

# Building Large Frame Relay Networks with OSPF

By Chuck Semeria and Tom Maufer

## Introduction

This paper describes recommendations for building large Frame Relay networks using OSPF. NETBuilder Software Release 8.2 provides support for the soon-to-be-standardized IETF method for running OSPF over a non-broadcast multiple-access (NBMA) network. The term "non-broadcast network" describes a network supporting many (more than two) routers, but having no native broadcast/multicast capability. Neighboring routers are maintained on these networks using OSPF's Hello Protocol. However, due to the lack of a broadcast/multicast capability, some configuration information may be necessary to aid in the discovery of neighbors. Frame Relay and X.25 networks are examples of non-broadcast networks.

Frame Relay is a service for which 3Com offers its customers many different solutions. Boundary routing is a low-end solution whose scalability is limited to 75 virtual ports. Even if the virtual port limit was increased, Boundary Routing is not designed to support the construction of extremely large Frame Relay networks. Access routing has the benefit of no virtual port limitations, and the network can be as large as the routing protocols can support. Large Frame Relay networks built using access routing require the deployment of OSPF to obtain the required scalability.

Note that Software Release 8.2 allows a single Frame Relay interface to support both access routing and Boundary Routing using OSPF. This permits full access routing employing OSPF to large sites and Boundary Routing employing Virtual Ports to smaller sites.

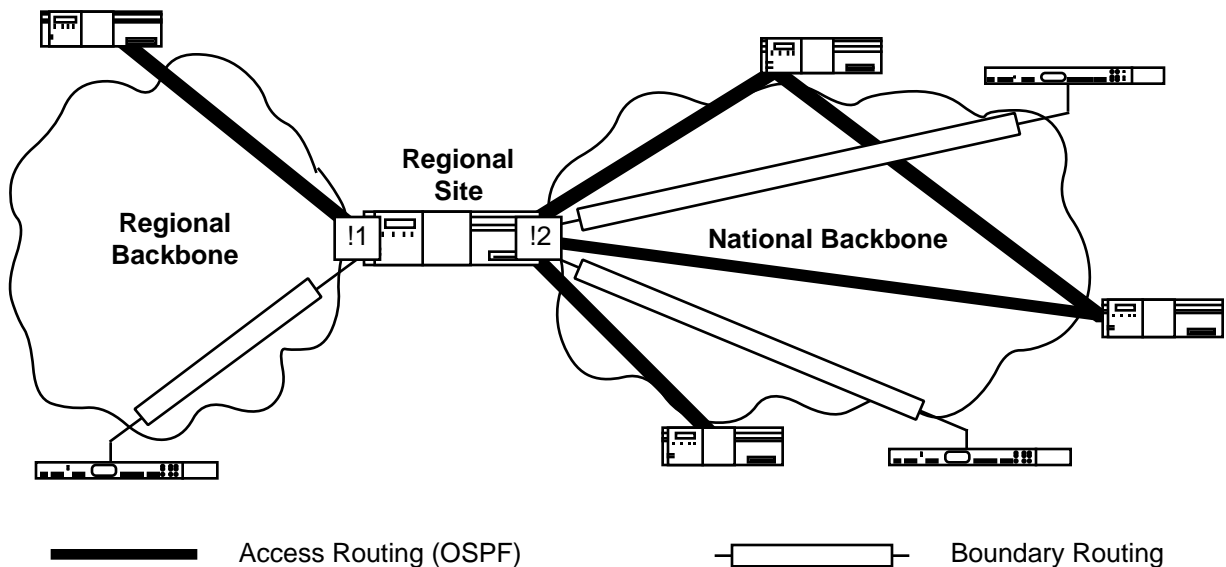


Figure 1: Access Routing (OSPF) and Boundary Routing over Frame Relay

Figure 1 illustrates a topology where a combination of access routing and Boundary Routing are deployed across Frame Relay networks. On !2 of the regional site router, three IP subnetworks must be used: one for each of the two Boundary Routing virtual ports, and one for the nonmesh OSPF access network.

