第 8 章 帧中继

帧中继线路是中小企业常用的广域网线路，其通信费用较低。由于帧中继技术的一些特殊性使得帧中继的配置较为复杂，特别是在帧中继上运行路由协议时更是如此。作为入门，对帧中继的理解应着重放在 DLCI、PVC、帧中继映射和子接口等概念上。本章通过几个实验详细介绍了帧中继的关键概念。

8.1 帧中继简介

8.1.1 什么是帧中继

帧中继（Frame Relay，FR）是面向连接的第二层传输协议，帧中继是典型的包交换技术。相比而言，同样带宽的帧中继通信费用比 DDN 专线要低，而且允许用户在帧中继交换网络比较空闲的时候以高于 ISP 所承诺的速率进行传输。

8.1.2 帧中继的合理性

用户经常需要租用线路把分散在各地的网络连接起来，如图 8-1（1），如果采用点到点的专用线路（例如 DDN），ISP 需要给每个地方的路由器拉 4 对物理线路，同时每个路由器需要有 4 个串口。而帧中继网络拓扑如 8-1（2）所示，每个路由器只通过一条线路连接到帧中继云上，线路的代价大大减低，每个路由器也只需要一个串行接口了。

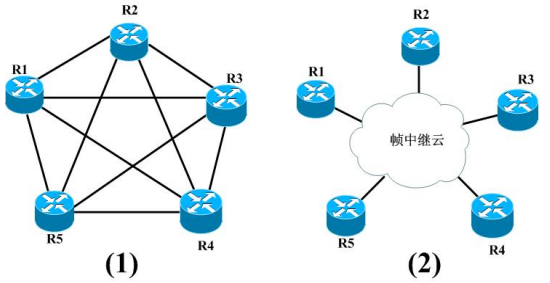

图 8-1　（1）用专线连接用户设备　（2）帧中继网络拓扑

8.1.3 DLCI

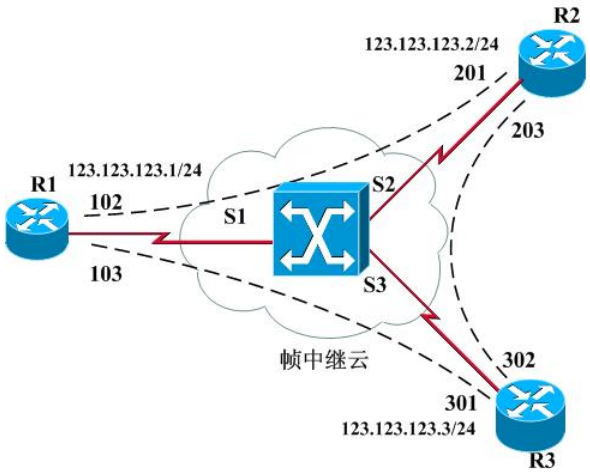

图 8-2 帧中继网络

DLCI（Data Link Circiut Identification，数据链路连接标识符）实际上就是帧中继网络中的第 2 层地址。如图 8-2，当路由器 R1 要把数据发向路由器 R2（IP 为 123.123.123.2）时，路由器 R1 可以用 DLCI=102 来对 IP 数据包进行第 2 层的封装。数据帧到了帧中继交换机，帧中继交换机根据帧中继交换表进行交换：从 S1 接口收到一个 DLCI 为 102 的帧时，交换机将把帧从 S2 接口发送出去，并且发送出去的帧的 DLCI 改为 201。这样路由器 R2 就会接收到 R1 发来的数据包。而当路由器 R2 要发送数据给 R1（IP 为 123.123.123.1）时，路由器 R2 可以用 DLCI=201 来对 IP 数据包进行第 2 层的封装，数据帧到了帧中继交换机，帧中继交换机同样根据帧中继交换表进行交换：从 S2 接口收到一个 DLCI 为 201 的帧时，交换机将把帧从 S1 接口发送出去，并且发送出去的帧的 DLCI 改为 102。这样路由器 R1 就会接收到 R2 发来的数据包。

通过以上分析可以知道DLCI实际上就是IP数据包在帧中继链路上进行封装时所需的第 2 层地址。图 8-2 各路由器中的第 3 层地址和第 2 层地址映射如下：

R1： 123.123.123.2→102
　　123.123.123.3→103
R2： 123.123.123.1→201
　　123.123.123.3→203
R3： 123.123.123.1→301
　　123.123.123.2→302

