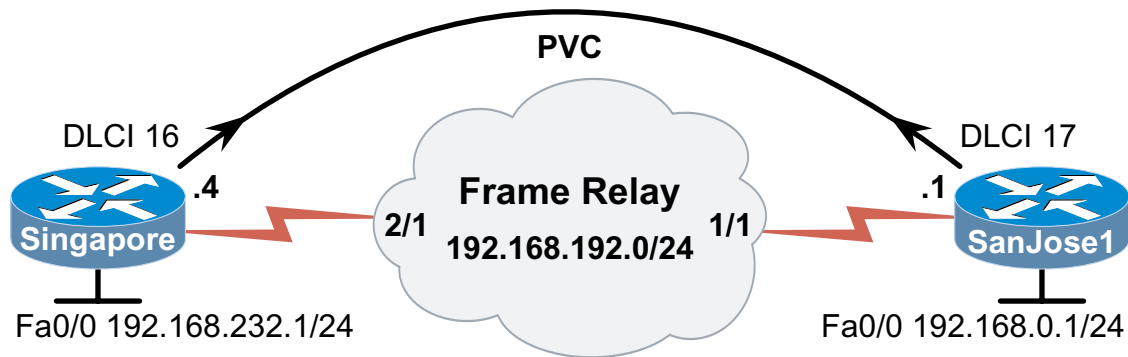


Lab 10.4.3: Custom Queuing



Objective

In this lab, you are to implement Custom Queuing.

Scenario

As the network engineer for the International Travel Agency, you are responsible for WAN connectivity. As ITA has grown, traffic has increased on your Frame Relay network. ITA partners, including airlines and hotel chains, have begun to complain about the speed of your web site. Also, file uploads, downloads, Web sessions, and database synchronizations time out on a regular basis. Though additional bandwidth has been ordered from the service provider, you have been directed to ensure partners and customers are not affected by bandwidth problems in the interim. If necessary, internal users will suffer during peak traffic times. After monitoring the network, you decide to allocate bandwidth, giving Voice over IP (VoIP) 40 percent, FTP 20 percent, and HTTP 20 percent, with all other traffic sharing the remaining 20 percent.

Step 1

Build the network as shown in the diagram. If you are using the Atlas 550 as a WAN emulator, be sure to use the ports as indicated in the diagram. Before beginning this lab, it is recommended that you reload each router after erasing its startup configuration. This prevents you from having problems caused by residual configurations.

Step 2

Configure the network as shown in the diagram with Frame Relay, Ethernet and IGRP with autonomous system number 234.

Step 3

Custom queuing attempts to remedy some of the potentially devastating effects other queuing methods can have on low-priority traffic. Recall that some applications may never get serviced when priority queuing is used.

Custom queuing will 'multiplex' bandwidth to assure that all queues get the pre-defined proportion of bandwidth. Seventeen different queues exist, of which 1-16 can be associated with traffic types. Queue 0 is the system queue, used by essential traffic, such as interface keepalives. Queue 0 is always serviced first. The remaining queues are serviced in a round-robin fashion, with each queue completely emptied in turn. Proportions of bandwidth are expressed as the maximum number of bytes a queue can hold. Queues that can buffer a relatively large number of bytes will have the opportunity to transmit more data per unit of time than queues buffering a relatively small number of bytes. Since custom queuing allocates bandwidth based on the maximum number of bytes rather than on time, bandwidth is not wasted. As soon as a queue is empty, the router moves to the next queue.

Define traffic for each queue with queue lists, as shown here:

```
Singapore(config)#queue-list 1 protocol ip 1 tcp ftp
Singapore(config)#queue-list 1 protocol ip 2 tcp www
Singapore(config)#queue-list 1 default 3
```

Because VoIP uses TCP and UDP, you need to reference an extended access list that will define VoIP traffic. Create the access list first.

```
Singapore(config)#access-list 111 permit tcp any any eq 1720
Singapore(config)#access-list 111 permit udp any any range 16380
16480
```

Now reference the access list from the queue list.

```
Singapore(config)#queue-list 1 protocol ip 4 list 111
```

The queue number within the queue list is only a unique identifier and not an indication of any hierarchy or preference. What matters is the relative number of bytes a queue can hold.

