

# Distributed Weighted Random Early Detection

---

## Feature Summary

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. Assuming the packet source is using TCP, it will decrease its transmission rate until all the packets reach their destination, indicating that the congestion is cleared.

Weighted RED (WRED) generally drops packets selectively based on IP precedence. Packets with a higher IP precedence are less likely to be dropped than packets with a lower precedence. Thus, higher priority traffic is delivered with a higher probability than lower priority traffic. However, you can also configure WRED to ignore IP precedence when making drop decisions so that non-weighted RED behavior is achieved.

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. Edge routers assign IP precedences to packets as they enter the network. WRED uses these precedences to determine how it treats different types of traffic.

The Distributed WRED (DWRED) feature uses the VIP rather than the RSP to perform the queuing; therefore, it requires a Cisco 7500 series router or Cisco 7000 series router with RSP7000.

The Release 11.1 CC version of DWRED differs from the previously existing RSP-based version of WRED. This document describes only the Release 11.1 CC version of DWRED. Refer to the Cisco IOS Release 11.3 *Configuration Fundamentals Configuration Guide* and *Configuration Fundamentals Command Reference* for information on the RSP version of WRED.

## Benefits

When RED is not configured, output buffers fill during periods of congestion. When the buffers are full, tail drop occurs; all additional packets are dropped. Since the packets are dropped all at once, global synchronization of TCP hosts can occur as multiple TCP hosts reduce their transmission rates. The congestion clears, and the TCP hosts increase their transmission rates, resulting in waves of congestion followed by periods where the transmission link is not fully used.

RED reduces the chances of tail drop by selectively dropping packets when the output interface begins to show signs of congestion. By dropping some packets early rather than waiting until the buffer is full, RED avoids dropping large numbers of packets at once and minimizes the chances of global synchronization. Thus, RED allows the transmission line to be used fully at all times.

In addition, RED statistically drops more packets from large users than small. Therefore, traffic sources that generate the most traffic are more likely to be slowed down than traffic sources that generate little traffic.

WRED provides separate thresholds and weights for different IP precedences, allowing you to provide different qualities of service for different traffic. Standard traffic may be dropped more frequently than premium traffic during periods of congestion.

## List of Terms

**DWRED**—Distributed WRED, an implementation of WRED. DWRED performs all of the functions of WRED, but all of the processing takes place on the VIP.

**RED**—Random Early Detection.

**VIP**—Versatile Interface Processor.

**WRED**—Weighted RED.

## Restrictions

DWRED has the following restrictions:

- 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.
- WRED treats non-IP traffic as precedence 0, the lowest precedence. Therefore, non-IP traffic will, in general, be more likely to be dropped than IP traffic.
- You cannot configure DWRED on the same interface as RSP-based custom queuing, priority queuing, or weighted fair queuing (WFQ). However, you can configure both DWRED and DWFQ on the same interface.
- DWRED is only available on a per-interface basis. You cannot configure DWRED on a subinterface.
- DWRED is not supported with the ATM encapsulations AAL5-MUX and AAL5-NLPID.
- DWRED is not supported on Fast EtherChannel or Tunnel interfaces.

## Platforms

This feature is supported on the following routers with a VIP2-40 card or better:

- Cisco 7000 series with RSP7000
- Cisco 7500 series

A VIP2-50 card is strongly recommended when the aggregate line rate of the port adapters on the VIP is greater than DS-3. A VIP2-50 card is required for OC-3 rates.

## Prerequisites

In order to use DWRED, Distributed Cisco Express Forwarding switching must be enabled on the interface. Refer to the Cisco Express Forwarding feature documentation for configuration information.

## Supported MIBs and RFCs

This feature supports the CISCO-WRED-MIB.

For descriptions of supported MIBs and how to use MIBs, see Cisco's MIB website on CCO at <http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>.

No RFCs are supported by this feature.

## Functional Description

When a packet arrives, the following events occur:

- The average queue size is calculated. See the “Average Queue Size” section for details.
- If the average is less than the minimum queue threshold, the arriving packet is queued.
- If the average is between the minimum queue threshold and the maximum threshold, the packet is either dropped or queued, depending on the packet drop probability. See the “Packet-Drop Probability” section for details.
- If the average queue size is greater than the maximum threshold, the packet is automatically dropped.

## Average Queue Size

The average queue size is based on the previous average and the current size of the queue. The formula is

$$\text{average} = (\text{old\_average} * (1 - 1/2^n)) + (\text{current\_queue\_size} * 1/2^n)$$

where  $n$  is the exponential weight factor, a user-configurable value.

---

**Note** Cisco recommends using the default value for the exponential weight factor. Change this value from the default only if you have determined that your situation would benefit from using a different value.

