **random-detect precedence** precedence {*min-threshold max-threshold* | *min-threshold* **bytes** *max-threshold* **bytes**} max-probability-denominator

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map main-interface`	Specifies the name of the traffic policy that you want to configure or modify and enters policy-map configuration mode.
Step 4	**class** *class-name* **Example:** `Router(config-pmap)# class AF1`	Specifies the name of the traffic class and enters policy-map class configuration mode.
Step 5	*qos-queueing-feature* **Example:** `Router(config-pmap-c)# bandwidth remaining ratio 90`	Enters a QoS configuration command. Some of the queueing features that are currently supported are **bandwidth**, **priority**, and **shape**. **Note** Multiple QoS queueing commands can be entered at this step. However, due to dependencies between the queue limit and WRED thresholds, you should configure WRED after you configure the queue limit.
Step 6	**queue-limit** *queue-limit-size* [**bytes** | **packets**] **Example:** `Router(config-pmap-c)# queue-limit 547500 bytes`	Specifies the maximum number (from 1 to 8192000) of bytes or packets that the queue can hold for this class.
Step 7	**random-detect** [**dscp-based** | **prec-based**] **Example:** `Router(config-pmap-c)# random-detect dscp-based`	Enables WRED in either DSCP-based mode or precedence-based mode.
Step 8	Do one of the following: • **random-detect dscp** dscp-value {*min-threshold max-threshold* | *min-threshold* **bytes** *max-threshold* **bytes**} [*max-probability-denominator*]	Configures WRED parameters for a particular DSCP value or IP precedence.

	Command or Action	Purpose
	• • • **random-detect precedence** precedence {*min-threshold max-threshold* | *min-threshold* **bytes** *max-threshold* **bytes**} max-probability-denominator **Example:** `Router(config-pmap-c)# random-detect precedence 8 750000 bytes 750000 bytes`	**Note** Use the *min-threshold max-threshold* arguments without the **bytes** keyword to configure packet-based thresholds, when the **queue-limit** unit is also packets (the default). Alternatively, use these arguments with the **bytes** keyword when the **queue-limit** unit is configured in bytes.

Examples

Correct Configuration

Invalid Configuration

Correct Configuration

Invalid Configuration

The following examples show both correct and invalid configurations to demonstrate some of the restrictions.

The following example shows the correct usage of setting the queue limit in bytes mode after the **bandwidth remaining ratio** queueing feature has been configured for a traffic class:

```
class AF1
 bandwidth remaining ratio 90
  queue-limit 750000 bytes
```

The following example shows an invalid configuration for the queue limit in bytes mode before the **bandwidth remaining ratio** queueing feature has been configured for a traffic class:

```
class AF1
 queue-limit 750000 bytes
  bandwidth remaining ratio 90
```

The following example shows the correct usage of setting the queue limit in bytes mode after the **bandwidth remaining ratio** queueing feature has been configured for a traffic class, followed by the setting of the thresholds for WRED in compatible byte mode:

```
class AF1
 bandwidth remaining ratio 90
  queue-limit 750000 bytes
   random-detect dscp-based
    random-detect dscp 8 750000 bytes 750000 bytes
```

This example shows an invalid configuration of the WRED threshold in bytes without any queue limit configuration, which therefore defaults to a packet-based queue depth. Therefore, the WRED threshold must also be in packets:

```
class AF1
 bandwidth remaining ratio 90
 random-detect dscp-based
 random-detect dscp 8 750000 bytes 750000 bytes
```

