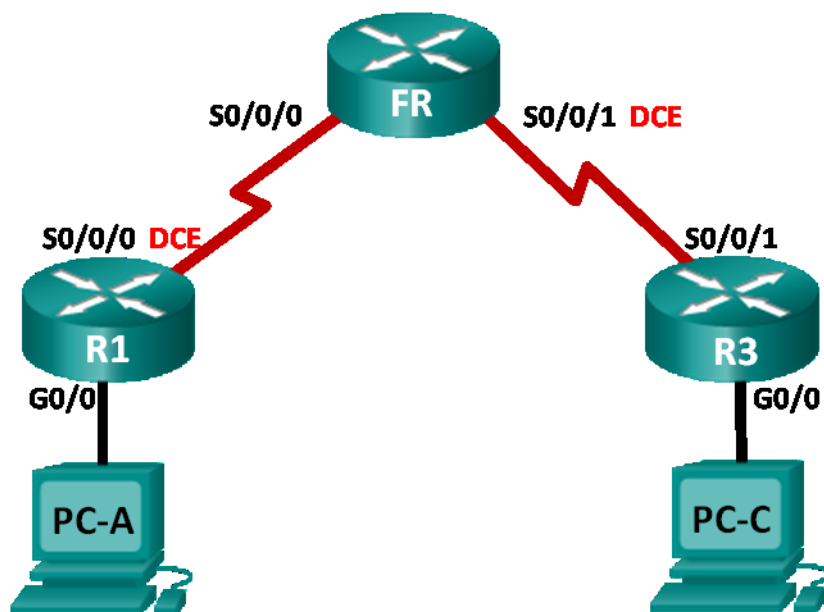# Lab10-01: Configuring Frame Relay and Subinterfaces

Topology

Member list:
1. นาย ธนพล โรจนะวิชานนท์ 5710503398
2. นาย ภารุจ สิงหพันธุ์ 5710503495

Addressing Table

Device	Interface	IPv4 and IPv6 Address	Default Gateway
R1	G0/0	192.168.1.1/24 2001:DB8:ACAD:A::1/64 FE80::1 link-local	N/A
	S0/0/0 (DCE)	10.1.1.1/30 2001:DB8:ACAD:B::1/64 FE80::1 link-local	N/A
FR	S0/0/0	N/A	N/A
	S0/0/1 (DCE)	N/A	N/A
R3	G0/0	192.168.3.1/24 2001:DB8:ACAD:C::3/64 FE80::3 link-local	N/A
	S0/0/1	10.1.1.2/30 2001:DB8:ACAD:B::3/64 FE80::3 link-local	N/A
PC-A	NIC	192.168.1.3/24 2001:DB8:ACAD:A::A/64	192.168.1.1 FE80::1
PC-C	NIC	192.168.3.3/24 2001:DB8:ACAD:C::C/64	192.168.3.1 FE80::3

Objectives

Part 1: Build the Network and Configure Basic Device Settings

Part 2: Configure a Frame Relay Switch

Part 3: Configure Basic Frame Relay

Part 4: Troubleshoot Frame Relay

Part 5: Configure a Frame Relay Subinterface

Background / Scenario

Frame Relay is a high-performance WAN protocol that operates at the physical and data link layers of the OSI reference model. Unlike leased lines, Frame Relay requires only a single access circuit to the Frame Relay provider to communicate with multiple sites that are connected to the same provider.

Frame Relay was one of the most extensively used WAN protocols, primarily because it was relatively inexpensive compared to dedicated lines. In addition, configuring user equipment in a Frame Relay network is fairly simple. With the advent of broadband services such as DSL and cable modem, GigaMAN (point-to-point Ethernet service over fiber-optic cable), VPN, and Multiprotocol Label Switching (MPLS), Frame Relay has become a less desirable solution for accessing the WAN. However, some rural areas do not have access to these alternative solutions and still rely on Frame Relay for connectivity to the WAN.

In this lab, you will configure Frame Relay encapsulation on serial links. You will also configure a router to simulate a Frame Relay switch. You will review Cisco standards and open standards that apply to Frame Relay. You will also configure Frame Relay point-to-point subinterfaces.

Note: The routers used with CCNA hands-on labs are Cisco 1941 Integrated Services Routers (ISRs) with Cisco IOS Release 15.2(4)M3 (universalk9 image). Other routers and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and output produced might vary from what is shown in the labs. Refer to the Router Interface Summary Table at the end of this lab for the correct interface identifiers.

Note: Make sure that the routers have been erased and have no startup configurations. If you are unsure, contact your instructor.

