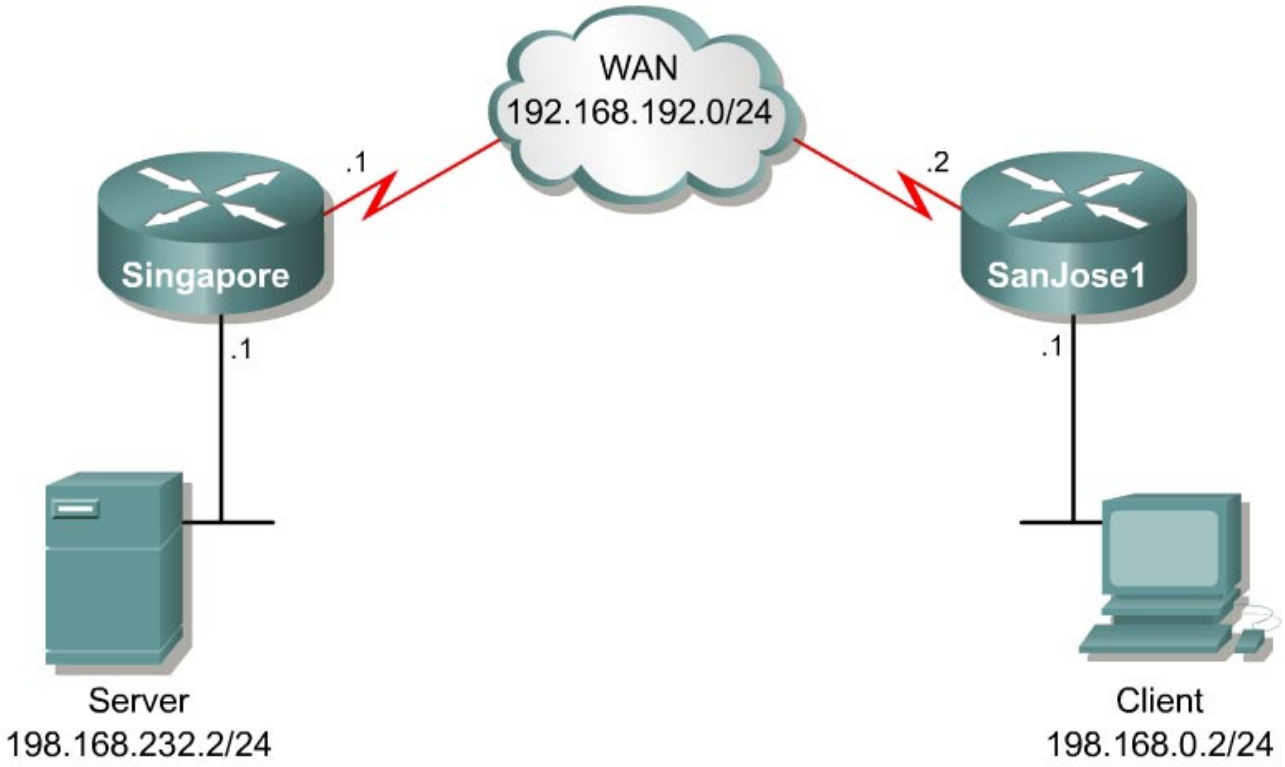
Lab 8.9.4 Weighted Fair Queuing



Objective

In this lab, weighted fair queuing (WFQ) is configured and optimized.

Scenario

The network engineer for the International Travel Agency (ITA) is responsible for WAN connectivity. As ITA has grown, traffic has increased on the WAN link. The network technicians have recently reported unreliable Telnet access between San Jose and regional sites. The network engineer discovers that average WAN link utilization between Singapore and SanJose1 is near saturation. While investigating ways to increase bandwidth, the engineer optimizes WFQ as a temporary solution to meet the needs of all users.

Step 1

Build the physical topology as shown in the diagram.

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.