帧中继的一个非常重要特性是 NBMA（非广播多录访问）。在图 8-2 中，如果路由器在 DLCI 为 102 的 PVC 上发送一个广播，R2 路由器可以收到，然而 R3 是无法收到的。如果 R1 想发送的广播让 R2 和 R3 都收到，必须分别在 DLCI 为 102 和 103 的 PVC 上各发送一次，这就是非广播的含义。多路访问的意思是帧中继网络是多个设备接在同一网络介质上，以太网也是多路访问网络。

8.1.4 帧中继术语

(1) 永久虚电路（PVC）：虚电路是永久建立的链路，由 ISP 在其帧中继交换机静态配置交换表实现。不管电路两端的设备是否连接上，它总是为它保留相应的带宽。
(2) 数据链路连接标识符（DLCI）：一个在路由器和帧中继交换机之间标识 PVC 或者 SVC 的数值。
(3) 本地管理接口（LMI）：是路由器和帧中继交换机之间的一种信令标准，负责管理设备之间的连接及维护其连接状态。
(4) 承诺信息速率（CIR，Committed Information Rate）：也叫保证速率，是服务提供商承诺将要提供的有保证的速率，一般为一段时间内（承诺速率测量间隔 T）的平均值，其单位为 bps。
(5) 超量突发（EB，Excess Brust）：在承诺信息速率之外，帧中继交换机试图发送而未被准许的最大额外数据量，单位为 bit。超量突发依赖于服务提供商提供的服务状况，但它通常受到本地接入环路端口速率的限制。

8.1.5 LMI

LMI（Local Management Interface）提供了一个帧中继交换机和路由器之间的简单信令。在帧中继交换机和路由器之间必须采用相同的 LMI 类型，Cisco 路由器在较高版本(11.2 以后)的 IOS 中具有自动检测 LMI 类型的功能。配置接口 LMI 类型的命令为“**encapsulation frame-relay [*cisco* | *ietf*]**” 。路由器从帧中继交换机收到 LMI 信息后，可以得知 PVC 状态。三种 PVC 状态是：

- 激活状态（Active）：本地路由器与帧中继交换机的连接是启动且激活的。可以与帧中继交换机交换数据。
- 非激活状态（Inactive）：本地路由器与帧中继交换机的连接是启动且激活的，但 PVC 另一端的路由器未能与它的帧中继交换机通信。
- 删除状态（Deleted）：本地路由器没有从帧中继交换机上收到任何 LMI，可能线路或网络有问题，或者配置了不存在的 PVC。

8.1.6 帧中继映射

DLCI 是帧中继网络中的第 2 层地址。路由器要通过帧中继网络把 IP 数据包发到下一跳路由器时，它必须知道 IP 和 DLCI 的映射才能进行帧的封装。有两种方法可以获得该映射：一种是静态映射，由管理员手工输入；另一种是动态映射。默认时，路由器帧中继接口是开启动态映射的。

