

计算机网络实验报告

|  |  |
| --- | --- |
| **实验** | 实验4 路由协议实验 |
| **专业** | 计算机科学与技术 |
| **班级** | 1班 |
| **姓名** | 姚怀聿 |
| **学号** | 22920202204632 |

**2022年11月10日**

**目 录**

[计算机网络实验报告 1](#_Toc119614239)

[一、 实验目的 3](#_Toc119614240)

[二、 实验内容 3](#_Toc119614241)

[1. 任务1:配置指定网络拓扑，分析RIP路由协议收敛具体过程。 3](#_Toc119614242)

[2. 任务2:在任务1基础上断开某段链路，观察并分析各节点路由的变化。 18](#_Toc119614243)

[3. 任务3: 配置指定网络拓扑，分析OSPF路由协议收敛具体过程。 22](#_Toc119614244)

[三、 实验结果分析 34](#_Toc119614245)

[四、 实验小结与感想 34](#_Toc119614246)

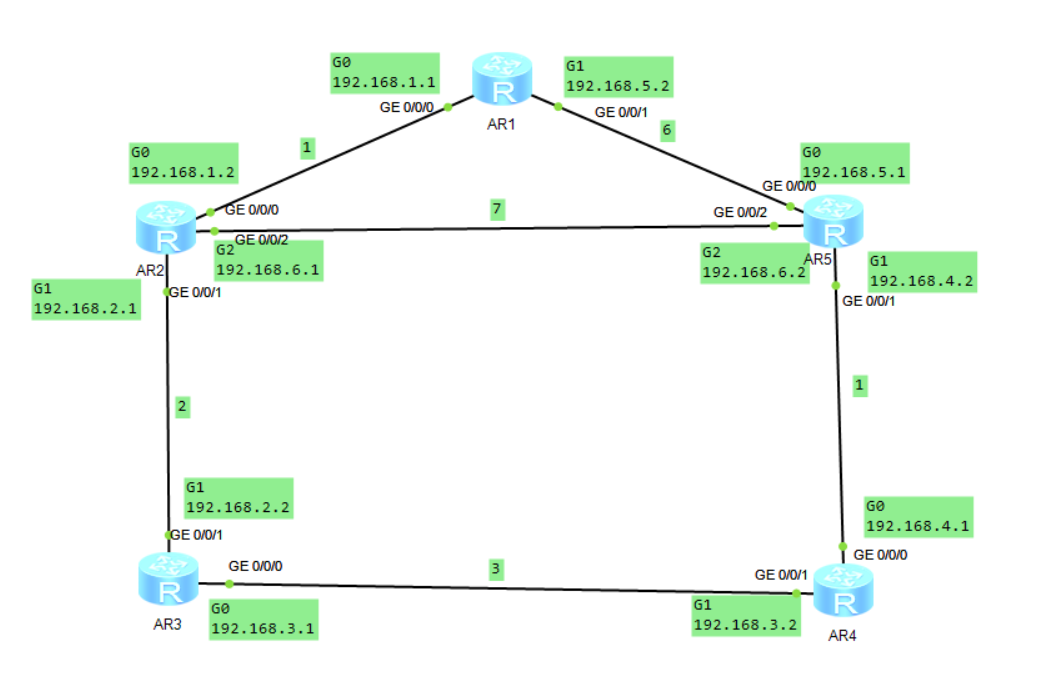
# 实验目的

1. 理解动态路由协议RIP的工作原理；掌握采用动态路由协议RIP进行网络设计的基本原则和方法
2. 理解动态路由协议OSPF的工作原理；掌握采用动态路由协议OSPF进行网络设计的基本原则和方法