Required Resources

- 3 Routers (Cisco 1941 with Cisco IOS Release 15.2(4)M3 universal image or comparable)
- 2 PCs (Windows 7, Vista, or XP with terminal emulation program, such as Tera Term)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet and serial cables as shown in the topology

Part 1: Build the Network and Configure Basic Device Settings

In Part 1, you will set up the network topology and configure basic settings on the PC hosts and routers.

Step 1: Cable the network as shown in the topology.

Step 2: Initialize and reload the routers as necessary.

Step 3: Configure basic settings for each router.

a. Disable DNS lookup.

b. Configure device names as shown in the topology.

c. Assign **class** as the privileged EXEC mode password.

d. Assign **cisco** as the console and vty passwords and enable login.

e. Configure **logging synchronous** for the console line.

f. Encrypt the plain text passwords.

g. Configure a MOTD banner to warn users that unauthorized access is prohibited.

h. Set the clocking rate for all DCE serial interfaces to **128000**.

i. Configure the IPv4 and IPv6 addresses listed in the Addressing Table for all interfaces. Do not activate the serial interfaces at this time.

j. Copy the running configuration to the startup configuration.

Step 4: Configure PC hosts.

Refer to the Addressing Table for PC host address information.

Step 5: Test connectivity.

At this point, the PCs will not be able to ping each other, but they should be able to ping their default gateway. Test both protocols, IPv4 and IPv6. Verify and troubleshoot if necessary.

Part 2: Configure a Frame Relay Switch

In Part 2, you will configure a Frame Relay switch. You will create permanent virtual circuits (PVCs) and assign Data Link Connection Identifiers (DLCIs). This configuration creates two PVCs: one from R1 to R3 (DLCI 103), and one from R3 to R1 (DLCI 301).

Step 1: Configure the FR router as a Frame Relay switch.

The **frame-relay switching** command enables Frame Relay switching globally on a router, allowing it to forward frames based on the incoming DLCI rather than an IP address.

```
FR(config)# frame-relay switching
```

Step 2: Change the interface encapsulation on S0/0/0.

Change the interface encapsulation type to Frame Relay. Like HDLC or PPP, Frame Relay is a data-link layer protocol that specifies the framing of Layer 2 traffic.

```
FR(config)# interface s0/0/0
FR(config-if)# encapsulation frame-relay
```

Step 3: Change the interface type to DCE.

Changing the interface type to DCE tells the router to send Local Management Interface (LMI) keepalives and allows Frame Relay route statements to be applied.

Note: Frame Relay interface types do not need to match the underlying physical interface type. A physical DTE serial interface can act as a Frame Relay DCE interface, and a physical DCE interface can act as a logical Frame Relay DTE interface.

```
FR(config)# interface s0/0/0
FR(config-if)# frame-relay intf-type dce
```

Step 4: Configure DLCI.

Configure the router to forward incoming traffic on interface S0/0/0 with DLCI 103 to S0/0/1 with an output of DLCI of 301.

```
FR(config-if)# frame-relay route 103 interface s0/0/1 301
FR(config-if)# no shutdown
```

Step 5: Configure Frame Relay on S0/0/1.

```
FR(config)# interface s0/0/1
FR(config-if)# encapsulation frame-relay
FR(config-if)# frame-relay intf-type dce
FR(config-if)# frame-relay route 301 interface s0/0/0 103
FR(config-if)# no shutdown
```

Step 6: Verify Frame Relay configuration.

a. Use the **show frame-relay pvc** command to verify that Frame Relay is configured correctly.

```
FR# show frame-relay pvc

PVC Statistics for interface Serial0/0/0 (Frame Relay DCE)
```

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

DLCI = 103, DLCI USAGE = SWITCHED, PVC STATUS = INACTIVE, INTERFACE = Serial0/0/0

```
    input pkts 0           output pkts 0          in bytes 0
    out bytes 0            dropped pkts 0         in pkts dropped 0
    out pkts dropped 0           out bytes dropped 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
    30 second input rate 0 bits/sec, 0 packets/sec
    30 second output rate 0 bits/sec, 0 packets/sec
    switched pkts 0
    Detailed packet drop counters:
    no out intf 0          out intf down 0        no out PVC 0
    in PVC down 0          out PVC down 0         pkt too big 0
    shaping Q full 0       pkt above DE 0         policing drop 0
    connected to interface Serial0/0/1 301
    pvc create time 00:00:53, last time pvc status changed 00:00:53
```

```
PVC Statistics for interface Serial0/0/1 (Frame Relay DCE)

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

DLCI = 301, DLCI USAGE = SWITCHED, PVC STATUS = INACTIVE, INTERFACE = Serial0/0/1

```
    input pkts 0           output pkts 0          in bytes 0
    out bytes 0            dropped pkts 0         in pkts dropped 0
    out pkts dropped 0           out bytes dropped 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
    30 second input rate 0 bits/sec, 0 packets/sec
    30 second output rate 0 bits/sec, 0 packets/sec
    switched pkts 0
    Detailed packet drop counters:
    no out intf 0          out intf down 0        no out PVC 0
    in PVC down 0          out PVC down 0         pkt too big 0
    shaping Q full 0       pkt above DE 0         policing drop 0
    connected to interface Serial0/0/0 103
    pvc create time 00:00:16, last time pvc status changed 00:00:16
```

b. Issue the **show frame-relay route** command. This is the Layer 2 route that Frame Relay traffic takes through the network. (Do not confuse this with Layer 3 IP routing.)