1．静态映射

管理员手工输入的映射就为静态映射，其命令为：

frame-relay map ip ***protocol address dlci*** **[broadcast]**

其中：

protocol：协议类型

address：网络地址

dlci：为所需要交换逆向 ARP 信息的本地接口的 DLCI 号

broadcast：参数表示允许在帧中继线路上传送路由广播或组播信息

例子：R1(config-if)**#frame map ip 123.123.123.2 102 broadcast**

2．动态映射

IARP（Inverse ARP，逆向 ARP）允许路由器自动建立帧中继映射，其工作原理如图 8-3 所示：

(1) R1 路由器从 DLCI=102 的 PVC 上发送 IARP 包,IARP 包中有 R1 的 IP 地址 12.12.12.1 ；

(2) 帧中继云对数据包进行交换，最终把 IARP 包通过 DLCI=201 的 PVC 发送给 R2；

(3) 由于 R2 是从 201 的 PVC 上接收到该 IARP 包，R2 就自动建立一个映射：

12.12.12.1——→201

(4) 同样 R2 也发送 IARP 数据包，R1 收到该 IARP 包，也会自动建立一个映射：

12.12.12.2——→102

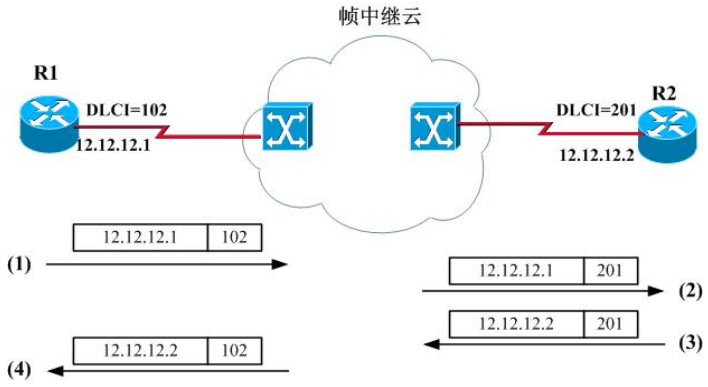

图 8-3 动态映射(IARP)工作示意图

8.1.7 子接口

子接口实际上是一个逻辑的接口，并不存在真正物理上的子接口。子接口有两种类型：

点到点、点到多点。采用点到点子接口时，每一个子接口用来连接一条 PVC，每条 PVC 的另一端连接到另一路由器的一个子接口或物理接口。这种子接口的连接与通过物理接口连接的点对点连接效果是一样的。每一对点对点的连接都是在不同的子网。

一个点到多点子接口被用来建立多条 PVC，这些 PVC 连接到远端路由器的多个子接口或物理接口。这时，所有加入连接的接口（不管是物理接口还是子接口）都应该在同一个子网上。点到多点子接口和一个没有配置子接口的物理主接口相同，路由更新要受到水平分割的限制。默认时多点子接口水平分割是开启的。

8.2 实验 1:把一台 Cisco 路由器配置为帧中继交换机

1. 实验目的

通过本实验，读者可以掌握如下技能：

（1） 理解帧中继交换表的工作原理

（2） 理解 PVC 的概念

（3） 用路由器充当帧中继交换机的配置

2. 实验拓扑

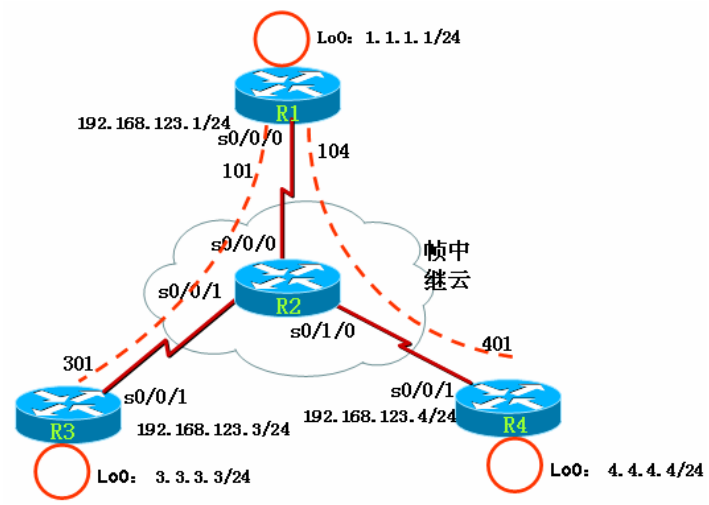

图 8-4 实验 1--实验 4 拓扑图

3. 实验步骤

我们这里只关心 R2 的配置。

(1) 步骤 1：开启帧中继交换功能

```
R2(config)#frame-relay switching  //注：把该路由器当成帧中继交换机
```

(2) 步骤 2：配置接口封装

```
R2(config)#int s0/0/0
R2(config-if)#no shutdown
R2(config-if)#clock rate 128000    //注：该接口为 DCE，要配置时钟
R2(config-if)#encapsulation frame-relay
```

//“encapsulation frame-relay [ietf]”命令用来配置接口封装成帧中继，如果不加 ietf 参数，帧类型为 cisco；如果加 ietf 参数，则帧类型为 ietf。

```
R2(config)#int s0/0/1
R2(config-if)#no shutdown
R2(config-if)#clock rate 128000
R2(config-if)#encapsulation frame-relay
R2(config)#int s0/1/0
R2(config-if)#no shutdown
R2(config-if)#clock rate 128000
R2(config-if)#encapsulation frame-relay
```

(3) 步骤 3：配置 LMI 类型

```
R2(config)#int s0/0/0
R2(config-if)#frame-relay lmi-type cisco
```

//命令“frame-relay lmi-type { ansi | cisco | q933a }用来配置 LMI 的类型，默认时是 cisco 。

```
R2(config-if)#frame-relay intf-type dce
```

//命令“frame-relay intf-type { dce | dte }”用来配置接口是帧中继的 DCE 还是 DTE，要注意的是：这里的帧中继接口 DCE 和 s0/0/0 接口是 DCE 还是 DTE 无关，也就是说即使 s0/0/0 是 DTE，也可以把它配置成帧中继的 DCE。

```
R2(config)#int s0/0/1
R2(config-if)#frame-relay lmi-type cisco
R2(config-if)#frame-relay intf-type dce
R2(config)#int s0/1/0
R2(config-if)#frame-relay lmi-type cisco
R2(config-if)#frame-relay intf-type dce
```

