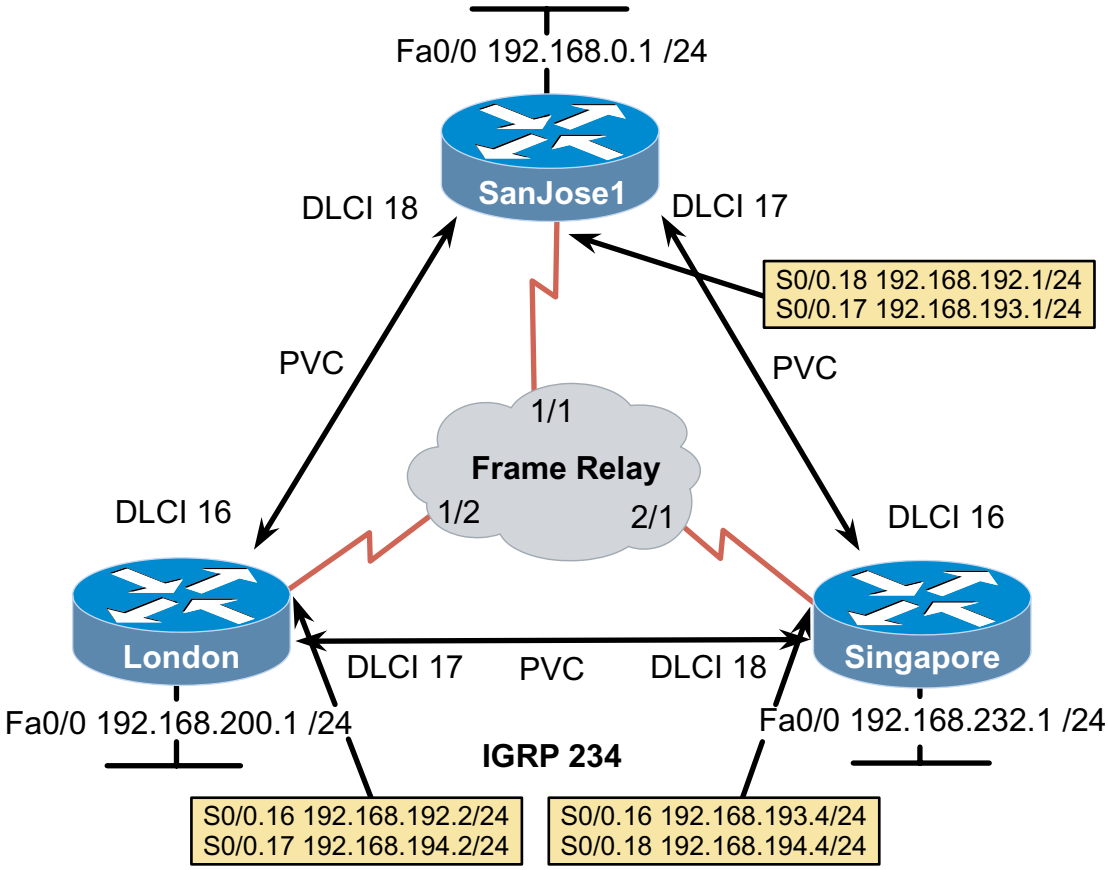
Configuring Full-Mesh Frame Relay With Subinterfaces



Objective

Configure three routers with Frame Relay in a full mesh using subinterfaces.

Scenario

While looking forward to fine-tuning traffic flow among regional sites connected to the International Travel Agency WAN, you realize greater control could be implemented if there was an individual physical leased line between each site. Unfortunately, your managers will not approve the cost of an additional leased line.

However, you can configure separate logical connections between the sites by using subinterfaces. Each router will have one physical interface connected to the Frame Relay switch. The single physical interface (s0/0) with have two logical subinterfaces configured, one for each regional site. Physical redundancy is achieved in the carrier's Frame Relay cloud, enabled by multiple PVCs per physical interface on the routers. Although your provider charges for each additional PVC, the fee is very small, especially when compared with the cost of a leased line.

Step 1

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. After you prepare the equipment, proceed with Step 2.

Build the network according to the above diagram. This lab assumes use of an Adtran Atlas 550 as the Frame Relay cloud. You may use other WAN emulators or a router as a Frame Relay switch. If you are using the Atlas 550, be sure to connect each router's serial interface to the correct port on the Atlas (labeled in the diagram) using a V.35 cable.

Step 2

Configure each router's FastEthernet interface and IGRP (AS 234) as shown in the diagram. Configure serial interfaces to use Frame Relay with subinterfaces. Enter the following commands on SanJose1:

```
SanJose1(config)#interface serial 0/0
SanJose1(config-if)#no ip address
SanJose1(config-if)#encapsulation frame-relay
SanJose1(config-if)#frame-relay lmi-type ansi
SanJose1(config-if)#No shutdown
```

The **`no ip address`** command is only necessary if the interface has already been configured with an IP address.

Issue the following commands to configure a point-to-point subinterface using DLCI 17. Note that the subinterface number has been chosen so that you can easily determine which PVC each subinterface is using (for example, subinterface 0.17 uses DLCI 17). Although this practice is not required, it is very common.