```
FR# show frame-relay route
Input Intf      Input Dlci      Output Intf     Output Dlci     Status
Serial0/0/0     103             Serial0/0/1     301             inactive
Serial0/0/1     301             Serial0/0/0     103             inactive
```

Part 3: Configure Basic Frame Relay

In Part 3, you will configure Frame Relay on routers R1 and R3. After Frame Relay is configured, you will enable the EIGRP routing protocol to provide end-to-end connectivity.

Step 1: Configure R1 for Frame Relay.

Inverse ARP allows distant ends of a Frame Relay link to discover each other dynamically, and provides a dynamic method of mapping IP addresses to DLCIs. Although Inverse ARP is useful, it is not always reliable. The best practice is to map IP addresses to DLCIs statically and disable Inverse ARP.

a. Change the encapsulation on S0/0/0 to Frame Relay.

```
R1(config)# interface s0/0/0
R1(config-if)# encapsulation frame-relay
```

b. Use the **no frame-relay inverse-arp** command to disable Inverse ARP.

```
R1(config)# interface s0/0/0
R1(config-if)# no frame-relay inverse-arp
```

c. Use the **frame-relay map** command to map an IP address to a DLCI statically. In addition to mapping an IP to a DLCI, Cisco IOS software allows several other Layer 3 protocol addresses to be mapped. In the following command, the **broadcast** keyword sends any multicast or broadcast traffic destined for this link over the DLCI. Most routing protocols require the **broadcast** keyword to function properly over Frame Relay. You can use the **broadcast** keyword on multiple DLCIs on the same interface. The traffic is replicated to all PVCs.

Note: The IPv6 Frame Relay map to a global unicast address does not include the **broadcast** keyword. However, the **broadcast** keyword is used in the mapping to the link-local address. IPv6 routing protocols use link-local addresses for multicast routing updates; therefore, only the link-local address map requires the **broadcast** keyword to forward multicast packets.

```
R1(config)# interface s0/0/0
R1(config-if)# frame-relay map ip 10.1.1.2 103 broadcast
R1(config-if)# frame-relay map ipv6 2001:db8:acad:b::3 103
R1(config-if)# frame-relay map ipv6 fe80::3 103 broadcast
```

d. For the router to ping its own interface, the DLCI must be created to map to the local interface.

```
R1(config)# interface s0/0/0
R1(config-if)# frame-relay map ip 10.1.1.1 103
R1(config-if)# frame-relay map ipv6 2001:db8:acad:b::1 103
```

e. Use the **no shutdown** command to activate S0/0/0.

```
R1(config-if)# no shutdown
```

Step 2: Configure R3 for Frame Relay.

```
R3(config)# interface s0/0/1
R3(config-if)# encapsulation frame-relay
R3(config-if)# no frame-relay inverse-arp
```

```
R3(config-if)# frame-relay map ip 10.1.1.1 301 broadcast
R3(config-if)# frame-relay map ipv6 2001:db8:acad:b::1 301
R3(config-if)# frame-relay map ipv6 fe80::1 301 broadcast
R3(config-if)# frame-relay map ip 10.1.1.2 301
R3(config-if)# frame-relay map ipv6 2001:db8:acad:b::3 301
R3(config-if)# no shutdown
```

Why is the **no shutdown** command used after the **no frame-relay inverse-arp** command?

_____เพื่อActive Interface _____

_____

_____

_____

Step 3:   Verify that Frame Relay is active.

a.   You should now be able to ping R3 from R1. It may take several seconds after bringing up the interfaces for the PVCs to become active.

```
R1# ping 10.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/30/40 ms
R1# ping 2001:db8:acad:b::3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:ACAD:B::3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
```

b.   Ping R1 from R3.

```
R3# ping 10.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
R3# ping 2001:db8:acad:b::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:ACAD:B::1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/26/28 ms
```

c.   Issue the **show frame-relay pvc** command to display PVC status information on R1 and R3.

