



警示

1. 实验报告如有雷同，雷同各方当次实验成绩均以 0 分计。
2. 当次小组成员成绩只计学号、姓名登录在下表中的。
3. 在规定时间内未上交实验报告的，不得以其他方式补交，当次成绩按 0 分计。
4. 实验报告文件以 PDF 格式提交。

院系	计算机学院	班 级	计科_1_班	组长	郝裕玮
学号	18329015	18325071	19335153		
学生	郝裕玮	张闯	马淙升		
实验分工					
郝裕玮	张闯	马淙升			
共同完成实验与实验报告	共同完成实验与实验报告	共同完成实验与实验报告			

【实验题目】RIP 路由协议实验

【实验目的】

1. 掌握在路由器上配置 RIPv2
2. 分析 RIPv1 和 RIPv2 的区别
3. 学习使用 debug 相关指令分析具体信息

【实验内容】

1. 在实验设备上完成 P243 实验 7-2 并测试实验网连通性。
2. 通过实验观察 RIP V1 和 V2 的区别（重点在 VLSM 上）给出分析过程与结果（实验 IP 采用 10.10.x.0 网段）
3. 学会使用 Debug ip packet 和 Debug ip rip 命令，并对 debug 信息做分析。
4. 观察试验拓扑中链路状态发生改变时路由表的前后信息对比及 debug 信息的变化。

【实验要求】

重要信息需给出截图，注意实验步骤的前后对比。

【实验记录】(如有实验拓扑请自行画出)

1. 在实验设备上完成 P243 实验 7-2 并测试实验网连通性。

【实验设备】

三层交换机 1 台, 路由器 2 台。

【实验拓扑】



本实验以 2 台路由器和 1 台三层交换机为例。交换机上划分有 VLAN 10 和 VLAN50, 其中 VLAN 10 用于连接路由器 R1,VLAN 50 用于连接校园网主机。

路由器分别命名为 R1 和 R2,路由器之间的串口采用 V35 DCE/DTE 电缆连接,DCE 端连接到路由器 R1。其拓扑结构如图 7-20 所示。

PC1 的 IP 地址和默认网关分别为 192.168.5.11 和 192.168.5.1,PC2 的 IP 地址和默认网关分别为 192.168.3.22 和 192.168.3.1,子网掩码都是 255.255.255.0。

【实验步骤】

分析:本实验的预期目标是通过配置动态路由协议 RIP,自动学习网段的路由信息,实现网络的互连互通。

步骤 1:

(1) 按照拓扑图配置 PC1 和 PC2 的 IP 地址、子网掩码,网关,并测试它们的连通性。

```
C:\Users\Administrator>ping 192.168.3.22
正在 Ping 192.168.3.22 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。

192.168.3.22 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),
```

因为尚未配置 RIP 协议, 所以显然 PC1 和 PC2 之间无法连通。

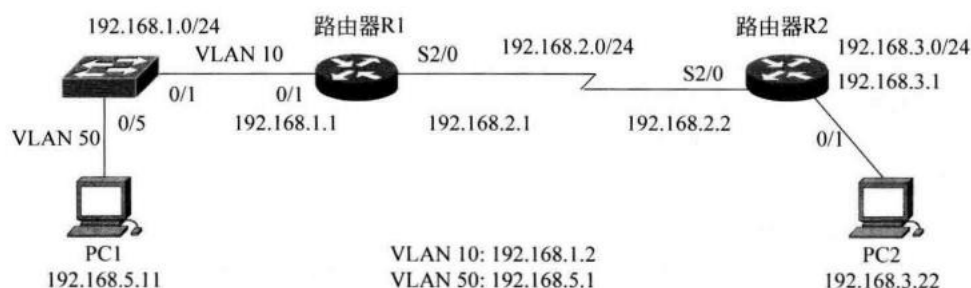


图 7-20 RIP 路由实验拓扑

(2) 在路由器 R1(或 R2)上执行 show ip route 命令,记录路由表信息。



```
5-RSR20-1>en 14
Password:
5-RSR20-1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
5-RSR20-1(config)#show ip route

Codes: C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        I - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
```

步骤 2：三层交换机的基本配置（见下页）。

```
switch#configure terminal
switch(config)#hostname S5750          !将交换机更名为 S5750
S5750(config)#vlan 10
S5750(config-vlan)#exit
S5750(config)#vlan 50
S5750(config-vlan)#exit
S5750(config)#interface gigabitethernet 0/1
S5750(config-if)#switchport access vlan 10
S5750(config-if)#exit
S5750(config)#interface gigabitethernet 0/5
S5750(config-if)#switchport access vlan 50
S5750(config-if)#exit
S5750(config)#interface vlan 10        !创建 VLAN 虚拟端口,并配置 IP 地址
S5750(config-if)#ip address 192.168.1.2 255.255.255.0
S5750(config-if)#no shutdown
S5750(config-if)#exit
S5750(config)#interface vlan 50        !创建 VLAN 虚拟端口,并配置 IP 地址
S5750(config-if)#ip address 192.168.5.1 255.255.255.0
S5750(config-if)#no shutdown
S5750(config-if)#exit
```



```
9-S5750-1>
9-S5750-1>en 14
Password:
9-S5750-1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
9-S5750-1(config)#hostname S7570
S7570(config)#vlan 10
S7570(config-vlan)#exit
S7570(config)#vlan 50
S7570(config-vlan)#exit
S7570(config)#*Jan 15 13:57:51: %LINK-3-UPDOWN: Interface GigabitEthernet 0/2, changed state to down.
*Jan 15 13:57:51: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet 0/2, changed state to down.
S7570(config)#*Jan 15 13:58:27: %LINK-3-UPDOWN: Interface GigabitEthernet 0/2, changed state to up.
*Jan 15 13:58:27: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet 0/2, changed state to up.
*Jan 15 13:58:45: %LINK-3-UPDOWN: Interface GigabitEthernet 0/2, changed state to down.
*Jan 15 13:58:45: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet 0/2, changed state to down.
S7570(config)#*Jan 15 13:59:07: %LINK-3-UPDOWN: Interface GigabitEthernet 0/1, changed state to up.
*Jan 15 13:59:07: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet 0/1, changed state to up.
S7570(config)#interface gi 0/1
S7570(config-if-GigabitEthernet 0/1)#switchport access vlan 10
S7570(config-if-GigabitEthernet 0/1)#exit
S7570(config)#interface gi 0/5
S7570(config-if-GigabitEthernet 0/5)#switchport access vlan 50
S7570(config-if-GigabitEthernet 0/5)#exit
S7570(config)#interface vlan 10
S7570(config-if-VLAN 10)#*Jan 15 14:00:05: %LINEPROTO-5-UPDOWN: Line protocol on Interface VLAN 10, changed state to up.
S7570(config-if-VLAN 10)#ip address 192.168.1.2 255.255.255.0
S7570(config-if-VLAN 10)#no shutdown
S7570(config-if-VLAN 10)#exit
S7570(config)#interface vlan 50
S7570(config-if-VLAN 50)#*Jan 15 14:01:08: %LINEPROTO-5-UPDOWN: Line protocol on Interface VLAN 50, changed state to up.
S7570(config-if-VLAN 50)#ip address 192.168.5.1 255.255.255.0
S7570(config-if-VLAN 50)#no shutdown
S7570(config-if-VLAN 50)#exit
S7570(config)#
```

