计算机网络技术实践

实验报告

实验名称 实验三 RIP 和 OSPF 路由配置协议流程分析

姓 名 陈朴炎 实 验 日 期：2023.11.05

学 号 2021211138 实验报告日期：2023.12.06

报 告 退 发:（ 订正 、 重做 ）

目录

一、环境

1.1 运行的操作系统以及网络平台环境

Windows 11
Dynamips 0.2.7 BY N.L.F.E

GNS3windows 版本：22.44.1

VM Version：0.15.0

Ubuntu version focal
Qemu version 4.2.1

1.2 网络拓扑图及说明

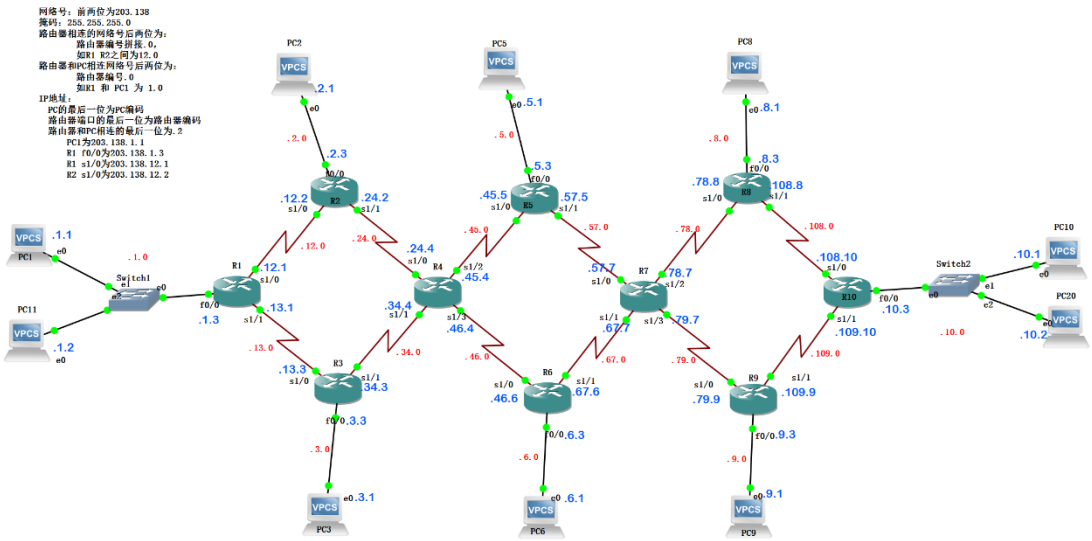


图 1-1 网络拓扑图

拓扑图说明：

图 1-1 中，用红色标明的是网络号，用蓝色标明的是 IP 地址。

路由器 10 个，PC 机 10 个，交换机 2 个，网段数量 20 个。

（1）网络号

所有的网络号前两位都是 203.138，子网掩码都是 255.255.255.0

路由器相连的网络号第 3 位为：路由器编号拼接的数字，如 R1 连 R2，之间的网络就是 203.138.12.0，R8 和 R10 连接的网络就是 203.138.108.0。

路由器和 PC 相连网络号后两位为：<路由器编号>.0。比如，R1 和 PC1 的网络号就是 203.138.1.0。

（2）IP 地址：

所有的 ip 地址前三位都对应着它们所在的网络号，只有第四位有区别：

和 PC 相连的网络中，PC 的最后一位是 1 或者 2，路由器的端口最后一位是 3，PC 设置的默认网关是它们的网络号.3，即和路由器相连的端口 IP。

在路由器和路由器相连的网络中，路由器端口的最后一位为路由器编码。

示例如下：

PC1 为 203.138.1.1，PC11 为 203.138.1.2

R1 f0/0 为 203.138.1.3

R1 s1/0 为 203.138.12.1

R2 s1/0 为 203.138.12.2

二、实验内容及目的

2.1 实验内容

（1）在第二次实验的基础上实现 RIP 和 OSPF 路由协议

（2）自己设计网络物理拓扑和逻辑网段，并在其上实现 RIP 和 OSPF 协议（不能少于 4 台路由器，要求 IP 地址第一位是 203，第二位是学号后三位%255）

（3）通过 debug 信息详细描述 RIP 和 OSPF 协议的工作过程，包括初始信息交

互、路由计算、链路故障处理等部分。(要修改部分链路，观察工作过程)

(4) RIP 协议中观察没有配置水平分割和配置水平分割后协议的工作流程，和路由消息传递方式；(要修改部分链路，观察区别，默认有水平分割)

(5) OSPF 中数据库同步信息的格式和同步对象？链路改变信息如何发送，具体格式（要修改部分链路，观察消息传递过程）

2.2 实验目的

(1) 理解和掌握 RIP 和 OSPF 路由协议的基本原理：

(2) 设计和实现网络拓扑：

(3) 实现 RIP 和 OSPF 协议：

(4) 调试和观察协议工作流程：

(5) 观察水平分割的影响：

(6) 深入理解 OSPF 的数据库同步和链路改变信息：

三、实验步骤及分析

3.1 ip 配置实验步骤

设计完网络拓扑后便开始配置 ip 和网络号。

首先将所有设备开启，并双击想要配置的设备。

(1) PC 的配置方法如下：

```
ip <ip 地址>/<子网掩码位数> <默认网关>
```

比如，PC1 的配置语句就是 ip 203.138.1.1/24 203.138.1.3，如图 3-1 所示，当终端报告该 PC 的 ip、掩码、网关时便说明配置完成。

```
PC1> ip 203.138.1.1/24 203.138.1.3
Checking for duplicate address...
PC1 : 203.138.1.1 255.255.255.0 gateway 203.138.1.3

PC1>
```

图 3-1 PC1 的 ip 配置示意图

（2）路由器和 PC 连接的接口 ip 配置如下：

首先点开路由器终端，输入 conf t，进入配置命令模式，如图 3-2。

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
```

图 3-2 R1 的 ip 配置步骤 1 示意图

进入待配置的端口，比如我的 R1 要配置端口 f0/0 的 IP 地址，我就输入 interface f0/0，如图 3-3 所示，进入接口配置。

```
R1(config)#interface f0/0
R1(config-if)#
```

图 3-3 R1 的 ip 配置步骤 2 示意图

在接口里，配置路由器该接口的 ip，并将该接口开启，如下图 3-4 所示。

```
R1(config)#interface f0/0
R1(config-if)#ip add 203.138.1.3 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#^Z
```

图 3-4 R1 的 ip 配置步骤 3 示意图

检查配置是否成功，用 PC1 来 ping R1 的 f0/0 接口，看是否 ping 通，如下图：

```
PC1> ping 203.138.1.3
203.138.1.3 icmp_seq=1 timeout
84 bytes from 203.138.1.3 icmp_seq=2 ttl=255 time=15.650 ms
84 bytes from 203.138.1.3 icmp_seq=3 ttl=255 time=15.919 ms
84 bytes from 203.138.1.3 icmp_seq=4 ttl=255 time=15.638 ms
84 bytes from 203.138.1.3 icmp_seq=5 ttl=255 time=15.793 ms
```

图 3-5 检查 ip 配置正确性示意图

（3）路由器和路由器相连的接口 ip 配置：

路由器之间连接通过串口，属于广域网连接，一端是 DCE，一端是 DTE，DCE 端配上时钟，数据链路层采用 PPP 协议。

我们在两个路由器之间选择一个路由器配 DCE，比如我的 R1 和 R2 相连，我选择 R1 的 S1/0 端口配 DCE，步骤如图 3-6：

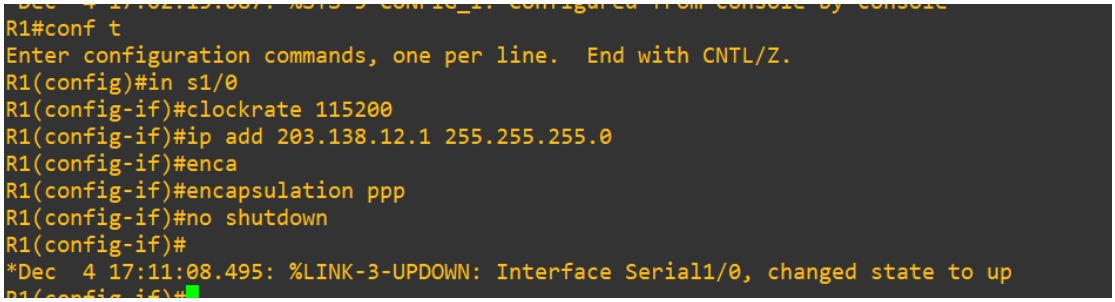


图 3-6 路由器接口配 DCE 及 IP 示意图

首先进入相应接口，配 DCE 就是加上时钟，输入 clockrate 115200，之后再配上 ip 地址，给数据链路层封装一个 PPP 协议，并启动该接口。当看到哦最后一行的提示 changed state to up 时就说明接口启动完成。

给 R2 配 DTE 时，只需少掉配时钟的步骤，其余步骤一样，如图 3-7：

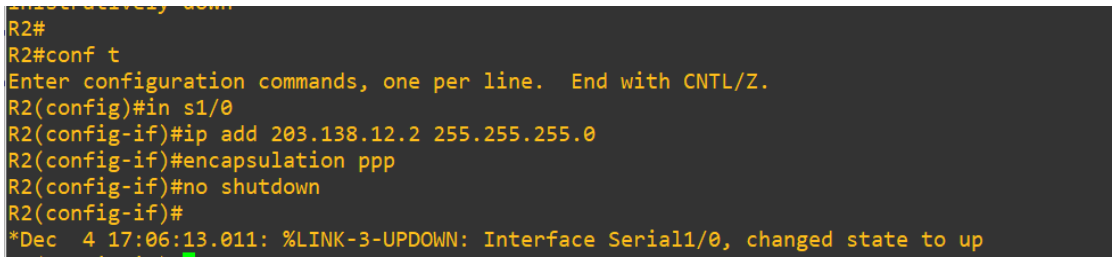


图 3-7 R2 路由器配 DTE 及 IP 示意图

首先进入串口，配上 ip，并封装好 ppp 协议，开启该接口，便配置完成。

其余的路由器和 PC 配置和上述步骤相差不大，我就不罗列了。

3.2 RIP 协议

3.2.1 RIP 协议配置过程

首先，将路由器的 debug 信息打开，如图 3-8。

```
R1#debug ip rip
RIP protocol debugging is on
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router rip
```

图 3-8 R1 开 debug 信息

在 conf t 下配置 rip 协议：

（1）router ip rip

（2）version 2

（3）network 203.138.1.0

（4）neighbor 203.138.12.2

具体步骤如图 3-9、图 3-10 所示：

```
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#network 203.138.1.0
R1(config-router)#network 203.138.1.0
*Dec  4 18:17:16.315: RIP: sending request on FastEthernet0/0 to 224.0.0.9
R1(config-router)#network 203.138.
*Dec  4 18:17:18.315: RIP: sending v2 flash update to 224.0.0.9 via FastEthernet0/0 (203.138.1.3)
*Dec  4 18:17:18.315: RIP: build flash update entries - suppressing null update
R1(config-router)#network 203.138.12.0
R1(config-router)#network 203.138.13.0
```

图 3-9 rip 协议配置图

```
*Dec  4 18:20:35.463:    203.138.13.0/24 via 0.
R1(config-router)#neighbor 203.138.12.2
R1(config-router)#neighbor 203.138.13.3
```

图 3-10 rip 协议配置图

network 命令用来设置该路由器连接的网络号。

neighbor 命令用来设置和该路由器相邻的路由器 ip。

输入完上述命令后，会循环出现如下 debug 信息：