Changing the Queue Depth and WRED Threshold Unit Modes

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no service-policy output** *policy-map-name*
5. **exit**
6. **policy-map** *policy-map-name*
7. **class** *class-name*
8. **queue-limit** *queue-limit-size* [**bytes** | **packets**]
9. Do one of the following:
   - **no random-detect dscp** *dscp-value* {*min-threshold max-threshold* | *min-threshold* **bytes** *max-threshold* **bytes**} [*max-probability-denominator*]
   -
   -
   - **no random-detect precedence** *precedence* {*min-threshold max-threshold* | *min-threshold* **bytes** *max-threshold* **bytes**} *max-probability-denominator*
10. Do one of the following:
    - **random-detect dscp** *dscp-value* {*min-threshold max-threshold* | *min-threshold* **bytes** *max-threshold* **bytes**} [*max-probability-denominator*]
    -
    -
    - **random-detect precedence** *precedence* {*min-threshold max-threshold*| *min-threshold* **bytes** *max-threshold* **bytes**} *max-probability-denominator*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Router(config)# policy-map main-interface`	Specifies the interface where you want to remove a service policy, and enters interface configuration mode.
Step 4	**no service-policy output** *policy-map-name* **Example:** `Router(config-if)# no service-policy output main-interface-policy`	Removes a service policy applied to the specified interface.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns you to global configuration mode.
Step 6	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map main-interface-policy`	Specifies the name of the Traffic policy that you want to modify and enters policy-map configuration mode.
Step 7	**class** *class-name* **Example:** `Router(config-pmap)# class AF1`	Specifies the name of the traffic class and enters policy-map class configuration mode.
Step 8	**queue-limit** *queue-limit-size* [**bytes** | **packets**] **Example:** `Router(config-pmap-c)# queue-limit 5000 packets`	Specifies the maximum number (from 1 to 8192000) of bytes or packets that the queue can hold for this class.
Step 9	Do one of the following: • **no random-detect dscp** *dscp-value* {*min-threshold max-threshold* | *min-threshold* **bytes** *max-threshold* **bytes**} [*max-probability-denominator*] • • • **no random-detect precedence** *precedence* {*min-threshold max-threshold* | *min-threshold* **bytes** *max-threshold* **bytes**} *max-probability-denominator* **Example:** `Router(config-pmap-c)# no random-detect dscp 8 750000 bytes 750000 bytes`	Removes the previously configured WRED parameters for a particular DSCP value or IP precedence.

	Command or Action	Purpose
Step 10	Do one of the following: • **random-detect dscp** *dscp-value* {*min-threshold max-threshold* | *min-threshold* **bytes** *max-threshold* **bytes**} [*max-probability-denominator*] • • • **random-detect precedence** *precedence* {*min-threshold max-threshold*| *min-threshold* **bytes** *max-threshold* **bytes**} *max-probability-denominator* **Example:** `Router(config-pmap-c)# random-detect dscp 8 4000 4000`	Configures WRED parameters for a particular DSCP value or IP precedence. **Note** Use the *min-threshold max-threshold* arguments without the **bytes** keyword to configure packet-based thresholds, when the queue-limit unit is also packets (the default). Alternatively, use these arguments with the **bytes** keyword when the queue-limit unit is configured in bytes.

Examples

The following example shows how to change the queue depth and WRED thresholds to packet-based values once a service policy has been applied to an interface:

```
interface GigabitEthernet1/2/0
no service-policy output main-interface-policy
end
policy-map main-interface-policy
class AF1
 queue-limit 5000 packets
 no random-detect dscp 8 750000 bytes 750000 bytes
 random-detect dscp 8 4000 4000
```

Verifying the Configuration for Byte-Based WRED

SUMMARY STEPS

1. **show policy-map**
2. The **show policy-map interface** command shows output for an interface that is configured for byte-based WRED.

DETAILED STEPS

Step 1 **show policy-map**

The **show policy-map** command shows the output for a service policy called pol1 that is configured for byte-based WRED.

Example:

```
Router# show policy-map
Policy Map pol1
```

```
  Class class c1
Bandwidth 10 (%)
exponential weight 9
        class   min-threshold(bytes)   max-threshold(bytes)  mark-probability
        ----------------------------------------------------------------------

        0       -                     -                     1/10
        1       20000                 30000                 1/10
        2       -                     -                     1/10
        3       -                     -                     1/10
        4       -                     -                     1/10
        5       -                     -                     1/10
        6       -                     -                     1/10
        7       -                     -                     1/10
        rsvp    -                     -                     1/10
```

Step 2 The **show policy-map interface** command shows output for an interface that is configured for byte-based WRED.

