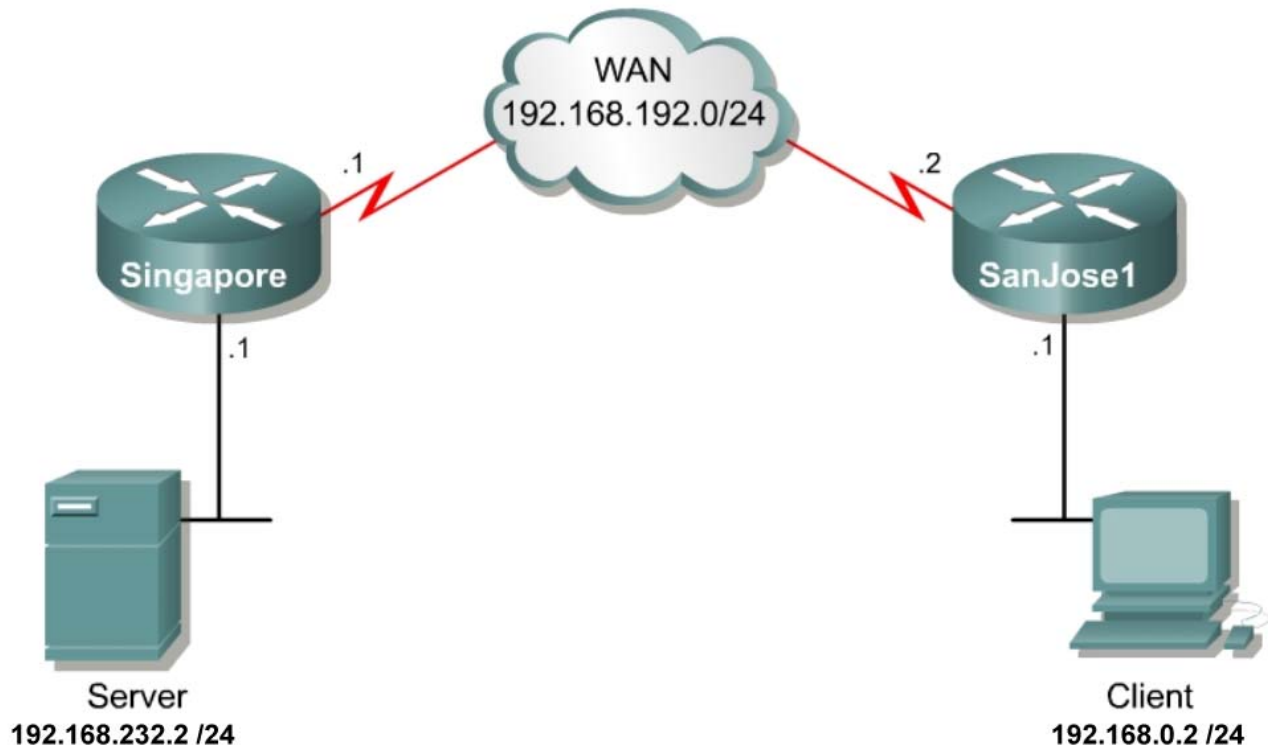


## Lab 8.1.10.5 Configuring WRED on an Interface



### Objective

The Cisco implementation of Random Early Detection (RED) is called Weighted Random Early Detection (WRED). WRED differs from other congestion-avoidance techniques because it attempts to anticipate and avoid congestion instead of controlling congestion after it occurs. WRED uses TCP congestion control and tries to control the average queue size by notifying end hosts when they should temporarily stop sending packets. WRED will randomly drop packets before periods of high congestion to instruct the packet source to decrease its transmission rate. If the packet source is using TCP, WRED will instruct it to decrease its transmission rate until all the packets reach their destination and the congestion is cleared.

WRED drops more packets from large users than small users. Therefore, sources that generate a lot of traffic are more likely to be slowed down than sources that generate limited amounts of traffic.

In this lab, WRED will be configured in its simplest form. The default IP Precedence bits in a packet will be used to determine the weighting.

### Scenario

The performance of the WAN link between Singapore and SanJose1 is not optimal. During a quiet period, large files are copied across the link to test the throughput of the link. This reveals that the throughput is considerably less than the 128-kbps bandwidth suggests. Network analysis indicates that

the 128-kbps bottleneck causes the egress queue on the Singapore router to overflow when a large file is requested by SanJose1. This causes TCP/IP to drastically reduce its transmission speed and reduce the unnecessary retransmission of data. This problem can be solved by using WRED.

### Step 1

Build the physical topology as shown in the diagram. Before beginning a lab, the configurations on all the routers should be cleared and then reloaded or power cycled to reset their default configurations. Delete the **vlan.dat** and startup configuration files on the switches before reloading them.