(4) 步骤 4：配置帧中继交换表

```
R2(config)#int s0/0/0
R2(config-if)#frame-relay route 103 interface s0/0/1 301
R2(config-if)#frame-relay route 104 interface s0/1/0 401
```

//命令“frame-relay route 103 interface s0/0/1 301”是配置帧中继交换表的，告诉路由器如果从该接口收到 DLCI=103 的帧，要从 s0/0/1 交换出去，并且 DLCI 改为 301。

```
R2(config)#int Serial0/0/1
R2(config-if)#frame-relay route 301 interface Serial0/0/0 103
R2(config)#int Serial0/1/0
R2(config-if)#frame-relay route 401 interface Serial0/0/0 104
```

4. 实验调试

可以使用“show frame-relay route”、“show frame pvc”、“show frame lmi”等命令检查帧中继交换机是否正常。

（1） “show frame-relay route”

```
R2#show frame-relay route
Input Intf      Input Dlci      Output Intf     Output Dlci     Status
```

```
Serial0/0/0      103             Serial0/0/1      301             inactive
Serial0/0/0      104             Serial0/1/0      401             inactive
Serial0/0/1      301             Serial0/0/0      103             inactive
Serial0/1/0      401             Serial0/0/0      104             inactive
```

（2） “show frame pvc”

```
R2#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             2             0            0
  Unused         0             0             0            0
DLCI = 103, DLCI USAGE = SWITCHED, PVC STATUS = INACTIVE, INTERFACE = Serial0/0/0
//由于 PVC 还未被使用，所以状态为 inactive
  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
  pvc create time 00:06:05, last time pvc status changed 00:06:05
```

（此处省略）

8.3 实验 2:帧中继基本配置、帧中继映射

1. 实验目的

通过本实验，读者可以掌握如下技能：

（1） 帧中继的基本配置

（2） 帧中继的动态映射

（3） 帧中继的静态映射

2. 实验拓扑

如图 8-4。

3. 实验步骤

在实验 1 的基础上进行实验 2。图 8-4 中，我们已经模拟出了帧中继交换机，现配置 R1、R3、R4，使得它们能够互相通信，配置步骤如下：

(1)　帧中继接口基本配置

```
R1(config)#int s0/0/0
R1(config-if)#ip address 192.168.123.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#encapsulation frame-relay
```

//使用命令“encapsulation frame-relay [*ietf*]”。帧中继有两种封装类型：cisco 和 ietf（Internet Engineering Task Force)。对于 cisco 路由器，cisco 是它的默认值；对于非 cisco 路由器，须选用 ietf 类型。但国内帧中继线路一般为 ietf 类型的封装，我们这里由于上面的帧中继交换机中封装类型是 cisco，所以这里选择 cisco。

```
R1(config-if)#frame-relay lmi-type cisco
```

//如果采用的是 cisco 路由器且 IOS 是 11.2 及以后版本的，路由器可以自动适应 LMI 的类型，则本步骤可不做。国内帧中继线路一般采用 ansi 的 LMI 信令类型，我们这里采用的是 cisco。

```
R3(config)#int s0/0/1
R3(config-if)#ip address 192.168.123.3 255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#encapsulation frame-relay

R4(config)#int s0/0/1
R4(config-if)#ip address 192.168.123.4 255.255.255.0
R4(config-if)#no shutdown
R4(config-if)#encapsulation frame-relay
```

(2)　测试连通性

从各个路由器 ping 其他路由器：

```
R1#ping 192.168.123.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.123.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
R1#ping 192.168.123.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.123.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms

R1#show frame-relay map
Serial0/0/0 (up): ip 192.168.123.3 dlci 103(0x67,0x1870), dynamic,
              broadcast,, status defined, active
Serial0/0/0 (up): ip 192.168.123.4 dlci 104(0x68,0x1880), dynamic,
              broadcast,, status defined, active
```

//默认时，帧中继接口开启了动态映射，会自动建立帧中继映射，“dynamic”表明这是动态映射。

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

//可以看到DLCI=103的PVC的状态为active

input pkts 11 output pkts 11 in bytes 1074

out bytes 1074 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 1 out bcast bytes 34