Specify the subinterfaces with the keyword **point-to-point** as shown below, and not **multipoint**. Some versions of the IOS allow you to create a subinterface without specifying either type, which results in a multipoint subinterface (the default).

```
SanJose1(config)#interface s0/0.17 point-to-point
SanJose1(config-subif)#ip address 192.168.193.1 255.255.255.0
SanJose1(config-subif)#frame-relay interface-dlci 17
```

The **`frame-relay interface-dlci`** command is used to specify which DLCI is used by each subinterface. Point-to-point subinterfaces can each use only one DLCI. If you do not specify a subinterface for a DLCI, it will be associated with the major interface (s0/0 in this case). Now issue the commands needed to configure the subinterface that will use DLCI 18, as shown:

```
SanJose1(config)#interface s0/0.18 point-to-point
SanJose1(config-subif)#ip address 192.168.192.1 255.255.255.0
SanJose1(config-subif)#frame-relay interface-dlci 18
```

Configure London and Singapore according to the diagram. Use the command syntax in this step as a guide.

Step 3

Verify PVC status by issuing the **`show frame-relay pvc`** command on SanJose1.

```
SanJose1#show frame-relay pvc

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

                Active     Inactive      Deleted       Static
  Local            2            1            0            0
  Switched         0            0            0            0
  Unused           0            0            0            0
```

```
DLCI = 16, DLCI USAGE = LOCAL, PVC STATUS = INACTIVE, INTERFACE =
       Serial0/0

  input pkts 0             output pkts 0            in bytes 0
  out bytes 0              dropped pkts 0           in FECN pkts 0
  in BECN pkts 0           out FECN pkts 0         out BECN pkts 0
  in DE pkts 0             out DE pkts 0
  out bcast pkts 0          out bcast bytes 0
  pvc create time 00:48:21, last time pvc status changed 00:48:21

DLCI = 17, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE =
       Serial0/0.17

  input pkts 167           output pkts 179          in bytes 21552
  out bytes 19871          dropped pkts 5           in FECN pkts 0
  in BECN pkts 0           out FECN pkts 0         out BECN pkts 0
  in DE pkts 0             out DE pkts 0
  out bcast pkts 59         out bcast bytes 10890
  pvc create time 00:48:34, last time pvc status changed 00:29:13

DLCI = 18, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE =
       Serial0/0.18

  input pkts 156           output pkts 135          in bytes 27757
  out bytes 18758          dropped pkts 5           in FECN pkts 0
  in BECN pkts 0           out FECN pkts 0         out BECN pkts 0
  in DE pkts 0             out DE pkts 0
  out bcast pkts 55         out bcast bytes 10718
  pvc create time 00:48:35, last time pvc status changed 00:07:14
```

The output of this command shows the PVC status according to the router's perspective. The different states are ACTIVE, INACTIVE, and DELETED. ACTIVE is a successful end-to-end (DTE to DTE) circuit. INACTIVE is a successful connection to the switch (DTE to DCE) without a DTE detected on the other end of the PVC. This can occur if the router dynamically learns of a DLCI not intended for its network, due to a residual or incorrect configuration on the Frame Relay switch. The DELETED state is when the DTE is configured for a DLCI the switch does not recognize as valid for that interface.

If you see INACTIVE DLCIs that you do not intend to use, check with your service provider regarding their status. In this configuration, SanJose1 detects DLCI 16, which is not used in this lab. To keep your router from dynamically creating a PVC with whatever DTE is on the other end, disable Inverse-ARP for DLCI 16, as shown here:

```
SanJose1(config)#interface serial 0/0
SanJose1(config-if)#no frame-relay inverse-arp ip 16
```

Issue the **show frame-relay map** command on SanJose1 to determine if Frame Relay has mapped the appropriate DLCIs to the correct IP address.

```
SanJose1#show frame-relay map
Serial0/0.17 (up): point-to-point dlci, dlci 17(0x11,0x410),
       broadcast
          status defined, active
Serial0/0.18 (up): point-to-point dlci, dlci 18(0x12,0x420),
       broadcast
          status defined, active
```

1. According to the output, are DLCI 17 and DLCI 18 mapped to next-hop IP addresses?

With point-to-point subinterfaces, there is no mapping between a local DLCI and a next-hop address. Each subinterface is treated as if it were a separate physical interface and each Frame Relay PVC contains only two hosts. There is no need to explicitly identify the next-hop. Point-to-point subinterfaces can also be implemented using IP unnumbered.

Step 4

Make sure IGRP is working by issuing the **show ip route** command on SanJose1.

```
SanJose1#show ip route

Gateway of last resort is not set

C    192.168.192.0/24 is directly connected, Serial0/0.18
C    192.168.193.0/24 is directly connected, Serial0/0.17
I    192.168.194.0/24 [100/82125] via 192.168.192.2, 00:00:00,
       Serial0/0.18
                      [100/82125] via 192.168.193.4, 00:01:25,
       Serial0/0.17
I    192.168.200.0/24 [100/80135] via 192.168.192.2, 00:00:00,
       Serial0/0.18
I    192.168.232.0/24 [100/80135] via 192.168.193.4, 00:01:25,
       Serial0/0.17
C    192.168.0.0/24 is directly connected, FastEthernet0/0
```

Verify round-trip connectivity with an extended **ping** from SanJose1 (sourced from 192.168.0.1) to London (192.168.200.1). This **ping** should be successful. Troubleshoot as necessary.