步骤 3: 路由器 R1 的基本配置。

```
Router1(config)#interface gigabitethernet 0/1
Router1(config-if)#ip address 192.168.1.1 255.255.255.0
Router1(config-if)#no shutdown
Router1(config-if)#exit
Router1(config)#interface serial 2/0
Router1(config-if)#ip address 192.168.2.1 255.255.255.0
Router1(config-if)#no shutdown
```

```
5-RSR20-1(config)#interface gi 0/1
5-RSR20-1(config-if-GigabitEthernet 0/1)#ip address 192.168.1.1 255.255.255.0
5-RSR20-1(config-if-GigabitEthernet 0/1)#no shutdown
5-RSR20-1(config-if-GigabitEthernet 0/1)#exit
5-RSR20-1(config)#interface serial 2/0
5-RSR20-1(config-if-Serial 2/0)#ip address 192.168.2.1 255.255.255.0
5-RSR20-1(config-if-Serial 2/0)#no shutdown
```

步骤 4: 路由器 R2 的基本配置。

```
Router2(config)#interface gigabitethernet 0/1
Router2(config-if)#ip address 192.168.3.1 255.255.255.0
Router2(config-if)#no shutdown
Router2(config-if)#exit
Router2(config)#interface serial 2/0
Router2(config-if)#ip address 192.168.2.2 255.255.255.0
Router2(config-if)#no shutdown
```

```
5-RSR20-2(config)#interface gi 0/1
5-RSR20-2(config-if-GigabitEthernet 0/1)#ip address 192.168.3.1 255.255.255.0
5-RSR20-2(config-if-GigabitEthernet 0/1)#no shutdown
5-RSR20-2(config-if-GigabitEthernet 0/1)#exit
5-RSR20-2(config)#interface serial 2/0
5-RSR20-2(config-if-Serial 2/0)#ip address 192.168.2.2 255.255.255.0
5-RSR20-2(config-if-Serial 2/0)#no shutdown
```

步骤 5: 交换机 S5750 配置 RIPv2 路由协议。



```
S5750(config)#router rip          !开启 RIP 进程
S5750(config)#version 2
S5750(config-router)#network 192.168.1.0 !申明本设备的直连网段
S5750(config-router)#network 192.168.5.0 !RIP 发布网段地址为有类地址
```

```
S7570#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
S7570(config)#router rip
S7570(config-router)#version 2
S7570(config-router)#network 192.168.1.0/24
% Unknown command.

S7570(config-router)#network 192.168.1.0/24
^
% Invalid input detected at '^' marker.

S7570(config-router)#network 192.168.1.0 255.255.255.0
S7570(config-router)#network 192.168.5.0 255.255.255.0
S7570(config-router)#
```

需要额外添加掩码

步骤 6:路由器 R1 配置 RIPv2 路由协议。

```
Router1(config)#router rip
Router1(config)#version 2
Router1(config)#no auto-summary      !关闭路由信息的自动汇总功能
Router1(config-router)#network 192.168.1.0
Router1(config-router)#network 192.168.2.0
```

```
5-RSR20-1(config)#router rip
5-RSR20-1(config-router)#version 2
5-RSR20-1(config-router)#no auto-summary
5-RSR20-1(config-router)#network 192.168.2.0 255.255.255.0
5-RSR20-1(config-router)#network 192.168.1.0 255.255.255.0
```

需要额外添加掩码

步骤 7:路由器 R2 配置 RIPv2 路由协议。

```
Router2(config)#router rip
Router2(config)#version 2
Router2(config)#no auto-summary      !关闭路由信息的自动汇总功能
Router2(config-router)#network 192.168.2.0
Router2(config-router)#network 192.168.3.0
```

```
5-RSR20-2(config)#router rip
5-RSR20-2(config-router)#version 2
5-RSR20-2(config-router)#no auto-summary
^
% Invalid input detected at '^' marker.

5-RSR20-2(config-router)#no auto-summary
5-RSR20-2(config-router)#network 192.168.2.0
5-RSR20-2(config-router)#network 192.168.3.0
5-RSR20-2(config-router)#no network 192.168.2.0
5-RSR20-2(config-router)#no network 192.168.3.0
5-RSR20-2(config-router)#network 192.168.2.0 255.255.255.0
5-RSR20-2(config-router)#network 192.168.3.0 255.255.255.0
```

需要额外添加掩码

验证 3 台路由设备的路由表,查看是否自动学习了其他网段的路由信息。注意观察 R 标签项。

```
S5750#show ip route
```

分析交换机 S5750 的路由表,表中有 R 条目吗?是怎样产生的?



```
S7570#show ip route

Codes: C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
C    192.168.1.0/24 is directly connected, VLAN 10
C    192.168.1.2/32 is local host.
R    192.168.2.0/24 [120/1] via 192.168.1.1, 00:02:06, VLAN 10
R    192.168.3.0/24 [120/2] via 192.168.1.1, 00:00:25, VLAN 10
C    192.168.5.0/24 is directly connected, VLAN 50
C    192.168.5.1/32 is local host.
S7570#
```