```
R1# show frame-relay pvc

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

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

```
Serial0/0/0 (up): ip 10.1.1.1 dlci 103(0x67,0x1870), static,
              CISCO, status defined, active
Serial0/0/0 (up): ipv6 2001:DB8:ACAD:B::3 dlci 103(0x67,0x1870), static,
              CISCO, status defined, active
Serial0/0/0 (up): ip 10.1.1.2 dlci 103(0x67,0x1870), static,
              broadcast,
              CISCO, status defined, active


R3# show frame-relay map
Serial0/0/1 (up): ipv6 FE80::1 dlci 301(0x12D,0x48D0), static,
              broadcast,
              CISCO, status defined, active
Serial0/0/1 (up): ipv6 2001:DB8:ACAD:B::3 dlci 301(0x12D,0x48D0), static,
              CISCO, status defined, active
Serial0/0/1 (up): ip 10.1.1.2 dlci 301(0x12D,0x48D0), static,
              CISCO, status defined, active
Serial0/0/1 (up): ipv6 2001:DB8:ACAD:B::1 dlci 301(0x12D,0x48D0), static,
              CISCO, status defined, active
Serial0/0/1 (up): ip 10.1.1.1 dlci 301(0x12D,0x48D0), static,
              broadcast,
              CISCO, status defined, active
```

Note: The FR router acts as a Layer 2 device, so there is no need to map Layer 3 addresses to Layer 2 DLCIs.

Step 4: Configure EIGRP on R1 and R3.

a. Enable IPv6 routing on R1 and R3.

b. Using AS 1, enable EIGRP for IPv4 and IPv6 on R1 and R3 for all networks. Set the router ID for R1 as 1.1.1.1 and 3.3.3.3 for R3.

Step 5: Verify end-to-end connectivity.

Ping PC-C from PC-A. If your pings were unsuccessful, troubleshoot until you have end-to-end connectivity.

Note: It may be necessary to disable the PC firewall for pings to be successful.

Part 4: Troubleshoot Frame Relay

In Part 4, you will break the Frame Relay connection established earlier and use some tools to troubleshoot Frame Relay. A variety of tools are available for troubleshooting Frame Relay connectivity issues.

Step 1: Debug Local Management Interface (LMI).

a. Issue the **debug frame-relay lmi** command on R1. The output gives detailed information on all LMI data. Keepalives are sent every 10 seconds by default, so you may have to wait until you see any output. The output shows an outgoing LMI packet with a sequence number of 50. The last LMI message received from FR had a sequence number of 49. The output is also showing an incoming LMI message from FR to R1 with a sequence number of 50. DLCI 103 is the only DLCI on this link, and it is currently active.

```
R1# debug frame-relay lmi
Frame Relay LMI debugging is on
Displaying all Frame Relay LMI data
R1#
```

```
*Jun 26 18:28:45.922: Serial0/0/0(out): StEnq, myseq 50, yourseen 49, DTE up
*Jun 26 18:28:45.922: datagramstart = 0xC318D54, datagramsize = 13
*Jun 26 18:28:45.922: FR encap = 0xFCF10309
*Jun 26 18:28:45.922: 00 75 01 01 01 03 02 32 31
*Jun 26 18:28:45.922:
*Jun 26 18:28:45.922: Serial0/0/0(in): Status, myseq 50, pak size 13
*Jun 26 18:28:45.922: RT IE 1, length 1, type 1
*Jun 26 18:28:45.922: KA IE 3, length 2, yourseq 50, myseq 50
*Jun 26 18:28:45.922: PVC IE 0x7 , length 0x6 , dlci 103 , status 0x2 , bw 0
```

b. Issue the **undebug all** command to turn off debugging.

Note: This command can be abbreviated to **u all**. This is useful to know when debug information is flooding the screen.

```
R1# undebug all
All possible debugging has been turned off
```

Step 2: Remove the IPv4 frame map from R1.

a. Issue the **no frame-relay map** command to remove the IPv4 frame map on R1.

```
R1(config)# interface s0/0/0
R1(config-if)# no frame-relay map ip 10.1.1.2 103 broadcast
```

b. Issue the **debug ip icmp** command on R1.

```
R1# debug ip icmp
ICMP packet debugging is on
```

c. Ping R1 from R3. Pings should not be successful. However, debug messages on R1 show that the ICMP packets from R3 are reaching R1.

Note: You should see console messages reporting the EIGRP adjacency going up and down. This is sometimes called flapping.

```
R3# ping 10.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

R1#
*Jun 26 20:12:35.693: ICMP: echo reply sent, src 10.1.1.1, dst 10.1.1.2, topology
BASE, dscp 0 topoid 0
R1#
*Jun 26 20:12:37.689: ICMP: echo reply sent, src 10.1.1.1, dst 10.1.1.2, topology
BASE, dscp 0 topoid 0
R1#
*Jun 26 20:12:39.689: ICMP: echo reply sent, src 10.1.1.1, dst 10.1.1.2, topology
BASE, dscp 0 topoid 0
R1#
*Jun 26 20:12:41.689: ICMP: echo reply sent, src 10.1.1.1, dst 10.1.1.2, topology
BASE, dscp 0 topoid 0
R1#
*Jun 26 20:12:43.689: ICMP: echo reply sent, src 10.1.1.1, dst 10.1.1.2, topology
BASE, dscp 0 topoid 0
```

Why does the ping fail?