5 minute input rate 0 bits/sec, 0 packets/sec

5 minute output rate 0 bits/sec, 0 packets/sec

pvc create time 00:07:31, last time pvc status changed 00:06:01

（此处省略）

(3) 手工配置帧中继映射

默认情况下，路由器支持逆向ARP。若逆向ARP未打开，可以用下列命令设置：

R1(config-if)#frame-relay inverse-arp

我们也可以关闭IARP，使用静态映射，命令如下：

“frame-relay map ip *address dlci* [broadcast]”

这里的broadcast参数是允许该帧中继链路通过多播或广播包,如果帧中继链路上要运行路由协议，该参数非常重要。

R1(config)#int s0/0/0

R1(config-if)#no frame-relay inverse-arp //关闭自动映射

R1(config-if)#frame-relay map ip 192.168.123.3 103 broadcast

R1(config-if)#frame-relay map ip 192.168.123.4 104 broadcast

R3(config)#int s0/0/1

R3(config-if)#no frame-relay inverse-arp

R3(config-if)#frame-relay map ip 192.168.123.1 301 broadcast

R4(config)#int s0/0/1

R4(config-if)#no frame-relay inverse-arp

R4(config-if)#frame-relay map ip 192.168.123.1 401 broadcast

4. 实验调试

可以使用“show frame-relay map”、“show frame pvc”、“show frame lmi”等命令检查帧中继交换机是否正常。

R1#show frame-relay map

Serial0/0/0 (up): ip 192.168.123.3 dlci 103(0x67,0x1870), dynamic,

broadcast,, status defined, active

从命令输出中可以得到的信息有：

- 192.168.123.3 映射到 103
- Dynamic：表明是动态映射
- Broadcast：该 PVC 允许广播包的通过
- Active：该 PVC 是激活的

该命令是很重要的一条命令，如果在映射表中不存在映射，路由器将无法通信。可以使用名命令“clear frame-relay inarp”命令清除无效的帧中继映射表。

```
R1#show frame-relay pvc
PVC Statistics for interface Serial0/0 (Frame Relay DTE)
DLCI = 103, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0
    input pkts 102024        output pkts 116191        in bytes 13974906
    out bytes 14707805       dropped pkts 0           in FECN pkts 287
    in BECN pkts 290         out FECN pkts 0          out BECN pkts 0
    in DE pkts 102024        out DE pkts 0
      pvc create time 1w1d, last time pvc status changed 1w1d
     ……
```

从命令输出中可以得到的信息有：

- DLCI = 103：表明该 PVC 的 DLCI 为 103
- PVC STATUS = ACTIVE：表明 PVC 的状态是激活的；若 PVC STATUS = INACTIVE——表明远端路由器没正确配置；若 PVC STATUS = DELETED——表明输入了错误的 DLCI，该 PVC 不存在。

```
R1#show frame-relay lmi
    LMI Statistics for interface Serial0/0 (Frame Relay DTE) LMI TYPE = CISCO
    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 74859             Num Status msgs Rcvd 74857
    Num Update Status Rcvd 0               Num Status Timeouts 2
```

从命令输出中可以得到的信息有：

- LMI TYPE = CISCO：表明帧中继 LMI 类型为 cisco；
- Frame Relay DTE：这是帧中继 DTE
- Num Status Enq. Sent 74859：表明路由器向帧中继交换机发送的 LMI 状态查询消息的数量；
- Num Status msgs Rcvd 74857：表明路由器从帧中继交换机收到的 LMI 状态信息的数量。

8.4 实验 3:帧中继上的 RIP

1. 实验目的

通过本实验，读者可以掌握如下技能：

（1） 帧中继上路由协议运行的特殊性

（2） 水平分割

2. 实验拓扑

如图 8-4。

3. 实验步骤

在实验 2 的基础上继续本实验。

（1） 配置 RIP：

```
R1(config)#interface Loopback0
R1(config-if)#ip address 1.1.1.1 255.255.255.0
R1(config)#router rip
R1(config-router)#network 1.0.0.0
R1(config-router)#network 192.168.123.0

R3(config)#interface Loopback0
R3(config-if)#ip address 3.3.3.3 255.255.255.0
R3(config)#router rip
R3(config-router)#network 3.0.0.0
R3(config-router)#network 192.168.123.0

R4(config)#interface Loopback0
R4(config-if)#ip address 4.4.4.4 255.255.255.0
R4(config)#router rip
R4(config-router)#network 4.0.0.0
R4(config-router)#network 192.168.123.0
```