有 R 条目，产生方式是自动学习了 R1 和 R2 的路由信息。

```
Router1#show ip route
```

分析路由器 R1 的路由表,表中有 R 条目吗?是怎样产生的?

```
5-RSR20-1#show ip route

Codes: C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
C    192.168.1.0/24 is directly connected, GigabitEthernet 0/1
C    192.168.1.1/32 is local host.
C    192.168.2.0/24 is directly connected, Serial 2/0
C    192.168.2.1/32 is local host.
R    192.168.3.0/24 [120/1] via 192.168.2.2, 00:01:03, Serial 2/0
R    192.168.5.0/24 [120/1] via 192.168.1.2, 00:02:44, GigabitEthernet 0/1
5-RSR20-1#
```

有 R 条目，产生方式是自动学习了 R2 和交换机的路由信息。

```
Router2#show ip route
```

分析路由器 R2 的路由表,表中有 R 条目吗?是怎样产生的?

```
5-RSR20-2#show ip route

Codes: C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
R    192.168.1.0/24 [120/1] via 192.168.2.1, 00:00:52, Serial 2/0
C    192.168.2.0/24 is directly connected, Serial 2/0
C    192.168.2.2/32 is local host.
C    192.168.3.0/24 is directly connected, GigabitEthernet 0/1
C    192.168.3.1/32 is local host.
R    192.168.5.0/24 [120/2] via 192.168.2.1, 00:00:52, Serial 2/0
5-RSR20-2#
```

有 R 条目，产生方式是自动学习了 R1 和交换机的路由信息。

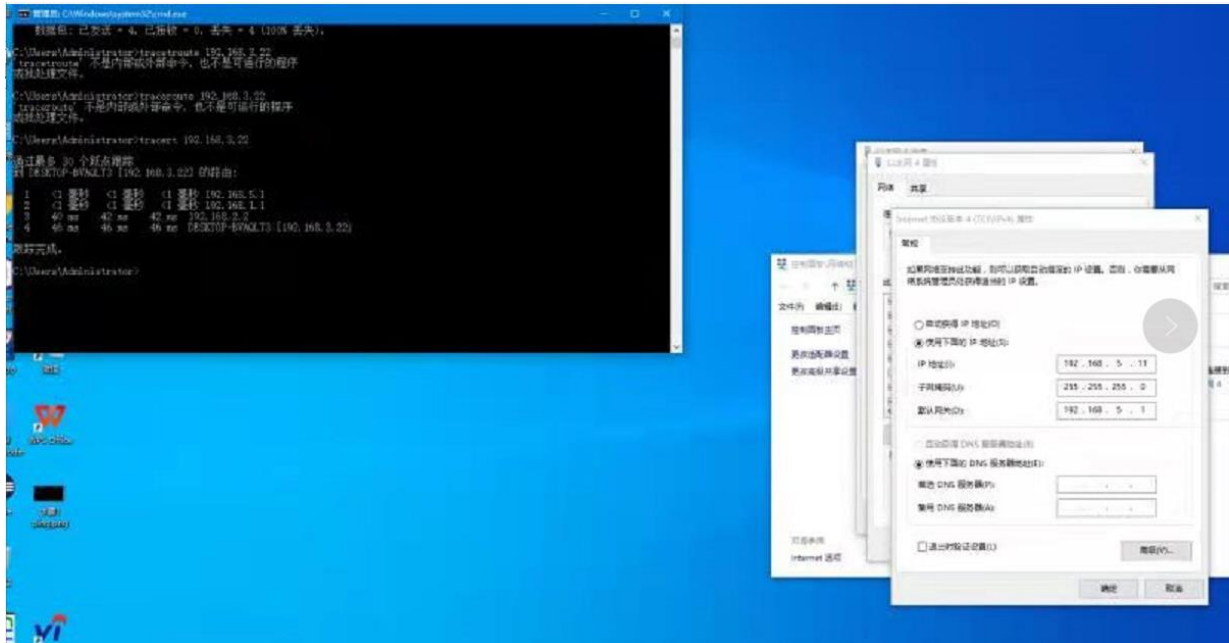
步骤 8:测试网络的连通性。

(1) 将此时的路由表与步骤 1 的路由表进行比较,有什么结论?

答：步骤 1 中的路由表没有 R 条目，而现在的路由表中有 R 条目。原因是 RIP 协议可以动态更新路由信息。

(2) 分析 traceroute PC1(或 PC2)的结果。

由于微信截图的时效性，未来得及保存原图，导致图片较模糊，但仍可以大致看出结果。



以上为 traceroute PC2 的结果，具体路径为 192.168.5.11 到 192.168.5.1 到 192.168.1.1 到 192.168.2.2 最后到 192.168.3.22。

(3) 进行拔线实验,通过 Wireshark 测试报文变化的时间差,路由有没有出现毒性反转现象?

拔线前后报文的时间差如下图所示:

55	16.388032	192.168.3.1	224.0.0.9	RIPv2	110	Response
100	30.750255	192.168.3.1	224.0.0.9	RIPv2	70	Response
149	46.380294	192.168.3.1	224.0.0.9	RIPv2	110	Response
189	76.380744	192.168.3.1	224.0.0.9	RIPv2	110	Response
231	106.381158	192.168.3.1	224.0.0.9	RIPv2	110	Response
293	136.381612	192.168.3.1	224.0.0.9	RIPv2	110	Response

我们发现拔线前后，时间差从一开始的十几秒左右变为稳定在 30 秒，说明拔线确实影响到了传输（因为 RIP 每隔 30s 就会向其相邻路由器广播本地路由表）。

对于毒性反转, 我们可观察 RIP 协议中的跳数是否达到或超过最大限定值 16 (因为 RIP 是以跳数衡量到达目的网络的度量值(Metric)。在 RIP 中, 路由器到与它直接相连网络的跳数为 0, 通过一个路由器的网络跳数为 1。RIP 规定度量值取 0~15 之间的整数, 大于或等于 16 的跳数被定义为无穷大, 即目的主机或网络不可达):



拔线前:

```
Address Family: IP (2)
Route Tag: 0
IP Address: 192.168.1.0
Netmask: 255.255.255.0
Next Hop: 0.0.0.0
Metric: 2
```

拔线后:

```
Address Family: IP (2)
Route Tag: 0
IP Address: 192.168.5.0
Netmask: 255.255.255.0
Next Hop: 0.0.0.0
Metric: 16
```

