



Configuring Weighted Random Early Detection

This chapter describes the tasks for configuring Weighted Random Early Detection (WRED), VIP-Distributed WRED (DWRED), and flow-based WRED on a router.

For complete conceptual information, see the section “Weighted Random Early Detection” in the chapter “Congestion Avoidance Overview” in this book.

For a complete description of the WRED and DWRED commands in this chapter, refer to the *Cisco IOS Quality of Service Solutions Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index, or search online.

The RSVP-ATM QoS Interworking and IP to ATM Class of Service features also use WRED. For information on how to configure these features with WRED, see the chapters “Configuring RSVP-ATM QoS Interworking” and “Configuring IP to ATM Class of Service” in this book.

The WRED feature is supported on the following Cisco router platforms:

- Cisco 1600 series
- Cisco 2500 series
- Cisco 3600 series
- Cisco 4000 series (including 4500 and 4700 series)
- Cisco 7200 series
- Cisco 7500 series with Route Switch Processor (RSP) interface card

The DWRED feature is only supported on Cisco 7000 series routers with an RSP-based RSP7000 interface processor and Cisco 7500 series routers with a Versatile Interface Processor-based VIP2-40 or greater interface processor. A VIP2-50 interface processor is strongly recommended when the aggregate line rate of the port adapters on the VIP is greater than DS3. A VIP2-50 interface processor is required for OC-3 rates.

To use DWRED, distributed Cisco Express Forwarding (dCEF) switching must first be enabled on the interface. For more information on dCEF, refer to the *Cisco IOS Switching Services Configuration Guide* and the *Cisco IOS Switching Services Command Reference*.

**Note**

WRED is useful with adaptive traffic such as TCP/IP. 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.

WRED treats non-IP traffic as precedence 0, the lowest precedence. Therefore, non-IP traffic is more likely to be dropped than IP traffic.

You cannot configure WRED on the same interface as Route Switch Processor (RSP)-based custom queueing, priority queueing, or weighted fair queueing (WFQ). However, you can configure both DWRED and DWFQ on the same interface.

Weighted Random Early Detection Configuration Task List

Random Early Detection (RED) is a congestion avoidance mechanism that 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. WRED drops packets selectively based on IP Precedence. Edge routers assign IP Precedences to packets as they enter the network. WRED is useful on any output interface where you expect to have congestion. However, WRED is usually used in the core routers of a network, rather than at the edge. WRED uses these precedences to determine how it treats different types of traffic.

When a packet arrives, the following events occur:

1. The average queue size is calculated.
2. If the average is less than the minimum queue threshold, the arriving packet is queued.
3. If the average 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.
4. If the average queue size is greater than the maximum threshold, the packet is dropped.

See the section “About WRED” in the chapter “Congestion Avoidance Overview” in this book for more details on the queue calculations and how WRED works.

To configure WRED or DWRED on an interface, perform the tasks in the following sections. The first section is required; the remaining sections are optional.

- Enabling WRED (Required)
- Changing WRED Parameters (Optional)
- Monitoring WRED and DWRED (Optional)

To use DWRED, dCEF switching must first be enabled on the interface.

See the end of this chapter for the section “WRED and DWRED Configuration Examples.”

Enabling WRED

To enable WRED, use the following command in interface configuration mode:

Command	Purpose
<code>random-detect</code>	Enables WRED. If you configure this command on a VIP interface, DWRED is enabled.

You need not specify any other commands or parameters in order to configure WRED on the interface. WRED will use the default parameter values.

Changing WRED Parameters

To change WRED parameters, use one of the following commands in interface configuration mode:

Command	Purpose
<code>random-detect exponential-weighting-constant exponent</code>	Configures the weight factor used in calculating the average queue length.
<code>random-detect precedence precedence min-threshold max-threshold mark-prob-denominator</code>	Configures parameters for packets with a specific IP Precedence. The minimum threshold for IP Precedence 0 corresponds to half the maximum threshold for the interface. Repeat this command for each precedence. To configure RED, rather than WRED, use the same parameters for each precedence.

When you enable WRED with the **random-detect** interface configuration command, the parameters are set to their default values. The weight factor is 9. For all precedences, the mark probability denominator is 10, and maximum threshold is based on the output buffering capacity and the transmission speed for the interface.

The default minimum threshold depends on the precedence. The minimum threshold for IP Precedence 0 corresponds to half of the maximum threshold. The values for the remaining precedences fall between half the maximum threshold and the maximum threshold at evenly spaced intervals.



Note

The default WRED parameter values are based on the best available data. We recommend that you do not change the parameters from their default values unless you have determined that your applications will benefit from the changed values.