```
R1(config-router)#
*Dec  4 18:22:10.711: RIP: sending v2 update to 224.0.0.9 via Serial1/1 (203.138.13.1)
*Dec  4 18:22:10.711: RIP: build update entries
*Dec  4 18:22:10.711:   203.138.1.0/24 via 0.0.0.0, metric 1, tag 0
*Dec  4 18:22:10.715:   203.138.12.0/24 via 0.0.0.0, metric 1, tag 0
*Dec  4 18:22:10.715: RIP: sending v2 update to 203.138.13.3 via Serial1/1 (203.138.13.1)
*Dec  4 18:22:10.715: RIP: build update entries
*Dec  4 18:22:10.715:   203.138.1.0/24 via 0.0.0.0, metric 1, tag 0
*Dec  4 18:22:10.719:   203.138.12.0/24 via 0.0.0.0, metric 1, tag 0
R1(config-router)#
*Dec  4 18:22:25.947: RIP: sending v2 update to 224.0.0.9 via FastEthernet0/0 (203.138.1.3)
*Dec  4 18:22:25.947: RIP: build update entries
*Dec  4 18:22:25.947:   203.138.12.0/24 via 0.0.0.0, metric 1, tag 0
*Dec  4 18:22:25.951:   203.138.13.0/24 via 0.0.0.0, metric 1, tag 0
```

图 3-11 R1debug 信息示意图 1

```
*Dec  4 18:26:10.747: RIP: sending v2 update to 224.0.0.9 via Serial1/0 (203.138.12.1)
*Dec  4 18:26:10.747: RIP: build update entries
*Dec  4 18:26:10.747:   203.138.1.0/24 via 0.0.0.0, metric 1, tag 0
*Dec  4 18:26:10.751:   203.138.13.0/24 via 0.0.0.0, metric 1, tag 0
*Dec  4 18:26:10.751: RIP: sending v2 update to 203.138.12.2 via Serial1/0 (203.138.12.1)
*Dec  4 18:26:10.751: RIP: build update entries
*Dec  4 18:26:10.751:   203.138.1.0/24 via 0.0.0.0, metric 1, tag 0
*Dec  4 18:26:10.755:   203.138.13.0/24 via 0.0.0.0, metric 1, tag 0
```

图 3-12 R1debug 信息示意图 2

sending v2 update 表示路由器在使用 RIP 的 version2。

224.0.0.9 是 RIP version2 的组播地址，via Serial1/1（203.138.13.1）表示发送路由信息通过的串口是 S1/1，该串口的 ip 地址为 203.138.13.1。

build update entries 表示路由器在创建路由信息，203.138.1.0/24 via 0.0.0.0, metric 1, tag 0 这句话表示网络 203.138.1.0/24 是通过路由器直连的，度量为 1，标记值为 0。0.0.0.0 表示的是路由器直连。

同理，配置其他的路由器，R2 配置过程如下：

```
R2(config)#router rip
R2(config-router)#version 2
R2(config-router)#network 203.138.2.0
R2(config-router)#network 203.138.12.0
R2(config-router)#network 203.138.24.0
R2(config-router)#neighbor 203.138.24.4
R2(config-router)#neighbor 203.138.12.1
```

图 3-13 R2 的 RIP 协议配置过程图

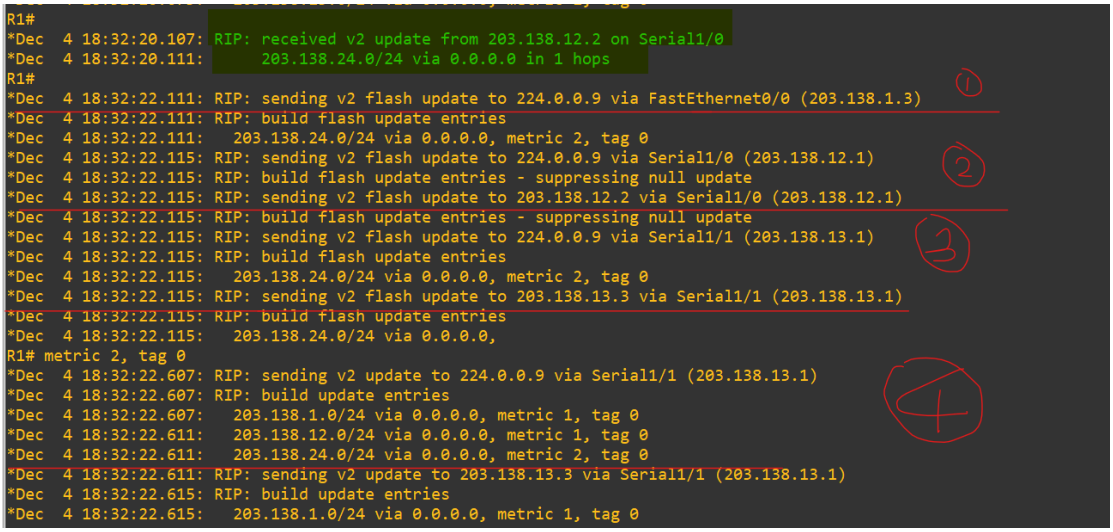


图 3-14 R2 配置好后 R1 的 debug 信息图

RIP 协议的工作过程

首先，标出绿色的那两行表示 R1 收到了 R2 发来的 update 信息。R2 告诉 R1：从我这里到网络 203.138.24.0/24 需要 1 跳。R1 收到后，更新自己的路由表，将 203.138.24.0/24 加入到自己的路由表中，并将跳数从 R2 来的 1 跳加 1 变成 2 跳，也就是图 3-14 ②中的 2 metric。之后，R1 通过组播，将该路由信息变化告知通过串口 S1/0、S1/1，还有接口 F0/0 通知给它的邻居，如图 3-14 的①②③。在④中，R1 已经构建好新的路由表，并将它们定期发送给自己的邻居。

3.2.2 RIP 协议路由构建过程分析

以 R10 为例，因为 R10 我是最后配置的，信息比较完整。

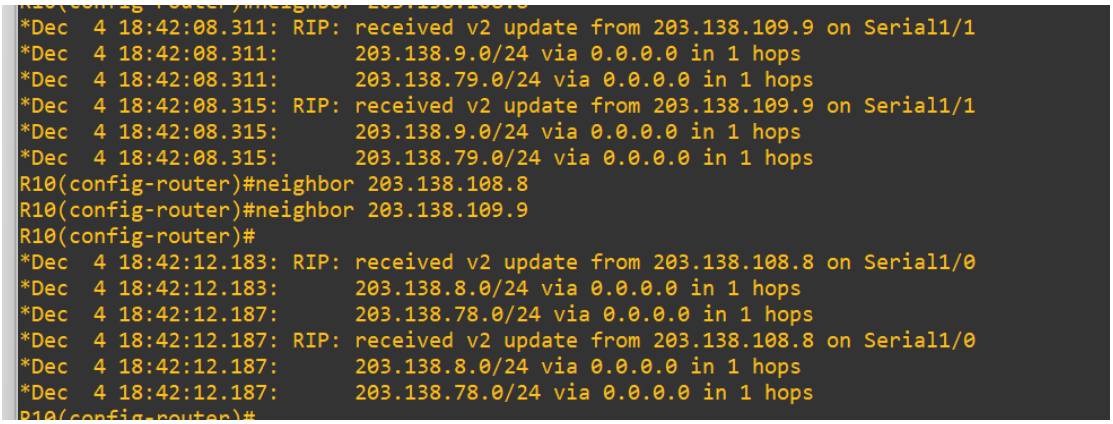


图 3-15 R10 构建路由表 1

首先，R10 收到从 203.138.109.9 和 203.138.108.8 发来的路由更新信息，告诉 R10 它们各自直连的网络有哪些，R10 更新完路由表后，将新的路由表信息发送给和自己直连的各个接口，如图 3-16。

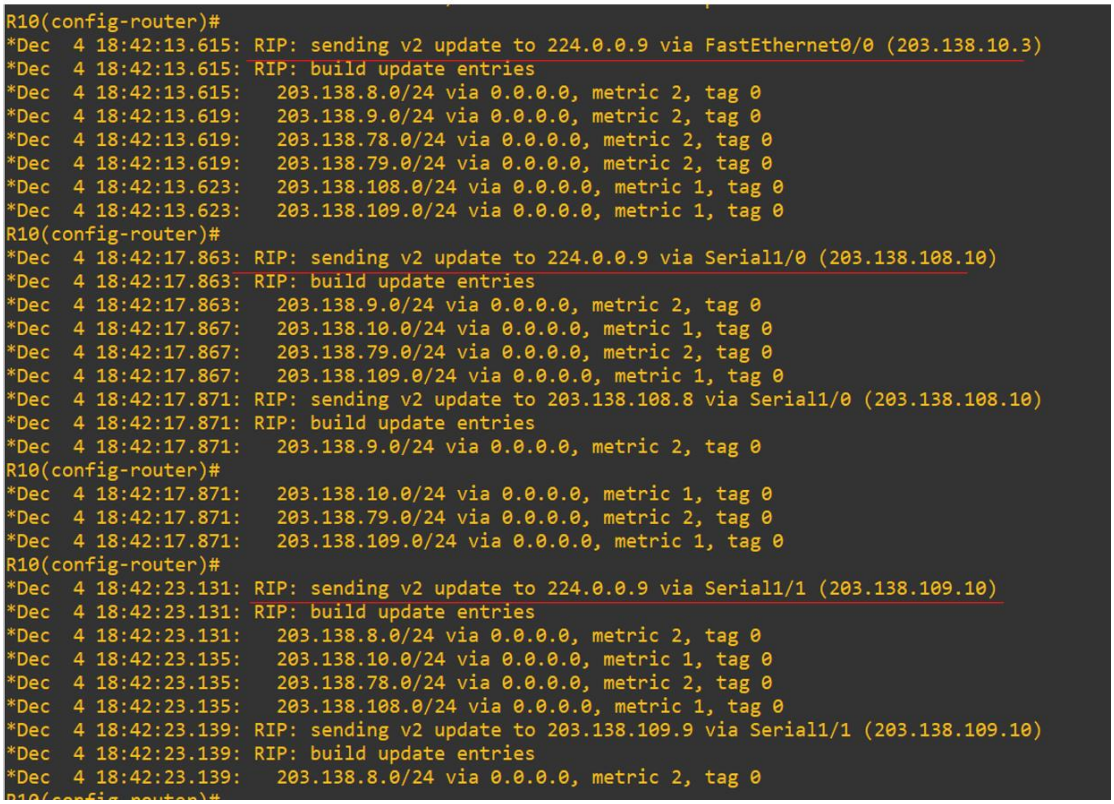


图 3-16 R10 发送路由表信息图 1

作为 R10 的邻居 R8 或 R9，在收到 R10 的消息后，将会更新各自的路由表信息，比如将 R10 连接的网络 203.138.10.0 记录到自己的路由表中，并设置跳

数为 2。更新完这个路由表信息后，它们会将更新的数据发送给自己的邻居。