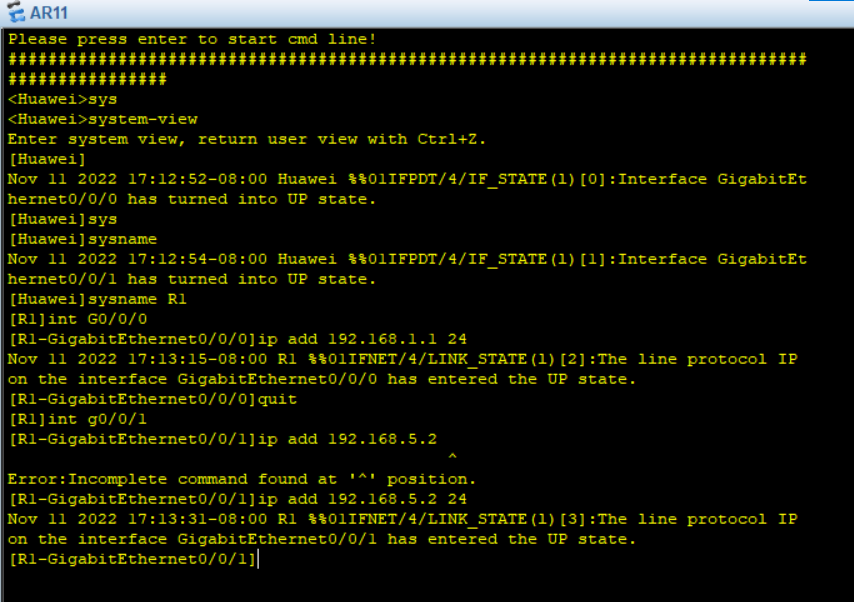
# 实验内容

## 任务1:配置指定网络拓扑，分析RIP路由协议收敛具体过程。

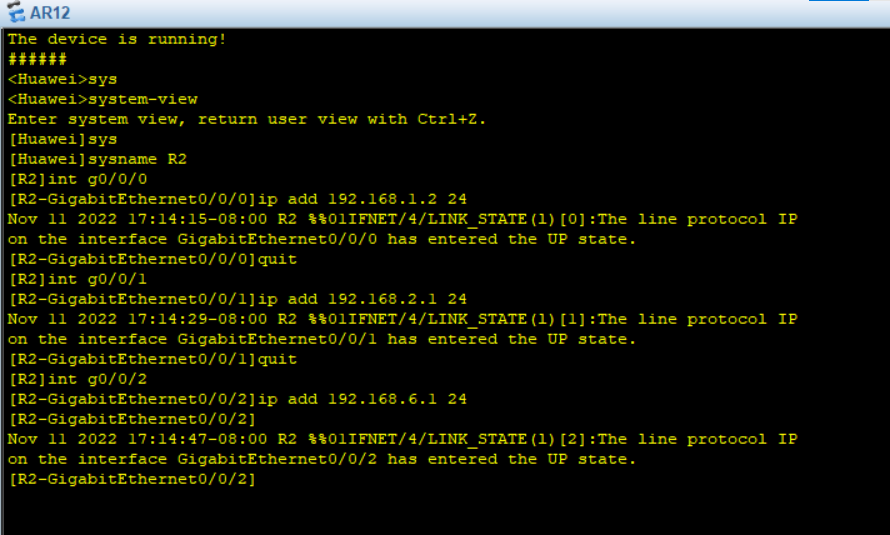
* 1. 在eNSP中搭建如下图拓扑，配置IP地址，配置RIPv2。完成后观察链路是否连通。



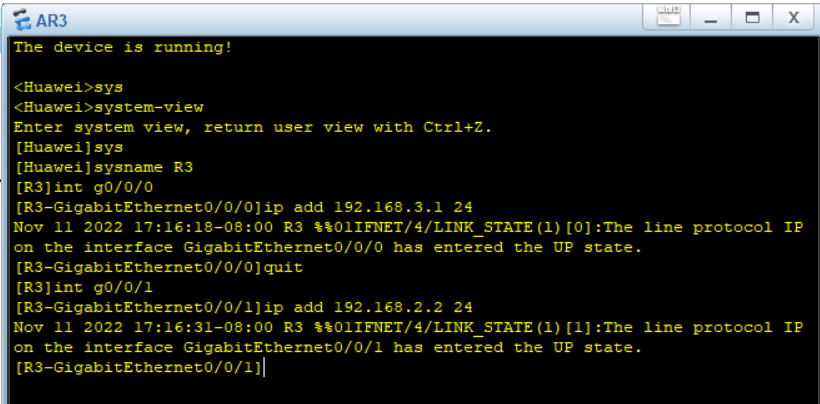
* + 1. 从R1到R5，为路由器的两个接口配置IP地址。

R1: 

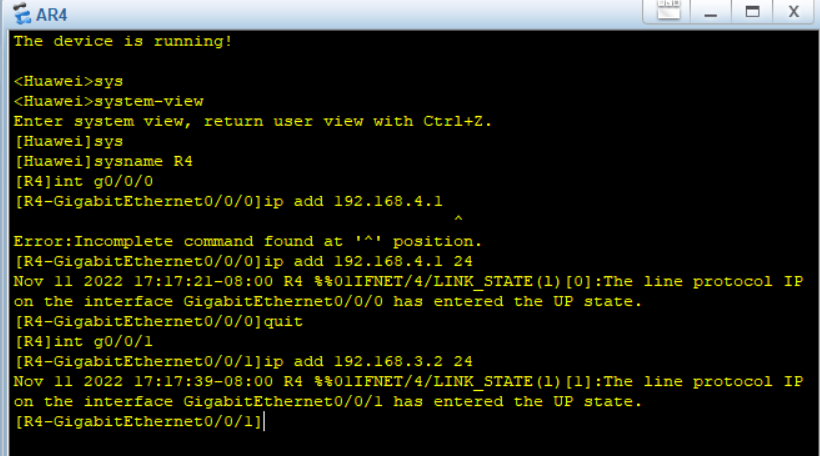
R2:



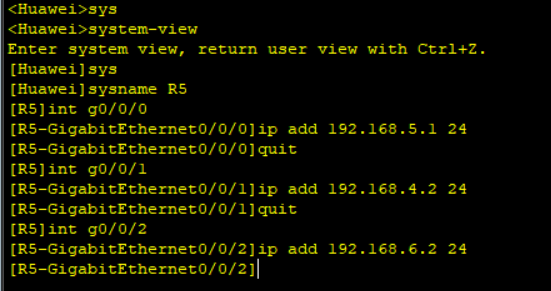
R3:



R4:

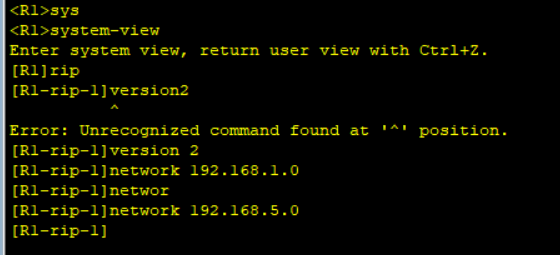


R5:

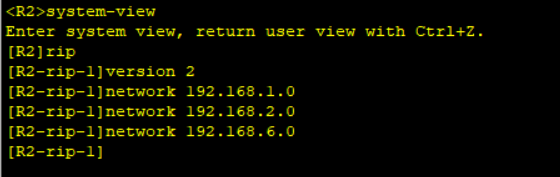


* + 1. 从R1到R5为五个路由器分别配置动态路由协议(RIPv2)。配置后使用disp ip routing-table检查路由表。

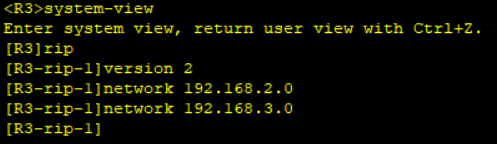
R1:



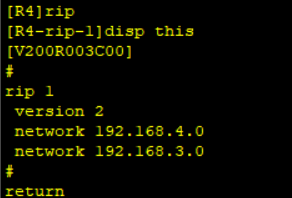
R2:



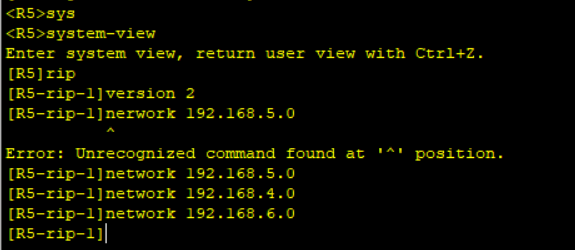
R3:



R4:

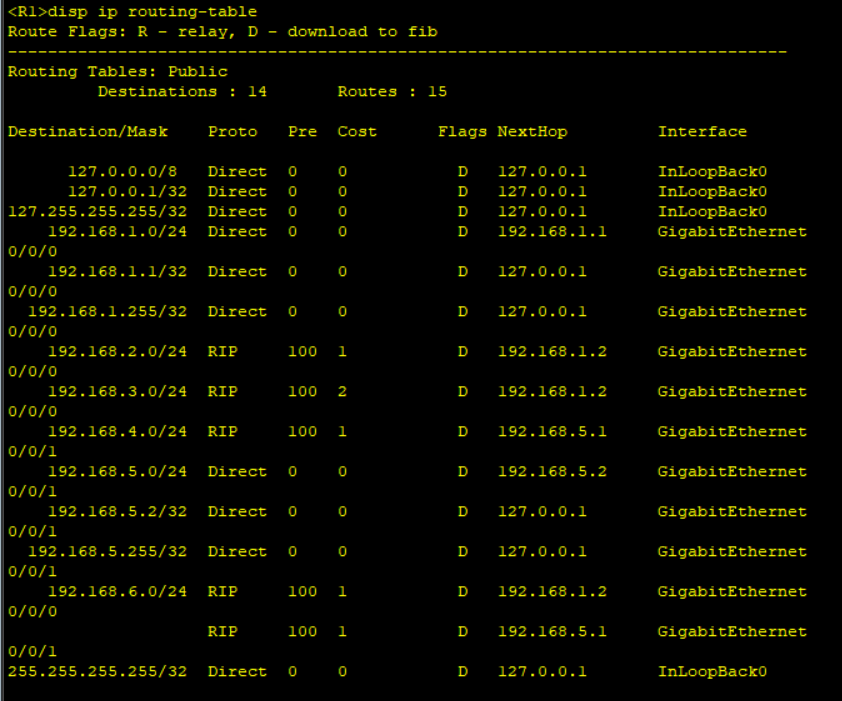


R5:

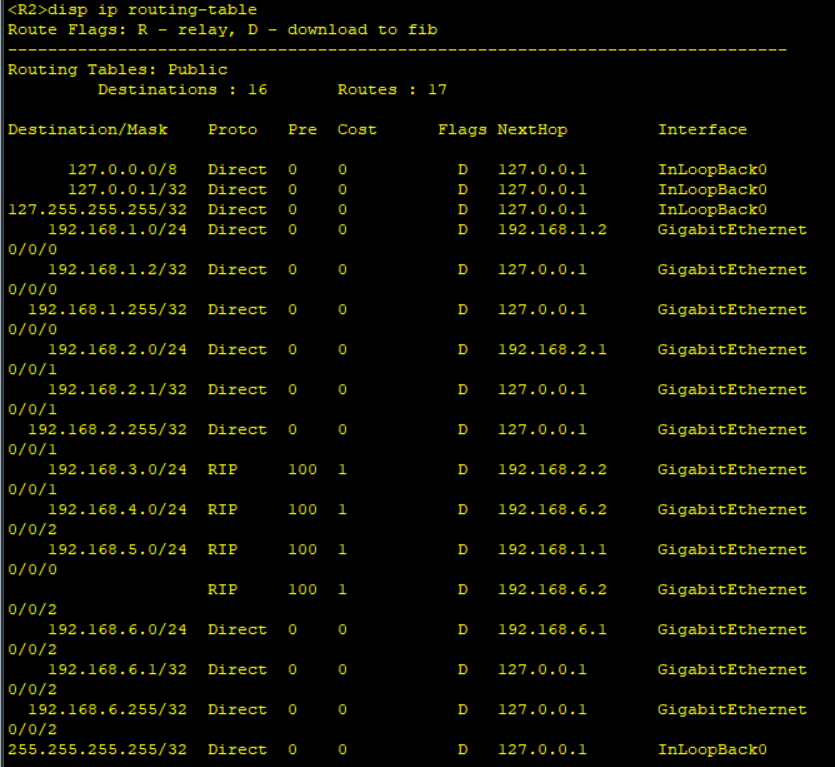


使用disp ip routing-table检查路由表

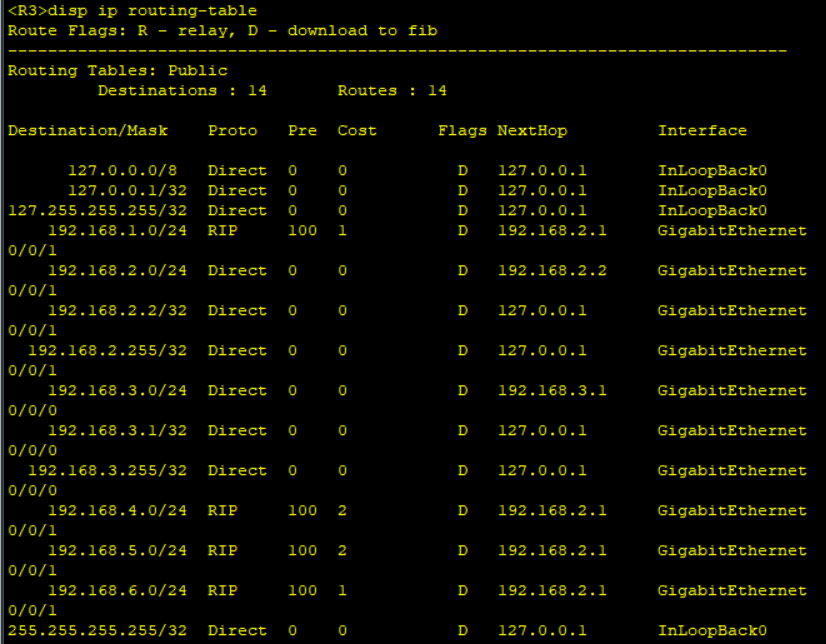
R1:



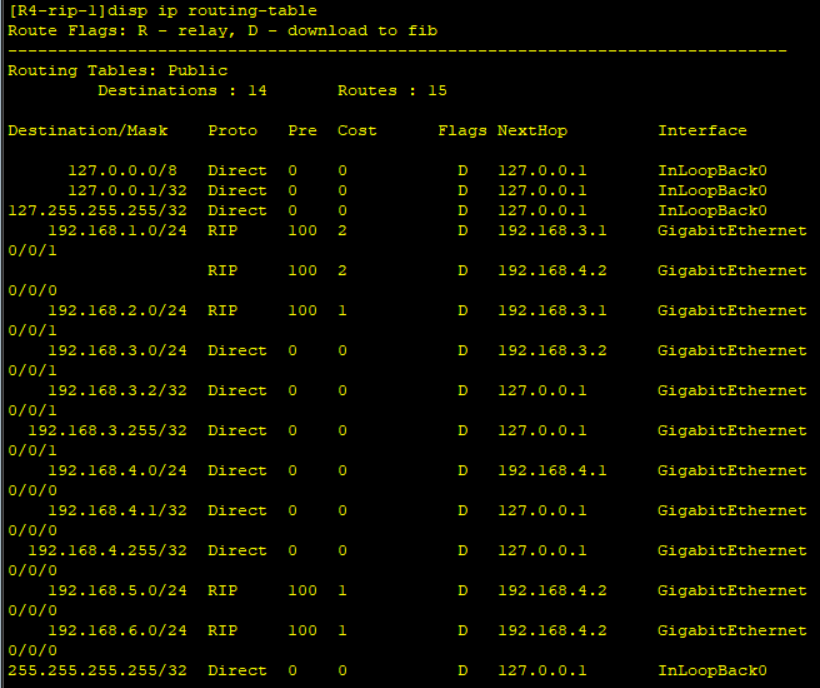
R2:



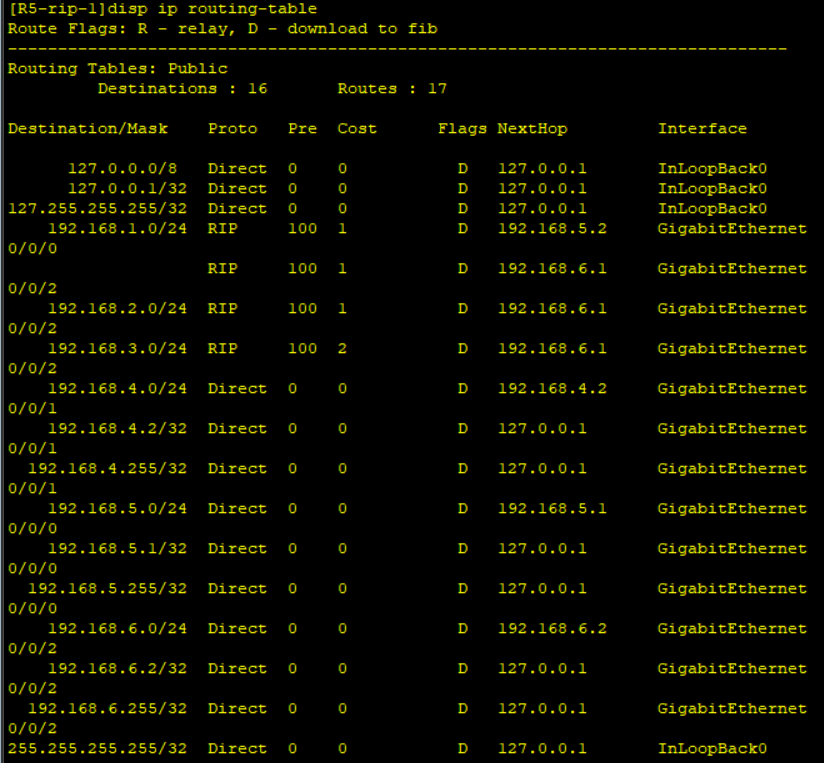
R3:



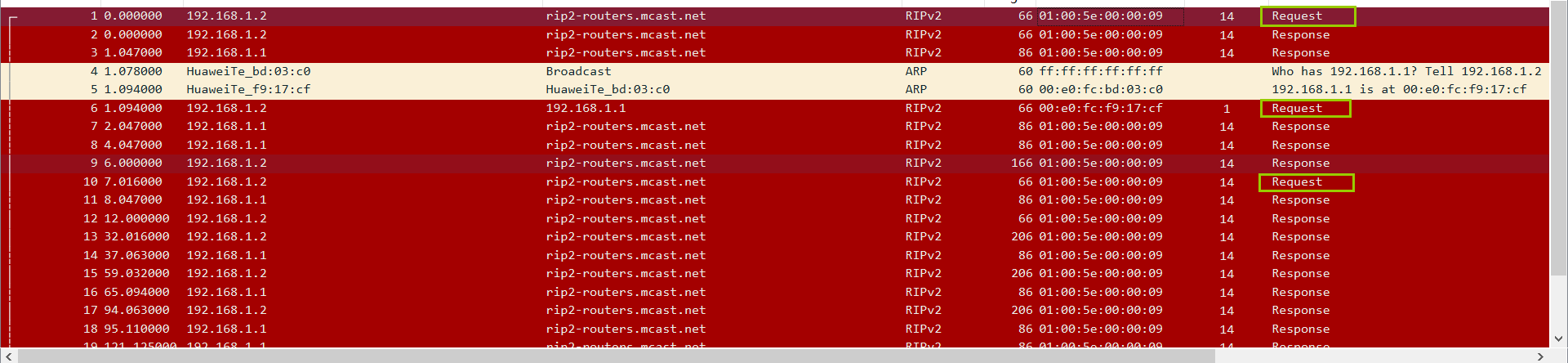
R4:



R5:

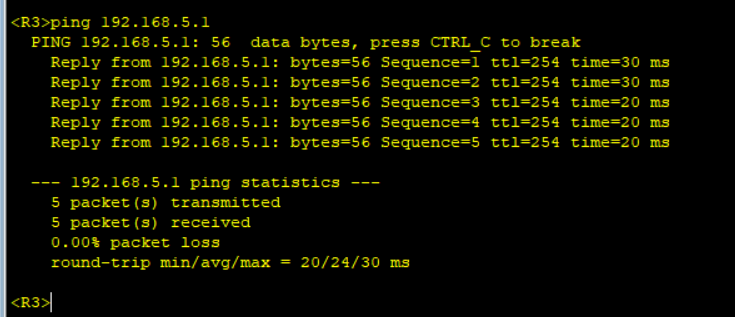


* 1. 通过Wireshark和各种路由器RIP命令观察路由动态更新，将关键信息分析并截图。
     1. 在打开路由器的同时对路由器端口抓包，可以观察到路由器Request报文:



* + 1. 检查连通性

在路由器R3上ping 192.168.5.1

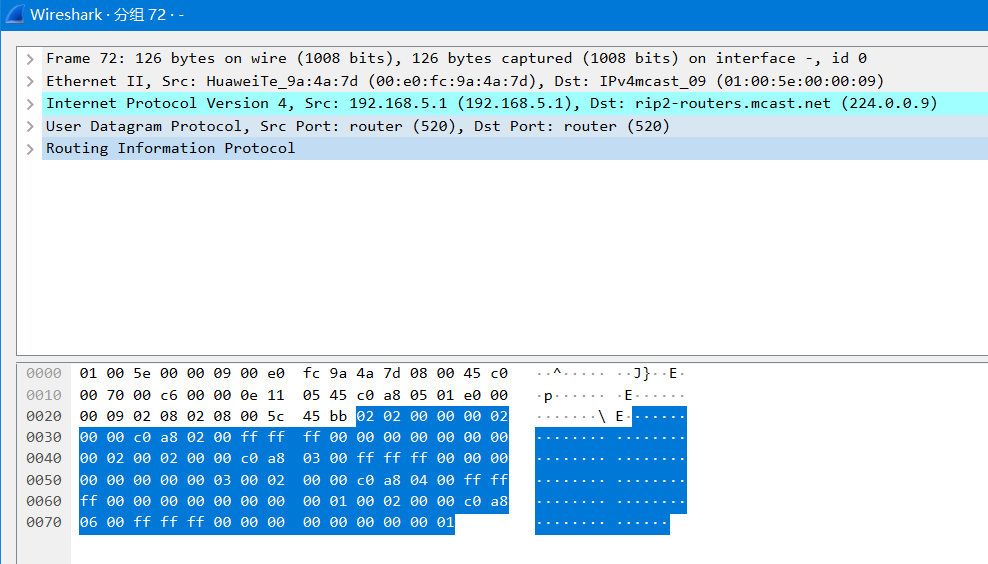


wireshark抓包



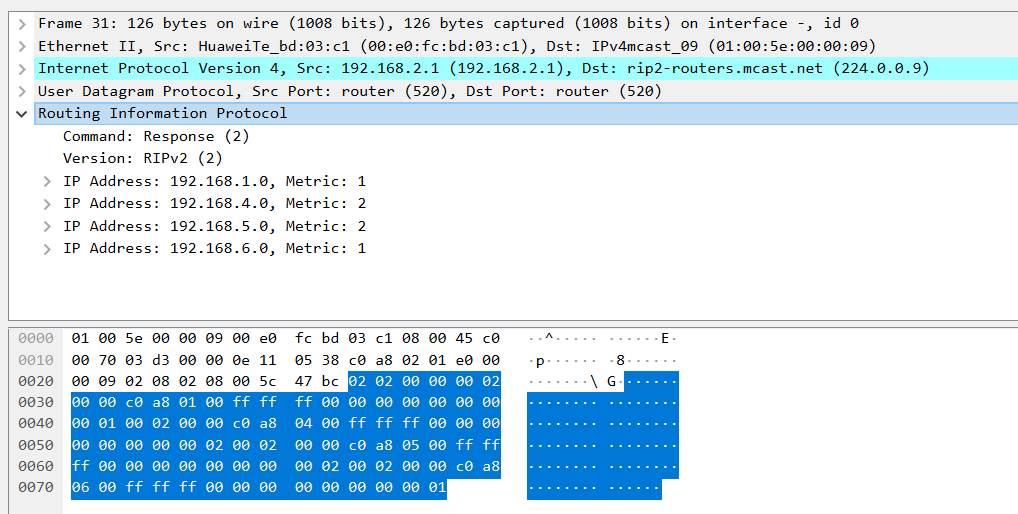
网络已连通

* + 1. 观察RIP交换报文的信息