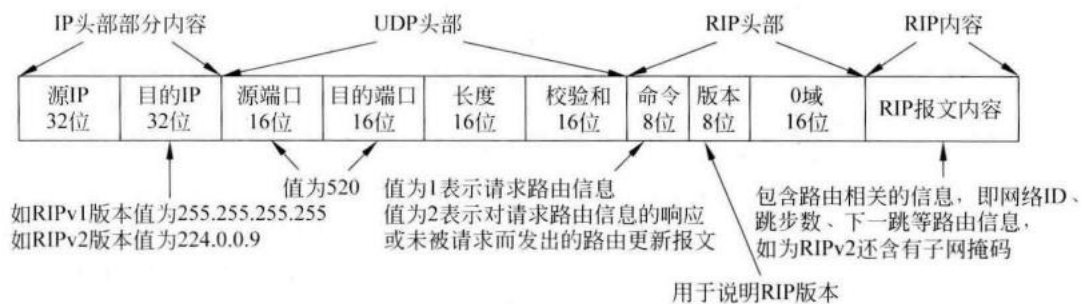
由上述结果可知,确实发生了毒性反转现象。

(4) 捕获数据包,分析 RIP 封装结构。RIP 包在 PC1 或 PC2 上能捕获到吗?如希望 2 台主机都能捕获到 RIP 包,请描述实现方法。

首先, PC1 和 PC2 均可以捕获到 RIP 包, 无需改进方法。

所以以下的 RIP 包分析均以 PC1 上捕获的为准:

RIP 封装结构如下:



```
▼ Frame 208: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface \Device\NPF_{F79B1DFF-B47D-45C5-8AFD-605A02562A6C}, id 0
  ▼ Interface id: 0 (\Device\NPF_{F79B1DFF-B47D-45C5-8AFD-605A02562A6C})
    Interface name: \Device\NPF_{F79B1DFF-B47D-45C5-8AFD-605A02562A6C}
    Interface description: 以太网 4
    Encapsulation type: Ethernet (1)
    Arrival Time: Jun 7, 2021 21:14:02.019542000 中国标准时间
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1623071642.019542000 seconds
    [Time delta from previous captured frame: 0.173658000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 25.935761000 seconds]
    Frame Number: 208
    Frame Length: 110 bytes (880 bits)
    Capture Length: 110 bytes (880 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:udp:rip]
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]
```




▼ Ethernet II, Src: RuijieNe_27:bf:9a (58:69:6c:27:bf:9a), Dst: IPv4mcast_09 (01:00:5e:00:00:09)

- ▼ Destination: IPv4mcast_09 (01:00:5e:00:00:09)
Address: IPv4mcast_09 (01:00:5e:00:00:09)
.... ..0. = LG bit: Globally unique address (factory default)
.... ...1. = IG bit: Group address (multicast/broadcast)
- ▼ Source: RuijieNe_27:bf:9a (58:69:6c:27:bf:9a)
Address: RuijieNe_27:bf:9a (58:69:6c:27:bf:9a)
.... ..0. = LG bit: Globally unique address (factory default)
.... ...0. = IG bit: Individual address (unicast)

Type: IPv4 (0x0800)
Frame check sequence: 0x97965fb4 [unverified]
[FCS Status: Unverified]

▼ Internet Protocol Version 4, Src: 192.168.3.1, Dst: 224.0.0.9

0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)

▼ Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
1100 00.. = Differentiated Services Codepoint: Class Selector 6 (48)
.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)

Total Length: 92
Identification: 0x00f4 (244)

▼ Flags: 0x00
0... = Reserved bit: Not set
.0.. = Don't fragment: Not set
..0. = More fragments: Not set

Fragment Offset: 0
Time to Live: 1
Protocol: UDP (17)
Header Checksum: 0x142b [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.3.1
Destination Address: 224.0.0.9

▼ User Datagram Protocol, Src Port: 520, Dst Port: 520

Source Port: 520
Destination Port: 520
Length: 72
Checksum: 0xe90 [unverified]
[Checksum Status: Unverified]
[Stream index: 3]

▼ [Timestamps]
[Time since first frame: 0.00000000 seconds]
[Time since previous frame: 0.00000000 seconds]

UDP payload (64 bytes)

▼ Routing Information Protocol

Command: Response (2)
Version: RIPv2 (2)

▼ IP Address: 192.168.1.0, Metric: 2
Address Family: IP (2)
Route Tag: 0
IP Address: 192.168.1.0
Netmask: 255.255.255.0
Next Hop: 0.0.0.0
Metric: 2

▼ IP Address: 192.168.2.0, Metric: 1
Address Family: IP (2)
Route Tag: 0
IP Address: 192.168.2.0
Netmask: 255.255.255.0
Next Hop: 0.0.0.0
Metric: 1

▼ IP Address: 192.168.5.0, Metric: 3
Address Family: IP (2)
Route Tag: 0
IP Address: 192.168.5.0
Netmask: 255.255.255.0
Next Hop: 0.0.0.0
Metric: 3



【实验思考】

(1) 查看交换机端口 0/1 所属 VLAN 应使用哪条命令?

答: show vlan 即可。

```
S7570(config)#show vlan
```

VLAN Name	Status	Ports
1 VLAN0001	STATIC	Gi0/2, Gi0/3, Gi0/4, Gi0/6 Gi0/7, Gi0/8, Gi0/9, Gi0/10 Gi0/11, Gi0/12, Gi0/13, Gi0/14 Gi0/15, Gi0/16, Gi0/17, Gi0/18 Gi0/19, Gi0/20, Gi0/21, Gi0/22 Gi0/23, Gi0/24, Gi0/25, Gi0/26 Gi0/27, Gi0/28
10 VLAN0010	STATIC	Gi0/1
50 VLAN0050	STATIC	Gi0/5

由上图易知交换机端口 0/1 属于 VLAN 10。

(2) 如何查看 RIP 的版本号和发布到的网段?

答: show ip protocols 即可。

(3) RIPv1 的广播地址是什么?RIPv2 的组播地址是什么?