```
*Dec  4 19:00:31.959: RIP: received v2 update from 203.138.108.8 on Serial1/0
*Dec  4 19:00:31.963:      203.138.1.0/24 via 0.0.0.0 in 6 hops
*Dec  4 19:00:31.967:      203.138.2.0/24 via 0.0.0.0 in 5 hops
*Dec  4 19:00:31.967:      203.138.3.0/24 via 0.0.0.0 in 5 hops
*Dec  4 19:00:31.971:      203.138.5.0/24 via 0.0.0.0 in 3 hops
*Dec  4 19:00:31.971:      203.138.6.0/24 via 0.0.0.0 in 3 hops
*Dec  4 19:00:31.971:      203.138.12.0/24 via 0.0.0.0 in 5 hops
*Dec  4 19:00:31.971:      203.138.13.0/24 via 0.0.0.0 in 5 hops
*Dec  4 19:00:31.971:      203.138.24.0/24 via 0.0.0.0 in 4 hops
*Dec  4 19:00:31.971:      203.138.34.0/24 via 0.0.0.0 in 4 hops
*Dec  4 19:00:31.971:      203.138.45.0/24 via 0.0.0.0 in 3 hops
*Dec  4 19:00:31.971:      203.138.46.0/24 via 0.0.0.0 in 3 hops
*Dec  4 19:00:31.971:      203.138.57.0/24 via 0.0.0.0 in 2 hops
*Dec  4 19:00:31.971:      203.138.67.0/24 via 0.0.0.0 in 2 hops
*Dec  4 19:00:31.971: RIP: received v2 update from 203.138.108.8
```

图 3-17 R10 收到 R8 路由表示意图

然后 R10 收到了来自 R8 的路由表信息，里面记载着 R8 能到达的网络号和所需要的跳数。如图 2-18，从 R8 经过到 203.138.1.0 需要 6 跳，到达 203.138.2.0 需要 5 跳……到达 203.138.67.0 需要 2 跳。

查看路由表信息：在 R10 中输入 show ip route，结果如下。其中 C 代表直连，R 代表是通过 RIP 协议动态获取的路由信息。可以看到，这个通过 RIP 协议，整个网络拓扑的所有网络都包含在路由表里了。

```
RIP protocol debugging is off
R10#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, 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, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

R    203.138.67.0/24 [120/2] via 203.138.109.9, 00:00:25, Serial1/1
                     [120/2] via 203.138.108.8, 00:00:08, Serial1/0
R    203.138.6.0/24 [120/3] via 203.138.109.9, 00:00:25, Serial1/1
                    [120/3] via 203.138.108.8, 00:00:08, Serial1/0
R    203.138.5.0/24 [120/3] via 203.138.109.9, 00:00:25, Serial1/1
                    [120/3] via 203.138.108.8, 00:00:08, Serial1/0
R    203.138.3.0/24 [120/5] via 203.138.109.9, 00:00:25, Serial1/1
                    [120/5] via 203.138.108.8, 00:00:08, Serial1/0
R    203.138.2.0/24 [120/5] via 203.138.109.9, 00:00:25, Serial1/1
                    [120/5] via 203.138.108.8, 00:00:08, Serial1/0
R    203.138.1.0/24 [120/6] via 203.138.109.9, 00:00:25, Serial1/1
                    [120/6] via 203.138.108.8, 00:00:08, Serial1/0
R    203.138.34.0/24 [120/4] via 203.138.109.9, 00:00:25, Serial1/1
                     [120/4] via 203.138.108.8, 00:00:08, Serial1/0
R    203.138.45.0/24 [120/3] via 203.138.109.9, 00:00:26, Serial1/1
                     [120/3] via 203.138.108.8, 00:00:09, Serial1/0
R    203.138.13.0/24 [120/5] via 203.138.109.9, 00:00:26, Serial1/1
                     [120/5] via 203.138.108.8, 00:00:09, Serial1/0
R    203.138.46.0/24 [120/3] via 203.138.109.9, 00:00:26, Serial1/1
                     [120/3] via 203.138.108.8, 00:00:09, Serial1/0
R    203.138.12.0/24 [120/5] via 203.138.109.9, 00:00:26, Serial1/1
                     [120/5] via 203.138.108.8, 00:00:09, Serial1/0
R    203.138.79.0/24 [120/1] via 203.138.109.9, 00:00:26, Serial1/1
     203.138.109.0/24 is variably subnetted, 2 subnets, 2 masks
C       203.138.109.9/32 is directly connected, Serial1/1
C       203.138.109.0/24 is directly connected, Serial1/1
C    203.138.10.0/24 is directly connected, FastEthernet0/0
R    203.138.57.0/24 [120/2] via 203.138.109.9, 00:00:26, Serial1/1
                     [120/2] via 203.138.108.8, 00:00:09, Serial1/0
R    203.138.78.0/24 [120/1] via 203.138.108.8, 00:00:09, Serial1/0
     203.138.108.0/24 is variably subnetted, 2 subnets, 2 masks
C       203.138.108.8/32 is directly connected, Serial1/0
C       203.138.108.0/24 is directly connected, Serial1/0
R    203.138.9.0/24 [120/1] via 203.138.109.9, 00:00:26, Serial1/1
R    203.138.24.0/24 [120/4] via 203.138.109.9, 00:00:26, Serial1/1
                     [120/4] via 203.138.108.8, 00:00:09, Serial1/0
  More
```

图 3-18 R10 路由信息图

最后检查网络之间能否互通，这里我选择使用相邻最远的 PC1 和 PC10 进行测试，测试结果如下：

```
PC1> ping 203.138.10.1
84 bytes from 203.138.10.1 icmp_seq=1 ttl=57 time=211.380 ms
84 bytes from 203.138.10.1 icmp_seq=2 ttl=57 time=210.461 ms
84 bytes from 203.138.10.1 icmp_seq=3 ttl=57 time=211.166 ms
84 bytes from 203.138.10.1 icmp_seq=4 ttl=57 time=210.828 ms
84 bytes from 203.138.10.1 icmp_seq=5 ttl=57 time=211.251 ms
```

图 3-19 PC1 ping 通 PC10

3.2.3 RIP 协议的水平分割

在 R9 的 S1/1 中，将水平分割关闭，步骤如下：

```
R9(config)#in s1/1
R9(config-if)#no ip split
R9(config-if)#no ip split-horizon
R9(config-if)#
```

图 3-20 关闭 R9 S1/1 的水平分割图

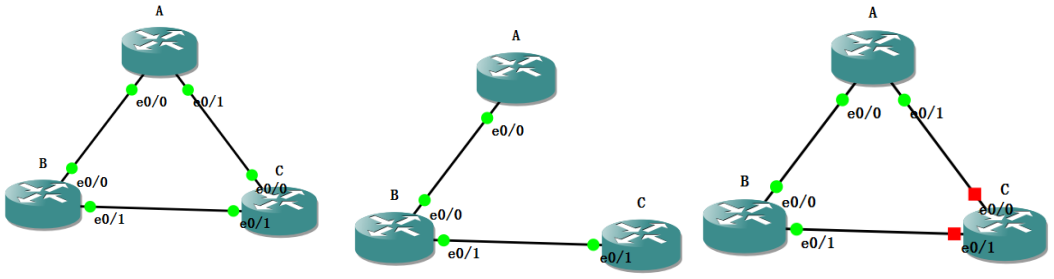
```
 received v2 update from 203.138.109.9
  203.138.1.0/24 via 0.0.0.0 in 6 hops
  203.138.2.0/24 via 0.0.0.0 in 5 hops
  203.138.3.0/24 via 0.0.0.0 in 5 hops
  203.138.5.0/24 via 0.0.0.0 in 3 hops
  203.138.6.0/24 via 0.0.0.0 in 3 hops
  203.138.9.0/24 via 0.0.0.0 in 1 hops
  203.138.12.0/24 via 0.0.0.0 in 5 hops
  203.138.13.0/24 via 0.0.0.0 in 5 hops
  203.138.24.0/24 via 0.0.0.0 in 4 hops
  203.138.34.0/24 via 0.0.0.0 in 4 hops
  203.138.45.0/24 via 0.0.0.0 in 3 hops
  203.138.46.0/24 via 0.0.0.0 in 3 hops
  203.138.57.0/24 via 0.0.0.0 in 2 hops
  203.138.67.0/24 via 0.0.0.0 in 2 hops
```

```
RIP: received v2 update from 203.138.109.9 on Serial1/1
     203.138.1.0/24 via 0.0.0.0 in 6 hops
     203.138.2.0/24 via 0.0.0.0 in 5 hops
     203.138.3.0/24 via 0.0.0.0 in 5 hops
     203.138.5.0/24 via 0.0.0.0 in 3 hops
     203.138.6.0/24 via 0.0.0.0 in 3 hops
     203.138.8.0/24 via 0.0.0.0 in 3 hops
     203.138.9.0/24 via 0.0.0.0 in 1 hops
     203.138.10.0/24 via 0.0.0.0 in 2 hops
     203.138.12.0/24 via 0.0.0.0 in 5 hops
     203.138.13.0/24 via 0.0.0.0 in 5 hops
     203.138.24.0/24 via 0.0.0.0 in 4 hops
     203.138.34.0/24 via 0.0.0.0 in 4 hops
     203.138.45.0/24 via 0.0.0.0 in 3 hops
     203.138.46.0/24 via 0.0.0.0 in 3 hops
```

图 3-21 R10 接收信息对比图

如图 3-21，左边为关闭水平分割前从 R9 收到的信息，右边为关闭水平分割后从 R9 收到的信息。它们的差别就在于：关闭水平分割后，R9 会向 R10 发送从 R10 发给 R9 的消息。水平分割的含义指的是 RIP 从某个接口接收到的路由信息，不会从该接口再发给邻居设备。它的好处是，不但减少了带宽消耗，还可以防止路由环路。

路由环路解析



(a) (b) (c)

图 3-22 路由环路示意图

路由环路是这样一种情况，拓扑图如图 3-22。

在状态（a）时，A 和 B 都直连 C。当 A 和 C 之间断开后，如图状态（b），B 通知 A，到达 C 的距离是 1，A 此时只能通过 B 到达 C。当 B 和 C 的连接也断开后，如图状态（c）。由于没有水平分割，A 告诉 B 到达 C 需要 2 跳，因此 B 将自己到 C 的路由信息设置为 3。B 又告诉 A 到达 C 需要 3 跳，A 将自己的路由信息修改成 4 跳并告诉 B……就这样形成了一个路由环路。这会大大影响拓扑的性能。

3.3 OSPF 协议

3.3.1 OSPF 协议配置过程

首先，将原来的 RIP 协议配置从所有路由器内删除干净，具体步骤为：conf t 进入路由器的配置模式，输入 no router rip，将 rip 协议删除。

```
                    [120/5] via 203.138.12.2, 00:00:22, Serial1/0
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#no router rip
```

图 3-23 删除 rip 协议示意图

在还没配协议前，打开 OSPF 的 debug 信息，用来观察网络中路由器建立路由信息的过程。打开 debug 信息命令为：

```
debug ip ospf events                -- 打开事件

debug ip ospf flood                 -- 打开洪泛信息

sh ip ospf neighbor                 -- 展示邻居信息
```

```
R1#debug ip ospf events
OSPF events debugging is on
R1#debug ip ospf flood
OSPF flooding debugging is on
```

图 3-24 打开 R1 ospf 的 debug 信息

由于打开 debug 信息后，配置过程难以截图，因此这一过程没有步骤图片。

打开 R1 的 debug 信息后，要配置 R1 的 OSPF 协议，步骤如下：

```
conf t
   router ospf 1
      network 203.138.1.0 255.255.255.0 area 0
      network 203.138.12.0 255.255.255.0 area 0
      network 203.138.13.0 255.255.255.0 area 0
   in s1/0
      ip ospf hello-interval 5
      ip ospf dead-interval 20
   in s1/1
      ip ospf hello-interval 5
      ip ospf dead-interval 20
```

步骤说明：

（1）router ospf 1 表示给 R1 配上 ospf 路由，放在进程 1 里

（2）network <网络号> <子网掩码> <区域号> 表示和 R1 直接相连的网络号，并将区域设置为 0。在本次实验中，并不涉及区域的设计，所以都配置成 area 0。

（3）进入相应的接口，给为该接口的 hello 包配置一个间隔。并设置一个间隔 20，表示多久没收到这个邻居的消息就说明这个邻居宕机了。

配完 R1 之后就会显示如下 debug 信息：