---

For high values of  $n$ , 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 drastic swings in size. The WRED process will be 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 will accommodate temporary bursts in traffic.

If the value of  $n$  gets too high, WRED will not react to congestion. Packets will be transmitted or dropped as if WRED were not in effect.

For low values of  $n$ , 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 will stop dropping packets.

If the value of  $n$  gets too low, WRED will overreact to temporary traffic bursts and drop traffic unnecessarily.

## Packet-Drop Probability

The packet drop probability is based on the minimum threshold, maximum threshold, and mark probability denominator.

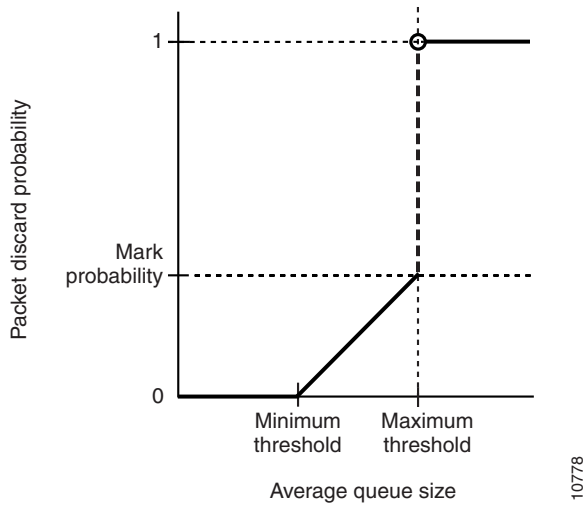
When the average queue depth is above the minimum threshold, RED starts dropping packets. The rate of packet drop increases linearly as the average queue size increases until the average queue size reaches the maximum threshold.

The mark probability denominator is the fraction of packets dropped when the average queue depth is at the maximum threshold. For example, if the denominator is 512, one out of every 512 packets is dropped when the average queue is at the maximum threshold.

When the average queue size is above the maximum threshold, all packets are dropped.

Figure 1 summarizes the packet drop probability.

**Figure 1** WRED Packet Drop Probability



The minimum threshold value should be set high enough to maximize the link utilization. If the minimum threshold is too low, packets may be dropped unnecessarily, and the transmission link will not be fully used.

The difference between the maximum threshold and the minimum threshold should be large enough to avoid global synchronization. If the difference is too small, many packets may be dropped at once, resulting in global synchronization.

## Configuration Tasks

To configure DWRED on an interface, perform the following tasks. The first task is required; the second is optional.

- Enable WRED
- Change WRED Parameters

## Enable WRED

To enable WRED, perform the following task in interface configuration mode:

Task	Command
Enable WRED.	<b>random-detect</b>

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

Use the **show interfaces random-detect EXEC** command to view the current WRED configuration and status.

## Change WRED Parameters

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 one half of the maximum threshold. The values for the remaining precedences fall between one 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 would benefit from the changed values.

---

To change WRED parameters, perform one of the following tasks in interface configuration mode:

Task	Command
Configure the weight factor used in calculating the average queue length.	<b>random-detect exponential-weighting-constant</b> <i>exponent</i>
Configure parameters for packets with a specific IP precedence. Repeat this command for each precedence. To configure RED, rather than WRED, use the same parameters for each precedence.	<b>random-detect precedence</b> <i>precedence min-threshold max-threshold mark-prob-denominator</i>

Use the **show interfaces random-detect EXEC** command to view the current WRED configuration and status.

## Configuration Examples

This section contains the following examples:

- Default DWRED Example
- DRED Example

## Default DWRED Example

The following example enables DWRED with the default parameter values:

```
interface Hssi0/0/0
  description 45Mbps to R1
  ip address 200.200.14.250 255.255.255.252
  random-detect
```

Use the **show interfaces random-detect** command to verify the configuration and view the default settings for the different precedences.

```
Router# show interfaces random-detect

Hssi0/0/0 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)
```

The following is sample **show interfaces** command output for this configuration. Notice that the Packet Drop Strategy is listed as “VIP-based weighted RED.”

```
Router# show interfaces hssi0/0/0

Hssi0/0/0 is up, line protocol is up
  Hardware is cyBus HSSI
  Description: 45Mbps to R1
  Internet address is 200.200.14.250/30
  MTU 4470 bytes, BW 45045 Kbit, DLY 200 usec, rely 255/255, load 1/255
  Encapsulation HDLC, loopback not set, keepalive set (10 sec)
  Last input 00:00:02, output 00:00:03, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Packet Drop strategy: VIP-based weighted RED
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  1976 packets input, 131263 bytes, 0 no buffer
  Received 1577 broadcasts, 0 runts, 0 giants
  0 parity
  4 input errors, 4 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  1939 packets output, 130910 bytes, 0 underruns
  0 output errors, 0 applique, 3 interface resets
  0 output buffers copied, 0 interrupts, 0 failures
```