Monitoring WRED and DWRED

To monitor WRED and DWRED services in your network, use one or more of the following commands in EXEC mode:

Command	Purpose
<code>show queue interface-type interface-number</code>	Shows the header information of the packets inside a queue. This command does not support DWRED.
<code>show queueing interface interface-number [vc [[vpi/] vci]]</code>	Shows the WRED/DWRED statistics of a specific VC on an interface.
<code>show queueing random-detect</code>	Shows the queueing configuration for WRED/DWRED.
<code>show interfaces [type slot port-adapter port]</code>	Shows WRED/DWRED configuration on an interface.

Flow-Based WRED Configuration Task List

To configure flow-based WRED on an interface, perform the tasks in the following section:

- Configuring Flow-Based WRED

See the end of this chapter for the section “Flow-Based WRED Configuration Example.”

Configuring Flow-Based WRED

Before you can configure flow-based WRED, you must enable WRED and configure it. For information on how to configure WRED, see the section “Weighted Random Early Detection Configuration Task List” in this chapter.

To configure an interface for flow-based WRED, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	<code>random-detect flow</code>	Enables flow-based WRED.
Step 2	<code>random-detect flow average-depth-factor scaling-factor</code>	Sets the flow threshold multiplier for flow-based WRED.
Step 3	<code>random-detect flow count number</code>	Sets the maximum flow count for flow-based WRED.

WRED and DWRED Configuration Examples

The following sections provide WRED and DWRED configuration examples:

- DWRED and WRED Configuration Example
- Parameter-Setting DWRED Example
- Parameter-Setting WRED Example

DWRED and WRED Configuration Example

The following example enables WRED or DWRED with default parameter values:

```
interface Serial5/0
  description to qos1-75a
  ip address 200.200.14.250 255.255.255.252
  random-detect
```

Use the **show interfaces** command output to verify the configuration. Notice that the “Queueing strategy” report lists “random early detection (RED).”

```
router# show interfaces serial 5/0

Serial5/0 is up, line protocol is up
  Hardware is M4T
  Description: to qos1-75a
  Internet address is 200.200.14.250/30
  MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 237/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive not set
  Last input 00:00:15, output 00:00:00, output hang never
  Last clearing of "show interface" counters 00:05:08
  Input queue: 0/75/0 (size/max/drops); Total output drops: 1036
  Queueing strategy: random early detection(RED)
  5 minutes input rate 0 bits/sec, 2 packets/sec
  5 minutes output rate 119000 bits/sec, 126 packets/sec
    594 packets input, 37115 bytes, 0 no buffer
    Received 5 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    37525 packets output, 4428684 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 output buffer failures, 0 output buffers swapped out
    0 carrier transitions      DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

Use the **show queue** command output to view the current contents of the interface queue. Notice that there is only a single queue into which packets from all IP precedences are placed after dropping has taken place. The output has been truncated to show only three of the five packets.

```
router# show queue serial 5/0

Output queue for Serial5/0 is 5/0

Packet 1, linktype: ip, length: 118, flags: 0x288
  source: 190.1.3.4, destination: 190.1.2.2, id: 0x0001, ttl: 254,
  TOS: 128 prot: 17, source port 11111, destination port 22222
  data: 0x2B67 0x56CE 0x005E 0xE89A 0xCBA9 0x8765 0x4321
        0x0FED 0xCBA9 0x8765 0x4321 0x0FED 0xCBA9 0x8765

Packet 2, linktype: ip, length: 118, flags: 0x288
  source: 190.1.3.5, destination: 190.1.2.2, id: 0x0001, ttl: 254,
  TOS: 160 prot: 17, source port 11111, destination port 22222
  data: 0x2B67 0x56CE 0x005E 0xE89A 0xCBA9 0x8765 0x4321
        0x0FED 0xCBA9 0x8765 0x4321 0x0FED 0xCBA9 0x8765

Packet 3, linktype: ip, length: 118, flags: 0x280
  source: 190.1.3.6, destination: 190.1.2.2, id: 0x0001, ttl: 254,
  TOS: 192 prot: 17, source port 11111, destination port 22222
  data: 0x2B67 0x56CE 0x005E 0xE89A 0xCBA9 0x8765 0x4321
        0x0FED 0xCBA9 0x8765 0x4321 0x0FED 0xCBA9 0x8765
```

Use the **show queuing** command output to view the current settings for each of the precedences. Also notice that the default minimum thresholds are spaced evenly between half and the entire maximum threshold. Thresholds are specified in terms of packet count.

