

# 实验(十二)：帧中继配置实验

## 一.实验目的

- 广域网是另一类主要有线物理网络，可以实现跨地域的网络连接，承担着骨干传输网络作用，只有ISP(Internet Service Provider，互联网服务提供商)才会拥有，如中国电信、移动等，普通企业很难见到此类设备。因此本书没有对广域网进行详尽介绍，但了解广域网网络和局域网互联，有助于深入理解网际网的异构特性。本实验利用路由器模拟帧中继交换机，用于远程连接两个以太网，实现网络互联。
  - 熟悉帧中继(Frame Relay)的配置方法
  - 理解帧中继的基本原理和特点
  - 验证帧中继在网络中的工作效果

## 二.实验原理

- 帧中继配置实验的原理主要包括以下几点：
  - **面向连接的数据链路技术**：帧中继技术基于虚电路（VC）概念。每个虚电路通过唯一的数据链路连接标识符（DLCI）进行标识，确保了数据传输前必须建立连接，然后在此连接上交换数据。
  - **DLCI的局部意义**：DLCI用于在帧中继网络中本地标识虚电路。它标识路由器的接口和交换机中的路径，但DLCI在帧中继网络中并非全局唯一，其标识作用是局部的。
  - **多路复用**：帧中继允许多个虚电路在同一物理链路上多路复用，有效地使用带宽资源，减少通信成本。
  - **虚电路类型**：帧中继网络支持两种虚电路类型：永久虚电路（PVC）和交换虚电路（SVC）。PVC是持久存在的，而SVC是根据需要建立并在数据传输完成后撤销的临时连接。
  - **数据转发和地址映射**：帧中继交换机使用映射表来将收到的帧中的DLCI映射到相应的输出端口。同时，路由器间的地址映射可以是手动配置的或通过动态地址映射实现，后者通常通过帧中继ARP（反向ARP）完成。
  - **封装选项**：在连接Cisco路由器时，默认封装是Cisco专有格式。然而，与非Cisco路由器连接时应选择国际标准组织IETF推荐的封装方式。
- 综合上述原理，帧中继为广域网络提供了一种高效、灵活且可靠的数据传输方案，尤其适用于需要高带宽和多路复用功能的场合。实验中，配置帧中继就涉及到设定DLCI值、确定封装类型、配置地址映射，以及理解和应用PVC或SVC虚电路。
- 帧中继网络拓扑结构示意图如下：