Example:

```
Router# show policy-map interface
serial3/1
Service-policy output: pol
Class-map: silver (match-all)
366 packets, 87840 bytes
30 second offered rate 15000 bps, drop rate 300 bps
Match: ip precedence 1
Queueing
Output Queue: Conversation 266
Bandwidth 10 (%)
(pkts matched/bytes matched) 363/87120
depth/total drops/no-buffer drops) 147/38/0
exponential weight: 9
mean queue depth: 25920
class     Transmitted       Random drop      Tail drop     Minimum Maximum Mark
          pkts/bytes        pkts/bytes       pkts/bytes    thresh  thresh  prob
                                                           (bytes)  (bytes)
0             0/0               0/0              0/0       20000    40000  1/10
1           328/78720         38/9120            0/0       22000    40000  1/10
2             0/0               0/0              0/0       24000    40000  1/10
3             0/0               0/0              0/0       26000    40000  1/10
4             0/0               0/0              0/0       28000    40000  1/10
```

Configuration Examples for Byte-Based Weighted Random Early Detection

Example Configuring Byte-Based WRED

The following example shows a service policy called wred-policy that sets up byte-based WRED for a class called prec2 and for the default class. The policy is then applied to Fast Ethernet interface 0/0/1.

```
policy wred-policy
 class prec2
   bandwidth 1000
```

```
   random-detect
   random-detect precedence 2 100 bytes 200 bytes 10
class class-default
   random-detect
   random-detect precedence 4 150 bytes 300 bytes 15
   random-detect precedence 6 200 bytes 400 bytes 5
interface fastethernet0/0/1
  service-policy output wred-policy
```

The following example shows the byte-based WRED results for the service policy attached to Ethernet interface 0/0/1.

```
Router# show policy-map interface
 Ethernet0/0/1
Service-policy output: wred-policy (1177)
Class-map: prec2 (match-all) (1178/10)
0 packets, 0 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: ip precedence 2  (1179)
Queueing
queue limit 62500 bytes
(queue depth/total drops/no-buffer drops) 0/0/0
(pkts queued/bytes queued) 0/0
bandwidth 1000 (kbps)
Exp-weight-constant: 9 (1/512)
Mean queue depth: 0 bytes
class     Transmitted       Random drop      Tail drop Minimum        Maximum     Mark
          pkts/bytes        pkts/bytes       pkts/bytes thresh         thresh     prob
                                                        bytes         bytes
0             0/0               0/0              0/0     15625         31250     1/10
1             0/0               0/0              0/0     17578         31250     1/10
2             0/0               0/0              0/0       100           200     1/10
3             0/0               0/0              0/0     21484         31250     1/10
4             0/0               0/0              0/0     23437         31250     1/10
5             0/0               0/0              0/0     25390         31250     1/10
6             0/0               0/0              0/0     27343         31250     1/10
7             0/0               0/0              0/0     29296         31250     1/10
Class-map: class-default (match-any) (1182/0)
0 packets, 0 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: any  (1183)
0 packets, 0 bytes
5 minute rate 0 bps
queue limit 562500 bytes
(queue depth/total drops/no-buffer drops) 0/0/0
(pkts queued/bytes queued) 0/0
Exp-weight-constant: 9 (1/512)
Mean queue depth: 0 bytes
class     Transmitted       Random drop      Tail drop Minimum        Maximum     Mark
          pkts/bytes        pkts/bytes       pkts/bytes thresh         thresh     prob
                                                        bytes         bytes
0             0/0               0/0              0/0    140625        281250     1/10
1             0/0               0/0              0/0    158203        281250     1/10
2             0/0               0/0              0/0    175781        281250     1/10
3             0/0               0/0              0/0    193359        281250     1/10
4             0/0               0/0              0/0       150           300     1/15
5             0/0               0/0              0/0    228515        281250     1/10
6             0/0               0/0              0/0       200           400     1/5
7             0/0               0/0              0/0    263671        281250     1/10
```

Additional References

Related Documents

Related Topic	Document Title
QoS Commands	*Cisco IOS Quality of Service Solutions Command Reference*
Modular QoS CLI	Modular Quality of Service Command-Line Interface module

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Byte-Based Weighted Random Early Detection

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for Byte-Based Weighted Random Early Detection

Feature Name	Releases	Feature Information
Byte-Based Weighted Random Early Detection	Cisco IOS XE Release 2.4	The Byte-Based Weighted Random Early Detection feature extends the functionality of WRED. In previous releases, you specified the WRED actions based on the number of packets. With the byte-based WRED, you can specify WRED actions based on the number of bytes. This feature was introduced on Cisco ASR 1000 Series Routers. The following commands were introduced or modified: **random-detect**, **random-detect precedence**, **show policy-map, show policy-map interface**.