Step 4

When the Frame Relay link is saturated, assuming all queues are full, bandwidth will be divided relative to queue size. Again, you want to allocate bandwidth to applications based on the following percentages: VoIP 40 percent, FTP 20 percent, HTTP 20 percent, and all other traffic sharing the remaining 20 percent.

The default queue size is 1500 bytes. If 1500 bytes are assumed to equal 20 percent of bandwidth, then doubling the VoIP queue will yield 40 percent. To figure out what percentage of bandwidth a queue is receiving in times of congestion, divide the queue size by the sum of all queues. For example, 3000 bytes/7500 bytes = 0.4 or 40 percent.

Define queue limits on **both** routers, as shown here:

```
Singapore(config)#queue-list 1 queue 1 byte-count 1500
Singapore(config)#queue-list 1 queue 2 byte-count 1500
Singapore(config)#queue-list 1 queue 3 byte-count 1500
Singapore(config)#queue-list 1 queue 4 byte-count 3000
```

Step 5

Now that the traffic is defined and associated with a queue, and the maximum bytes are defined for each queue, configure **both** Frame Relay interfaces to use custom queuing.

```
Singapore(config)#interface serial 0/0
Singapore(config-if)#custom-queue-list 1
```

Step 6

Verify your configuration with the **show queueing custom** and **show interface** command, as shown here:

Note: Remember to use the incorrect spelling, "queueing."

```
SanJose1#show queueing custom
Current custom queue configuration:
```

| List | Queue | Args |
|------|-------|--------------------------|
| 1 | 3 | default |
| 1 | 1 | protocol ip tcp port ftp |
| 1 | 2 | protocol ip tcp port www |
| 1 | 4 | protocol ip list 111 |
| 1 | 4 | byte-count 3000 |

```
Singapore#show interfaces serial 0/0
Serial0/0 is up, line protocol is up
  Hardware is PowerQUICC Serial
  Internet address is 192.168.192.4/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 5/255, rxload 5/255
  Encapsulation FRAME-RELAY, loopback not set
  Keepalive set (10 sec)
  LMI enq sent 1917, LMI stat recvd 1919, LMI upd recvd 0, DTE
    LMI up
  LMI enq recvd 0, LMI stat sent 0, LMI upd sent 0
  LMI DLCI 0 LMI type is ANSI Annex D frame relay DTE
  Broadcast queue 0/64, broadcasts sent/dropped 794/0, interface
    broadcasts 240
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters 05:19:42
  Input queue: 1/75/0 (size/max/drops); Total output drops: 0
  Queueing strategy: custom-list 1
  Output queues: (queue #: size/max/drops)
    0: 0/20/0 1: 0/20/0 2: 0/20/0 3: 0/20/0 4: 0/20/0
    5: 0/20/0 6: 0/20/0 7: 0/20/0 8: 0/20/0 9: 0/20/0
    10: 0/20/0 11: 0/20/0 12: 0/20/0 13: 0/20/0 14: 0/20/0
    15: 0/20/0 16: 0/20/0
  5 minute input rate 31000 bits/sec, 4 packets/sec
  5 minute output rate 31000 bits/sec, 4 packets/sec
    6383 packets input, 1956197 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    6534 packets output, 1960423 bytes, 0 underruns
    0 output errors, 0 collisions, 7 interface resets
    0 output buffer failures, 0 output buffers swapped out
    0 carrier transitions
  DCD=up DSR=up DTR=up RTS=up CTS=up
```

The command output shows statistics for each queue. Queueing will become active when Frame Relay becomes saturated and thresholds have to be enforced.

You have successfully implemented custom queueing.