### Step 2

Configure the network as shown in the diagram and enable EIGRP with an autonomous system number of 100 as the routing protocol. Confirm connectivity by pinging between the hosts.

```
Router(config)#hostname Singapore
Singapore(config)#interface fastethernet 0/0
Singapore(config-if)#ip address 192.168.232.1 255.255.255.0
Singapore(config-if)#no shutdown
Singapore(config-if)#interface serial 0/0
Singapore(config-if)#ip address 192.168.192.1 255.255.255.0
Singapore(config-if)#clock rate 128000
Singapore(config-if)#no shutdown
Singapore(config-if)#router eigrp 100
Singapore(config-router)#network 192.168.192.0
Singapore(config-router)#network 192.168.232.0

Router(config)#hostname SanJose1
SanJose1(config)#interface fastethernet 0/0
SanJose1(config-if)#ip address 192.168.0.1 255.255.255.0
SanJose1(config-if)#no shutdown
SanJose1(config-if)#interface serial 0/0
SanJose1(config-if)#ip address 192.168.192.2 255.255.255.0
SanJose1(config-if)#clock rate 128000
SanJose1(config-if)#no shutdown
SanJose1(config-if)#router eigrp 100
SanJose1(config-router)#network 192.168.0.0
SanJose1(config-router)#network 192.168.192.0
```

### Step 3

Use the interface **random-detect** command to enable WRED on the exit queues of each router.

```
Singapore(config)#interface serial 0/0
Singapore(config-if)#random-detect

SanJose1(config)#interface serial 0/0
SanJose1(config-if)#random-detect
```

No other commands or parameters need to be specified to configure WRED on the interface with the default parameter values.

### Before Turning On RED

```
Singapore#show interfaces serial 0/0
Serial0/0 is up, line protocol is up
  Hardware is PowerQUICC Serial
  Internet address is 192.168.192.1/24
  MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
```

```

Encapsulation HDLC, loopback not set
Keepalive set (10 sec)
Last input 00:00:04, output 00:00:00, output hang never
Last clearing of "show interface" counters 01:25:35
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue :0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
  23892 packets input, 2705156 bytes, 0 no buffer
    Received 604 broadcasts, 0 runts, 0 giants, 0 throttles
    3 input errors, 0 CRC, 3 frame, 0 overrun, 0 ignored, 0 abort
  36829 packets output, 50930429 bytes, 0 underruns
    0 output errors, 0 collisions, 7 interface resets
    0 output buffer failures, 0 output buffers swapped out
    15 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up

```

Singapore#

SanJose1#**show interfaces serial 0/0**

```

Serial0/0 is up, line protocol is up
  Hardware is PowerQUICC Serial
  Internet address is 192.168.192.2/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  Last input 00:00:01, output 00:00:04, output hang never
  Last clearing of "show interface" counters 00:03:34
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue :0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    24 packets input, 2712 bytes, 0 no buffer
      Received 10 broadcasts, 0 runts, 0 giants, 0 throttles
      0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    37 packets output, 2203 bytes, 0 underruns
      0 output errors, 0 collisions, 2 interface resets
      0 output buffer failures, 0 output buffers swapped out
      31 carrier transitions
  DCD=up DSR=up DTR=up RTS=up CTS=up

```

## After Turning On RED

Singapore#**show interfaces serial 0/0**

```

Serial0/0 is up, line protocol is down
  Hardware is PowerQUICC Serial
  Internet address is 192.168.192.1/24
  MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  Last input 00:01:37, output 00:01:38, output hang never
  Last clearing of "show interface" counters 00:07:31
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: random early detection(RED)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    94 packets input, 6503 bytes, 0 no buffer
      Received 34 broadcasts, 0 runts, 0 giants, 0 throttles
      0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort

```

```
126 packets output, 9325 bytes, 0 underruns
0 output errors, 0 collisions, 7 interface resets
0 output buffer failures, 0 output buffers swapped out
12 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up
```

#### **SanJose1#show interfaces serial 0/0**

```
Serial0/0 is up, line protocol is up
Hardware is PowerQUICC Serial
Internet address is 192.168.192.2/24
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation HDLC, loopback not set
Keepalive set (10 sec)
Last input 00:00:02, output 00:00:00, output hang never
Last clearing of "show interface" counters 00:04:58
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: random early detection (RED)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
 52 packets input, 4628 bytes, 0 no buffer
Received 20 broadcasts, 0 runts, 0 giants, 0 throttles
 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
67 packets output, 4209 bytes, 0 underruns
 0 output errors, 0 collisions, 3 interface resets
 0 output buffer failures, 0 output buffers swapped out
31 carrier transitions

DCD=up DSR=up DTR=up RTS=up CTS=up
```

## **Step 4**

Use the **show interfaces** command to verify the configuration and operation of WRED.