## NETBuilder Software Release 8.2

For NETBuilder Software Release 8.2, the syntax for the -OSPF CONTROL parameter has been changed from:

```
SETDefault !<port> -OSPF CONTROL = ([Enable | Disable],  
                                     [P2Pmode | NBMAmode])
```

to:

```
SETDefault !<port> -OSPF CONTROL = ([Enable | Disable],  
                                     [DynamicNbr | NoDynamicNbr],  
                                     [FullMesh | NonMesh])
```

### DynamicNbr | NoDynamicNbr

This option determines how OSPF obtains its Frame Relay neighbor list. If DynamicNbr is selected, OSPF will attempt to establish an adjacency with all routers whose IP-to-DLCI mappings are contained in the Address Translation Table. The entries are placed in the Address Translation Table via Inverse ARP (InARP) or through static configuration using the ADD -IP ADDRESS <IP address> <media address> syntax. If NoDynamicNbr is selected, OSPF will attempt to become adjacent with the list of static neighbors defined using the ADD !<port> -OSPF Neighbor <IP address> syntax.

### FullMesh | NonMesh

#### FullMesh Option

Selecting the FullMesh option requires that all of the neighboring routers be fully connected with each other with a complete mesh of virtual circuits. When this option is selected, OSPF emulates the operation of OSPF over a broadcast network. This means that a Designated Router (DR) and Backup Designated Router (BDR) are elected for the non-broadcast network, and the Designated Router originates a Network-LSA for the non-broadcast network.

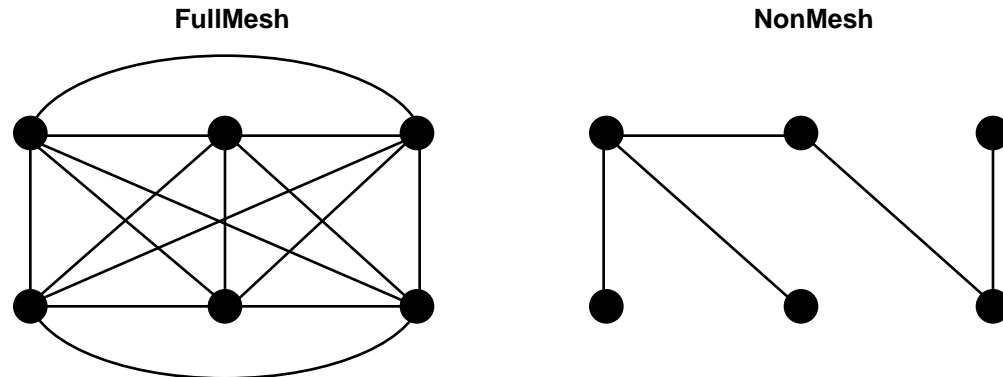


Figure 2: FullMesh vs. NonMesh Topologies

It should be emphasized that FullMesh topologies are impracticable for large Frame Relay networks because they do not scale:

- \* A large, fully-meshed Frame Relay network requires a separate virtual circuit for each link between neighbors. A FullMesh topology requires  $[n(n-1)]/2$  virtual circuits, where  $n$  is the number of routers connected to the Frame Relay network. This means that a FullMesh topology consisting of 5 routers requires 10 virtual circuits while a FullMesh topology consisting of 7 routers requires 21 virtual circuits - a 110% increase in the number of virtual circuits that must be purchased from the service provider!
- \* The lack of a network broadcast/multicast capability requires that a router unicast standard protocol broadcast/multicast frames to each virtual circuit for that Frame Relay interface. This occurs whenever OSPF Hellos, LSD Updates, etc. must be transmitted. As the number of virtual circuits increases, router broadcast/multicast replication can consume a considerable amount of the Frame Relay network's bandwidth.

For these reasons, FullMesh topologies should only be deployed in small Frame Relay networks. Economics dictate that large Frame Relay networks be built using NonMesh topologies.

### NonMesh Option

Selecting the NonMesh option causes OSPF to run in the soon-to-be-standardized Point-to-MultiPoint mode. If this option is selected, OSPF treats each router-to-router connection as if it were a separate point-to-point link. Point-to-MultiPoint networks have been designed to work simply and naturally when faced with partial mesh connectivity. This means that:

- \* A DR and BDR are not elected for the Frame Relay network.
- \* A Network-LSA is not originated for the Frame Relay network.
- \* Point-to-point adjacencies are established between each set of directly communicating neighboring routers. A point-to-point adjacency in steady state will be either Full or Down, but never 2-Way.

- \* Separate Hello packets are sent to each attached neighbor every HelloTime interval. To improve scalability in Software Release 9.0, the Hello transmissions will be sent out evenly over time so that there will be a steady stream of Hellos rather than a large burst every 10 seconds. Note that Software Release 8.3 allows you use the SETDefault !<port> -OSPF DemandInterface = Enable syntax to eliminate the transmission of Hellos and periodic LSAs, but real topology changes will still be flooded over the network.
- \* When originating a Router-LSA, the Point-to-MultiPoint interface is reported as a collection of "point-to-point links" to each of the interface's adjacent neighbors.