（2） 检查路由表、测试

在各个路由器上检查路由表，并测试从环回口之间的互相 ping。

```
R3#show ip route
（此处省略）
C    192.168.123.0/24 is directly connected, Serial0/0/1
R    1.0.0.0/8 [120/1] via 192.168.123.1, 00:00:26, Serial0/0/1
     3.0.0.0/24 is subnetted, 1 subnets
C       3.3.3.0 is directly connected, Loopback0
R    4.0.0.0/8 [120/2] via 192.168.123.1, 00:00:26, Serial0/0/1
//看到正常的路由表，注意 RIP 路由表中的下一跳

R3#ping 4.4.4.4 source loopback 0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
Packet sent with a source address of 3.3.3.3
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/54/56 ms
```

//以上表明从 R3 的 loopback0 接口能 ping 通 R4 的 loopback0 接口

R3#ping 4.4.4.4

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
....
```

//这里 ping 命令没指明源地址，则 ICMP 数据包的源 IP 为 192.168.123.3，目标为 4.4.4.4。R3 路由表查询路由表得知该数据包应该发送给 192.168.123.1，而 192.168.123.1 的帧中继映射 DLCI 为 301；数据包到达 R1，R1 路由表查询路由表得知该数据包应该发送给 192.168.123.4，而 192.168.123.4 的帧中继映射 DLCI 为 104。R4 收到数据包，进行响应，ICMP 数据包的源 IP 为 4.4.4.4，目标为 192.168.123.3；R4 有 192.168.123.0/24 的直连路由，然而却没有 192.168.123.3 的帧中继映射，因此无法进行封装。为了解决该问题，可以在 R4 中增加映射：

R4(config)#int s0/0/1

R4(config-if)#frame-relay map ip 192.168.123.3 401

这样从 R3 就可以直接 ping 通 R4 的 loopback0 接口了。

（3） 水平分割问题

R1#show ip int s0/0/0

```
Serial0/0/0 is up, line protocol is up
  Internet address is 192.168.123.1/24
（此处省略）
  Security level is default
  Split horizon is disabled      //接口封装了帧中继后，水平分割被自动关闭
  ICMP redirects are always sent
   （此处省略）
```

在 R1 上重新打开水平分割，在各路由器上检查路由表。

R1(config)#int Serial0/0/0

R1(config-if)#ip split-horizon

R1#clear ip route *   //清除路由表

R1#show ip route

```
C    192.168.123.0/24 is directly connected, Serial0/0/0
     1.0.0.0/24 is subnetted, 1 subnets
C       1.1.1.0 is directly connected, Loopback0
R    3.0.0.0/8 [120/1] via 192.168.123.3, 00:00:01, Serial0/0/0
R    4.0.0.0/8 [120/1] via 192.168.123.4, 00:00:01, Serial0/0/0
```

//R1 可以获得 R3 和 R4 的环回口路由

R3#clear ip route *

R3#show ip route

```
C    192.168.123.0/24 is directly connected, Serial0/0/1
R    1.0.0.0/8 [120/1] via 192.168.123.1, 00:00:00, Serial0/0/1
     3.0.0.0/24 is subnetted, 1 subnets
C       3.3.3.0 is directly connected, Loopback0
```

//R3 只能获得 R1 的环回口路由，这是由于 R1 上的水平分割开启后，R1 从 R4 接收到 R4 公告的路由后，不从帧中继口发送出来，导致 R3 没有接收到 R4 上公告的路由

```
R3#clear ip route *
R4#show ip route
C    192.168.123.0/24 is directly connected, Serial0/0/1
R    1.0.0.0/8 [120/1] via 192.168.123.1, 00:00:01, Serial0/0/1
     4.0.0.0/24 is subnetted, 1 subnets
C       4.4.4.0 is directly connected, Loopback0
```

//R4 也只能获得 R1 的环回口路由

8.5 实验 4:帧中继点到多点子接口

1. 实验目的

通过本实验，读者可以掌握如下技能：

（1） 点到多点子接口的配置

2. 实验拓扑

如图 8-4，R3 和 R4 使用帧中继主接口，在 R1 上创建点到多点子接口。

3. 实验步骤

在实验 1 的基础上继续本实验。

(1) 对主接口进行配置

```
R1(config)#interface serial0/0/0
R1(config-if)#no ip address     //注：主接口下不需要 IP 地址
R1(config-if)#encap frame-relay      //注：封装帧中继
R1(config-if)#no frame-relay inverse-arp   //注：通常需要关闭主接口下的 IARP
R1(config-if)#no shutdown
```

(2) 创建点到多点子接口

```
R1(config)#int s0/0/0.1 multipoint     //注：创建点到多点子接口
```

//这里命令“interface serial *slot-number/interface-number.subinterface-number* { multipoint | point-to-point }”用来创建子接口，其中：

- *slot-number/interface-number*：即本物理接口的号码
- *subinterface-number*：是子接口号，用户可以根据自己喜好来确定
- multipoint 和 point-to-point：属于必须选择的项，是子接口的类型，要么是点到多点，要么是点到点。