<u>เพราะได้ทำการ remove frame-relay ของ10.1.1.2ออกไปแล้วทำให้ping</u>

<u>10.1.1.1ไม่ได้</u>

d. Issue the **show frame-relay map** command on R1. The IPv4 map for R3 is missing from the list.

```
R1# show frame-relay map
Serial0/0/0 (up): ipv6 FE80::3 dlci 103(0x67,0x1870), static,
                broadcast,
                CISCO, status defined, active
Serial0/0/0 (up): ipv6 2001:DB8:ACAD:B::1 dlci 103(0x67,0x1870), static,
                CISCO, status defined, active
Serial0/0/0 (up): ip 10.1.1.1 dlci 103(0x67,0x1870), static,
                CISCO, status defined, active
Serial0/0/0 (up): ipv6 2001:DB8:ACAD:B::3 dlci 103(0x67,0x1870), static,
                CISCO, status defined, active
```

e. Issue the **undebug all** command to turn off debugging on R1.

```
R1# undebug all
All possible debugging has been turned off
```

f. Re-apply the **frame-relay map ip** command to S0/0/0 on R1, but without using the **broadcast** keyword.

```
R1(config)# interface s0/0/0
R1(config-if)# frame-relay map ip 10.1.1.2 103
```

g. Ping R1 from R3. Pings should be successful, but the EIGRP adjacency continues to flap. It may take a few minutes between each message because of the EIGRP timers.

```
R3# ping 10.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms

R1(config-if)#
*Jun 26 20:25:10.871: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 10.1.1.2 (Serial0/0/0)
is down: Interface PEER-TERMINATION received
*Jun 26 20:28:13.673: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 10.1.1.2 (Serial0/0/0)
is up: new adjacency
R1(config-if)#
*Jun 26 20:31:18.185: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 10.1.1.2 (Serial0/0/0)
is down: retry limit exceeded
R1(config-if)#
*Jun 26 20:32:00.977: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 10.1.1.2 (Serial0/0/0)
is up: new adjacency
R1(config-if)#
*Jun 26 20:35:05.489: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 10.1.1.2 (Serial0/0/0)
is down: retry limit exceeded
```

Why does the EIGRP adjacency continue to flap?

__Multicast traffic is no being forwarded over the DLCI specified in the frame-relay map statement_____

__เนื่องจากไม่มีการ broadcast_____

h.   Replace the Frame Relay map statement and include the **broadcast** keyword this time.

```
R1(config-if)# frame-relay map ip 10.1.1.2 103 broadcast
```

i.   Verify that the full routing table is restored and that you have end-to-end connectivity.

Step 3: Change the Frame Relay encapsulation type.

Cisco IOS software supports two types of Frame Relay encapsulation: the default Cisco encapsulation and the standards-based IETF encapsulation.

a.   Change the Frame Relay encapsulation on S0/0/1 on R3 to IETF.

```
R3(config)# interface s0/0/1
R3(config-if)# encapsulation frame-relay ietf
```

b.   Issue the **show interfaces s0/0/1** command on R3 and FR. Even though the encapsulation is different on each interface, the link is still active. This is because Cisco routers understand both types of incoming frames. However, if you have routers from different vendors and you are using Frame Relay, then the IETF standard must be used.

```
R3# show interfaces s0/0/1
Serial0/0/1 is up, line protocol is up
  Hardware is WIC MBRD Serial
  Internet address is 10.1.1.2/30
  MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY IETF, loopback not set
  Keepalive set (10 sec)
  LMI enq sent  1898, LMI stat recvd 1900, LMI upd recvd 0, DTE LMI up
<output omitted>

FR# show interfaces s0/0/1
Serial0/0/1 is up, line protocol is up
  Hardware is WIC MBRD Serial
  MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY, loopback not set
  Keepalive set (10 sec)
  LMI enq sent  0, LMI stat recvd 0, LMI upd recvd 0
```

c.   Reset the R3 Frame Relay encapsulation back to Cisco (the default).

```
R3(config)# interface s0/0/1
R3(config-if)# encapsulation frame-relay
```

Step 4: Change the LMI type.