```
R1#
*Dec  5 13:28:30.067: OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet0/0 from 203.138.1.3
R1#
*Dec  5 13:28:31.431: OSPF: Send hello to 224.0.0.5 area 0 on Serial1/0 from 203.138.12.1
*Dec  5 13:28:31.635: OSPF: Send hello to 224.0.0.5 area 0 on Serial1/1 from 203.138.13.1
R1#
```

图 3-25 R1 刚配置好的 debug 信息图

同理，为 R2 配置 OSPF 协议，debug 信息如下：

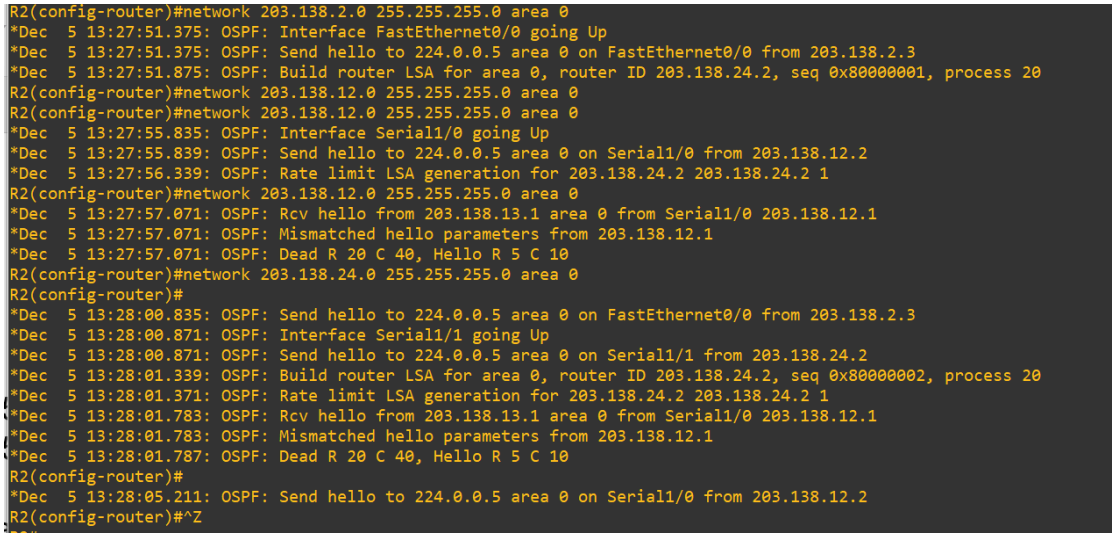


图 3-26 R2 配置 OSPF 的 debug 信息图

debug 信息说明：

(1) OSPF: Interface FastEthernet0/0 going Up

OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet0/0 from 203.138.2.3

这表示 FastEthernet0/0 接口状态变为 Up，同时 OSPF 发送了 Hello 消息到 224.0.0.5（OSPF 的多播地址）以通知其他 OSPF 路由器。

(2) OSPF: Build router LSA for area 0, router ID 203.138.24.2, seg 0x80000001, process 2

表示 OSPF 正在为区域 0 构建 Router LSA(Link State Advertisement)。

(3) *Dec 5 13:35:21.039: OSPF: Rcv hello from 203.138.13.1 area 0 from Serial1/0 203.138.12.1

表示 OSPF 协议从 Serial1/0 接口收到了来自 203.138.13.1 的 Hello 消息。这是指有一个 OSPF 相邻路由器（203.138.13.1）通过 Serial1/0 接口发送了 Hello 消息。

在 R1 中输入 sh ip ospf neighbor，得到如下信息：

```
R1#sh ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
203.138.34.3      0   FULL/  -        00:00:17    203.138.13.3    Serial1/1
203.138.24.2      0   FULL/  -        00:00:18    203.138.12.2    Serial1/0
R1#
```

图 3-27 R1 的 neighbor 表

其中有两个 neighbor 信息，第一条表示 邻居 203.138.34.3 优先级是 0，状态是 FULL，还有 17 秒没收到邻居信息就说明它宕机了，邻居的路由 ip 地址是 203.138.13.3，通过接口 S1/1 进行连接。

将所有路由的 OSPF 协议配置好后，检查所有路由器的路由表。

在 R1 下输入 show ip route，显示 R1 的路由表，如下图所示，

```
R1#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, 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, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

O    203.138.67.0/24 [110/256] via 203.138.13.3, 00:32:17, Serial1/1
                     [110/256] via 203.138.12.2, 00:32:17, Serial1/0
O    203.138.6.0/24 [110/193] via 203.138.13.3, 00:32:17, Serial1/1
                    [110/193] via 203.138.12.2, 00:32:17, Serial1/0
O    203.138.5.0/24 [110/193] via 203.138.13.3, 00:34:35, Serial1/1
                    [110/193] via 203.138.12.2, 00:34:35, Serial1/0
O    203.138.3.0/24 [110/65] via 203.138.13.3, 00:48:03, Serial1/1
O    203.138.2.0/24 [110/65] via 203.138.12.2, 00:57:15, Serial1/0
C    203.138.1.0/24 is directly connected, FastEthernet0/0
O    203.138.34.0/24 [110/128] via 203.138.13.3, 00:48:03, Serial1/1
O    203.138.45.0/24 [110/192] via 203.138.13.3, 00:38:04, Serial1/1
                     [110/192] via 203.138.12.2, 00:38:14, Serial1/0
     203.138.13.0/24 is variably subnetted, 2 subnets, 2 masks
C       203.138.13.0/24 is directly connected, Serial1/1
C       203.138.13.3/32 is directly connected, Serial1/1
O    203.138.46.0/24 [110/192] via 203.138.13.3, 00:38:05, Serial1/1
                     [110/192] via 203.138.12.2, 00:38:15, Serial1/0
     203.138.12.0/24 is variably subnetted, 2 subnets, 2 masks
C       203.138.12.0/24 is directly connected, Serial1/0
C       203.138.12.2/32 is directly connected, Serial1/0
O    203.138.79.0/24 [110/320] via 203.138.13.3, 00:30:58, Serial1/1
                     [110/320] via 203.138.12.2, 00:30:58, Serial1/0
O    203.138.109.0/24 [110/384] via 203.138.13.3, 00:21:30, Serial1/1
                      [110/384] via 203.138.12.2, 00:21:30, Serial1/0
O    203.138.10.0/24 [110/385] via 203.138.13.3, 00:18:35, Serial1/1
                     [110/385] via 203.138.12.2, 00:18:35, Serial1/0
O    203.138.57.0/24 [110/256] via 203.138.13.3, 00:34:36, Serial1/1
                     [110/256] via 203.138.12.2, 00:34:36, Serial1/0
O    203.138.78.0/24 [110/320] via 203.138.13.3, 00:30:58, Serial1/1
                     [110/320] via 203.138.12.2, 00:30:58, Serial1/0
O    203.138.108.0/24 [110/384] via 203.138.13.3, 00:22:46, Serial1/1
                      [110/384] via 203.138.12.2, 00:22:46, Serial1/0
O    203.138.9.0/24 [110/321] via 203.138.13.3, 00:21:30, Serial1/1
                    [110/321] via 203.138.12.2, 00:21:30, Serial1/0
O    203.138.24.0/24 [110/128] via 203.138.12.2, 00:57:15, Serial1/0
O    203.138.8.0/24 [110/321] via 203.138.13.3, 00:22:46, Serial1/1
                    [110/321] via 203.138.12.2, 00:22:46, Serial1/0
```

图 3-28 R1 的 ospf 路由表

其中，前缀为 C 的是直接连接的，前缀为 O 的是通过 OSPF 协议动态更新的

路由信息。

尝试用网络中的每个 PC 来 ping 其他的 PC，看看网络是否每个节点都能 ping 通。下面展示 PC1 ping PC10，成功 ping 通，结果如下：

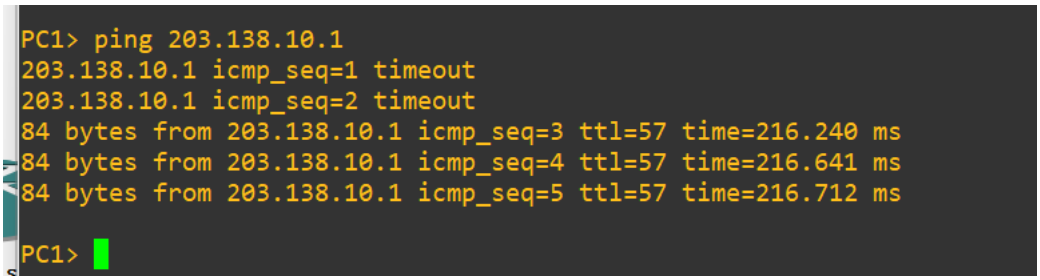


图 3-29 PC1 ping PC10 结果图

3.3.2 OSPF 链路状态信息更新过程

将 R1 一直到 R10 配置好 OSPF 协议，观察 R1 的 debug 信息。当 R10 刚配置好时，R1 收到的其中一条 debug 信息如下：

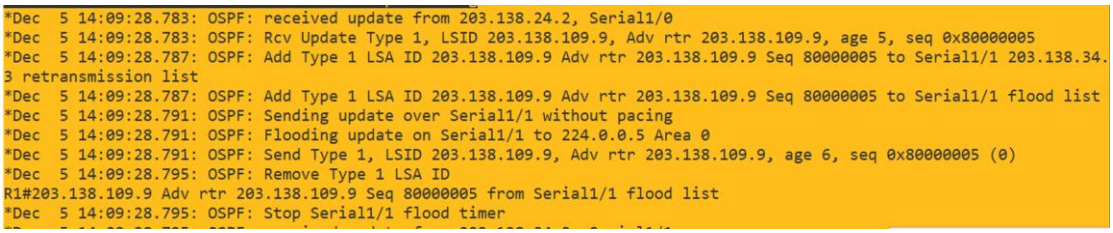


图 3-30 R1 的 update debug 信息

(1) OSPF: received update from 203.138.24.2, Serial1/0

这条语句表示从 203.138.24.2，串口 1/0 收到了一条链路状态更新信息。

(2) OSPF: Rcv Update Type 1, LSID 203.138.109.9, Adv rtr 203.138.109.9, age 5, seq 0x80000005

表示收到了类型为 1 的 OSPF 路由更新，LSA 的 ID 是 203.138.109.9，广播路由器是 203.138.109.9，路由信息的年龄是 5 秒，序列号是 0x80000005。

(3) OSPF: Add Type 1 LSA ID 203.138.109.9 Adv rtr 203.138.109.9 Seq 80000005 to Serial1/1 203.138.34.3 retransmission list

表示将 LSA ID 为 203.138.109.9 的路由信息添加到 Serial1/1 接口的重传列表中。

(4) OSPF: Sending update over Serial1/1 without pacing

表示通过 Serial1/1 接口发送 OSPF 路由更新，不使用 pacing。

(5) OSPF: Flooding update on Serial1/1 to 224.0.0.5 Area 0

表示在 Serial1/1 接口向 OSPF Area 0 的多播地址 224.0.0.5 广播更新。

(6) OSPF: Send Type 1, LSID 203.138.109.9, Adv rtr 203.138.109.9, age 6, seq 0x80000005 (0)

表示发送了类型为 1 的 OSPF 路由更新，其中广播的消息来源于 LSA ID 203.138.109.9 的信息。

(7) OSPF: Remove Type 1 LSA ID 203.138.109.9 Adv rtr 203.138.109.9 Seq 80000005 from Serial1/1 flood list

表示将 LSA ID 为 203.138.109.9 的路由信息从 Serial1/1 接口的洪泛列表中移除。

(8) OSPF: Stop Serial1/1 flood timer

表示停止 Serial1/1 接口的洪泛计时器。

上述 debug 信息详细描述了 OSPF 协议在链路状态信息更改时一个路由器所作的工作。首先，路由器收到了一个状态更新信息，里面包含了是什么类型的更新，以及发送该路由信息的路由 ip 地址。接着，路由器更新自己的路由表。然后，路由器通过另一个串口发送更新信息的洪泛包，并设置洪泛计时器。当计时器到时的时候，就将该洪泛信息从洪泛表中删除，并停止该计时器。

3.4 RIP 和 OSPF 协议的区别

3.4.1 RIP 协议特点

RIP 协议每隔一段时间就送出自己完整的路由表到所有激活的接口。(不管网络有没有变换，周期性的。将自己所连接的网段和学到的路由信息，通过这些口告诉其他的路由器。如果路由表条数多，那么可能需要发好几个包才能将路由表通告出去，如果条数较少，可能一个数据包就可以发送完毕)

RIP 协议选择最佳路径的标准就是跳数，认为到达目标网络经过的路由器最少的路径就是最佳路径。

默认它所允许的最大跳数为 15 跳，也就是说 16 跳的距离将被认为是不可达的。在小型网络中，RIP 会运转良好，但是对于使用慢速 WAN 连接的大型网络或者安装有大量路由器的网络来说，它的效率就很低了。

3.4.2 RIP 协议工作过程

路由器在刚刚开始工作时，只知道到直接连接的网络的距离（此距离定义为 0)。它的路由表是空的。

以后，每一个路由器也只和数目非常有限的相邻路由器交换并更新路由信息。

经过若干次更新后，所有的路由器最终都会知道到达本自治系统中任何一个网络的最短距离和下一跳路由器的地址。(最后运行 RIP 协议的路由器，经过一段时间更新之后，都知道网络当中有多少网段，下一跳给谁)

距离矢量算法:

路由器收到相邻路由器（其地址为 X）的一个 RIP 报文：

（1） 先修改此 RIP 报文中的所有项目：把“下一跳”字段中的地址都改为 X，并把所有的“距离”字段的值加 1。

（2）对修改后的 RIP 报文中的每一个项目，重复以下步骤：

若项目中的目的网络不在路由表中，则把该项目加到路由表中。

若在网络中，且下一跳字段给出的路由器地址是同样的，则把收到的项目替换原路由表中的项目。若收到项目中的距离小于路由表中的距离，则进行更新。

否则，什么也不做。

（3）若 3 分钟还没有收到相邻路由器的更新路由表，则把此相邻路由器记为不可达路由器，即将距离置为 16，即不可达，并且将到该网段的路由删除。

3.4.1 OSPF 协议的特点

OSPF 协议使用 Dijkstra 算法来选择最短路径并将其加入到路由表中。

OSPF 在网络中建立一个联通域，该域中的所有路由器都互相交换信息，以便维护一张完整的拓扑图。当路径发生变化时，路由器会通过 LSA（Link State Advertisements）报告给同一联通域内的其他路由器。

每个路由器都使用 LSA 来更新它的链路状态数据库并重新计算出最短路径。这些信息则会被同步到整个联通域内的其他路由器。

运行 OSPF 路由器之间交互的是 LS（Link State，链路状态）信息，而不是直接交互路由。LS 信息是 OSPF 能够正常进行拓扑及路由计算的关键信息。

OSPF 路由器将网络中的 LS 信息收集起来，存储在 LSDB 中。路由器都清楚

区域内的网络拓扑结构，这有助于路由器计算无环路径。

3.4.2 OSPF 协议的工作过程

（1）OSPF 路由器启动后，便会通过 OSPF 接口向外发送 Hello 报文用于发现邻居。收到 Hello 报文的 OSPF 路由器会检查报文中所定义的一些参数，如果双方的参数一致，就会彼此形成邻居关系，状态到达 2-way 即可称为建立了邻居关系。

Hello 报文用来发现和维持 OSPF 邻居关系。

Hello 报文的作用：

1°邻居发现：自动发现邻居路由器。

2°邻居建立：完成 Hello 报文中的参数协商，建立邻居关系。

3°邻居保持：通过 Keepalive 机制，检测邻居运行状态。

（2）建立邻接关系，形成链路状态数据库

形成邻居关系的双方不一定都能形成邻接关系，这要根据网络类型而定。当双方成功交换 DD 报文，并同步 LSDB 后，才形成真正意义上的邻接关系。路由器一旦建立了邻居关系，就可以创建链路状态数据包。

（3）交换链路信息

路由器将描述链路状态的 LSA 泛洪到邻居，最终形成包含网络完整链路状态信息的链路状态数据库。

（4）计算最短路径

路由区域内的每台路由器都可以使用 SPF 算法来独立计算路由。

（5）链路状态更新

当链路状态更新时，新加入的路由器或者宕机的路由器的邻居会察觉到链路状态更新，并先更新自己的链路状态数据库。然后构建链路状态改变的广播包，将该广播包放到自己的洪泛列表中。在接下去一段时间内朝自己的各个启动的邻居发送洪泛包。当洪泛时间到达后，该路由器会把该洪泛包从洪泛列表里删除，并停止洪泛计时器。（具体请看 3.3.2 节图 3-30 以及下方解析过程）

四、实验结果（包括最终实验结果，需要截图）

4.1 RIP 协议实验结果

以 R10 为例，因为 R10 我是最后配置的，信息比较完整。