NonMesh is ideal for building large Frame Relay networks since it does not require that all routers to be able to communicate directly.

## Example Configuration

The following figure shows the topology that will be used for the configuration example:

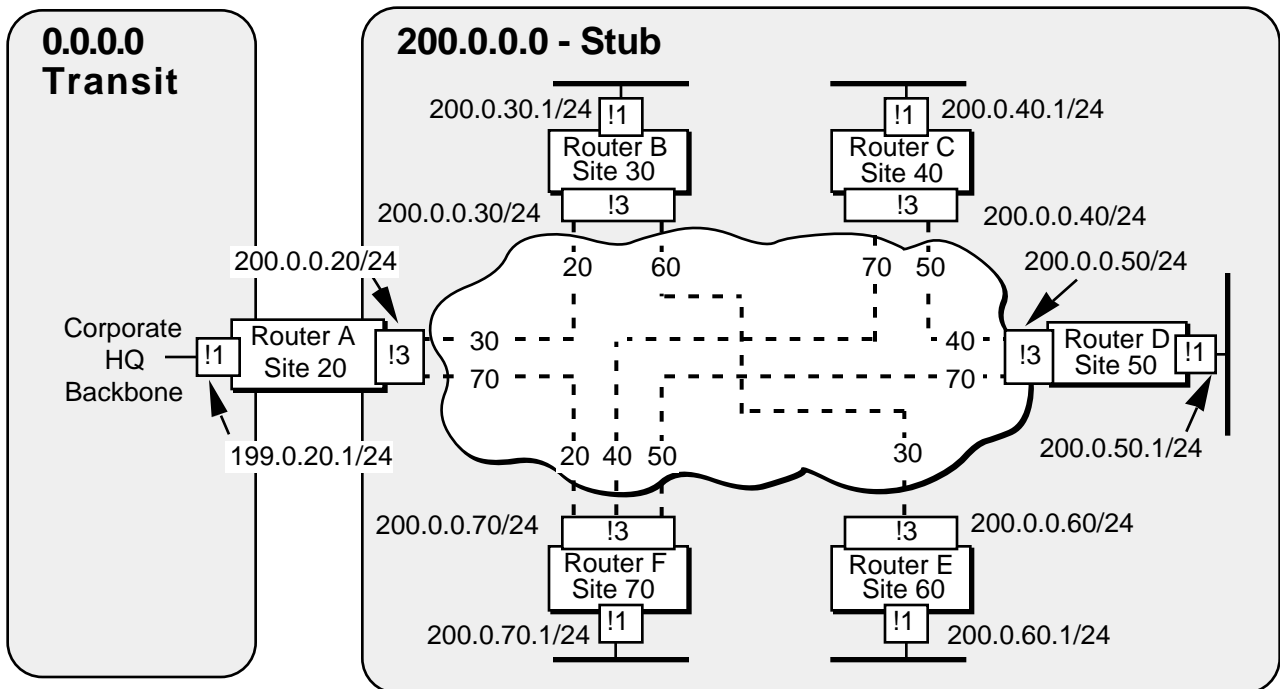


Figure 3: Example Topology

## General Comments

Some comments about the topology, assumptions, and specific design considerations used in this example:

1. A relatively simple IP addressing scheme is used for clarity. The notation "200.0.30.1/24" represents the four octet Class C IP address "200.0.30.1" with a 24 bit subnetwork mask (255.255.255.0). A real-world example would probably use a much more complicated addressing scheme.