a.   Issue the **frame-relay lmi-type ansi** command on interface S0/0/1 on R3.

```
R3(config-if)# frame-relay lmi-type ansi
```

b.   After at least 60 seconds, issue the **show interfaces s0/0/1** command on R3. When 60 seconds have passed, the interface changes its state to up, then down, because R3 is expecting ANSI LMI, and FR is sending Cisco LMI.

```
R3# show interfaces s0/0/1
Serial0/0/1 is up, line protocol is down
```

```
    Hardware is WIC MBRD Serial
    Internet address is 10.1.1.2/30
    MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
       reliability 255/255, txload 1/255, rxload 1/255
    Encapsulation FRAME-RELAY, loopback not set
    Keepalive set (10 sec)
    LMI enq sent  2157, LMI stat recvd 2136, LMI upd recvd 0, DTE LMI down
    LMI enq recvd 0, LMI stat sent  0, LMI upd sent  0
    LMI DLCI 0  LMI type is ANSI Annex D  frame relay DTE  segmentation inactive
    FR SVC disabled, LAPF state down
    Broadcast queue 0/64, broadcasts sent/dropped 733/0, interface broadcast
<output omitted>
```

c. On R3, issue the **show frame-relay lmi** command to display LMI information, including LMI type, number of timeouts, and the amount of time since the last full update.

```
R3# show frame-relay lmi

LMI Statistics for interface Serial0/0/1 (Frame Relay DTE) LMI TYPE = ANSI
  Invalid Unnumbered info 0          Invalid Prot Disc 0
  Invalid dummy Call Ref 0           Invalid Msg Type 0
  Invalid Status Message 0           Invalid Lock Shift 0
  Invalid Information ID 0           Invalid Report IE Len 0
  Invalid Report Request 0           Invalid Keep IE Len 0
  Num Status Enq. Sent 2158          Num Status msgs Rcvd 2136
  Num Update Status Rcvd 0           Num Status Timeouts 23
  Last Full Status Req 00:00:05      Last Full Status Rcvd 00:04:35
```

d. On R3, issue the **debug frame-relay lmi** command. The LMI packets no longer display in pairs. While all outgoing LMI messages are logged, no incoming messages display because R3 is expecting ANSI LMI, and FR is sending Cisco LMI.

```
R3# debug frame-relay lmi
Frame Relay LMI debugging is on
Displaying all Frame Relay LMI data
R3#
*Jun 26 21:49:10.829: Serial0/0/1(out): StEnq, myseq 104, yourseen 0, DTE down
*Jun 26 21:49:10.829: datagramstart = 0xC313554, datagramsize = 14
*Jun 26 21:49:10.829: FR encap = 0x00010308
*Jun 26 21:49:10.829: 00 75 95 01 01 00 03 02 68 00
*Jun 26 21:49:10.829:
R3#
*Jun 26 21:49:20.829: Serial0/0/1(out): StEnq, myseq 105, yourseen 0, DTE down
*Jun 26 21:49:20.829: datagramstart = 0xC317554, datagramsize = 14
*Jun 26 21:49:20.829: FR encap = 0x00010308
*Jun 26 21:49:20.829: 00 75 95 01 01 00 03 02 69 00
*Jun 26 21:49:20.829:
```

e. Restore the LMI type back to Cisco on R3. Notice that the debug messages change after you issue this command. The LMI sequence number has been reset to 1. R3 began to understand the LMI messages coming in from FR. After R3 and FR have successfully exchanged LMI messages, the interface changed state to up.

```
R3(config)# interface s0/0/1
R3(config-if)# frame-relay lmi-type cisco
R3(config-if)#
*Jun 26 21:51:20.829: Serial0/0/1(out): StEnq, myseq 117, yourseen 0, DTE down
*Jun 26 21:51:20.829: datagramstart = 0xC31F254, datagramsize = 14
*Jun 26 21:51:20.829: FR encap = 0x00010308
*Jun 26 21:51:20.829: 00 75 95 01 01 00 03 02 75 00
*Jun 26 21:51:20.829:
R3(config-if)#
*Jun 26 21:51:30.829: Serial0/0/1(out): StEnq, myseq 1, yourseen 0, DTE down
*Jun 26 21:51:30.829: datagramstart = 0xC31F3D4, datagramsize = 13
*Jun 26 21:51:30.829: FR encap = 0xFCF10309
*Jun 26 21:51:30.829: 00 75 01 01 00 03 02 01 00
*Jun 26 21:51:30.829:
*Jun 26 21:51:30.829: Serial0/0/1(in): Status, myseq 1, pak size 21
*Jun 26 21:51:30.829: RT IE 1, length 1, type 0
*Jun 26 21:51:30.829: KA IE 3, length 2, yourseq 1 , myseq 1
*Jun 26 21:51:30.829: PVC IE 0x7 , length 0x6 , dlci 301, stat
R3(config-if)#us 0x2 , bw 0
R3(config-if)#
*Jun 26 21:51:40.829: Serial0/0/1(out): StEnq, myseq 2, yourseen 1, DTE down
*Jun 26 21:51:40.829: datagramstart = 0xC313B54, datagramsize = 13
*Jun 26 21:51:40.829: FR encap = 0xFCF10309
*Jun 26 21:51:40.829: 00 75 01 01 01 03 02 02 01
*Jun 26 21:51:40.829:
*Jun 26 21:51:40.829: Serial0/0/1(in): Status, myseq 2, pak size 21
*Jun 26 21:51:40.829: RT IE 1, length 1, type 0
*Jun 26 21:51:40.829: KA IE 3, length 2, yourseq 2 , myseq 2
*Jun 26 21:51:40.829: PVC IE 0x7 , length 0x6 , dlci 301, stat
R3(config-if)#us 0x2 , bw 0
*Jun 26 21:51:51.829: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial0/0/1, changed state to up
R3(config-if)#
```