```
R1(config-subif)#ip address 192.168.123.1 255.255.255.0
R1(config-subif)#frame-relay map ip 192.168.123.3 103 broadcast
R1(config-subif)#frame-relay map ip 192.168.123.4 104 broadcast
```

//以上是配置帧中继映射

(3) R1 上配置路由协议：

```
R1(config)#router rip
```

```
R1(config-router)#network 1.0.0.0
R1(config-router)#network 192.168.123.0
```
(4)　R3、R4 完整的配置如下：
```
R3(config)#interface serial 0/0/1
R3(config-if)#ip address 192.168.123.3 255.255.255.0
R3(config-if)#encapsulation frame-relay
R3(config-if)#no frame-relay inverse-arp
R3(config-if)#frame-relay map ip 192.168.123.1 301 broadcast
R3(config-if)＃no shutdown
R3(config)#router rip
R3(config-router)#network 3.0.0.0
R3(config-router)#network 192.168.123.0

R4(config)#interface serial 0/0/1
R4(config-if)#ip address 192.168.123.4 255.255.255.0
R4(config-if)#encapsulation frame-relay
R4(config-if)#no frame-relay inverse-arp
R4(config-if)#frame-relay map ip 192.168.123.1 401 broadcast
R4(config-if)＃no shutdown
R4(config)#router rip
R4(config-router)#network 4.0.0.0
R4(config-router)#network 192.168.123.0
```

【提示】可以使用“no interface s0/0/0.1”命令来删除子接口，然而需要重新启动路由器，该子接口才真正被删除。

4. 实验调试

在各个路由器上检查路由表，注意路由的下一跳。
```
R1#show ip route
（此处省略）
C    192.168.123.0/24 is directly connected, Serial0/0/0.1
     1.0.0.0/24 is subnetted, 1 subnets
C       1.1.1.0 is directly connected, Loopback0
R    3.0.0.0/8 [120/1] via 192.168.123.3, 00:00:19, Serial0/0/0.1
R    4.0.0.0/8 [120/1] via 192.168.123.4, 00:00:16, Serial0/0/0.1
```
//R1 可以获得 R3 和 R4 的环回口路由

```
R3#show ip route
（此处省略）
C    192.168.123.0/24 is directly connected, Serial0/0/1
R    1.0.0.0/8 [120/1] via 192.168.123.1, 00:00:01, Serial0/0/1
     3.0.0.0/24 is subnetted, 1 subnets
C       3.3.3.0 is directly connected, Loopback0
```
//R3 只能获得 R1 的环回口路由，这是由于默认时 R1 的点到多点子接口水平分割是开启的，

可以使用命令“ip split-horizon”在子接口下关闭水平分割。

```
R4#show ip route
（此处省略）
C     192.168.123.0/24 is directly connected, Serial0/0/1
R     1.0.0.0/8 [120/1] via 192.168.123.1, 00:00:01, Serial0/0/1
      4.0.0.0/24 is subnetted, 1 subnets
C        4.4.4.0 is directly connected, Loopback0
//R4 同样只能获得 R1 的环回口路由
```

8.6 实验 5:帧中继点到点子接口

1. 实验目的

通过本实验，读者可以掌握如下技能：

（1） 点到点子接口的配置

2. 实验拓扑

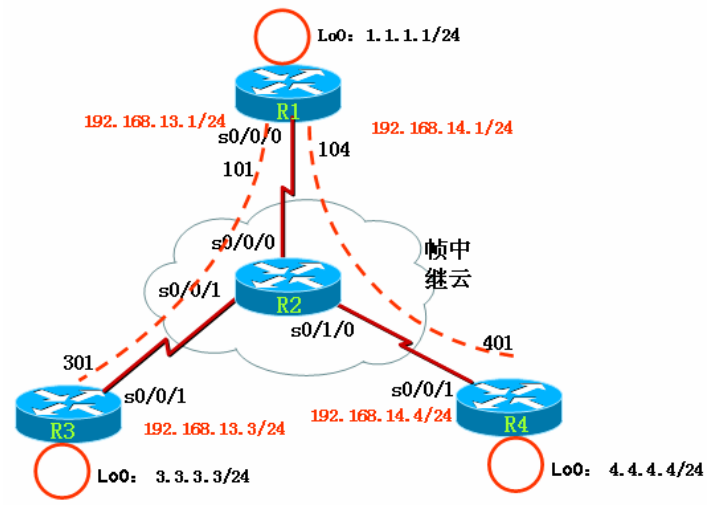

图 8-5 实验 5 拓扑图

3. 实验步骤

(1) 对主接口进行配置