2. The Frame Relay service provider's switches implement the ANSI LMI which supports the global addressing extension. Note that most switches auto-sense the LMI type used by the CPE.
3. DLCIs have been assigned using "site-based" addressing where each site has a unique DLCI. The DLCI that has been assigned to each site by the service provider is indicated in each of the router icons. The alternative would have been to use "PVC-based" addressing where each PVC has the same DLCI on each end. In the graphic, note that the DLCI label within the Frame Relay cloud at the end of each virtual circuit indicates the "site" address at the other end of the virtual circuit.
4. The example assumes that dynamic address resolution is employed across the Frame Relay network via Inverse ARP (InARP). If InARP is not supported, static mappings must be placed in the Address Translation Table using the `ADD -IP ADDRESS <IP address> @<DLCI#>` syntax. Each of the router configurations examples show the alternate steps required for static address resolution if it were to become part of the design specification.
5. The example assumes that OSPF will dynamically learn about neighboring routers from the mappings InARP places in the Address Translation Table. If security or other requirements demand static neighbor configuration, an OSPF neighbor list must be created using the `ADD !<port> -OSPF Neighbor <IP address>` syntax. For total control, `SETD !<port> -ARP CONTROL = NoInARP` and perform static address resolution using the `ADD -IP ADDRESS <IP address> @<DLCI#>` syntax. Each of the router configurations examples show the alternate steps required for static neighbor configuration if it were to become part of the design specification.
6. The networks located at the corporate headquarters are in Area 0.0.0.0-Transit while the Frame Relay network and each of the regional site networks are assigned to Area 200.0.0.0-Stub. One benefit of this design is that it eliminates the flooding of External-LSAs into the Frame Relay network since OSPF does not flood External-LSAs into Stub areas. Another benefit is that Router A functions as an Area Border Router (ABR) which keeps topology changes in Area 0.0.0.0 from causing a Dijkstra calculation in Area 200.0.0.0 and vice versa. Note that area 200.0.0.0 had to contain each of the remote site LANs or else Virtual Links would have had to be defined.

An alternate OSPF Area configuration would be to have the router interfaces on the Frame Relay network and the corporate headquarters networks all contained in Area 0.0.0.0. This permits the location of stub or transit areas at each remote site, but causes External-LSAs to be flooded throughout the Frame Relay network and will result in a larger number of routers performing a Dijkstra calculation if any topology change takes place in Area 0.0.0.0.

## Router A Configuration

Basic configuration parameters:

```
SETDefault !3 -Port Owner = FrameRelay
SETDefault !3 -ARP Control = InARP (Default setting)
SETDefault !3 -FrameRelay Control = ANsiLMI
SETDefault -IP Control = ROute
SETDefault !1 -IP NETaddr 199.0.20.1
SETDefault !3 -IP NETaddr 200.0.0.20
SETDefault !1 -OSPF AreaId = 0.0.0.0 Transit (Default setting)
SETDefault !3 -OSPF AreaId = 200.0.0.0 Stub
SETDefault !1 -OSPF Control = Enable
SETDefault !3 -OSPF Control = (Enable, DynamicNbr, NonMesh)
```

If the design requires that InARP be disabled and static address resolution be performed for the Frame Relay network. Note that Neighbors can still be dynamically learned for this network design:

```
SETDefault !3 -ARP Control = NoInARP
ADD -IP Address 200.0.0.30 @30
ADD -IP Address 200.0.0.70 @70
```

If security or other considerations required static neighbor configuration for the Frame Relay network:

```
SETDefault !3 -OSPF Control = (Enable, NoDynamicNbr, NonMesh)
ADD !3 -OSPF Neighbor 200.0.0.30
ADD !3 -OSPF Neighbor 200.0.0.70
SETDefault !3 -ARP Control = NoInARP
ADD -IP Address 200.0.0.30 @30
ADD -IP Address 200.0.0.70 @70
```

## Router B Configuration

Basic configuration parameters:

```
SETDefault !3 -Port Owner = FrameRelay
SETDefault !3 -ARP Control = InARP (Default setting)
SETDefault !3 -FrameRelay Control = ANsiLMI
SETDefault -IP Control = ROute
SETDefault !1 -IP NETaddr 200.0.30.1
SETDefault !3 -IP NETaddr 200.0.0.30
SETDefault !1 -OSPF AreaId = 200.0.0.0 Stub
SETDefault !3 -OSPF AreaId = 200.0.0.0 Stub
SETDefault !1 -OSPF Control = Enable
SETDefault !3 -OSPF Control = (Enable, DynamicNbr, NonMesh)
```

If the design required that InARP be disabled and static address resolution be performed for the Frame Relay network:

```
SETDefault !3 -ARP CONTrol = NoInARP
ADD -IP ADDRess 200.0.0.20 @20
ADD -IP ADDRess 200.0.0.60 @60
```

If security or other considerations required static neighbor configuration for the Frame Relay network:

```
SETDefault !3 -OSPF CONTrol = (Enable, NoDynamicNbr, NonMesh)
ADD !3 -OSPF Neighbor 200.0.0.20
ADD !3 -OSPF Neighbor 200.0.0.60
SETDefault !3 -ARP CONTrol = NoInARP
ADD -IP ADDRess 200.0.0.20 @20
ADD -IP ADDRess 200.0.0.60 @60
```