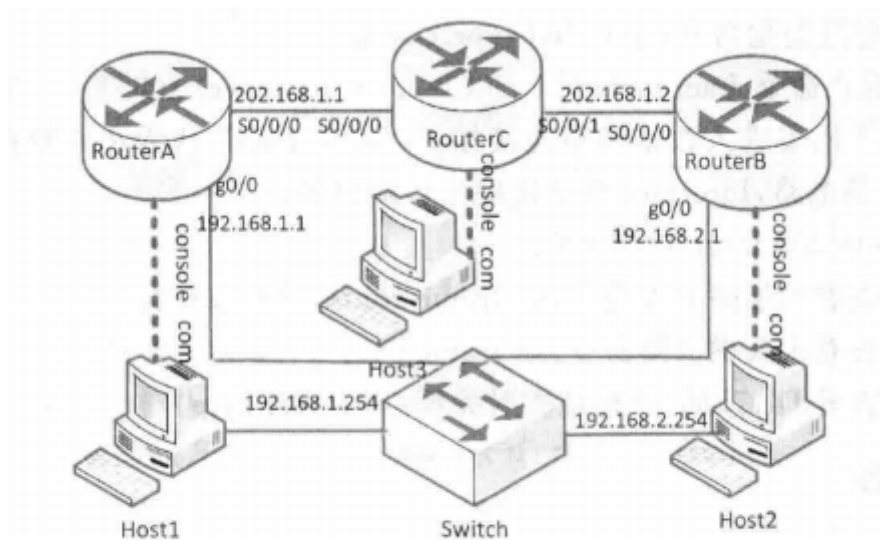


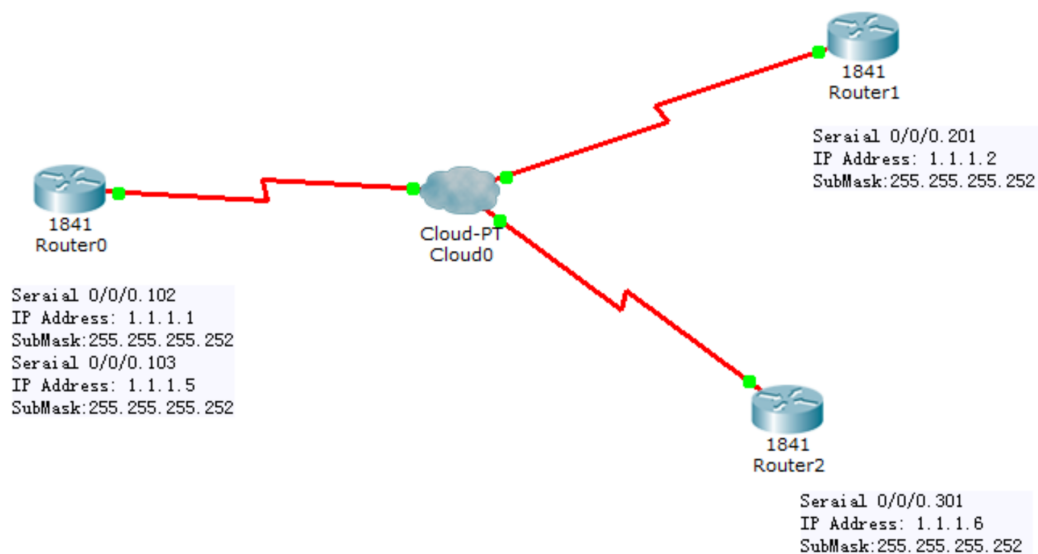
图 4-26 帧中继网络实验拓扑结构

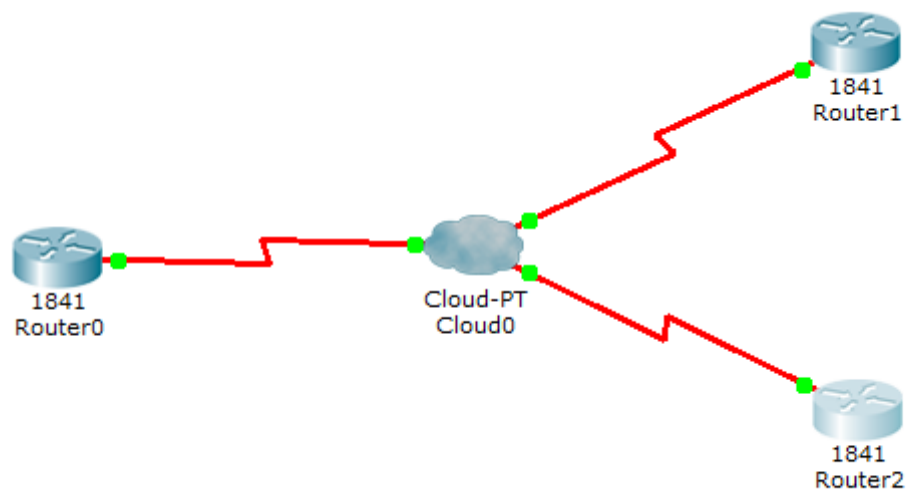
### 三.实验环境

- 操作系统：Windows 10
- 网络环境：局域网
- 软件：Cisco Packet Tracer虚拟实验环境

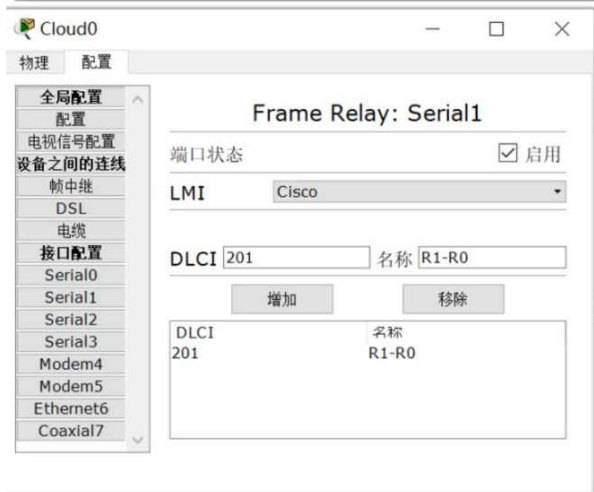
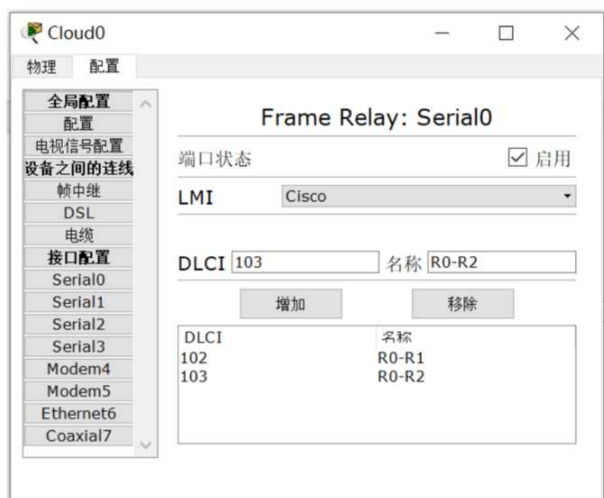
### 四.实验步骤

