Configuring Weighted Random Early Detection

This chapter describes the tasks for configuring Weighted Random Early Detection (WRED) and Distributed WRED (DWRED) on a router. For a complete description of the commands mentioned in this chapter, refer to the *Quality of Service Solutions Command Reference*; the commands are listed alphabetically within that guide. To locate documentation of specific commands, use the command reference, master index, or search online.

The WRED feature is supported on these 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 RSP interface card

The DWRED feature is only supported on Cisco 7000 series routers with a Route Switch Processor-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 Transmission Control Protocol/Internet Protocol (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 TCP's congestion control mechanism. 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 the edge. WRED uses these precedences to determine how it treats different types of traffic.

When a packet arrives, the following events occur:

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

See the section "About Weighted Random Early Detection" 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:

- Enable WRED
- Change WRED Parameters
- Monitor WRED and DWRED

The section "WRED and DWRED Configuration Examples" later in this chapter provides examples of configuring WRED.

Enable WRED

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

Command	Purpose
random-detect	Enable 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.

Change WRED Parameters

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

Command	Purpose
random-detect exponential-weighting-constant *exponent*	Configure the weight factor used in calculating the average queue length.
random-detect precedence *precedence min-threshold max-threshold mark-prob-denominator*	Configure 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** 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. Cisco recommends that you do not change the parameters from their default values unless you have determined that your applications will benefit from the changed values.

Monitor WRED and DWRED

To monitor WRED services in your network, use any of the following commands in EXEC mode:

Command	Purpose
show interfaces [*interface-type interface-number*] **random-detect**	Show information about WRED on a VIP-based interface.
show queue *interface-type interface-number*	Show WRED configuration and statistics for a particular interface.
show queueing [**red**]	Show the queueing configuration for WRED.
show interfaces [*type slot* | *port-adapter* | *port*]	Show DWRED statistics for Cisco 7500 series routers.

Note The **show interfaces random-detect** command can only be used on a Versatile Interface Processor (VIP)-based interface.

WRED and DWRED Configuration Examples

The following sections provide these 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:

```
router(config)# interface Serial5/0
router(config-if)# description to qos1-75a
router(config-if)# ip address 200.200.14.250 255.255.255.252
router(config-if)# random-detect
```

Use the **show interfaces** command output to verify the configuration and view the default settings for the different precedences. Notice that the "Queueing strategy" report lists "random early detection (RED)." 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 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)
    mean queue depth: 28
    drops: class  random   tail     min-th   max-th   mark-prob
           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
  30 second input rate 0 bits/sec, 2 packets/sec
  30 second 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.

```
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

Packet 4, linktype: ip, length: 118, flags: 0x280
  source: 190.1.3.7, destination: 190.1.2.2, id: 0x0001, ttl: 254,
  TOS: 224 prot: 17, source port 11111, destination port 22222
    data: 0x2B67 0x56CE 0x005E 0xE89A 0xCBA9 0x8765 0x4321
          0x0FED 0xCBA9 0x8765 0x4321 0x0FED 0xCBA9 0x8765

Packet 5, linktype: ip, length: 118, flags: 0x280
  source: 190.1.3.8, destination: 190.1.2.2, id: 0x0001, ttl: 254,
  TOS: 0 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 queueing** command output to view the current settings for each of the precedences.

```
router# show queueing

Current RED queue configuration:
  Interface: Serial5/0   Exp-weight-constant: 9
    Class    Min-th    Max-th   Mark-prob
    0        20        40       1/10
    1        22        40       1/10
    2        24        40       1/10
    3        26        40       1/10
    4        28        40       1/10
    5        31        40       1/10
    6        33        40       1/10
    7        35        40       1/10
    rsvp     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.

```
router(config)# interface FastEthernet1/0/0
router(config-if)# ip address 200.200.14.250 255.255.255.252
router(config-if)# random-detect
```

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

```
router# show interfaces random-detect

 FastEthernet1/0/0 queue size 0
        packets output 29692, drops 0
 WRED: queue average 0
       weight 1/512
     Precedence 0: 109 min threshold, 218 max threshold, 1/10 mark weight
       1 packets output, drops: 0 random, 0 threshold
     Precedence 1: 122 min threshold, 218 max threshold, 1/10 mark weight
        (no traffic)
     Precedence 2: 135 min threshold, 218 max threshold, 1/10 mark weight
       14845 packets output, drops: 0 random, 0 threshold
     Precedence 3: 148 min threshold, 218 max threshold, 1/10 mark weight
        (no traffic)
     Precedence 4: 161 min threshold, 218 max threshold, 1/10 mark weight
        (no traffic)
     Precedence 5: 174 min threshold, 218 max threshold, 1/10 mark weight
        (no traffic)
     Precedence 6: 187 min threshold, 218 max threshold, 1/10 mark weight
       14846 packets output, drops: 0 random, 0 threshold
     Precedence 7: 200 min threshold, 218 max threshold, 1/10 mark weight
        (no traffic)
```

Note The **show interfaces random-detect** command can only be used on a Versatile Interface Processor (VIP)-based interface.

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

```
router(config)# interface FastEthernet1/0/0
router(config-if)# random-detect precedence 0 100 218 10
router(config-if)# random-detect precedence 1 100 218 10
router(config-if)# random-detect precedence 2 100 218 10
router(config-if)# random-detect precedence 3 100 218 10
router(config-if)# random-detect precedence 4 100 218 10
router(config-if)# random-detect precedence 5 100 218 10
router(config-if)# random-detect precedence 6 100 218 10
router(config-if)# 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:

```
router(config)# interface Hssi0/0/0
router(config-if)# description 45Mbps to R1
router(config-if)# ip address 200.200.14.250 255.255.255.252
router(config-if)# random-detect
router(config-if)# random-detect precedence 0 32 256 100
router(config-if)# random-detect precedence 1 64 256 100
router(config-if)# random-detect precedence 2 96 256 100
router(config-if)# random-detect precedence 3 120 256 100
router(config-if)# random-detect precedence 4 140 256 100
router(config-if)# random-detect precedence 5 170 256 100
router(config-if)# random-detect precedence 6 290 256 100
router(config-if)# random-detect precedence 7 210 256 100
router(config-if)# random-detect precedence rsvp 230 256 100
```