## Router C Configuration

Basic configuration parameters:

```
SETDefault !3 -Port OWner = FrameRelay
SETDefault !3 -ARP CONTrol = InARP (Default setting)
SETDefault !3 -FrameRelay CONTrol = ANsiLMI
SETDefault -IP CONTrol = ROute
SETDefault !1 -IP NETaddr 200.0.40.1
SETDefault !3 -IP NETaddr 200.0.0.40
SETDefault !1 -OSPF AreaId = 200.0.0.0 Stub
SETDefault !3 -OSPF AreaId = 200.0.0.0 Stub
SETDefault !1 -OSPF CONTrol = Enable
SETDefault !3 -OSPF CONTrol = (Enable, DynamicNbr, NonMesh)
```

If the design required that InARP be disabled and static address resolution be performed for the Frame Relay network:

```
SETDefault !3 -ARP CONTrol = NoInARP
ADD -IP ADDRess 200.0.0.50 @50
ADD -IP ADDRess 200.0.0.70 @70
```

If security or other considerations required static neighbor configuration for the Frame Relay network:

```
SETDefault !3 -OSPF CONTrol = (Enable, NoDynamicNbr, NonMesh)
ADD !3 -OSPF Neighbor 200.0.0.50
ADD !3 -OSPF Neighbor 200.0.0.70
SETDefault !3 -ARP CONTrol = NoInARP
ADD -IP ADDRess 200.0.0.50 @50
ADD -IP ADDRess 200.0.0.70 @70
```

## Router D Configuration

Basic configuration parameters:

```
SETDefault !3 -Port Owner = FrameRelay
SETDefault !3 -ARP Control = InARP (Default setting)
SETDefault !3 -FrameRelay Control = ANsiLMI
SETDefault -IP Control = ROute
SETDefault !1 -IP NETaddr 200.0.50.1
SETDefault !3 -IP NETaddr 200.0.0.50
SETDefault !1 -OSPF AreaId = 200.0.0.0 Stub
SETDefault !3 -OSPF AreaId = 200.0.0.0 Stub
SETDefault !1 -OSPF Control = Enable
SETDefault !3 -OSPF Control = (Enable, DynamicNbr, NonMesh)
```

If the design required that InARP be disabled and static address resolution be performed for the Frame Relay network:

```
SETDefault !3 -ARP Control = NoInARP
ADD -IP Address 200.0.0.40 @40
ADD -IP Address 200.0.0.70 @70
```

If security or other considerations required static neighbor configuration for the Frame Relay network:

```
SETDefault !3 -OSPF Control = (Enable, NoDynamicNbr, NonMesh)
ADD !3 -OSPF Neighbor 200.0.0.40
ADD !3 -OSPF Neighbor 200.0.0.70
SETDefault !3 -ARP Control = NoInARP
ADD -IP Address 200.0.0.40 @40
ADD -IP Address 200.0.0.70 @70
```

## Router E Configuration

Basic configuration parameters:

```
SETDefault !3 -Port Owner = FrameRelay
SETDefault !3 -ARP Control = InARP (Default setting)
SETDefault !3 -FrameRelay Control = ANsiLMI
SETDefault -IP Control = ROute
SETDefault !1 -IP NETaddr 200.0.60.1
SETDefault !3 -IP NETaddr 200.0.0.60
SETDefault !1 -OSPF AreaId = 200.0.0.0 Stub
SETDefault !3 -OSPF AreaId = 200.0.0.0 Stub
SETDefault !1 -OSPF Control = Enable
SETDefault !3 -OSPF Control = (Enable, DynamicNbr, NonMesh)
```

If the design required that InARP be disabled and static address resolution be performed for the Frame Relay network:

```
SETDefault !3 -ARP Control = NoInARP
ADD -IP Address 200.0.0.30 @30
```



If security or other considerations required static neighbor configuration for the Frame Relay network:

```
SETDefault !3 -OSPF CONTrol = (Enable, NoDynamicNbr, NonMesh)
ADD !3 -OSPF Neighbor 200.0.0.30
SETDefault !3 -ARP CONTrol = NoInARP
ADD -IP ADDRESS 200.0.0.30 @30
```