## DRED Example

The following example configures DRED, rather than DWRED, by specifying the same parameters for each IP precedence. Thus, all IP precedences receive the same treatment. Start by enabling DRED:

```
interface FastEthernet1/0/0
 ip address 200.200.14.250 255.255.255.252
 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)
```

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 pick 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
```

## Command Reference

This section documents new or modified commands. All other commands used with this feature are documented in the Cisco IOS Release 11.1 command references and the Cisco Express Forwarding feature documentation.

- **random-detect**
- **random-detect exponential-weighting-constant**
- **random-detect precedence**

- **show interfaces**
- **show interfaces random-detect**

## random-detect

To enable Weighted Random Early Detection (WRED) on an interface, use the **random-detect** interface configuration command. The **no** form of this command disables WRED.

**random-detect**  
**no random-detect**

### Syntax Description

This command has no arguments or keywords.

### Default

Disabled

### Command Mode

Interface configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 11.1 CC.

WRED is a congestion avoidance mechanism that slows traffic by randomly dropping packets when there is congestion. WRED is only useful with protocols like TCP, which respond to dropped packets by decreasing the transmission rate.

The router automatically determines parameters to use in the WRED calculations. If you wish to change these parameters, use the **random-detect exponential-weighting-constant** or **random-detect precedence** command.

### Example

The following example configures WRED on the HSSI 0/0/0 interface:

```
interface Hssi0/0/0
 random-detect
```

### Related Commands

**random-detect exponential-weighting-constant**  
**random-detect precedence**  
**show interfaces random-detect**

## random-detect exponential-weighting-constant

To configure the exponential weight factor for the average queue size calculation, use the **random-detect exponential-weighting-constant** interface configuration command. The **no** form of this command returns the value to the default.

```
random-detect exponential-weighting-constant exponent  
no random-detect exponential-weighting-constant
```

### Syntax Description

*exponent* Exponent from 1 to 16 used in the average queue size calculation. The default is 9.

### Default

The weight factor is 9.

### Command Mode

Interface configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 11.1 CC.

Use this command to change the exponent used in the average queue size calculation. The average queue size is based on the previous average and the current size of the queue. The formula is

$$\text{average} = (\text{old\_average} * (1 - 1/2^n)) + (\text{current\_queue\_size} * 1/2^n)$$

where *n* is the exponential weight factor specified in this command. Thus, the higher the factor, the more dependent the average is on the previous average.

---

**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 would benefit from the changed values.

---

For high values of *n*, 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 drastic swings in size. The WRED process will be 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 will accommodate temporary bursts in traffic.

If the value of *n* gets too high, WRED will not react to congestion. Packets will be transmitted or dropped as if WRED were not in effect.

For low values of *n*, 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 will stop dropping packets.

If the value of *n* gets too low, WRED will overreact to temporary traffic bursts and drop traffic unnecessarily.

## Example

The following example configures WRED on an interface with weight factor of 10:

```
interface Hssi0/0/0
  description 45Mbps to R1
  ip address 200.200.14.250 255.255.255.252
  random-detect
  random-detect exponential-weighting-constant 10
```

## Related Commands

**random-detect**

**random-detect precedence**

**show interfaces random-detect**

## random-detect precedence

To configure WRED parameters for a particular IP precedence, use the **random-detect precedence** interface configuration command. The **no** form of this command returns the values to the default for the precedence.

```
random-detect precedence precedence min-threshold max-threshold mark-prob-denominator
no random-detect precedence precedence min-threshold max-threshold
mark-prob-denominator
```

### Syntax Description

<i>precedence</i>	IP precedence number. The value ranges from 0 to 7.
<i>min-threshold</i>	Minimum threshold in number of packets. The value ranges from 1 to 4096. When the average queue length reaches this number, RED begins to drop packets with the specified IP precedence.
<i>max-threshold</i>	Maximum threshold in number of packets. The value ranges from <i>min-threshold</i> to 4096. When the average queue length exceeds this number, WRED drops all packets with the specified IP precedence.
<i>mark-prob-denominator</i>	Denominator for the fraction of packets dropped when the average queue depth is <i>max-threshold</i> . For example, if the denominator is 512, one out of every 512 packets is dropped when the average queue is at the <i>max-threshold</i> . The value is from 1 to 65536. The default is 10; one out of every ten packets are dropped at the <i>max-threshold</i> .

### Default

For all precedences, the *mark-prob-denominator* is 10, and the *max-threshold* is based on the output buffering capacity and the transmission speed for the interface.

The default *min-threshold* depends on the precedence. The *min-threshold* for IP precedence 0 corresponds to one half of the *max-threshold*. The values for the remaining precedences fall between one half the *max-threshold* and the *max-threshold* at evenly spaced intervals. Table 1 lists the default minimum value for each precedence.

**Table 1** Default WRED Minimum Threshold Values

IP Precedence	Minimum Threshold Value (Fraction of Output Buffer)
0	9/18 (1/2)
1	10/18 (5/9)
2	11/18
3	12/18 (2/3)
4	13/18
5	14/18 (7/9)
6	15/18 (5/6)

**Table 1** Default WRED Minimum Threshold Values

IP Precedence	Minimum Threshold Value (Fraction of Output Buffer)
7	16/18 (8/9)

## Command Mode

Interface configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 11.1 CC.

When you configure the **random-detect** command on an interface, packets are given preferential treatment based on the IP precedence of the packet. Use the **random-detect precedence** command to adjust the treatment for different IP precedences.

If you want WRED to ignore the precedence when determining which packets to drop, enter this command with the same parameters for each precedence. Remember to use reasonable values for the minimum and maximum thresholds.

---

**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 would benefit from the changed values.

---

## 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 200.200.14.250 255.255.255.252
  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 3 128 256 100
  random-detect precedence 4 160 256 100
  random-detect precedence 5 192 256 100
  random-detect precedence 6 224 256 100
  random-detect precedence 7 256 256 100