```
*Dec  4 18:42:08.311: RIP: received v2 update from 203.138.109.9 on Serial1/1
*Dec  4 18:42:08.311:      203.138.9.0/24 via 0.0.0.0 in 1 hops
*Dec  4 18:42:08.311:      203.138.79.0/24 via 0.0.0.0 in 1 hops
*Dec  4 18:42:08.315: RIP: received v2 update from 203.138.109.9 on Serial1/1
*Dec  4 18:42:08.315:      203.138.9.0/24 via 0.0.0.0 in 1 hops
*Dec  4 18:42:08.315:      203.138.79.0/24 via 0.0.0.0 in 1 hops
R10(config-router)#neighbor 203.138.108.8
R10(config-router)#neighbor 203.138.109.9
R10(config-router)#
*Dec  4 18:42:12.183: RIP: received v2 update from 203.138.108.8 on Serial1/0
*Dec  4 18:42:12.183:      203.138.8.0/24 via 0.0.0.0 in 1 hops
*Dec  4 18:42:12.187:      203.138.78.0/24 via 0.0.0.0 in 1 hops
*Dec  4 18:42:12.187: RIP: received v2 update from 203.138.108.8 on Serial1/0
*Dec  4 18:42:12.187:      203.138.8.0/24 via 0.0.0.0 in 1 hops
*Dec  4 18:42:12.187:      203.138.78.0/24 via 0.0.0.0 in 1 hops
```

图 4-1 R10 构建路由表 1

首先，R10 收到从 203.138.109.9 和 203.138.108.8 发来的路由更新信息，告诉 R10 它们各自直连的网络有哪些，R10 更新完路由表后，将新的路由表信息发送给和自己直连的各个接口，如图 4-5。

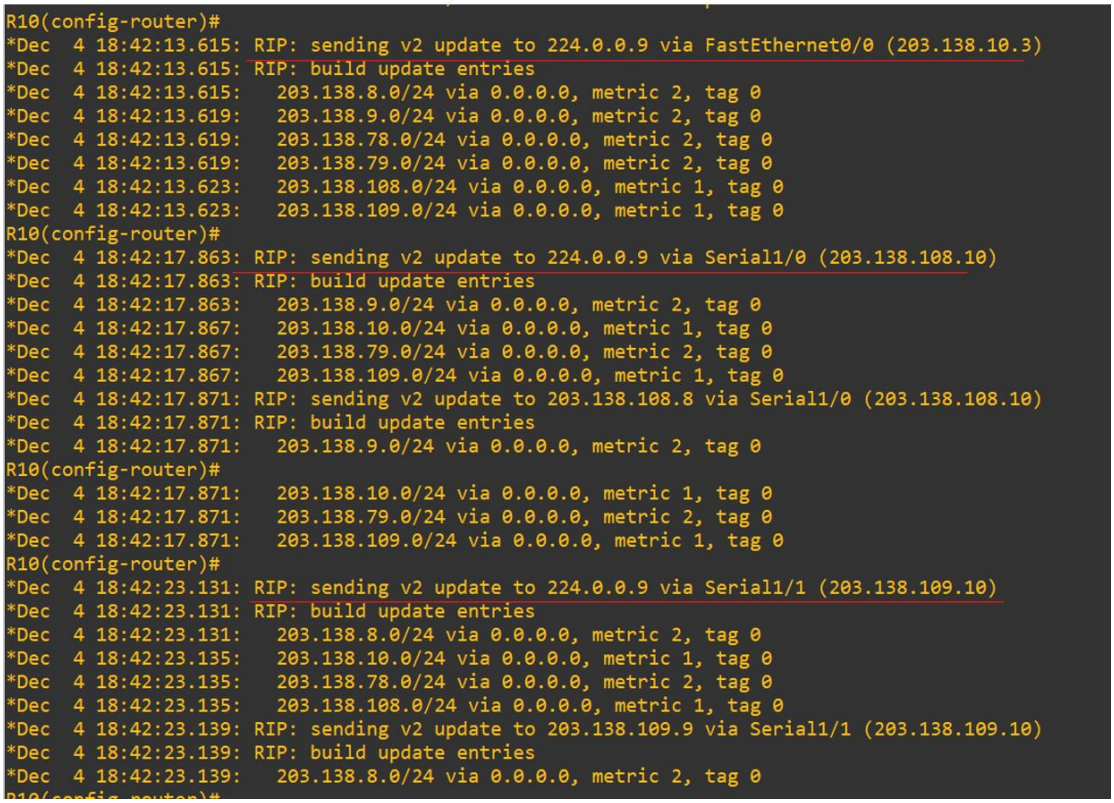


图 4-5 R10 发送路由表信息图 1

作为 R10 的邻居 R8 或 R9，在收到 R10 的消息后，将会更新各自的路由表信息，比如将 R10 连接的网络 203.138.10.0 记录到自己的路由表中，并设置跳数为 2。更新完这个路由表信息后，它们会将更新的数据发送给自己的邻居。

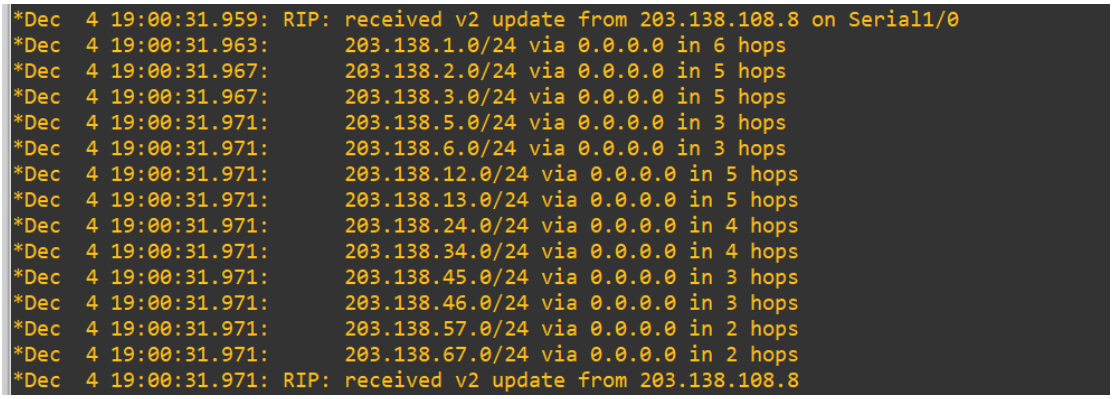


图 4-3 R10 收到 R8 路由表示意图

然后 R10 收到了来自 R8 的路由表信息，里面记载着 R8 能到达的网络号和所需要的跳数。如图 4-4，从 R8 经过到 203.138.1.0 需要 6 跳，到达 203.138.2.0

需要 5 跳……到达 203.138.67.0 需要 2 跳。

查看路由表信息：在 R10 中输入 show ip route，结果如下。其中 C 代表直连，R 代表是通过 RIP 协议动态获取的路由信息。可以看到，这个通过 RIP 协议，整个网络拓扑的所有网络都包含在路由表里了。