- 按照下图所示，连线构成网络；在连接时，需要注意，需要在路由器上增加WIC-1T接口卡，以提供串行端口





- 进行帧中继交换机配置（配置交换机帧中继）
  - 需要在 serial0、Serial11、Serial12 中配置，为 DLCI 命名，指示从一个端口的子链接到另一个端口的子链接的连接关系



### Frame Relay

Serial0 ▾ R0-R2 ▾ <-> Serial2 ▾ R2-R0 ▾

端口	子链路	端口	子链路
来源端口	子链路	目标端口	子链路
Serial0	R0-R1	Serial1	R1-R0
Serial0	R0-R2	Serial2	R2-R0

增加
删除

- 依次在CLI中输入以下命令以配置 Router0

```
// 进入特权模式。
R0>enable
// 进入全局配置模式。
R0#configure terminal
// 进入指定的串行接口配置模式。
R0(config)#interface serial 0/0/0
// 启动接口，使其变为活跃状态，这样接口就可以传输数据了。
R0(config-if)#no shutdown
// 为接口设置封装类型为帧中继。帧中继是一种WAN封装技术，用于在WAN链接上传输数据。
```

```

R0(config-if)#encapsulation frame-relay
// 退出接口配置模式，返回到全局配置模式。
R0(config-if)#exit
// 创建一个点对点类型的子接口，用于帧中继连接的配置。这里的.102表示子接口编号，通常与DLCI值相关
联。
R0(config)#interface Serial 0/0/0.102 point-to-point
// 为子接口配置IP地址和子网掩码。这个IP地址在该点对点链接的一端。
R0(config-subif)#ip address 1.1.1.1 255.255.255.252
// 为子接口设置帧中继DLCI编号，这是子接口在帧中继网络中的标识。
R0(config-subif)#frame-relay interface-dlci 102
// 退出子接口配置模式，返回到全局配置模式。
R0(config-subif)#exit
// 创建另一个点对点子接口，类似上面的步骤，为另一条帧中继虚拟连接进行配置。
R0(config)#interface Serial 0/0/0.103 point-to-point
// 为这个子接口配置IP地址和子网掩码，这个IP地址在这条点对点链接的另一端。
R0(config-subif)#ip address 1.1.1.5 255.255.255.252
// 为这个子接口设置帧中继DLCI编号。
R0(config-subif)#frame-relay interface-dlci 103

```

- 依次在CLI中输入以下命令以配置Router1

```

R1>enable
R1#configure terminal
R1(config)#interface Serial 0/0/0
R1(config-if)#no shutdown
R1(config-if)#encapsulation frame-relay
R1(config-if)#exit
R1(config)#interface Serial 0/0/0.201 point-to-point
R1(config-subif)#ip address 1.1.1.2 255.255.255.252
R1(config-subif)#frame-relay interface-dlci 201

```

- 依次在CLI中输入以下命令以配置Router2

```

R2>enable
R2#configure terminal
R2(config)#interface Serial 0/0/0
R2(config-if)#no shutdown
R2(config-if)#encapsulation frame-relay
R2(config-if)#exit
R2(config)#interface Serial 0/0/0.301 point-to-point
R2(config-subif)#ip address 1.1.1.6 255.255.255.252
R2(config-subif)#frame-relay interface-dlci 301