```

## Related Commands

**random-detect**  
**random-detect exponential-weighting-constant**  
**show interfaces random-detect**

## show interfaces

Use the **show interfaces** EXEC command to display statistics for all interfaces. The resulting output varies depending on the network for which an interface has been configured.

**show interfaces** [*type slot/port-adapter/port*] (for ports on VIP cards in the Cisco 7500 series routers)

### Syntax Description

<i>type</i>	(Optional) Interface type.
<i>slot</i>	(Optional) Refer to the appropriate hardware manual for slot information.
<i>port-adapter</i>	Refer to the appropriate hardware manual for information about port adapter compatibility.
<i>port</i>	(Optional) Refer to the appropriate hardware manual for port information.

### Command Mode

EXEC

### Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

The **show interfaces** command displays statistics for the network interfaces.

The output of the **show interfaces** command varied based in the interface type and configuration. This document describes the output related to the DWRED feature. Refer to the Cisco IOS Release 11.1 command references for a complete description of the entire output.

### Sample Display for DWRED

The following is sample output from the **show interfaces** command when DWRED is enabled on an interface. Notice that the packet drop strategy is listed as “VIP-based weighted RED.”

```
Router# show interfaces hssi0/0/0

Hssi0/0/0 is up, line protocol is up
  Hardware is cyBus HSSI
  Description: 45Mbps to R1
  Internet address is 200.200.14.250/30
  MTU 4470 bytes, BW 45045 Kbit, DLY 200 usec, rely 255/255, load 1/255
  Encapsulation HDLC, loopback not set, keepalive set (10 sec)
  Last input 00:00:02, output 00:00:03, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Packet Drop strategy: VIP-based weighted RED
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  1976 packets input, 131263 bytes, 0 no buffer
  Received 1577 broadcasts, 0 runts, 0 giants
  0 parity
```

---

```
4 input errors, 4 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
1939 packets output, 130910 bytes, 0 underruns
0 output errors, 0 applique, 3 interface resets
0 output buffers copied, 0 interrupts, 0 failures
```

## show interfaces random-detect

To display information about WRED for an interface, use the **show interfaces random-detect EXEC** command.

**show interfaces** [*interface*] **random-detect**

### Syntax Description

*interface* (Optional) Type and number of the interface.

### Command Mode

EXEC

### Usage Guidelines

This command first appeared in Cisco IOS Release 11.1 CC.

### Sample Display

The following is sample output from the **show interfaces random-detect** command:

```
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)
```

Table 2 describes the fields shown in this display.

**Table 2 Show Interfaces Random-Detect Field Descriptions**

Field	Description
queue size	Output queue size for this interface.
packets output	Number of packets transmitted out this interface.
drops	Number of packets dropped.
queue average	Average queue length.

**Table 2 Show Interfaces Random-Detect Field Descriptions (Continued)**

<b>Field</b>	<b>Description</b>
weight	Weighting factor used to determine the average queue size.
Precedence	WRED parameters for this precedence.
min threshold	Minimum threshold for this precedence.
max threshold	Maximum length of the queue. When the average queue is this long, any additional packets will be dropped.
mark weight	Probability of a packet being dropped if the average queue is at the maximum threshold.
packets output	Number of packets with this precedence that have been transmitted.
random	Number of packets dropped randomly through the WRED process.
threshold	Number of packets dropped automatically because the average queue was at the maximum threshold length.
(no traffic)	No packets with this precedence.

**Related Commands****random-detect****random-detect exponential-weighting-constant****random-detect precedence**