```
R10#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, 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, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

R    203.138.67.0/24 [120/2] via 203.138.109.9, 00:00:25, Serial1/1
                     [120/2] via 203.138.108.8, 00:00:08, Serial1/0
R    203.138.6.0/24 [120/3] via 203.138.109.9, 00:00:25, Serial1/1
                    [120/3] via 203.138.108.8, 00:00:08, Serial1/0
R    203.138.5.0/24 [120/3] via 203.138.109.9, 00:00:25, Serial1/1
                    [120/3] via 203.138.108.8, 00:00:08, Serial1/0
R    203.138.3.0/24 [120/5] via 203.138.109.9, 00:00:25, Serial1/1
                    [120/5] via 203.138.108.8, 00:00:08, Serial1/0
R    203.138.2.0/24 [120/5] via 203.138.109.9, 00:00:25, Serial1/1
                    [120/5] via 203.138.108.8, 00:00:08, Serial1/0
R    203.138.1.0/24 [120/6] via 203.138.109.9, 00:00:25, Serial1/1
                    [120/6] via 203.138.108.8, 00:00:08, Serial1/0
R    203.138.34.0/24 [120/4] via 203.138.109.9, 00:00:25, Serial1/1
                     [120/4] via 203.138.108.8, 00:00:08, Serial1/0
R    203.138.45.0/24 [120/3] via 203.138.109.9, 00:00:26, Serial1/1
                     [120/3] via 203.138.108.8, 00:00:09, Serial1/0
R    203.138.13.0/24 [120/5] via 203.138.109.9, 00:00:26, Serial1/1
                     [120/5] via 203.138.108.8, 00:00:09, Serial1/0
R    203.138.46.0/24 [120/3] via 203.138.109.9, 00:00:26, Serial1/1
                     [120/3] via 203.138.108.8, 00:00:09, Serial1/0
R    203.138.12.0/24 [120/5] via 203.138.109.9, 00:00:26, Serial1/1
                     [120/5] via 203.138.108.8, 00:00:09, Serial1/0
R    203.138.79.0/24 [120/1] via 203.138.109.9, 00:00:26, Serial1/1
     203.138.109.0/24 is variably subnetted, 2 subnets, 2 masks
C       203.138.109.9/32 is directly connected, Serial1/1
C       203.138.109.0/24 is directly connected, Serial1/1
C    203.138.10.0/24 is directly connected, FastEthernet0/0
R    203.138.57.0/24 [120/2] via 203.138.109.9, 00:00:26, Serial1/1
                     [120/2] via 203.138.108.8, 00:00:09, Serial1/0
R    203.138.78.0/24 [120/1] via 203.138.108.8, 00:00:09, Serial1/0
     203.138.108.0/24 is variably subnetted, 2 subnets, 2 masks
C       203.138.108.8/32 is directly connected, Serial1/0
C       203.138.108.0/24 is directly connected, Serial1/0
R    203.138.9.0/24 [120/1] via 203.138.109.9, 00:00:26, Serial1/1
R    203.138.24.0/24 [120/4] via 203.138.109.9, 00:00:26, Serial1/1
                     [120/4] via 203.138.108.8, 00:00:09, Serial1/0
```

图 4-4 R10 路由信息图

最后检查网络之间能否互通，这里我选择使用相邻最远的 PC1 和 PC10 进行测试，测试结果如下：

```
PC1> ping 203.138.10.1
84 bytes from 203.138.10.1 icmp_seq=1 ttl=57 time=211.380 ms
84 bytes from 203.138.10.1 icmp_seq=2 ttl=57 time=210.461 ms
84 bytes from 203.138.10.1 icmp_seq=3 ttl=57 time=211.166 ms
84 bytes from 203.138.10.1 icmp_seq=4 ttl=57 time=210.828 ms
84 bytes from 203.138.10.1 icmp_seq=5 ttl=57 time=211.251 ms
```

图 4-5 PC1 ping 通 PC10

对于水平分割协议，结果如下：

在 R9 的 S1/1 中，将水平分割关闭，步骤如下：

```
R9(config)#in s1/1
R9(config-if)#no ip split
R9(config-if)#no ip split-horizon
R9(config-if)#
```

图 4-6 关闭 R9 S1/1 的水平分割图

```
received v2 update from 203.138.109.9
 203.138.1.0/24 via 0.0.0.0 in 6 hops
 203.138.2.0/24 via 0.0.0.0 in 5 hops
 203.138.3.0/24 via 0.0.0.0 in 5 hops
 203.138.5.0/24 via 0.0.0.0 in 3 hops
 203.138.6.0/24 via 0.0.0.0 in 3 hops
 203.138.9.0/24 via 0.0.0.0 in 1 hops
 203.138.12.0/24 via 0.0.0.0 in 5 hops
 203.138.13.0/24 via 0.0.0.0 in 5 hops
 203.138.24.0/24 via 0.0.0.0 in 4 hops
 203.138.34.0/24 via 0.0.0.0 in 4 hops
 203.138.45.0/24 via 0.0.0.0 in 3 hops
 203.138.46.0/24 via 0.0.0.0 in 3 hops
 203.138.57.0/24 via 0.0.0.0 in 2 hops
 203.138.67.0/24 via 0.0.0.0 in 2 hops
```

```
RIP: received v2 update from 203.138.109.9 on Serial1/1
     203.138.1.0/24 via 0.0.0.0 in 6 hops
     203.138.2.0/24 via 0.0.0.0 in 5 hops
     203.138.3.0/24 via 0.0.0.0 in 5 hops
     203.138.5.0/24 via 0.0.0.0 in 3 hops
     203.138.6.0/24 via 0.0.0.0 in 3 hops
     203.138.8.0/24 via 0.0.0.0 in 3 hops
     203.138.9.0/24 via 0.0.0.0 in 1 hops
     203.138.10.0/24 via 0.0.0.0 in 2 hops
     203.138.12.0/24 via 0.0.0.0 in 5 hops
     203.138.13.0/24 via 0.0.0.0 in 5 hops
     203.138.24.0/24 via 0.0.0.0 in 4 hops
     203.138.34.0/24 via 0.0.0.0 in 4 hops
     203.138.45.0/24 via 0.0.0.0 in 3 hops
     203.138.46.0/24 via 0.0.0.0 in 3 hops
```

图 4-7 R10 接收信息对比图

如图 4-7，左边为关闭水平分割前从 R9 收到的信息，右边为关闭水平分割后从 R9 收到的信息。它们的差别就在于：关闭水平分割后，R9 会向 R10 发送从 R10 发给 R9 的消息。水平分割的含义指的是 RIP 从某个接口接收到的路由信息，不会从该接口再发给邻居设备。它的好处是，不但减少了带宽消耗，还可以防止路由环路（在实验报告 3.2.3 节有路由环路解析）。

RIP 协议的 debug 信息可以看实验报告的 RIP 协议的工作流程

4.2 OSPF 协议实验结果

将所有路由的 OSPF 协议配置好后，检查所有路由器的路由表。

在 R1 下输入 show ip route，显示 R1 的路由表，如下图所示，

```
R1#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, 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, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

O    203.138.67.0/24 [110/256] via 203.138.13.3, 00:32:17, Serial1/1
                     [110/256] via 203.138.12.2, 00:32:17, Serial1/0
O    203.138.6.0/24 [110/193] via 203.138.13.3, 00:32:17, Serial1/1
                    [110/193] via 203.138.12.2, 00:32:17, Serial1/0
O    203.138.5.0/24 [110/193] via 203.138.13.3, 00:34:35, Serial1/1
                    [110/193] via 203.138.12.2, 00:34:35, Serial1/0
O    203.138.3.0/24 [110/65] via 203.138.13.3, 00:48:03, Serial1/1
O    203.138.2.0/24 [110/65] via 203.138.12.2, 00:57:15, Serial1/0
C    203.138.1.0/24 is directly connected, FastEthernet0/0
O    203.138.34.0/24 [110/128] via 203.138.13.3, 00:48:03, Serial1/1
O    203.138.45.0/24 [110/192] via 203.138.13.3, 00:38:04, Serial1/1
                     [110/192] via 203.138.12.2, 00:38:14, Serial1/0
     203.138.13.0/24 is variably subnetted, 2 subnets, 2 masks
C       203.138.13.0/24 is directly connected, Serial1/1
C       203.138.13.3/32 is directly connected, Serial1/1
O    203.138.46.0/24 [110/192] via 203.138.13.3, 00:38:05, Serial1/1
                     [110/192] via 203.138.12.2, 00:38:15, Serial1/0
     203.138.12.0/24 is variably subnetted, 2 subnets, 2 masks
C       203.138.12.0/24 is directly connected, Serial1/0
C       203.138.12.2/32 is directly connected, Serial1/0
O    203.138.79.0/24 [110/320] via 203.138.13.3, 00:30:58, Serial1/1
                     [110/320] via 203.138.12.2, 00:30:58, Serial1/0
O    203.138.109.0/24 [110/384] via 203.138.13.3, 00:21:30, Serial1/1
                      [110/384] via 203.138.12.2, 00:21:30, Serial1/0
O    203.138.10.0/24 [110/385] via 203.138.13.3, 00:18:35, Serial1/1
                     [110/385] via 203.138.12.2, 00:18:35, Serial1/0
O    203.138.57.0/24 [110/256] via 203.138.13.3, 00:34:36, Serial1/1
                     [110/256] via 203.138.12.2, 00:34:36, Serial1/0
O    203.138.78.0/24 [110/320] via 203.138.13.3, 00:30:58, Serial1/1
                     [110/320] via 203.138.12.2, 00:30:58, Serial1/0
O    203.138.108.0/24 [110/384] via 203.138.13.3, 00:22:46, Serial1/1
                      [110/384] via 203.138.12.2, 00:22:46, Serial1/0
O    203.138.9.0/24 [110/321] via 203.138.13.3, 00:21:30, Serial1/1
                    [110/321] via 203.138.12.2, 00:21:30, Serial1/0
O    203.138.24.0/24 [110/128] via 203.138.12.2, 00:57:15, Serial1/0
O    203.138.8.0/24 [110/321] via 203.138.13.3, 00:22:46, Serial1/1
                    [110/321] via 203.138.12.2, 00:22:46, Serial1/0
```

图 4-8 R1 的 ospf 路由表

其中，前缀为 C 的是直接连接的，前缀为 O 的是通过 OSPF 协议动态更新的路由信息。

尝试用网络中的每个 PC 来 ping 其他的 PC，看看网络是否每个节点都能 ping 通。下面展示 PC1 ping PC10，成功 ping 通，结果如下：

```
PC1> ping 203.138.10.1
203.138.10.1 icmp_seq=1 timeout
203.138.10.1 icmp_seq=2 timeout
84 bytes from 203.138.10.1 icmp_seq=3 ttl=57 time=216.240 ms
84 bytes from 203.138.10.1 icmp_seq=4 ttl=57 time=216.641 ms
84 bytes from 203.138.10.1 icmp_seq=5 ttl=57 time=216.712 ms