```

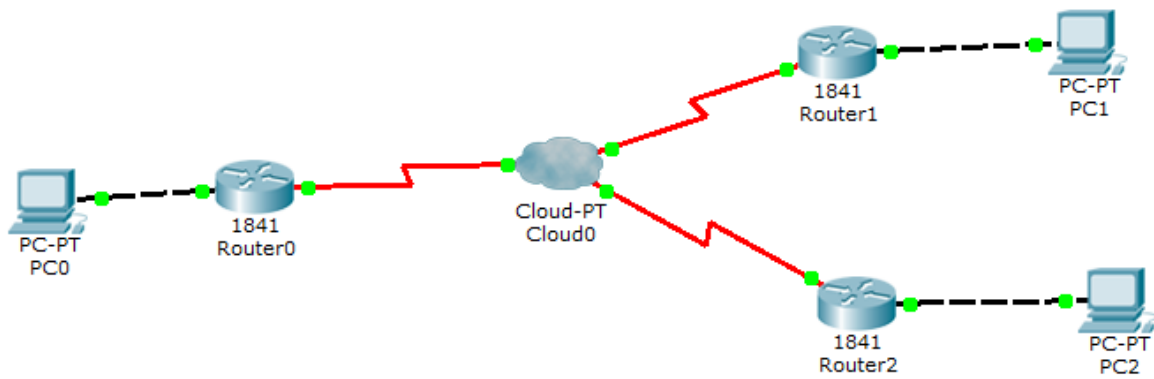
- 检查接口间能否相互ping通，观察现象
- 为Router1、Router2进行静态路由配置

```

ip route 1.1.1.4 255.255.255.252 1.1.1.1
ip route 1.1.1.0 255.255.255.252 1.1.1.5

```

- 再次检查接口间能否相互ping通，观察现象
- 创建三个主机PC0、PC1、PC2，分别连接至Router0、Router1、Router2的FastEthernet0/0口上