```
R1(config)#interface serial0/0/0
R1(config-if)#no ip address
R1(config-if)#encap frame-relay
R1(config-if)#no frame-relay inverse-arp
R1(config-if)#no shutdown
```

(2) 创建两个点到点子接口

```
R1(config)#int s0/0/0.3 point-to-point     //注：创建点到点子接口
R1(config-subif)#ip address 192.168.13.1 255.255.255.0
R1(config-subif)#frame-relay interface-dlci 103
```

//在接口下不能使用“frame-relay map ip”命令来配置帧中继的映射，而是改用命令“frame-relay interface-dlci 103” 。

```
R1(config)#int s0/0/0.4 point-to-point
R1(config-subif)#ip address 192.168.14.1 255.255.255.0
R1(config-subif)#frame-relay interface-dlci 104
```

(3) R1 上配置路由协议：

```
R1(config)#router rip
R1(config-router)#network 1.0.0.0
R1(config-router)#network 192.168.13.0
R1(config-router)#network 192.168.14.0
```

(4) R3、R4 完整的配置如下：

```
R3(config)#interface serial 0/0/1
R3(config-if)#no ip address
R3(config-if)#encapsulation frame-relay
R3(config-if)#no frame-relay inverse-arp
R3(config-if)#no shutdown
R3(config-if)#exit
R3(config)#interface serial 0/0/1.1 point-to-point
R3(config-subif)#ip address 192.168.13.3 255.255.255.0
R3(config-subif)#frame-relay interface-dlci 301
R3(config-subif)#exit
R3(config)#router rip
R3(config-router)#network 3.0.0.0
R3(config-router)#network 192.168.13.0

R4(config)#interface serial 0/0/1
R4(config-if)#no ip address
R4(config-if)#encapsulation frame-relay
R4(config-if)#no frame-relay inverse-arp
R4(config-if)#no shutdown
R4(config-if)#exit
R4(config)#interface serial 0/0/1.1 point-to-point
R4(config-subif)#ip address 192.168.14.4 255.255.255.0
R4(config-subif)#frame-relay interface-dlci 401
R4(config-subif)#exit
R4(config)#router rip
R4(config-router)#network 4.0.0.0
R4(config-router)#network 192.168.14.0
```

4. 实验调试

在各个路由器上检查路由表，注意路由的下一跳。

```
R3#show ip route
```

（此处省略）

```
R    1.0.0.0/8 [120/1] via 192.168.13.1, 00:00:14, Serial0/0/1.1
C    192.168.13.0/24 is directly connected, Serial0/0/1.1
R    192.168.14.0/24 [120/1] via 192.168.13.1, 00:00:14, Serial0/0/1.1
     3.0.0.0/24 is subnetted, 1 subnets
C       3.3.3.0 is directly connected, Loopback0
R    4.0.0.0/8 [120/2] via 192.168.13.1, 00:00:14, Serial0/0/1.1
```

8.7 本章小结

本章简要介绍了帧中继技术的使用场合，重点介绍理解帧中继的关键术语：DLCI、PVC和帧中继映射。DLCI 实际上就是数据链路层地址，路由器在帧中继链路上发送数据包，要获得该地址才能封装帧。可以手工或者使用 Inverse-arp 来把网络层地址和 DLCI 进行映射。子接口是逻辑接口，子接口的引入使得在帧中继链路上运行路由协议变得容易，也使得帧中继的配置更加灵活。表 8-1 是本章出现的命令。

表 8-1 本章命令汇总

命令	作用
frame-relay switching	把路由器当成帧中继交换机
encapsulation frame-relay	接口封装成帧中继
frame-relay lmi-type cisco	配置 LMI 的类型
frame-relay intf-type dce	配置接口是帧中继的 DCE 还是 DTE
frame-relay route	配置帧中继交换表
show frame-relay route	显示帧中继交换表
show frame pvc	显示帧中继 PVC 状态
show frame lmi	显示帧中继 LMI 信息
show frame-relay map	查看帧中继映射
no frame-relay inverse-arp	关闭帧中继自动映射
ip split-horizon	打开水平分割
int s0/0/0.1 multipoint	创建点到多点子接口
int s0/0/0.3 point-to-point	创建点到点子接口
frame-relay interface-dlci 104	在点到点子接口上配置 DLCI