f. Issue the **undebug all** command to end debugging.

```
R3# undebug all
All possible debugging has been turned off
```

Part 5: Configure a Frame Relay Subinterface

Frame Relay supports two types of subinterfaces: point-to-point and point-to-multipoint. Point-to-multipoint subinterfaces support non-broadcast multiaccess topologies. For example, a hub and spoke topology would use a point-to-multipoint subinterface. In Part 5, you will create a point-to-point subinterface.

Step 1: On the FR router, create new PVCs between R1 and R3.

```
FR(config)# interface s0/0/0
FR(config-if)# frame-relay route 113 interface s0/0/1 311
FR(config-if)# interface s0/0/1
FR(config-if)# frame-relay route 311 interface s0/0/0 113
```

Step 2:   Create and configure a point-to-point subinterface on R1 and R3.

Note: Frame Relay encapsulation must be specified on the physical interface before subinterfaces can be created.

a.   Create subinterface 113 as a point-to-point interface on R1.

```
R1(config)# interface s0/0/0.113 point-to-point
R1(config-subif)# ip address 10.1.1.5 255.255.255.252
R1(config-subif)# ipv6 address 2001:db8:acad:d::1/64
R1(config-subif)# ipv6 address fe80::1 link-local
R1(config-subif)# frame-relay interface-dlci 113
R1(config-fr-dlci)#
```

b.   Create subinterface 311 as a point-to-point subinterface on R3.

```
R3(config)# interface s0/0/1.311 point-to-point
R3(config-subif)# ip address 10.1.1.6 255.255.255.252
R3(config-subif)# ipv6 address 2001:db8:acad:d::3/64
R3(config-subif)# ipv6 address fe80::3 link-local
R3(config-subif)# frame-relay interface-dlci 311
R3(config-fr-dlci)#
```

c.   Verify connectivity.

```
R1# ping 10.1.1.6
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
R1# ping 2001:db8:acad:d::3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:ACAD:D::3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms


R3# ping 10.1.1.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
R3# ping 2001:db8:acad:d::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:ACAD:D::1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
```

d.   Issue the **show frame-relay pvc** command on R1 and R3 to display the PVC status.

```
R1# show frame-relay pvc

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

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


 DLCI = 103, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0/0

   input pkts 1170        output pkts 1408        in bytes 92566
   out bytes 105327       dropped pkts 0          in pkts dropped 0
   out pkts dropped 0            out bytes dropped 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 1160    out bcast bytes 89034
   5 minute input rate 0 bits/sec, 0 packets/sec
   5 minute output rate 0 bits/sec, 0 packets/sec
   pvc create time 07:53:13, last time pvc status changed 00:35:58
```

<mark>DLCI = 113, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0/0.113</mark>

```
   input pkts 86          output pkts 494         in bytes 20916
   out bytes 45208        dropped pkts 0          in pkts dropped 0
   out pkts dropped 0            out bytes dropped 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 464     out bcast bytes 42088
   5 minute input rate 0 bits/sec, 0 packets/sec
   5 minute output rate 0 bits/sec, 0 packets/sec
   pvc create time 00:35:58, last time pvc status changed 00:35:58
```

