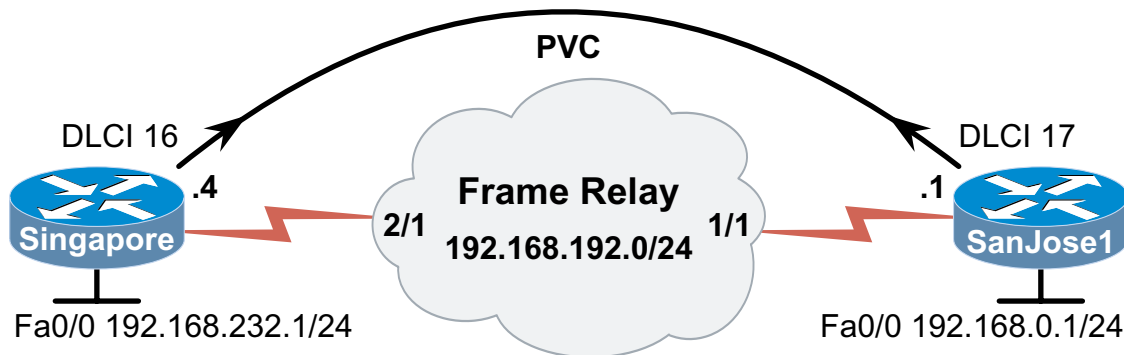


## Lab 10.4.2 : Priority Queuing



### Objective

In this lab, you are to implement priority 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. Additional bandwidth has not yet been provisioned for your Frame Relay permanent virtual circuits (PVC). At peak traffic times during the workday, the network becomes sluggish and unreliable. Immediate action needs to be taken until more bandwidth can be acquired. You decide to implement priority queuing.

### 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 will prevent you from having problems caused by residual configurations.

### Step 2

Priority queuing allows you to define types of traffic and associate each type of traffic with one of four queues: high, medium, normal, and low. Priority queuing will transmit all packets in the high queue first. When the high queue is empty, the medium queue is emptied, and the high queue is checked again for buffered packets. Only after all higher priority queues are emptied, lower queues are checked and emptied. It is possible that low queues will hold packets that never get transmitted, causing their associated sessions to time out.

There are three steps to configuring priority queuing. First, traffic is defined using access lists. Second, access lists are associated with queues using priority lists. Finally, priority lists are associated with interfaces.

You have defined three types of traffic you wish to assign priority to: TCP, UDP, and ICMP. All other traffic will be placed in the default queue, which is the normal queue.

Define three types of traffic with access lists on **both** routers, as shown here:

```
SanJose1(config)#access-list 111 permit tcp any any
SanJose1(config)#access-list 112 permit icmp any any
SanJose1(config)#access-list 113 permit udp any any

Singapore(config)#access-list 121 permit tcp any any
Singapore(config)#access-list 122 permit icmp any any
Singapore(config)#access-list 123 permit udp any any
```

### Step 3

Priority lists now need to be created, and each access list must be associated with a queue. As with access lists, packets are compared to a priority list one line at a time, starting with the first line of the priority list. The first priority list line that matches the packet sends it to the associated queue. The following priority list statements are compared to the packet. For efficiency, structure your priority list so that a majority of packets are matched to the first few lines.

Give TCP traffic high priority, ICMP medium priority, and UDP low priority on **both** routers as shown here:

```
Singapore(config)#priority-list 3 protocol ip high list 121
Singapore(config)#priority-list 3 protocol ip medium list 122
Singapore(config)#priority-list 3 protocol ip low list 123

SanJose1(config)#priority-list 5 protocol ip high list 111
SanJose1(config)#priority-list 5 protocol ip medium list 112
SanJose1(config)#priority-list 5 protocol ip high list 113
```

Most of the configuration is complete, but queuing will never be applied until the priority list is associated with an interface. Apply the priority lists to the serial interfaces, as shown here:

```
SanJose1(config)#interface serial 0/0
SanJose1(config-if)#priority-group 5

Singapore(config)#interface serial 0/0
Singapore(config-if)#priority-group 3
```

Traffic outbound interface Serial 0/0 will be compared with the **priority list** statements.

Confirm configuration with the **show queueing priority** command. Note that this command misspells the word "queuing." When using the Cisco IOS, you must spell this word incorrectly as "queueing" in order for the command to be recognized, as in the example here:

```
SanJose1#show queueing priority
```

Current priority queue configuration:

List	Queue	Args	
5	high	protocol ip	list 111
5	medium	protocol ip	list 112
5	high	protocol ip	list 113

```
Singapore#show queueing priority
Current priority queue configuration:
```

List	Queue	Args	
3	high	protocol ip	list 121
3	medium	protocol ip	list 122
3	low	protocol ip	list 123

To view the quantity of packets in each queue per interface, issue **show queue** on each router, as shown here:

```
Singapore#show queueing interface serial 0/0
Interface Serial0/0 queueing strategy: priority

Output queue utilization (queue/count)
    high/274 medium/560 normal/34 low/0

SanJose1#show queueing interface serial 0/0
Interface Serial0/0 queueing strategy: priority

Output queue utilization (queue/count)
    high/306 medium/560 normal/88 low/0
```

As the link becomes saturated and some traffic has to wait, you have ensured that TCP traffic will always be transmitted first, medium traffic should get through, and normal and low traffic might get through. Since more packets will likely be buffered in the lower-priority queues, they are larger by default. The default queue sizes are high 20, medium 40, normal 60, and low 80.

Recall that the default queue for all undefined traffic is the normal queue. You have decided to elevate the priority of undefined traffic by configuring the default queue as having medium priority on **both** routers:

```
SanJose1(config)#priority-list 5 default medium
```

Increase the number of records (packets) for the medium queue since that is now the default queue for all undefined traffic.

```
SanJose1(config)#priority-list 5 queue-limit 20 90 60 80
```

Confirm queuing configuration, as shown here:

```
Singapore#show queueing priority
Current priority queue configuration:

List  Queue  Args
3     high  protocol ip          list 121
3     medium protocol ip          list 122
3     low   protocol ip          list 123
3     medium limit 90
```

**Note:** Only non-default queue sizes appear in the output.

You have successfully implemented and optimized priority queuing.