## Router F Configuration

Basic configuration parameters:

```
SETDefault !3 -POrt OWner = FrameRelay
SETDefault !3 -ARP CONTrol = InARP (Default setting)
SETDefault !3 -FrameRelay CONTrol = ANsiLMI
SETDefault -IP CONTrol = ROute
SETDefault !1 -IP NETaddr 200.0.70.1
SETDefault !3 -IP NETaddr 200.0.0.70
SETDefault !1 -OSPF AreaId = 200.0.0.0 Stub
SETDefault !3 -OSPF AreaId = 200.0.0.0 Stub
SETDefault !1 -OSPF CONTrol = Enable
SETDefault !3 -OSPF CONTrol = (Enable, DynamicNbr, NonMesh)
```

If the design required that InARP be disabled and static address resolution be performed for the Frame Relay network:

```
SETDefault !3 -ARP CONTrol = NoInARP
ADD -IP ADDRESS 200.0.0.20 @20
ADD -IP ADDRESS 200.0.0.40 @40
ADD -IP ADDRESS 200.0.0.50 @50
```

If security or other considerations required static neighbor configuration for the Frame Relay network:

```
SETDefault !3 -OSPF CONTrol = (Enable, NoDynamicNbr, NonMesh)
ADD !3 -OSPF Neighbor 200.0.0.20
ADD !3 -OSPF Neighbor 200.0.0.40
ADD !3 -OSPF Neighbor 200.0.0.50
SETDefault !3 -ARP CONTrol = NoInARP
ADD -IP ADDRESS 200.0.0.20 @20
ADD -IP ADDRESS 200.0.0.40 @40
ADD -IP ADDRESS 200.0.0.50 @50
```

## Troubleshooting the Frame Relay Network Configuration

The following commands and parameters should help with the verification and troubleshooting of your Frame Relay network configuration:

## Path Service Verification

SHoW [!<path>|!\*] -PAth CONFiguration

Displays current Path Service configuration information. Verify that each physical interface is Enabled and that the BAud, CONNector, and Clock parameters are correctly configured.

SHoW -SYStem STATistics -PAth

Displays path related statistics. Look for CRC errors, discards, and congestion.

## Port Service Verification

SHoW [!<port>|!\*] -PORT CONFiguration

Displays current Port Service configuration information. Verify that the port is Enabled and that the OWNer is Frame Relay.

SHoW -SYStem STATistics -PORT

Displays port related statistics. Look for the number of packets Rcvd and Xmit, as well as discards due to buffer overruns and congestion.

## Frame Relay Service Verification

SHoW [!<port>|!\*] -FrameRelay CONFiguration

Displays current FR Service configuration information.

SHoW [!<port>|!\*] -FrameRelay AllDLCCi

Displays active DLCIs. Verify that the correct remote sites are available.

SHoW [!<port>|!\*] -FrameRelay DLciStat

Displays the DLCI status and statistics for all active Frame Relay interfaces. Verify that the DLCIs are correct for each remote site and that traffic is flowing on those DLCIs.

## IP Service Verification

SHoW [!<port>|!\*] -IP CONFiguration

Displays values for all IP related parameters. Verify that IP routing is Enabled and that the IP addressing information is correct on all ports.

SHoW [!<port>|!\*] -IP NETaddr

Displays the address information for the directly connected IP networks. Verify that the status for each port is Up.

SHoW -IP ADDRESS

Displays the Address Translation Table (the ARP Cache). Verify that the display contains the correct InARP-derived or statically-derived IP-to-DLCI address mappings.

SHoW -IP AllRoutes

Displays the IP routing table. Verify that none of the expected routes to remote sites are missing.

## OSPF Service Verification

Show [!<port>|!\*] -OSPF CONFIguration

Displays values for all OSPF related parameters. Verify that OSPF is Enabled on each port and that all routers are operating in NonMesh mode on their Frame Relay port. Verify that the Cost, AreaID, and PassWord match on neighbors.

SHow !<port> -OSPF Neighbors

Displays the current list of statically-configured neighbors for the specified port. Verify that all statically configured neighbors are displayed.

SHow !<port> -OSPF NeighborStatus

Displays the status of neighbor adjacencies for the specified port. These neighbors may come from static neighbor configuration or dynamic neighbor learning. Static neighbors are distinguished in the display by an asterisk (\*). Verify that all expected neighbor adjacencies are included in the display. For this example, each router should have 2 or 3 point-to-point adjacencies depending on which DLCI's it is connected to.

SHow [!<port>|!\*] -OSPF InterfaceStatus

Displays the status of a specified interface that is running OSPF.

SHow -SYSTem STATistics -OSPF

Displays OSPF related statistics. Look for excessive SPF calculations, DR elections, Adjacency UP and DOWN events, and errors. Finally, verify that the ratio of Hello Rcvd to Hello Xmit makes sense for the topology.