R3# **show frame-relay pvc**

```
 PVC Statistics for interface Serial0/0/1 (Frame Relay DTE)


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


 DLCI = 301, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0/1

   input pkts 1406        output pkts 1176        in bytes 105143
   out bytes 93110        dropped pkts 0          in pkts dropped 0
   out pkts dropped 0            out bytes dropped 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 1038    out bcast bytes 80878
   5 minute input rate 0 bits/sec, 0 packets/sec
   5 minute output rate 0 bits/sec, 0 packets/sec
```

```
    pvc create time 07:51:07, last time pvc status changed 00:37:16


  DLCI = 311, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0/1.311


    input pkts 513          output pkts 114         in bytes 47072
    out bytes 30360         dropped pkts 0          in pkts dropped 0
    out pkts dropped 0            out bytes dropped 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 74       out bcast bytes 26200
    5 minute input rate 0 bits/sec, 0 packets/sec
    5 minute output rate 0 bits/sec, 0 packets/sec
    pvc create time 01:11:06, last time pvc status changed 00:37:16
```

e. Issue the **show frame-relay route** command on FR to verify the status of the Frame Relay map statements.

```
FR# show frame-relay route
Input Intf      Input Dlci     Output Intf     Output Dlci     Status
Serial0/0/0     103            Serial0/0/1     301             active
Serial0/0/0     113            Serial0/0/1     311             active
Serial0/0/1     301            Serial0/0/0     103             active
Serial0/0/1     311            Serial0/0/0     113             active
```

f. Issue the **show frame-relay map** command on R1 and R3 to verify the status of the Frame Relay map statements.

```
R1# show frame-relay map
Serial0/0/0 (up): ip 10.1.1.2 dlci 103(0x67,0x1870), static,
              broadcast,
              CISCO, status defined, active
Serial0/0/0 (up): ipv6 FE80::3 dlci 103(0x67,0x1870), static,
              broadcast,
              CISCO, status defined, active
Serial0/0/0 (up): ipv6 2001:DB8:ACAD:B::1 dlci 103(0x67,0x1870), static,
              CISCO, status defined, active
Serial0/0/0 (up): ip 10.1.1.1 dlci 103(0x67,0x1870), static,
              CISCO, status defined, active
Serial0/0/0 (up): ipv6 2001:DB8:ACAD:B::3 dlci 103(0x67,0x1870), static,
              CISCO, status defined, active
Serial0/0.113 (up): point-to-point dlci, dlci 113(0x71,0x1C10), broadcast
          status defined, active


R3# show frame-relay map
Serial0/0/1 (up): ipv6 FE80::1 dlci 301(0x12D,0x48D0), static,
              broadcast,
              CISCO, status defined, active
Serial0/0/1 (up): ipv6 2001:DB8:ACAD:B::3 dlci 301(0x12D,0x48D0), static,
              CISCO, status defined, active
Serial0/0/1 (up): ip 10.1.1.2 dlci 301(0x12D,0x48D0), static,
              CISCO, status defined, active
Serial0/0/1 (up): ipv6 2001:DB8:ACAD:B::1 dlci 301(0x12D,0x48D0), static,
```

```
                CISCO, status defined, active
    Serial0/0/1 (up): ip 10.1.1.1 dlci 301(0x12D,0x48D0), static,
                broadcast,
                CISCO, status defined, active
    Serial0/0/1.311 (up): point-to-point dlci, dlci 311(0x137,0x4C70), broadcast
            status defined, active
```

Reflection

1. What is a PVC and how is it used?

   _Permanent Virtual Circuit เป็นสิ่งที่จะเกิดขึ้นเมื่อเริ่มใช้บริการระหว่างพอร์ต_____

   _แต่ละพอร์ตจะถูกกำหนดค่าที่เรียกว่า CIR (Committed Information Rate) ซึ่งเป็นค่าที่ผู้ให้บริการรับรองว่าความเร็วที่ได้ใน____

   _การรับ/ส่งข้อมูลต้องไม่น้อยไปกว่านี้ แม้ว่าการรับ/ส่งอาจจะทำได้ในความเร็วที่สูงกว่านี้ก็ได้_____

2. What is the purpose of a DLCI?

   _เป็นตัวตรวจสอบการรับส่งข้อมูลของแต่ละport ว่ามีความถูกต้องหรือไม่_____

3. What purpose does the Local Management Interface (LMI) serve in a Frame Relay network?

   _ใช้ตรวจสอบ status ของ PVC และช่วยทำความมั่นใจว่า data จะสามารถถูกส่งไประหว่าง Frame Relay DTE และ DCE ได้

   และ Lmi extension ยังทำให้ค่า DLCI เป็นค่าแบบ Global และยัง multicast capabilities ไปยัง Frame Relay networksด้วย

   _____

   _____

4. Why would you use subinterfaces with Frame Relay?

   _subinterfaces ทำให้ meshed network ถูกแบ่งออกไปเป็น จำนวนที่น้อยลงได้ โดยทั่วไปแล้วแต่ละ point-to-point subnetwork

   _จะถูกกำหนดด้วย unique network address ทำให้ packet ที่ได้รับจาก 1 physical interfaceถูกส่งต่อไปภายใน interface เดียวได้

   _____

   _____