答: RIPv1 的广播地址: 0.0.0.0

RIPv2 的组播地址: 224.0.0.9

(4)使用 10.10.X.0 的 IP 地址重做本次试验,注意网段间使用不同的子网掩码。当在 RIPv1 下设置不同网段时,配置后的端口实际上获得的子网掩码是什么?配合实验分析原因。

答: 实际获得掩码为 24 位掩码。实际我们将 PC1 的 IP 地址和掩码做更改, 将我们使用的 RIP 版本改为 RIP V1 版本, 我们发现, 在访问 PC2 时, 在原有的路由配置情况下, PC1 仍能顺利访问 PC2, 但我们发现在抓到的 RIP V1 的数据包中并没有掩码信息, 在我们的路由器 R2 上, 在转发给 PC2 的数据包中, 没有记录数据包来源地的掩码。在路由器的路由表项中, 路由器将数据来源的掩码视为与路由出口的掩码相同, 即 24 位掩码。

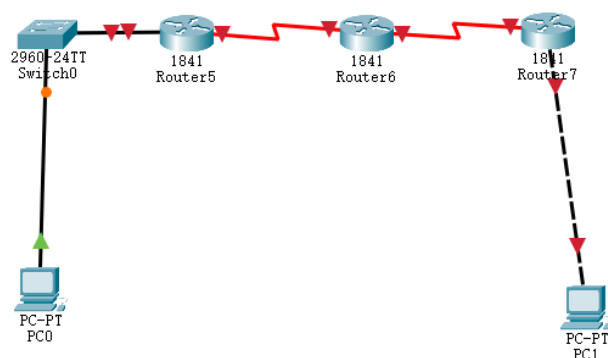
(5) RIPv1 必须使用自动汇总,不支持不连续网络,请实验验证。RIPv2 支持不连续网络吗?

答: 我们在上面的配置中, 我们修改了 PC1 和器默认路由端口的 IP 地址, 我们发现 PC1 仍能 ping 通 PC2, 具体原因我们已经解释。我们在原有的实验拓扑图的结构上, 在两个路由器之间增设一个路由器, 该路由器与两侧路由器相连的网段设置成 8 位掩码网



计算机网络实验报告

络, PC 的网络仍设位 24 位网络 (实验室一台机器只有 2 个路由器, 我们通过 packet tracer 模拟), 我们发现, 此时使用 RIPv1 两台 PC 无法互联, 而使用 RIPv2 则可以。这说明了 RIPv1 必须使用自动汇总, 不支持不连续网络, 而 RIPv2 支持不连续网络。



(6) RIPv1 对路由没有标记的功能, RIPv2 可以对路由打标记(tag), 用于过滤和做策略。请在实验中观察和分析。

答: 在本次实验捕获的数据包中, 我们分别选取用 RIPv1 和 RIPv2 做实验时捕获的数据包中的 RIPv1 的数据包和 RIPv2 的数据报, 观察其数据包记录的路由信息, 发现 RIPv2 数据包中有 tag 信息, 而 RIPv1 数据包中没有 tag 信息, 即 RIPv1 对路由没有标记的功能, RIPv2 可以对路由打标记(tag), 用于过滤和做策略。相应的报文截图在步骤 8 (4) 已经给出。

最后我们测试实验网的连通性:



```
C:\Users\Administrator>ping 192.168.5.11

正在 Ping 192.168.5.11 具有 32 字节的数据:
来自 192.168.5.11 的回复: 字节=32 时间=39ms TTL=61
来自 192.168.5.11 的回复: 字节=32 时间=40ms TTL=61
来自 192.168.5.11 的回复: 字节=32 时间=40ms TTL=61
来自 192.168.5.11 的回复: 字节=32 时间=40ms TTL=61

192.168.5.11 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 39ms, 最长 = 40ms, 平均 = 39ms

C:\Users\Administrator>ping 192.168.3.22

正在 Ping 192.168.3.22 具有 32 字节的数据:
来自 192.168.3.22 的回复: 字节=32 时间=37ms TTL=61
来自 192.168.3.22 的回复: 字节=32 时间=39ms TTL=61
来自 192.168.3.22 的回复: 字节=32 时间=40ms TTL=61
来自 192.168.3.22 的回复: 字节=32 时间=37ms TTL=61

192.168.3.22 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 37ms, 最长 = 40ms, 平均 = 38ms
```

由上图可知连通性正常。

2.通过实验观察 RIP V1 和 V2 的区别（重点在 VLSM 上）给出分析过程与结果（实验 IP 采用 10.10.x.0 网段）

答：我们在第一问中已经分析了，RIPv2 的报文中包含子网掩码信息，而 RIPv1 报文中则不包含掩码信息，也由此，RIP V1 协议不能区分不同掩码的网段，因而也无法支持 VLSM（可变长字段掩码），而 RIP V2 可以记录掩码信息，能够支持 VLSM，我们在第一问中也可以验证该结果。此外，RIPv1 通过广播更新，RIPv2 通过组播更新，我们在 RIP 数据包中的 IP 地址中可以看出这一点，RIPv1 对路由无标记，RIPv2 则有，这在我们分析 RIP 数据包内容时的 tag 元素有无可以看出。