PC1>
```

图 4-9 PC1 ping PC10 结果图

OSPF 的链路状态更新过程及分析可以查看 3.3.2 OSPF 链路状态更新过程

五、实验中的问题及心得

5.1 实验中的问题

（1)第一次做实验三时,我使用了 Dynamips 作为仿真工具。我的 PC3 死活 ping 不通别的设备，别的设备也 ping 不通 PC3。我检查了我的 ip 地址设置，检查了我的默认路由设置，检查了我的配置文件编写，几乎用了所有的查看命令，比如 show ip route、show run、show interface 等等……但是都没能发现实际的问题所在，只有 show interface 显示出我的 PC3 f0/0 接口和 R3 的 f0/0 接口 protocol 是 down 的状态，但是接口本身是 up 的状态。我用 PC3 自己 ping 自己倒是能 ping 通。我询问了老师原因，才知道原来 Dynamips 会有一些 bug，出现连接不通的情况下可以去配置文件中，把原来 R3 f0/0 连接到 PC3 的配置改到 PC3 f0/0 连到 R3 上。重新打开命令行和设备，才成功 ping 通。

（2）见识到 Dynamips 的险恶之后，我尝试使用 GNS3 作为我的仿真工具。但是在配 GNS3 的时候卡了半天。首先是 GNS3 的虚拟机老是装不成功，我又决定不装虚拟机，直接在本机上跑。但是老是出现 wait for localhost …… 的弹窗，让我烦不胜烦。我删掉重装了好几次 GNS3，最后沉下心来成功安装了虚拟机，并且设置好了环境，也导入了所需要的路由器，这才得以安心实验。

（3）在 GNS3 软件上，我被一个小 bug 折磨了好久。一开始，我想知道 GNS3 的简单工作流程，我设计了一个简单的拓扑，两台 PC 机和一台路由器，我设置好了 IP 和网关，但是两台 PC 怎么样也 ping 不通。我上网找也没找到有相关的问题，我还以为是我安装软件忘记配置什么东西了。

我想着抓包分析一下包在哪里被丢掉了，但是在抓包时我又犯难了。打开

Wireshark，几乎所有的网口我都尝试了一遍，但是都没有找到我想看到的包。机缘巧合之下，我在 GNS3 界面中右键点击了一条相连的线，看到了 start capture 的选项，这回才成功抓包了。

对于一开始设计的简单网络，我看了从 PC1 到 PC2 的数据包和 PC2 到 PC2 的数据包，发现 PC2 去 ping PC1 的时候根本没发数据包出来。我检查了好几次 PC2 的配置，觉得都没问题，有理由怀疑 PC2 坏了。我又拉出了一个 PC3，放到该网络里，这回 PC1 ping PC3 就成功了，PC3 ping PC1 也成功了，但是 PC2 还是 ping 不通，那就说明是 PC2 坏了，应该是这个软件的小 bug。

（4）做实验的时候我对于 RIP 协议和 OSPF 协议还是懵逼的状态，我忘记了什么是距离矢量算法，什么是链路状态信息，我也不记得 RIP 协议里路由器都发送什么消息，OSPF 里路由器都发送什么消息，以及两个协议里路由表是怎么维护的。我看了老师发的 ppt，找了网上的公开课，也看了好多博客，最后搭配着自己实验，才逐渐理清它们的工作过程。

5.2 心得体会

在遇到 Dynamips 时，遇到连接不通的情况，通过修改配置文件将连接反转是一种巧妙的解决方法。这种情况下，灵活运用实践经验和尝试不同的方法，最终找到解决方案是非常重要的。

在安装 GNS3 时，遇到虚拟机安装失败和本地连接问题，最终通过耐心多次尝试、重新安装以及不放弃的精神解决了问题。这强调了在面对技术挑战时需要具备坚韧不拔的毅力和解决问题的能力。

在解决实际网络问题中，我知道了常常可以采用以下的方法来找到问题：

(1)通过 ping 自己可以确认设备的本地网络配置是否正确,包括 IP 地址、子网掩码、默认网关等。如果设备无法 ping 通自己，说明可能存在本地配置错误或者接口状态问题。

（2）利用命令行工具查看各个设备各个端口的状态以及接口信息，比如使用命令行工具 show 命令，可以获取实时的设备状态信息，帮助定位问题。

（3）如果前两种还没定位到问题时，可以采用抓取每一个接口通过的数据包，通过抓包分析，可以确定数据包在网络中的流动情况，以及是否在某一特定接口上被丢弃。这有助于追踪问题发生的位置，确定是否是某个特定的链路或设备导致的问题。

同时,通过这次实验,我也掌握了 RIP 和 OSPF 以及 ARP 协议的工作流程。这次实验花了我好多时间，配置环境花了一天半，剩下一天半大多数时间都在熟悉命令，检查问题、排除问题。现在，我已经熟悉了整个配置的流程以及网络的工作流程，我可以很快的为设备配置好这两个协议，并分析每个包的传输过程。

还有部分心得我写在了 3.4 RIP 和 OSPF 协议的区别

六、实验思考

6.1 不同方式配置路由丢包解析

实验中，采用下一跳和转发接口这两种方式配置 PC11 和 PC10 的静态路由有什么区别？会导致在你的拓扑结构中从 PC11 ping PC10 时的丢包数有什么变化？需要用你的拓扑中 PC11 和 PC10 的 arp 表和路由表中的内容来解释，要附带截图。

采用下一跳的方式配置静态路由，步骤如下图 6-1 ~ 图 6-4：

```
PC11> show ip

NAME          : PC11[1]
IP/MASK       : 203.138.1.2/24
GATEWAY       : 203.138.1.3
DNS           :
MAC           : 00:50:79:66:68:08
LPORT         : 13128
RHOST:PORT    : 127.0.0.1:13129
MTU:          : 1500
```

```
PC10> show ip

NAME          : PC10[1]
IP/MASK       : 203.138.10.1/24
GATEWAY       : 203.138.10.3
DNS           :
MAC           : 00:50:79:66:68:07
LPORT         : 13126
RHOST:PORT    : 127.0.0.1:13127
MTU:          : 1500
```

图 6-1 PC11 和 PC10 下一跳配置图

```
PC10> arp

arp table is empty
```

```
PC11> arp

arp table is empty
```

图 6-2 PC11 和 PC10 的初始 arp 表

```
PC11> ping 203.138.10.1
203.138.10.1 icmp_seq=1 timeout
203.138.10.1 icmp_seq=2 timeout
84 bytes from 203.138.10.1 icmp_seq=3 ttl=57 time=216.901 ms
84 bytes from 203.138.10.1 icmp_seq=4 ttl=57 time=215.282 ms
84 bytes from 203.138.10.1 icmp_seq=5 ttl=57 time=215.659 ms

PC11> arp

ca:01:21:58:00:08  203.138.1.3 expires in 110 seconds
```

图 6-3 PC11 第一次 ping PC10

```
PC10> arp

ca:0a:4a:c8:00:08  203.138.10.3 expires in 92 seconds
```

图 6-4 PC10 被 ping 通后的 arp 表

根据上图所示，当用下一跳的方式配置 PC 的静态路由时，两个 PC 的初始 arp 表都是空的。当 PC11 ping PC10 时，PC11 先发给网关，但是 PC11 并不知道网关的 MAC 地址，这时候会丢一包。当网关返回它的 MAC 地址后，PC11 会把 ping 的包发给网关。之后由于路由器之间是点对点网络，不是广播型网络，路由器传输不会丢包。但是当 R10 给 PC10 转发这个 ping 的包时，也会丢包，

因为 R10 并不知道 PC10 的 MAC 地址，这是第二次丢包。PC10 给 PC11 返回回应包的时候就不会出现丢包的情况了，因为这时候 MAC 地址都知道了，PC 和路由器会根据自己的 arp 表进行相应的转发操作。

接下来分析用接口转发配置会出现什么情况。由于原先 GNS3 上我使用的是软件带的 PC 机，而不是用路由器模拟，所以我重新设置了一个拓扑图，在 Dynamips 仿真运行，里面的 PC 机用路由器来模拟，拓扑图如下：

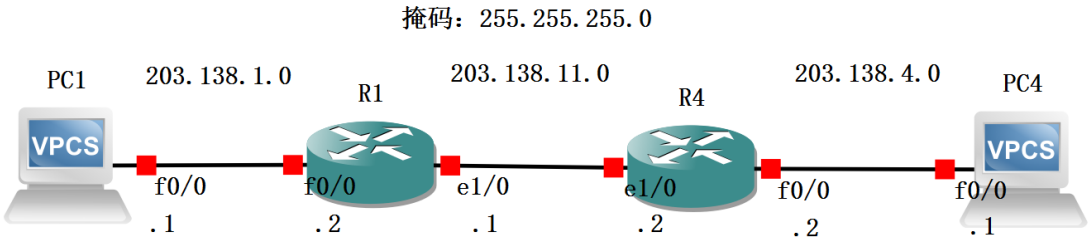


图 6-5 接口转发拓扑图

采用转发接口的方式配置静态路由，步骤及结果如下，PC1 ping PC4 丢了三次包。

```
ip classless
ip route 0.0.0.0 0.0.0.0 FastEthernet0/0
no ip http server
```

图 6-6 配置 PC1 和 PC4 的接口转发

```
PC1>en
PC1#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  203.138.1.1            -   c804.343c.0000  ARPA   FastEthernet0/0
PC1#
```

图 6-7 查看 PC1 的 arp 表

```
PC4#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  203.138.4.1            -   c807.343c.0000  ARPA   FastEthernet0/0
PC4#
```

图 6-8 查看 PC4 的 arp 表

```
PC1>ping 203.138.4.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 203.138.4.1, timeout is 2 seconds:
...!!
Success rate is 40 percent (2/5), round-trip min/avg/max = 76/90/104 ms
PC1>
```

图 6-9 PC1 ping PC4 结果

```
Success rate is 40 percent (2/5), round-trip min/avg/max = 72/82/92 ms
PC1#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  203.138.1.1             -   c804.343c.0000  ARPA   FastEthernet0/0
Internet  203.138.4.1             0   ca00.343c.0000  ARPA   FastEthernet0/0
```

图 6-10 查看 ping 通后 PC1 的 arp 表

```
PC4#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  203.138.1.1             0   ca03.343c.0000  ARPA   FastEthernet0/0
Internet  203.138.4.1             -   c807.343c.0000  ARPA   FastEthernet0/0
Internet  203.138.4.2             0   ca03.343c.0000  ARPA   FastEthernet0/0
```

图 6-11 查看 ping 通后的 PC4 的 arp 表

```
PC4>ping 203.138.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 203.138.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 104/112/120 ms
PC4>
```

图 6-12 第一次 ping 通后用 PC4 ping 通 PC1