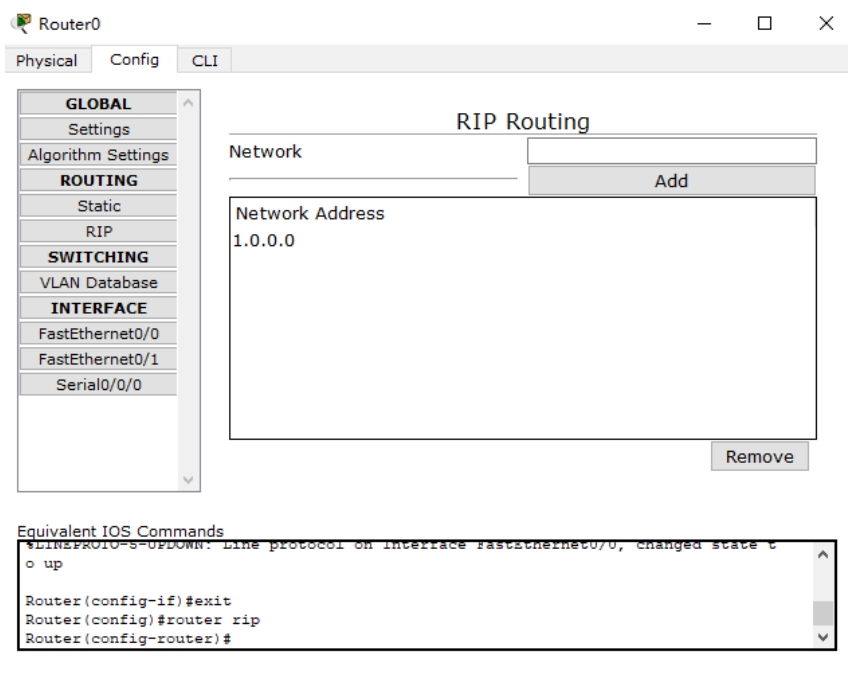
- 再次配置Router0、Router1、Router2

- Router0

- 在CLI中输入

```
interface FastEthernet0/0
ip address 1.1.1.9 255.255.255.252
```

- 在RIP路由协议中添加1.0.0.0



- Router1

- 在CLI中输入

```
interface FastEthernet0/0
ip address 1.1.1.13 255.255.255.252
```

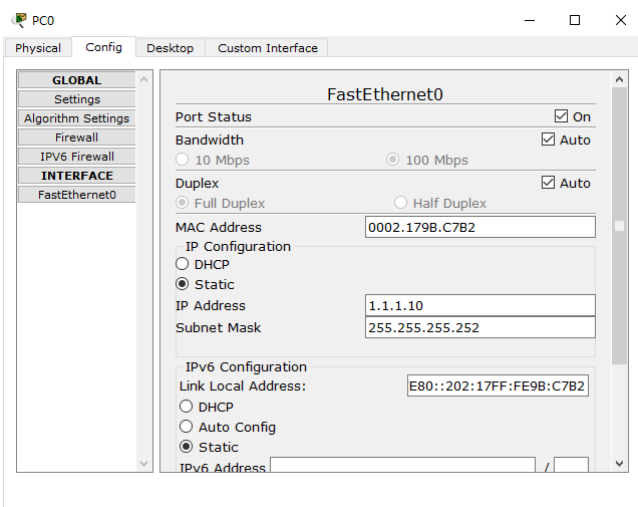
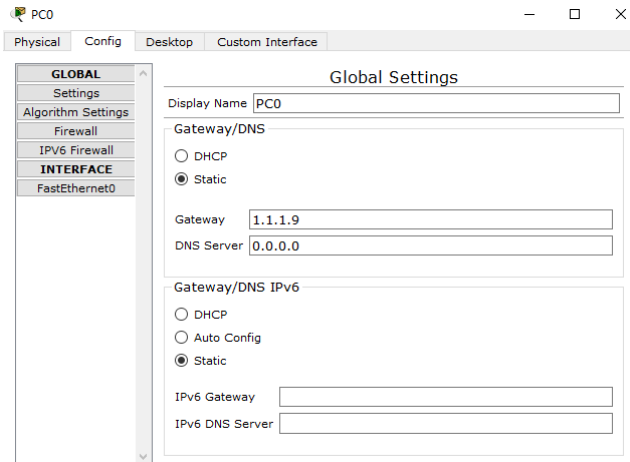
- 在RIP路由协议中添加1.0.0.0

- Router2

- 在CLI中输入

```
interface FastEthernet0/0
ip address 1.1.1.17 255.255.255.252
```

- 在RIP路由协议中添加1.0.0.0
- 对PC0、PC1、PC2进行配置
  - PC0
    - 网关：1.1.1.9
    - FastEthernet0：1.1.1.10 255.255.255.252



- PC1
  - 网关：1.1.1.13
  - FastEthernet0：1.1.1.14 255.255.255.252
- PC2
  - 网关：1.1.1.17
  - FastEthernet0：1.1.1.18 255.255.255.252
- PC0、PC1、PC2 相互 ping，观察现象

## 五、实验现象

- 最开始配置好交换机帧中继与路由器后发现：Router0 分别与 Router1 和 Router2 之间能够相互 ping 通，但是 Router1 与 Router2 之间不能够 ping 通

Router2

Physical Config CLI

### IOS Command Line Interface

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0.301, changed state t
o up

Router(config-subif)#ip address 1.1.1.6 255.255.255.252
Router(config-subif)#frame-relay interface-dlci 301
Router(config-subif)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#ping 1.1.1.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.5, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

Router#ping 1.1.1.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.5, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 9/12/18 ms

Router#
```

未启用时ping不成功

启用后ping成功

Copy Paste

Router1

Physical Config CLI

### IOS Command Line Interface

```
Success rate is 0 percent (0/5)

Router#IP
Router#ip route 1.1.1.4 255.255.255.252 1.1.1.1
^
% Invalid input detected at '^' marker.

Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 1.1.1.4 255.255.255.252 1.1.1.1
Router(config)#ping 1.1.1.6
^
% Invalid input detected at '^' marker.

Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#ping 1.1.1.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.6, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

Router#
```

Copy Paste

- 为 Router1 与 Router2 配置好静态路由后可以相互 ping 通



Router1

Physical Config CLI

### IOS Command Line Interface

```
Success rate is 0 percent (0/5)

Router#IP
Router#ip route 1.1.1.4 255.255.255.252 1.1.1.1
^
% Invalid input detected at '^' marker.

Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 1.1.1.4 255.255.255.252 1.1.1.1
Router(config)#ping 1.1.1.6
^
% Invalid input detected at '^' marker.

Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#ping 1.1.1.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.6, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/9/19 ms

Router#
```

Copy Paste

Router2

Physical Config CLI

### IOS Command Line Interface

```
Sending 5, 100-byte ICMP Echos to 1.1.1.5, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

Router#ping 1.1.1.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.5, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 9/12/18 ms

Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 1.1.1.0 255.255.255.252 1.1.1.5
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#ping 1.1.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.2, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/13/24 ms

Router#
```

Copy Paste

- PC0、PC1、PC2 之间相互ping均成功

PC0 PC1 PC2

Physical Config Desktop Custom Interface

### Command Prompt

```
Pinging 1.1.1.13 with 32 bytes of data:
Reply from 1.1.1.13: bytes=32 time=2ms TTL=254
Reply from 1.1.1.13: bytes=32 time=2ms TTL=254
Reply from 1.1.1.13: bytes=32 time=3ms TTL=254
Reply from 1.1.1.13: bytes=32 time=12ms TTL=254

Ping statistics for 1.1.1.13:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 12ms, Average = 6ms

PC>ping 1.1.1.17

Pinging 1.1.1.17 with 32 bytes of data:
Reply from 1.1.1.17: bytes=32 time=1ms TTL=254
Reply from 1.1.1.17: bytes=32 time=6ms TTL=254
Reply from 1.1.1.17: bytes=32 time=2ms TTL=254
Reply from 1.1.1.17: bytes=32 time=17ms TTL=254

Ping statistics for 1.1.1.17:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 17ms, Average = 9ms

PC>
```

PC1

```
Pinging 1.1.1.17 with 32 bytes of data:
Reply from 1.1.1.17: bytes=32 time=18ms TTL=253
Reply from 1.1.1.17: bytes=32 time=12ms TTL=253
Reply from 1.1.1.17: bytes=32 time=16ms TTL=253
Reply from 1.1.1.17: bytes=32 time=4ms TTL=253

Ping statistics for 1.1.1.17:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 4ms, Maximum = 18ms, Average = 12ms

PC>ping 1.1.1.9

Pinging 1.1.1.9 with 32 bytes of data:
Reply from 1.1.1.9: bytes=32 time=2ms TTL=254
Reply from 1.1.1.9: bytes=32 time=2ms TTL=254
Reply from 1.1.1.9: bytes=32 time=2ms TTL=254
Reply from 1.1.1.9: bytes=32 time=6ms TTL=254

Ping statistics for 1.1.1.9:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 6ms, Average = 3ms

PC>
```

PC2

```
Pinging 1.1.1.13 with 32 bytes of data:
Reply from 1.1.1.13: bytes=32 time=13ms TTL=253
Reply from 1.1.1.13: bytes=32 time=14ms TTL=253
Reply from 1.1.1.13: bytes=32 time=4ms TTL=253
Reply from 1.1.1.13: bytes=32 time=4ms TTL=253

Ping statistics for 1.1.1.13:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 4ms, Maximum = 14ms, Average = 8ms

PC>ping 1.1.1.9

Pinging 1.1.1.9 with 32 bytes of data:
Reply from 1.1.1.9: bytes=32 time=6ms TTL=254
Reply from 1.1.1.9: bytes=32 time=4ms TTL=254
Reply from 1.1.1.9: bytes=32 time=4ms TTL=254
Reply from 1.1.1.9: bytes=32 time=2ms TTL=254

Ping statistics for 1.1.1.9:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 6ms, Average = 4ms

PC>
```

## 六、实验结论

- 通过本次实验，我成功地配置了帧中继网络，模拟了广域网环境，并实现了不同局域网之间的互联。实验中，路由器通过配置了不同的DLCI和相应的子接口，确保了数据可以在不同的网络中正确地路由和交换。在实验的开始，我们观察到 Router0 能够与 Router1 和 Router2 分别进行通信，但是 Router1 与 Router2 之间无法直接通信。这是由于尚未配置静态路由，使得两个路由器无法找到对方的网络路径。配置了静态路由之后，Router1 与 Router2 之间可以成功通信，说明正确的路由配置对于网络互联是至关重要的。实验的最后阶段，在主机 PC0、PC1、PC2 之间的相互通信也验证了帧中继配置的正确性和网络互通性。通过正确配置，数据包能够在经过不同的网络设备后到达目的地，这显示了帧中继技术在连接异构网络中的有效性。
- 总结来说，帧中继技术是连接广域网中不同局域网的有效手段，能够处理不同网络间的数据传输。本实验加深了对帧中继技术工作原理的理解，并通过实际操作提升了网络配置和故障排除的实践技能。