3.学会使用 Debug ip packet 和 Debug ip rip 命令，并对 debug 信息做分析。

答：我们在路由器或交换机中执行 Debug ip packet 和 Debug ip rip 指令，就可以如下的信息。



```
5-RSR20-1#debug ip packet
5-RSR20-1#Jun 18 02:24:07: %7: [RIP] update timer expired via interface Serial 2/0[192.168.2.1/24]
Jun 18 02:24:07: %7: [RIP] update timer schedule via interface Serial 2/0[192.168.2.1/24]
Jun 18 02:24:07: %7: [RIP] Prepare to send MULTICAST response...
Jun 18 02:24:07: %7: [RIP] Building update entries on Serial 2/0
Jun 18 02:24:07: %7: 192.168.1.0/24 via 0.0.0.0 metric 1 tag 0
Jun 18 02:24:07: %7: 192.168.5.0/24 via 0.0.0.0 metric 2 tag 0
Jun 18 02:24:07: %7: [RIP] Send packet to 224.0.0.9 Port 520 on Serial 2/0
Jun 18 02:24:07: %7: IP: s=192.168.2.1 (local), d=224.0.0.9 (Serial 2/0), vrf=global(0), g=224.0.0.9, len=72, sent ip pkt to link_layer 222
Jun 18 02:24:08: %7: [RIP] update timer expired via interface GigabitEthernet 0/1[192.168.1.1/24]
Jun 18 02:24:08: %7: [RIP] update timer schedule via interface GigabitEthernet 0/1[192.168.1.1/24]
Jun 18 02:24:08: %7: [RIP] Prepare to send MULTICAST response...
Jun 18 02:24:08: %7: [RIP] Building update entries on GigabitEthernet 0/1
Jun 18 02:24:08: %7: 192.168.2.0/24 via 0.0.0.0 metric 1 tag 0
Jun 18 02:24:08: %7: 192.168.3.0/24 via 0.0.0.0 metric 2 tag 0
Jun 18 02:24:08: %7: [RIP] Send packet to 224.0.0.9 Port 520 on GigabitEthernet 0/1
Jun 18 02:24:08: %7: IP: s=192.168.1.1 (local), d=224.0.0.9 (GigabitEthernet 0/1), vrf=global(0), g=224.0.0.9, len=72, sent ip pkt to link_layer --> raw send
Jun 18 02:24:16: %7: IP: s=192.168.1.2 (GigabitEthernet 0/1), d=224.0.0.9, vrf=global(0), len=52, received
Jun 18 02:24:16: %7: [RIP] RIP received packet, sock=32979 src=192.168.1.2 len=52
Jun 18 02:24:16: %7: [RIP] Received version 2 response packet on GigabitEthernet 0/1
Jun 18 02:24:16: %7: [RIP] Cancel peer[192.168.1.2] remove timer
Jun 18 02:24:16: %7: [RIP] Peer[192.168.1.2] remove timer schedule...
Jun 18 02:24:16: %7: [RIP] Both do not need auth, Auth ok
Jun 18 02:24:16: %7: route-entry: family 2 tag 0 ip 192.168.5.0 mask 255.255.255.0 nhop 0.0.0.0 metric 1
Jun 18 02:24:16: %7: [RIP] [192.168.5.0/24] RIP route update, protocol(4)
Jun 18 02:24:16: %7: [RIP] Old path is: nhop=192.168.1.2 routesrc=192.168.1.2 intf=5
Jun 18 02:24:16: %7: [RIP] New path is: nhop=192.168.1.2 routesrc=192.168.1.2 intf=5
Jun 18 02:24:16: %7: [RIP] 192.168.5.0/24 RIP distance apply from 192.168.1.2!
Jun 18 02:24:16: %7: [RIP] 192.168.5.0/24 cancel route timer
Jun 18 02:24:16: %7: [RIP] 192.168.5.0/24 route timer schedule...
Jun 18 02:24:21: %7: IP: s=192.168.2.2 (Serial 2/0), d=224.0.0.9, vrf=global(0), len=52, received
Jun 18 02:24:21: %7: [RIP] RIP received packet, sock=32979 src=192.168.2.2 len=52
Jun 18 02:24:21: %7: [RIP] Received version 2 response packet on Serial 2/0
Jun 18 02:24:21: %7: [RIP] Cancel peer[192.168.2.2] remove timer
Jun 18 02:24:21: %7: [RIP] Peer[192.168.2.2] remove timer schedule...
Jun 18 02:24:21: %7: [RIP] Both do not need auth, Auth ok
Jun 18 02:24:21: %7: route-entry: family 2 tag 0 ip 192.168.3.0 mask 255.255.255.0 nhop 0.0.0.0 metric 1
Jun 18 02:24:21: %7: [RIP] [192.168.3.0/24] RIP route update, protocol(4)
Jun 18 02:24:21: %7: [RIP] Old path is: nhop=192.168.2.2 routesrc=192.168.2.2 intf=2
Jun 18 02:24:21: %7: [RIP] New path is: nhop=192.168.2.2 routesrc=192.168.2.2 intf=2
Jun 18 02:24:21: %7: [RIP] 192.168.3.0/24 RIP distance apply from 192.168.2.2!
Jun 18 02:24:21: %7: [RIP] 192.168.3.0/24 cancel route timer
Jun 18 02:24:21: %7: [RIP] 192.168.3.0/24 route timer schedule...
Jun 18 02:24:37: %7: [RIP] update timer expired via interface Serial 2/0[192.168.2.1/24]
Jun 18 02:24:37: %7: [RIP] update timer schedule via interface Serial 2/0[192.168.2.1/24]
Jun 18 02:24:37: %7: [RIP] Prepare to send MULTICAST response...
Jun 18 02:24:37: %7: [RIP] Building update entries on Serial 2/0
Jun 18 02:24:37: %7: 192.168.1.0/24 via 0.0.0.0 metric 1 tag 0
Jun 18 02:24:37: %7: 192.168.5.0/24 via 0.0.0.0 metric 2 tag 0
Jun 18 02:24:37: %7: [RIP] Send packet to 224.0.0.9 Port 520 on Serial 2/0
Jun 18 02:24:37: %7: IP: s=192.168.2.1 (local), d=224.0.0.9 (Serial 2/0), vrf=global(0), g=224.0.0.9, len=72, sent ip pkt to link_layer 222
```