Step 3

View the default queuing strategy on the Singapore WAN link. The default queuing may be WFQ or first-in, first-out (FIFO), depending on the router IOS version and the bandwidth of the interface.

```
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: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/3/32 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 96 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     325 packets input, 21083 bytes, 0 no buffer
     Received 105 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     341 packets output, 23164 bytes, 0 underruns
     0 output errors, 0 collisions, 11 interface resets
     0 output buffer failures, 0 output buffers swapped out
     4 carrier transitions
     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

Issue the **show queue serial 0/0** command to view the queuing configuration on an interface. This command is not supported with FIFO queuing.

```
Singapore#show queue serial 0/0
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/3/32 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 96 kilobits/sec
```

If the serial interface of a router is using FIFO, WFQ can be configured by issuing the **fair-queue** command.

```
Singapore(config)#interface serial 0/0
Singapore(config-if)#fair-queue
```

On a congested FIFO interface, a low-volume, interactive session like Telnet is subject to intolerable delays while high-bandwidth applications like FTP monopolize available bandwidth. WFQ identifies and gives equal access to a variety of application protocols. WFQ can be thought of as statistically multiplexing all applications. Low-volume sessions are given the necessary bandwidth while high-volume sessions share the remainder. However, there is no guarantee of reserved bandwidth. Interfaces that are overwhelmed with traffic may be forced to drop packets.

Each communication session between hosts creates a flow. The router understands a flow as a record of attributes such as source and destination addresses, port numbers, and the inbound interface. The router can then compare subsequent packets to existing flows. After packets are identified as belonging to a certain session, they are buffered accordingly.

To give each session equal router resources, a default maximum of 64 messages or packets can be buffered by any one session. The congestion threshold must be increased to 128 packets for the ITA network. This allows the router to buffer more packets per session, but decreases the number of sessions serviced at a time.

Queuing will not solve this problem because additional bandwidth is required. Queuing may increase performance problems because it demands additional router CPU cycles and forces the router to apply queuing logic to each packet. Therefore, queuing is either a temporary fix or a solution for times when interactive sessions fail due to latency or dropped packets.

> **Note:** Queuing is only active when congestion exists. Congestion exists when any interface has one or more packets buffered in its queue. If all interfaces are clear of buffered packets, queuing is idle.

Step 4

Use the following command syntax to increase the congestion threshold value to 128 packets on both router WAN links.

```
Singapore(config)#interface serial 0/0
Singapore(config-if)#fair-queue 128
SanJose1(config)#interface serial 0/0
SanJose1(config-if)#fair-queue 128
```

Review the WFQ parameters on Serial 0/0.

```
SanJose1#show interface 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 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/128/0 (size/max total/threshold/drops)
     Conversations  0/1/32 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 96 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     481 packets input, 30955 bytes, 0 no buffer
     Received 160 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     498 packets output, 33108 bytes, 0 underruns
     0 output errors, 0 collisions, 12 interface resets
     0 output buffer failures, 0 output buffers swapped out
     6 carrier transitions
     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up

Singapore#show queue serial 0/0
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/128/0 (size/max total/threshold/drops)
     Conversations  0/1/32 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 96 kilobits/sec
```

When the WAN link is saturated and queuing is activated, each session will be able to buffer up to 128 packets before dropping any incoming packets. After the 128-packet discard threshold limit is

reached for a particular flow, no packets will be buffered until the queue for the flow drops to 25 percent of the discard threshold. In this case, the queue must reach 32 packets, which is 25 percent of 128. If packets are dropped, upper-layer protocols such as TCP may compensate and retransmit undelivered packets. This will successfully change the behavior of WFQ.

Step 5

Configure Windows file sharing on the Singapore host to see the effect of WFQ. Copy a large file from the Singapore host to the SanJose1 host. The file should be large enough to take five to ten minutes to copy over the 128-kbps WAN link.

Step 6

Initiate a Telnet session between Singapore and SanJose1. The keystrokes should be echoed back in a timely fashion when WFQ is being used.

Step 7

Use the **`no fair-queue`** command to turn off WFQ on each serial interface. This will result in FIFO queuing on the WAN link.

Step 8

Initiate another Telnet session between Singapore and SanJose1. The keystrokes should be echoed back after some latency and may be erratic. This makes it difficult to correct typing mistakes by using the **Backspace** key.