```
PC1#ping 203.138.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 203.138.1.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 28/31/32 ms
PC1#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  203.138.1.1             -   c804.343c.0000  ARPA   FastEthernet0/0
Internet  203.138.1.2             0   ca00.343c.0000  ARPA   FastEthernet0/0
Internet  203.138.4.1             2   ca00.343c.0000  ARPA   FastEthernet0/0
```

图 6-13 用 PC1 ping R1，arp 表变化

在 ping 操作前，我查看了 PC1 和 PC4 的 arp 表，里面只有自己的表项，没有其他的表项。用 PC1 ping PC4，丢包数为 3，以下是丢包分析：

PC1 ping PC4，由于 PC1 没有网关 IP 地址， arp 包解析出来只有 PC4 的 IP 地址，却没有网关的 IP 地址。所以第一次 PC1 在网络内广播，R1 收到了 PC1 的 arp 请求，就返回一个回应，告诉 PC1：要去 PC4 就要先发到我的 f0/0 接口，在这里第一次丢包。

因此，PC1 的 arp 表中就多了一组对应关系，是 PC4 的 IP 地址对应 R1 的 f0/0 接口的 MAC 地址。同时，R1 的 arp 表中也多了 PC1 和它的 MAC 地址的对应关系。然后，PC1 在发包给 R1。路由器中由于是点对点网络，不是广播型网络，所以 R1 给 R4 直接转发，不会丢包。R4 给 PC4 时由于 R4 不知道 PC4MAC 地址所以这时候还会发一个广播包，询问 PC4 的 MAC 地址，得到 PC4 的 MAC 地址后才能有效转发，这里又丢一次包。

在 PC4 收到 PC1 的 ping 包后，PC4 需要返回回应，但是 PC4 的 arp 表中并没有 PC1 的 IP 和 MAC 地址对应关系，所以它会在网络内先发一个广播包，询问 PC1 的 MAC 地址。R4 收到 PC4 的 arp 请求后，告诉 PC4，发到 PC1 的 IP 地址需要走我的 f0/0 接口。所以 PC4 的 arp 表中多了一组对应关系，PC1 的 IP 对应 R4 的 f0/0 接口 MAC 地址。这是第三次丢包。然后 PC4 才能返回一个回应包，之后就不会再丢包了。因此是丢包三次。

在 PC1 ping 通 PC4 之后，两个主机都有对方的 IP 地址和 MAC 地址的对应关系，虽然是 PC 的 IP 和 Router 的接口 MAC，但是双方都知道到达对方那里需要怎么走了。所以再尝试 PC4 ping PC1 时，不会丢包。

最后我尝试了 PC1 ping R1 的 f0/0 接口。由于 PC1 的 arp 表项中并没有 R1 f0/0 的 ip 地址和 MAC 地址对应关系，所以它第一次会在网络内发一个 arp

广播包，询问 R1 f0/0 的 IP 地址对应的 MAC 地址，所以会丢一次包。而在 R1 返回给 PC1 时，由于在 R1 中已经建立了关于 PC1 的 arp 表对应关系，所以返回时并不会丢包。

6.2 RIP 和 OSPF 的协议工作过程

对照所截获的消息，说明 RIP 协议有无水平分割的工作流程；以及 OSPF 协议在点对点网络和广播型网络上工作流程。附截图和对消息的说明。

说明 RIP 协议有无水平分割的工作流程：请看 3.2.3 RIP 协议水平分割。

说明 OSPF 在点对点网络和广播型网络上工作流程。详细信息请看 3.3.2 OSPF 链路状态信息更新过程以及 3.4.2 OSPF 协议的工作过程。下面我也给出了简要的说明：

1、OSPF 在点对点网络上的工作流程：

（1）邻居发现：

在点对点网络中，OSPF 路由器直接连接到另一个路由器，因此邻居关系相对简单。当两个路由器相互连接时，它们通过发送 Hello 消息来发现彼此，建立邻居关系。

（2）邻居关系建立：

通过交换 Hello 消息，路由器确认彼此的 OSPF 参数，包括区域号、Hello 间隔、路由器 ID 等。如果相邻路由器的 OSPF 参数匹配，它们就建立邻居关系。

（3）链路状态数据库同步：

一旦建立邻居关系，路由器会交换链路状态数据库信息。路由器将自己的链路状态数据库信息发送给邻居，使两者保持同步。

（4）最短路径计算：

每个路由器使用链路状态数据库中的信息计算最短路径树。SPF 算法应用于链路状态数据库，确定最短路径并生成路由表。

（5）路由表更新：

路由器根据计算得到的最短路径更新其路由表。每个路由器的路由表都反映了网络拓扑的当前链路状态。

2、OSPF 在广播型网络上的工作流程：

（1）邻居发现：

在广播型网络中，路由器通过多播 Hello 消息来发现潜在的邻居。路由器会定期发送 Hello 消息，而其他路由器监听这些消息以识别潜在的邻居。

（2）邻居关系建立：

当一个路由器收到另一个路由器的 Hello 消息时，它会回应以建立邻居关系。邻居关系的建立还包括交换 OSPF 参数和验证邻居身份。

（3）链路状态数据库同步：

一旦建立邻居关系，路由器通过发送 LSA 来同步链路状态数据库。这种同步确保网络中的所有路由器都有相同的链路状态数据库。

（4）最短路径计算：

在广播型网络中，每个路由器都能够收到其他路由器的 LSA，从而构建完整的链路状态数据库。SPF 算法再次用于计算最短路径树。

（5）路由表更新：

路由器更新其路由表，以反映最新的最短路径信息。这确保了在广播型网络

中路由器都具有相同的最短路径信息。

总体而言，OSPF 在点对点网络和广播型网络上的基本工作流程相似，都涉及邻居发现、邻居关系建立、链路状态数据库同步、最短路径计算和路由表更新。不同之处在于邻居发现的方式和如何同步链路状态数据库。在点对点网络中，直接连接的两个路由器可以直接交换 Hello 消息；而在广播型网络中，路由器通过广播 Hello 消息来发现邻居。

6.3 数据包的发送过程

写出在你的拓扑中，数据包从某台 PC 机 A 发送给其他 PC 机 B 完整过程（这两台 PC 机间要跨越两台以上的路由器），包括 A 在网络内发送数据包的过程和跨越网络时路由器的处理过程，附相关截图。

在 6.1 不同方式配置路由丢包解析中，我们探讨了不同的路由配置方式导致的丢包数量，以及不同的路由方式数据包的收发过程和 arp 表的建立过程。但是我们只说明了 PC 到路由和路由到 PC 的过程。

而当数据包通过 PC 传到路由器之后，路由器会解析该数据包。我用图 1-1 的网络拓扑中 PC5 ping PC20，结果和在 PC5 的 f0/0 接口的抓包分析如下：

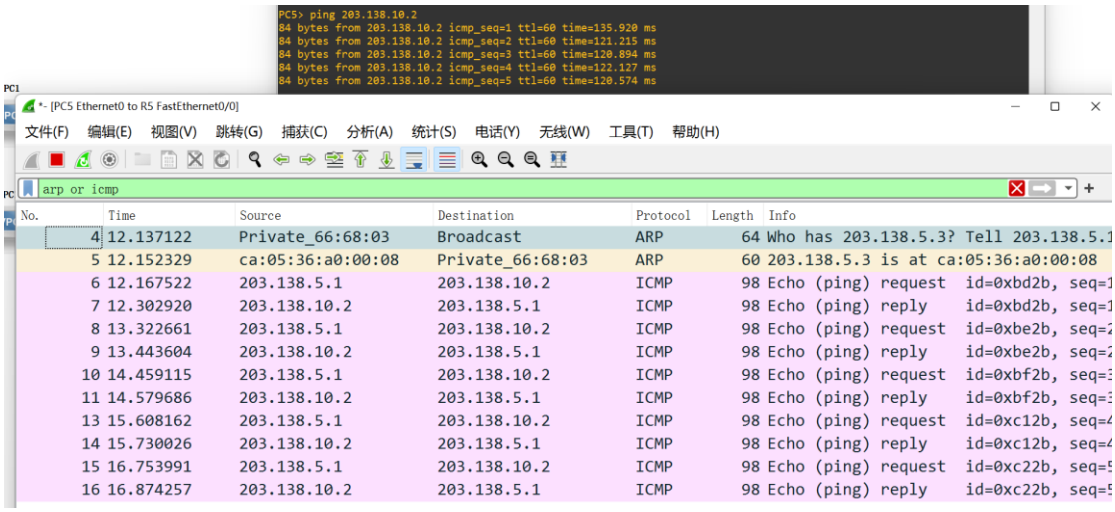


图 6-14 PC5 ping PC20 分析

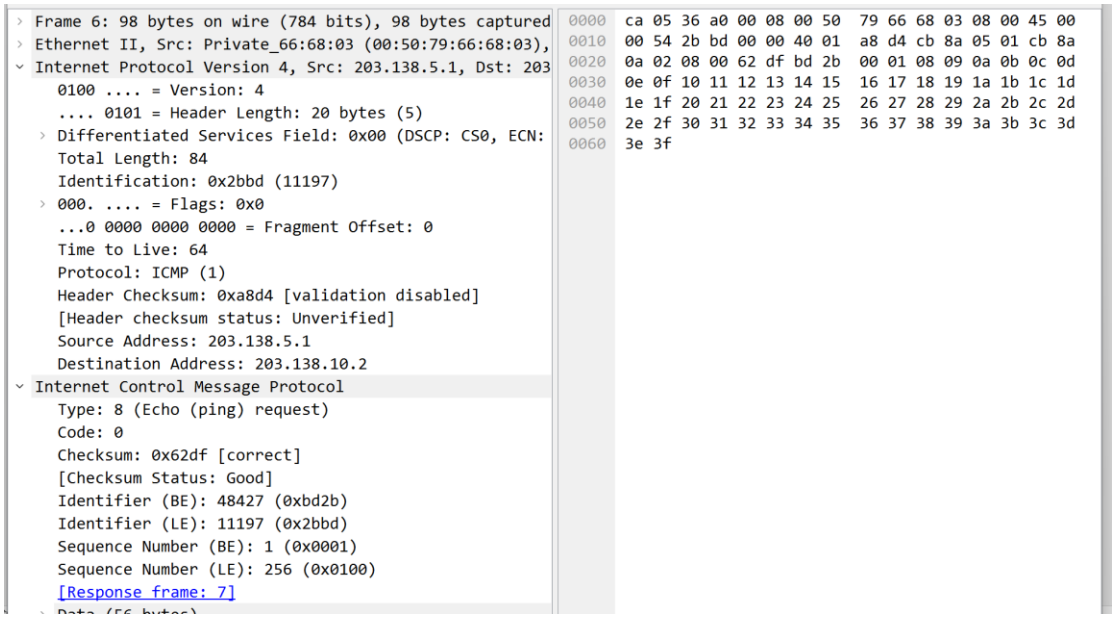


图 6-15 IP 包拆解图

（1）一个路由器在收到 ICMP 包时， 由于 ICMP 报文通常是与 IP 数据报一起传输的，路由器的网络层负责将其从链路层中提取出来。

（2）检查报文类型和代码，路由器检查 ICMP 报文的类型和代码字段，以确定报文的目的和所包含的信息类型。

（3）校验 ICMP 报文的校验和，路由器会对 ICMP 报文进行校验和的验证，确保报文的完整性。

（4）查找路由表：路由器根据 ICMP 报文中的目的 IP 地址查找路由表，确定下一跳的接口。如果目的地址是路由器本身，路由器将 ICMP 报文传递给本地进程进行处理。

（5）转发 ICMP 报文，路由器根据路由表找到的下一跳信息，将 ICMP 报文转发到相应的出接口。

（6）生成 ICMP 响应，如果 ICMP 报文需要路由器生成响应（例如

Destination Unreachable 或 Time Exceeded)，路由器会生成相应的 ICMP 响应报文。响应报文中包含了原始数据报文的一部分，以便通知源主机或路由器有关问题的详细信息。

（7）发送 ICMP 响应，生成的 ICMP 响应通过相应的出接口发送到下一跳或目的地。

（8）更新 TTL（Time to Live）或 Hop Limit，如果路由器是中间路由器，它会更新 TTL 或 Hop Limit 字段，并重新计算 ICMP 报文的校验和，然后将其传递到下一个路由器。

（9）当目的地址为本地时，路由器会将 ICMP 报文传递给本地协议栈，以便本地协议栈处理相应的 ICMP 消息。