由上图可知，每经过 30s，路由信息经历一个更新循环。

使用 Debug ip packet 指令，我们可以观察到我们的路由器获取和传出的路由表信息和一些 tag 信息等，如上图红色框部分所示。而使用 Debug ip rip 指令我们可以观察到我们的路由器的路由表更新过程。更新过程中，我们可以观察到原有的路由信息和修改后的路由信息，如下图红色部分所示。

```
5-RSR20-2#debug ip rip
5-RSR20-2#Jun 18 09:28:31: %7: [RIP] update timer expired via interface Serial 2/0[192.168.2.2/24]
Jun 18 09:28:31: %7: [RIP] update timer schedule via interface Serial 2/0[192.168.2.2/24]
Jun 18 09:28:31: %7: [RIP] Prepare to send MULTICAST response...
Jun 18 09:28:31: %7: [RIP] Building update entries on Serial 2/0
Jun 18 09:28:31: %7: 192.168.3.0/24 via 0.0.0.0 metric 1 tag 0
Jun 18 09:28:31: %7: [RIP] Send packet to 224.0.0.9 Port 520 on Serial 2/0
Jun 18 09:28:33: %7: [RIP] update timer expired via interface GigabitEthernet 0/1[192.168.3.1/24]
Jun 18 09:28:33: %7: [RIP] update timer schedule via interface GigabitEthernet 0/1[192.168.3.1/24]
Jun 18 09:28:33: %7: [RIP] Prepare to send MULTICAST response...
Jun 18 09:28:33: %7: [RIP] Building update entries on GigabitEthernet 0/1
Jun 18 09:28:33: %7: 192.168.1.0/24 via 0.0.0.0 metric 2 tag 0
Jun 18 09:28:33: %7: 192.168.2.0/24 via 0.0.0.0 metric 1 tag 0
Jun 18 09:28:33: %7: 192.168.5.0/24 via 0.0.0.0 metric 3 tag 0
Jun 18 09:28:33: %7: [RIP] Send packet to 224.0.0.9 Port 520 on GigabitEthernet 0/1
5-RSR20-2#debug ip packet
5-RSR20-2#Jun 18 09:28:44: %7: IP: s=192.168.3.22 (GigabitEthernet 0/1), d=192.168.3.255, vrf=global(0), len=1468, received
Jun 18 09:28:48: %7: IP: s=192.168.2.1 (Serial 2/0), d=224.0.0.9, vrf=global(0), len=72, received
Jun 18 09:28:48: %7: [RIP] RIP received packet, sock=32979 src=192.168.2.1 len=44
Jun 18 09:28:48: %7: [RIP] Received version 2 response packet on Serial 2/0
Jun 18 09:28:48: %7: [RIP] Cancel peer[192.168.2.1] remove timer
Jun 18 09:28:48: %7: [RIP] Peer[192.168.2.1] remove timer schedule...
Jun 18 09:28:48: %7: [RIP] Both do not need auth, Auth ok
Jun 18 09:28:48: %7: route-entry: family 2 tag 0 ip 192.168.1.0 mask 255.255.255.0 nhop 0.0.0.0 metric 1
Jun 18 09:28:48: %7: route-entry: family 2 tag 0 ip 192.168.5.0 mask 255.255.255.0 nhop 0.0.0.0 metric 2
Jun 18 09:28:48: %7: [RIP] [192.168.1.0/24] RIP route update, protocol(4)
Jun 18 09:28:48: %7: [RIP] Old path is: nhop=192.168.2.1 routesrc=192.168.2.1 intf=2
Jun 18 09:28:48: %7: [RIP] New path is: nhop=192.168.2.1 routesrc=192.168.2.1 intf=2
Jun 18 09:28:48: %7: [RIP] 192.168.1.0/24 RIP distance apply from 192.168.2.1!
Jun 18 09:28:48: %7: [RIP] 192.168.1.0/24 cancel route timer
Jun 18 09:28:48: %7: [RIP] 192.168.1.0/24 route timer schedule...
Jun 18 09:28:48: %7: [RIP] 192.168.5.0/24 RIP route update, protocol(4)
Jun 18 09:28:48: %7: [RIP] Old path is: nhop=192.168.2.1 routesrc=192.168.2.1 intf=2
Jun 18 09:28:48: %7: [RIP] New path is: nhop=192.168.2.1 routesrc=192.168.2.1 intf=2
Jun 18 09:28:48: %7: [RIP] 192.168.5.0/24 RIP distance apply from 192.168.2.1!
Jun 18 09:28:48: %7: [RIP] 192.168.5.0/24 cancel route timer
Jun 18 09:28:48: %7: [RIP] 192.168.5.0/24 route timer schedule...
Jun 18 09:28:52: %7: IP: s=192.168.3.22 (GigabitEthernet 0/1), d=192.168.3.255, vrf=global(0), len=1468, received
Jun 18 09:29:01: %7: IP: s=192.168.3.22 (GigabitEthernet 0/1), d=192.168.3.255, vrf=global(0), len=1468, received
Jun 18 09:29:01: %7: [RIP] update timer expired via interface Serial 2/0[192.168.2.2/24]
Jun 18 09:29:01: %7: [RIP] update timer schedule via interface Serial 2/0[192.168.2.2/24]
Jun 18 09:29:01: %7: [RIP] Prepare to send MULTICAST response...
Jun 18 09:29:01: %7: [RIP] Building update entries on Serial 2/0
Jun 18 09:29:01: %7: 192.168.3.0/24 via 0.0.0.0 metric 1 tag 0
Jun 18 09:29:01: %7: [RIP] Send packet to 224.0.0.9 Port 520 on Serial 2/0
Jun 18 09:29:01: %7: IP: s=192.168.2.2 (local), d=224.0.0.9 (Serial 2/0), vrf=global(0), g=224.0.0.9, len=52, sent ip pkt to link_layer 222
```