```
router# show queuing

Current random-detect configuration:
Serial5/0
  Queuing strategy:random early detection (WRED)
  Exp-weight-constant:9 (1/512)
  Mean queue depth:28

Class   Random   Tail   Minimum   Maximum   Mark
        drop   drop   threshold threshold probability
-----
0       330      0      20        40        1/10
1       267      0      22        40        1/10
2       217      0      24        40        1/10
3       156      0      26        40        1/10
4       61       0      28        40        1/10
5       6        0      31        40        1/10
6       0        0      33        40        1/10
7       0        0      35        40        1/10
rsvp    0        0      37        40        1/10
```

Parameter-Setting DWRED Example

The following example specifies the same parameters for each IP Precedence. Thus, all IP precedences receive the same treatment. Start by enabling DWRED.

```
interface FastEthernet1/0/0
ip address 200.200.14.250 255.255.255.252
random-detect
```

Next, enter the **show queueing random-detect** command to determine reasonable values to use for the precedence-specific parameters:

```
router# show queueing random-detect
```

```
Current random-detect configuration:
```

```
FastEthernet2/0/0
Queueing strategy:fifo
Packet drop strategy:VIP-based random early detection (DWRED)
Exp-weight-constant:9 (1/512)
Mean queue depth:0
Queue size:0          Maximum available buffers:6308
Output packets:5 WRED drops:0 No buffer:0
```

Class	Random drop	Tail drop	Minimum threshold	Maximum threshold	Mark probability	Output Packets
0	0	0	109	218	1/10	5
1	0	0	122	218	1/10	0
2	0	0	135	218	1/10	0
3	0	0	148	218	1/10	0
4	0	0	161	218	1/10	0
5	0	0	174	218	1/10	0
6	0	0	187	218	1/10	0
7	0	0	200	218	1/10	0

Complete the configuration by assigning the same parameter values to each precedence. Use the values obtained from the **show queueing random-detect** command output to choose reasonable parameter values.

```
interface FastEthernet1/0/0
random-detect precedence 0 100 218 10
random-detect precedence 1 100 218 10
random-detect precedence 2 100 218 10
random-detect precedence 3 100 218 10
random-detect precedence 4 100 218 10
random-detect precedence 5 100 218 10
random-detect precedence 6 100 218 10
random-detect precedence 7 100 218 10
```

Parameter-Setting WRED Example

The following example enables WRED on the interface and specifies parameters for the different IP precedences:

```
interface Hssi0/0/0
description 45Mbps to R1
ip address 10.200.14.250 255.255.255.252
random-detect
random-detect precedence 0 32 256 100
random-detect precedence 1 64 256 100
random-detect precedence 2 96 256 100
random-detect precedence 3 120 256 100
random-detect precedence 4 140 256 100
random-detect precedence 5 170 256 100
random-detect precedence 6 290 256 100
random-detect precedence 7 210 256 100
random-detect precedence rsvp 230 256 100
```

Flow-Based WRED Configuration Example

The following example enables WRED on the Serial1 interface and configures flow-based WRED. The **random-detect** interface configuration command is used to enable WRED. Once WRED is enabled, the **random-detect flow** command is used to enable flow-based WRED.

After flow-based WRED is turned on, the **random-detect flow average-depth-factor** command is used to set the scaling factor to 8 and the **random-detect flow count** command is used to set the flow count to 16. The scaling factor is used to scale the number of buffers available per flow and to determine the number of packets allowed in the output queue for each active flow.

```
configure terminal
interface Serial1
  random-detect
  random-detect flow
  random-detect flow average-depth-factor 8
  random-detect flow count 16
end
```

The following part of the example shows a sample configuration file after the above flow-based WRED commands are issued:

```
Router# more system:running-config
Building configuration...
Current configuration:
!
version 12.0
service timestamps debug datetime msec localtime
service timestamps log uptime
no service password-encryption
service tcp-small-servers
!
no logging console
enable password lab
!
clock timezone PST -8
clock summer-time PDT recurring
ip subnet-zero
no ip domain-lookup
!
interface Ethernet0
  no ip address
  no ip directed-broadcast
  no ip mroute-cache
  shutdown
!
interface Serial0
  no ip address
  no ip directed-broadcast
  no ip mroute-cache
  no keepalive
  shutdown
!
```

```
interface Serial1
 ip address 190.1.2.1 255.255.255.0
 no ip directed-broadcast
 load-interval 30
 no keepalive
 random-detect
 random-detect flow
 random-detect flow count 16
 random-detect flow average-depth-factor 8
!
router igrp 8
 network 190.1.0.0
!
ip classless
no ip http server
!
line con 0
 transport input none
line 1 16
 transport input all
line aux 0
 transport input all
line vty 0 4
 password lab
 login
!
end
```