#### **Singapore#show interfaces serial 0/0**

```
Serial0/0 is up, line protocol is up
Hardware is PowerQUICC Serial
Internet address is 192.168.192.1/24
MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation HDLC, loopback not set
Keepalive set (10 sec)
Last input 00:00:00, output 00:00:03, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 797
Queueing strategy: random early detection (RED)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
 72 packets input, 5041 bytes, 0 no buffer
Received 56 broadcasts, 0 runts, 0 giants, 0 throttles
 3 input errors, 0 CRC, 3 frame, 0 overrun, 0 ignored, 0 abort
151 packets output, 7317 bytes, 0 underruns
 0 output errors, 0 collisions, 3 interface resets
 0 output buffer failures, 0 output buffers swapped out
11 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up
```

The thresholds that WRED is currently using to determine packet drop can be viewed by using the **show queueing random-detect** command.

```
Singapore#show queueing random-detect
Current random-detect configuration:
```

```

Serial0/0
Queueing strategy: random early detection (WRED)
Exp-weight-constant: 9 (1/512)
Mean queue depth: 0

```

class	Random drop pkts/bytes	Tail drop pkts/bytes	Minimum thresh	Maximum thresh	Mark prob
0	0/0	0/0	20	40	1/10
1	0/0	0/0	22	40	1/10
2	0/0	0/0	24	40	1/10
3	0/0	0/0	26	40	1/10
4	0/0	0/0	28	40	1/10
5	0/0	0/0	31	40	1/10
6	0/0	0/0	33	40	1/10
7	0/0	0/0	35	40	1/10
rsvp	0/0	0/0	37	40	1/10

```

SanJose1#show queueing random-detect
Current random-detect configuration:
Serial0/0
Queueing strategy: random early detection (WRED)
Exp-weight-constant: 9 (1/512)
Mean queue depth: 0

```

class	Random drop pkts/bytes	Tail drop pkts/bytes	Minimum thresh	Maximum thresh	Mark prob
0	0/0	0/0	20	40	1/10
1	0/0	0/0	22	40	1/10
2	0/0	0/0	24	40	1/10
3	0/0	0/0	26	40	1/10
4	0/0	0/0	28	40	1/10
5	0/0	0/0	31	40	1/10
6	0/0	0/0	33	40	1/10
7	0/0	0/0	35	40	1/10
rsvp	0/0	0/0	37	40	1/10

## Step 5

Use the **random-detect** command to modify the default thresholds that WRED uses to determine packet drop. This command configures the weight factor that is used to calculate the average queue length. Set the exponential-weighting-constant factor to **5**.

```

Singapore(config)#interface serial 0/0
Singapore(config-if)#random-detect exponential-weighting-constant 5

```

```

SanJose1(config)#interface serial 0/0
SanJose1(config-if)#random-detect exponential-weighting-constant 5

```

The following command configures parameters for packets with a specific IP Precedence. Set the precedence to **5**, the minimum threshold to **100**, maximum threshold to **200** and the probability denominator to **1000**.

```

Singapore(config-if)#random-detect precedence 5 100 200 1000
SanJose1(config-if)#random-detect precedence 5 100 200 1000

```

Experiment with these commands and observe any changes with the **show queueing random-detect** command.

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 instead of WRED use the same parameters for each precedence.

**Note:** The default WRED parameter values should not be changed unless the applications will benefit from the changed values.

### Change To Weighting Constant

```
Singapore(config)#interface serial 0/0
Singapore(config-if)#random-detect exponential-weighting-constant 5
```

```
Singapore#show queueing random-detect
```

Current random-detect configuration:

```
Serial0/0
Queueing strategy: random early detection (WRED)
Exp-weight-constant: 5 (1/32)
Mean queue depth: 0
```

class	Random drop pkts/bytes	Tail drop pkts/bytes	Minimum thresh	Maximum thresh	Mark
prob					
0	0/0	0/0	20	40	1/10
1	0/0	0/0	22	40	1/10
2	0/0	0/0	24	40	1/10
3	0/0	0/0	26	40	1/10
4	0/0	0/0	28	40	1/10
5	0/0	0/0	31	40	1/10
6	0/0	0/0	33	40	1/10
7	0/0	0/0	35	40	1/10
rsvp	0/0	0/0	37	40	1/10

### Change To Precedence

```
Singapore(config)#interface serial 0/0
Singapore(config-if)#random-detect precedence ?
<0-7> IP precedence
rsvp rsvp traffic
```

```
Singapore(config-if)#random-detect precedence 5 ?
<1-4096> minimum threshold (number of packets)
```

```
Singapore(config-if)#random-detect precedence 5 100 ?
<1-4096> maximum threshold (number of packets)
```

```
Singapore(config-if)#random-detect precedence 5 100 200 ?
<1-65535> mark probability denominator
<cr>
```

```
Singapore(config-if)#random-detect precedence 5 100 200 1000
```

```
Singapore#show queueing random-detect
```

Serial0/0

Queueing strategy: random early detection (WRED)

Exp-weight-constant: 5 (1/32)

Mean queue depth: 0

class	Random drop pkts/bytes	Tail drop pkts/bytes	Minimum thresh	Maximum thresh	Mark
prob					
0	0/0	0/0	20	40	1/10
1	0/0	0/0	22	40	1/10
2	0/0	0/0	24	40	1/10
3	0/0	0/0	26	40	1/10
4	0/0	0/0	28	40	1/10
5	0/0	0/0	100	200	1/1000
6	0/0	0/0	33	40	1/10
7	0/0	0/0	35	40	1/10
rsvp	0/0	0/0	37	40	1/10