由上图红色方框部分可知，Debug ip rip 命令可显示 RIP 路由更新信息，且包含了 RIP 的版本信息。



4.观察试验拓扑中链路状态发生改变时路由表的前后信息对比及 debug 信息的变化。

链路断开前，我们的链路状态稳定，路由器之间虽然会相互发送路由信息，但是接受到邻居路由器的路由表后该路由器路由表并不会改变，如我们第 3 题截图所示。

链路断开后，路由表信息如下所示：

```
*Jun 18 02:40:38: %7: [RIP] Send packet to 224.0.0.9 Port 520 on GigabitEthernet 0/1
*Jun 18 02:40:38: %7: IP: s=192.168.1.1 (local), d=224.0.0.9 (GigabitEthernet 0/1), vrf=global(0), g=224.0.0.9, len=72, sent ip pkt to link_layer -->
*Jun 18 02:40:46: %7: [RIP] Received version 2 response packet on GigabitEthernet 0/1, len=52, received
*Jun 18 02:40:46: %7: [RIP] RIP received packet, sock=32979 src=192.168.1.2 len=24
*Jun 18 02:40:46: %7: [RIP] New path is: nhop=192.168.1.2 routesrc=192.168.1.2 intf=5
*Jun 18 02:40:46: %7: [RIP] Cancel peer[192.168.1.2] remove timer
*Jun 18 02:40:46: %7: [RIP] Peer[192.168.1.2] remove timer schedule...
*Jun 18 02:40:46: %7: [RIP] Both do not need auth, Auth ok
*Jun 18 02:40:46: %7: [RIP] route-entry: family 2 tag 0 ip 192.168.5.0 mask 255.255.255.0 nhop 0.0.0.0 metric 1
*Jun 18 02:40:46: %7: [RIP] [192.168.5.0/24] RIP route update, protocol(4)
*Jun 18 02:40:46: %7: [RIP] Old path is: nhop=192.168.1.2 routesrc=192.168.1.2 intf=5
*Jun 18 02:40:46: %7: [RIP] New path is: nhop=192.168.1.2 routesrc=192.168.1.2 intf=5
*Jun 18 02:40:46: %7: [RIP] [192.168.5.0/24] RIP distance apply from 192.168.1.2!
*Jun 18 02:40:46: %7: [RIP] [192.168.5.0/24] cancel Route timer
*Jun 18 02:40:46: %7: [RIP] [192.168.5.0/24] route timer schedule...
*Jun 18 02:40:51: %7: IP: s=192.168.2.2 (Serial 2/0), d=224.0.0.9, vrf=global(0), len=52, received
*Jun 18 02:40:51: %7: [RIP] RIP received packet, sock=32979 src=192.168.2.2 len=24
*Jun 18 02:40:51: %7: [RIP] Received version 2 response packet on Serial 2/0
*Jun 18 02:40:51: %7: [RIP] Cancel peer[192.168.2.2] remove timer
*Jun 18 02:40:51: %7: [RIP] Peer[192.168.2.2] remove timer schedule...
*Jun 18 02:40:51: %7: [RIP] Both do not need auth, Auth ok
*Jun 18 02:40:51: %7: [RIP] route-entry: family 2 tag 0 ip 192.168.3.0 mask 255.255.255.0 nhop 0.0.0.0 metric 1
*Jun 18 02:40:51: %7: [RIP] [192.168.3.0/24] RIP route update, protocol(4)
*Jun 18 02:40:51: %7: [RIP] Old path is: nhop=192.168.2.2 routesrc=192.168.2.2 intf=2
*Jun 18 02:40:51: %7: [RIP] New path is: nhop=192.168.2.2 routesrc=192.168.2.2 intf=2
*Jun 18 02:40:51: %7: [RIP] [192.168.3.0/24] RIP distance apply from 192.168.2.2!
*Jun 18 02:40:51: %7: [RIP] [192.168.3.0/24] cancel Route timer
*Jun 18 02:40:51: %7: [RIP] [192.168.3.0/24] route timer schedule...
*Jun 18 02:40:58: %7: IP: s=192.168.1.2 (GigabitEthernet 0/1), d=224.0.0.9, vrf=global(0), len=52, received
*Jun 18 02:40:58: %7: [RIP] RIP received packet, sock=32979 src=192.168.1.2 len=24
*Jun 18 02:40:58: %7: [RIP] Received version 2 response packet on GigabitEthernet 0/1
*Jun 18 02:40:58: %7: [RIP] Cancel peer[192.168.1.2] remove timer
*Jun 18 02:40:58: %7: [RIP] Peer[192.168.1.2] remove timer schedule...
*Jun 18 02:40:58: %7: [RIP] Both do not need auth, Auth ok
*Jun 18 02:40:58: %7: [RIP] route-entry: family 2 tag 0 ip 192.168.5.0 mask 255.255.255.0 nhop 0.0.0.0 metric 16
*Jun 18 02:40:58: %7: [RIP] [192.168.5.0/24] RIP route disabling...
*Jun 18 02:40:58: %7: [RIP] [192.168.3.0/24] cancel route timer
*Jun 18 02:40:58: %7: [RIP] [192.168.5.0/24] route timer schedule...
*Jun 18 02:40:58: %7: [RIP] Trigger timer schedule, by instance 0
*Jun 18 02:40:58: %7: [RIP] [192.168.5.0/24] ready to add into kernel...
*Jun 18 02:40:58: %7: [RIP] NSM delete: IPv4 Route 192.168.5.0/24
*Jun 18 02:41:00: %7: [RIP] Trigger timer expired, by instance 0
*Jun 18 02:41:00: %7: [RIP] Prepare to send MULTICAST response...
*Jun 18 02:41:00: %7: [RIP] Building update entries on Serial 2/0
*Jun 18 02:41:00: %7: [RIP] Skip route[192.168.1.0/24] in trigger
*Jun 18 02:41:00: %7: [RIP] Skip route[192.168.2.0/24] in trigger
*Jun 18 02:41:00: %7: [RIP] Skip route[192.168.3.0/24] in trigger
*Jun 18 02:41:00: %7: [RIP] 192.168.5.0/24 via 0.0.0.0 metric 16 tag 0
*Jun 18 02:41:00: %7: [RIP] Send packet to 224.0.0.9 Port 520 on Serial 2/0
*Jun 18 02:41:00: %7: [RIP] Prepare to send MULTICAST response...
*Jun 18 02:41:00: %7: [RIP] Building update entries on GigabitEthernet 0/1
*Jun 18 02:41:00: %7: [RIP] Skip route[192.168.1.0/24] in trigger
*Jun 18 02:41:00: %7: [RIP] Skip route[192.168.2.0/24] in trigger
*Jun 18 02:41:00: %7: [RIP] Skip route[192.168.3.0/24] in trigger
*Jun 18 02:41:00: %7: [RIP] Skip send response packet...
*Jun 18 02:41:07: %7: [RIP] Timer expired via interface Serial 2/0[192.168.2.1/24]
*Jun 18 02:41:07: %7: [RIP] Update timer schedule via interface Serial 2/0[192.168.2.1/24]
*Jun 18 02:41:07: %7: [RIP] Prepare to send MULTICAST response...
*Jun 18 02:41:07: %7: [RIP] Building update entries on Serial 2/0
*Jun 18 02:41:07: %7: [RIP] 192.168.1.0/24 via 0.0.0.0 metric 1 tag 0
*Jun 18 02:41:07: %7: [RIP] 192.168.5.0/24 via 0.0.0.0 metric 16 tag 0
*Jun 18 02:41:07: %7: [RIP] Send packet to 224.0.0.9 Port 520 on Serial 2/0
*Jun 18 02:41:07: %7: [RIP] Send packet to 224.0.0.9 Port 520 on Serial 2/0, vrf=global(0), g=224.0.0.9, len=72, sent ip pkt to link_layer 222
```

我们发现，在 PC1 链路断开后，根据我们毒性逆转的原理，我们的 R1 路由器收到了通向 PC1 的条数为 16 的信息，并据此开始更新路由表，将更新结果发往邻居路由器。

链路稳定后，我们由回到了类似于初始的网络路由结构，但是我们 VLAN 50 的网络由于不可达在一段时间后被移除出我们的网络路由结构。

学号	学生	自评分
18329015	郝裕玮	100
18325071	张闯	100
19335153	马淙升	100