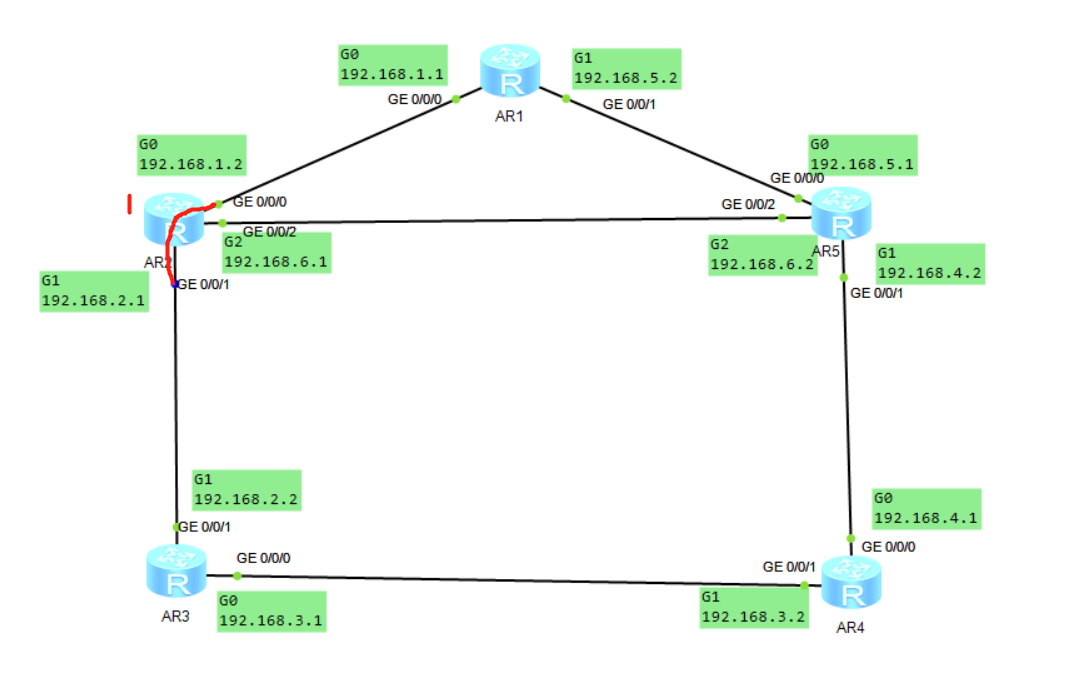


可以看到RIP报文封装在UDP中，每隔30s向周边的邻居结点交换自己的路由表信息

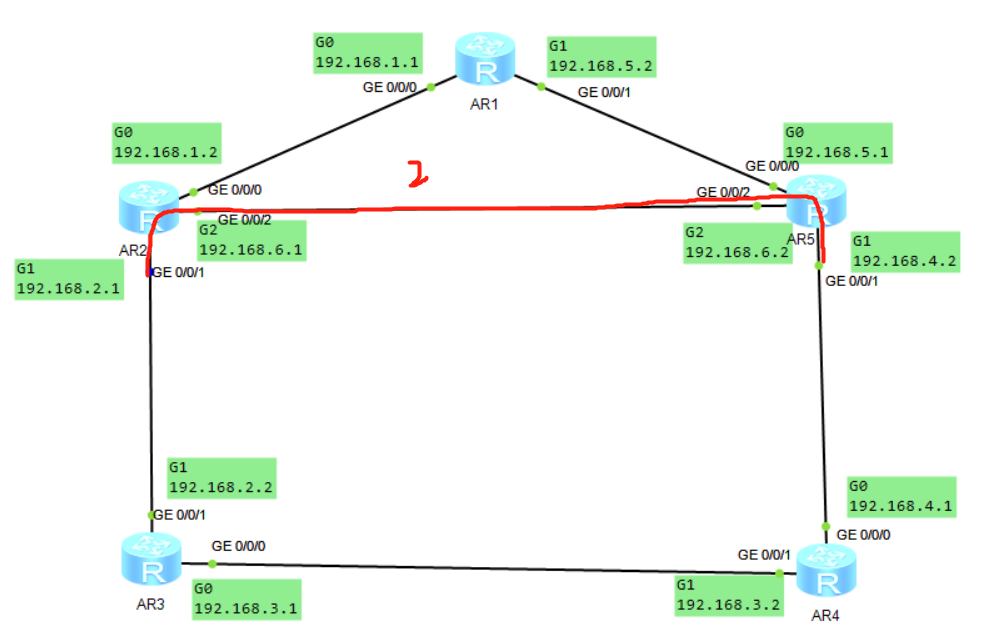
我对IP地址为192.168.2.1和IP地址为192.168.2.2的端口进行了数据抓包，观察到在网络收敛后，两个端口抓包得到的RIP报文如下:



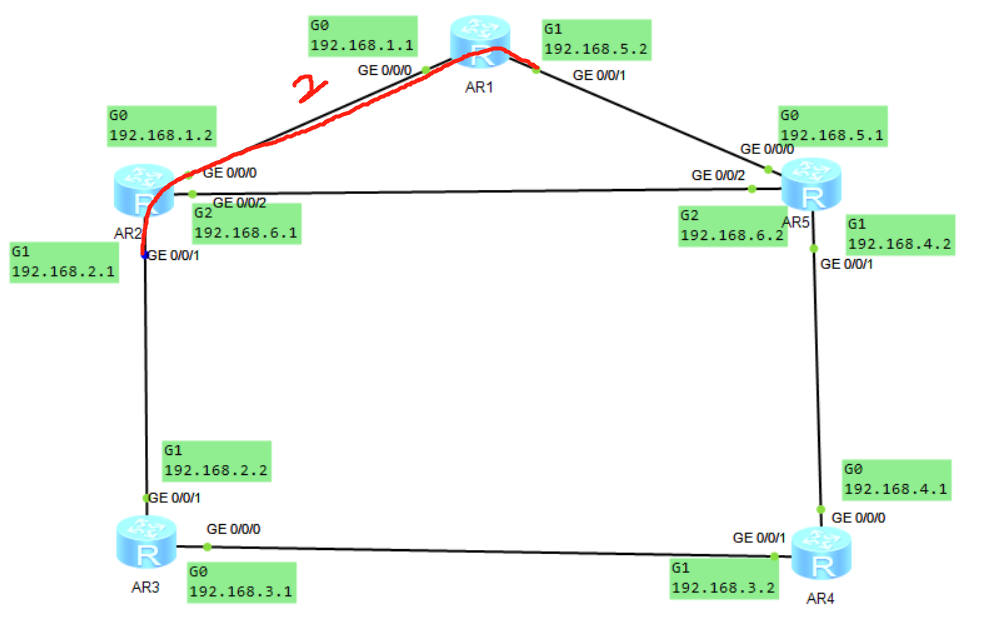
这是IP地址为192.168.2.1的端口抓到的RIP报文，可以看到，到达192.168.1.0网段需要metric为1:



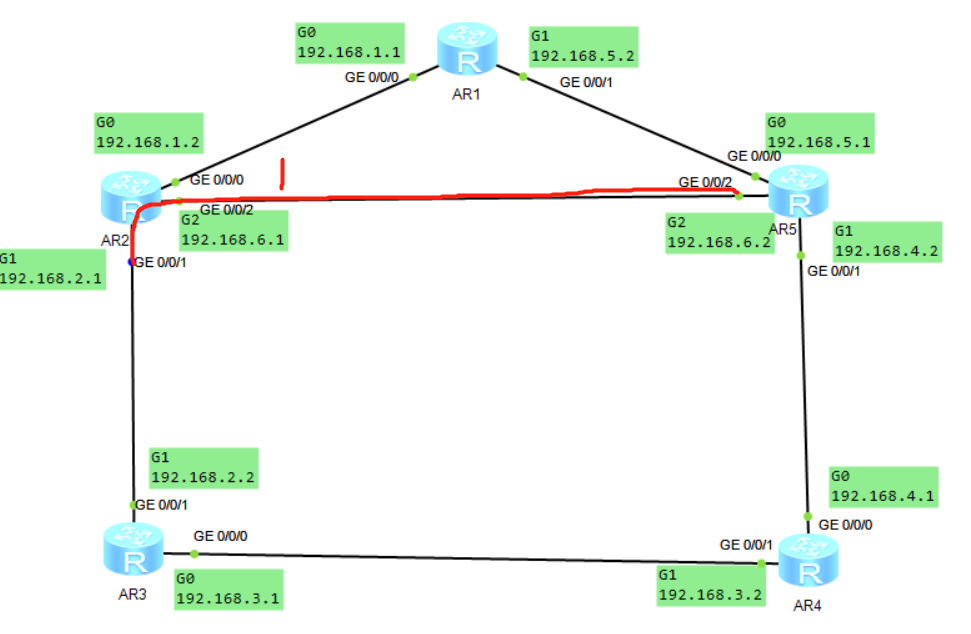
到达192.168.4.0网段需要metric为2:



到达192.168.5.0网段需要metric为2:



到达192.168.6.0网段需要metric为1:

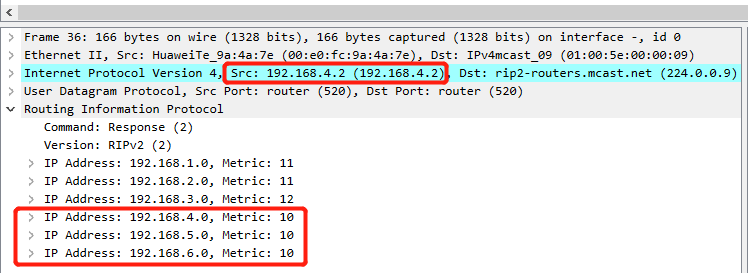


* + 1. 使用rip metricout 10

对R5的端口0/0/1使用rip metricout 10命令

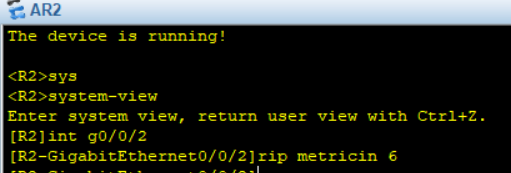


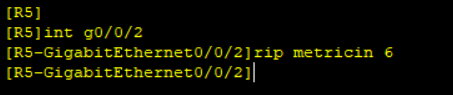
可以观察到到达网段192.168.4.0、192.168.5.0和192.168.6.0的metric均变为10:



* + 1. 改变RIP链路开销:

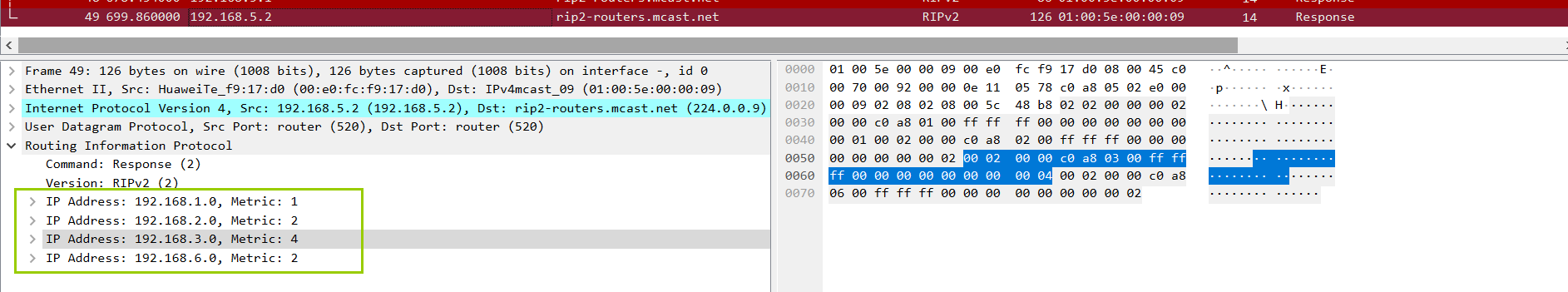
以开销为7的这条链路为例，设置路由器R2和R5

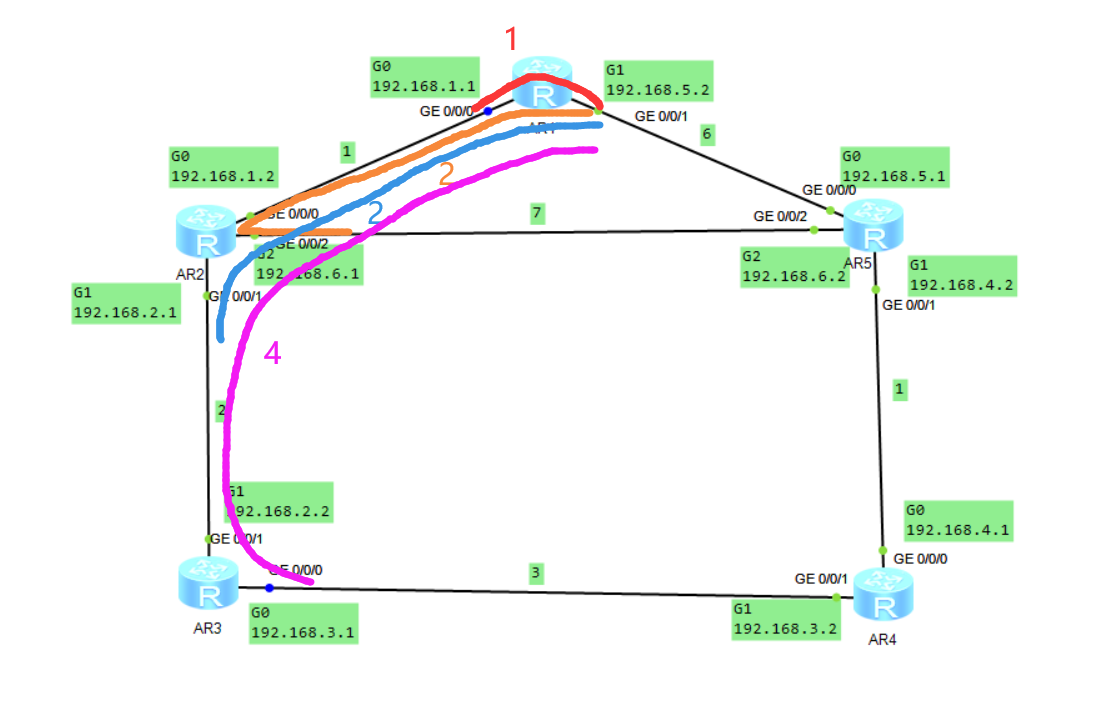




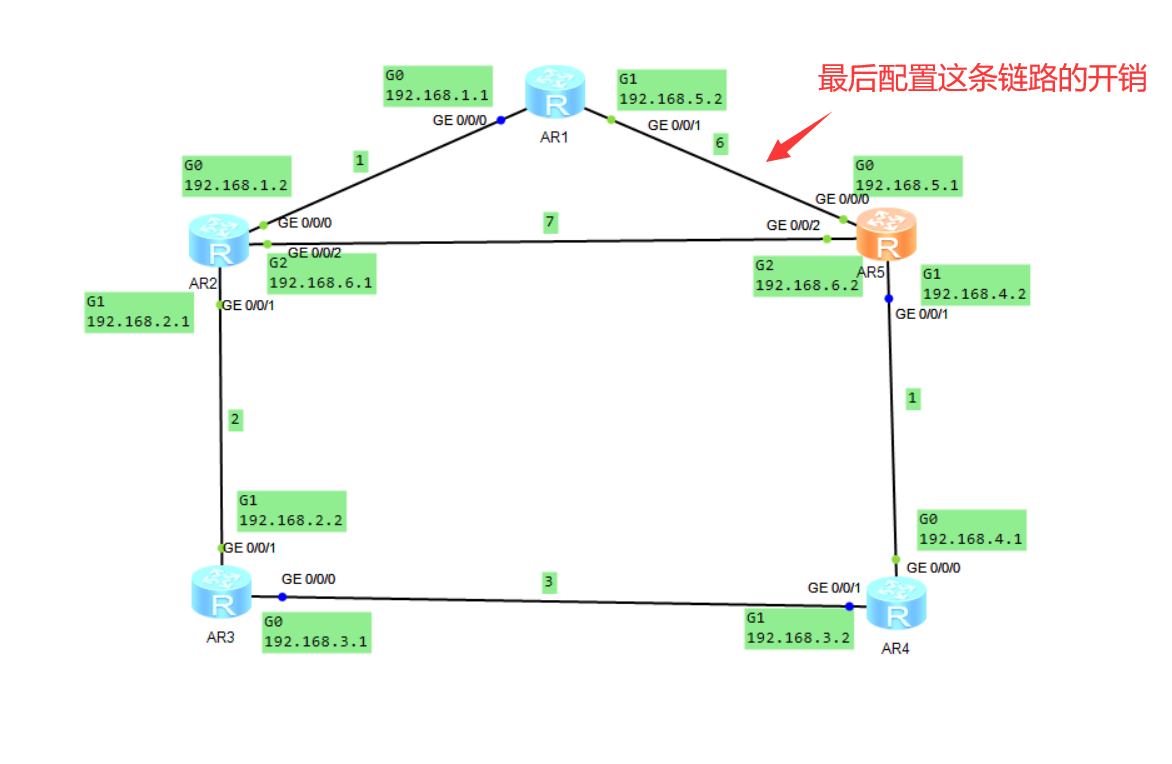
同理设置其他路由器；

全部设置完毕后，观察Wireshark抓包得到的RIP报文:

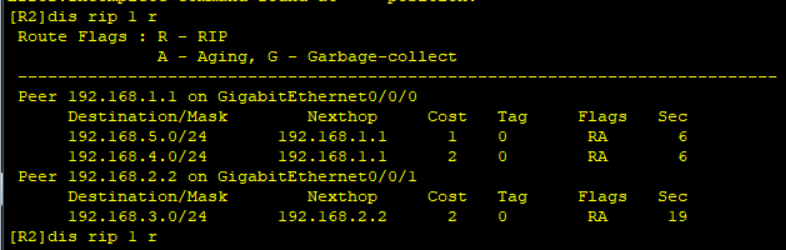


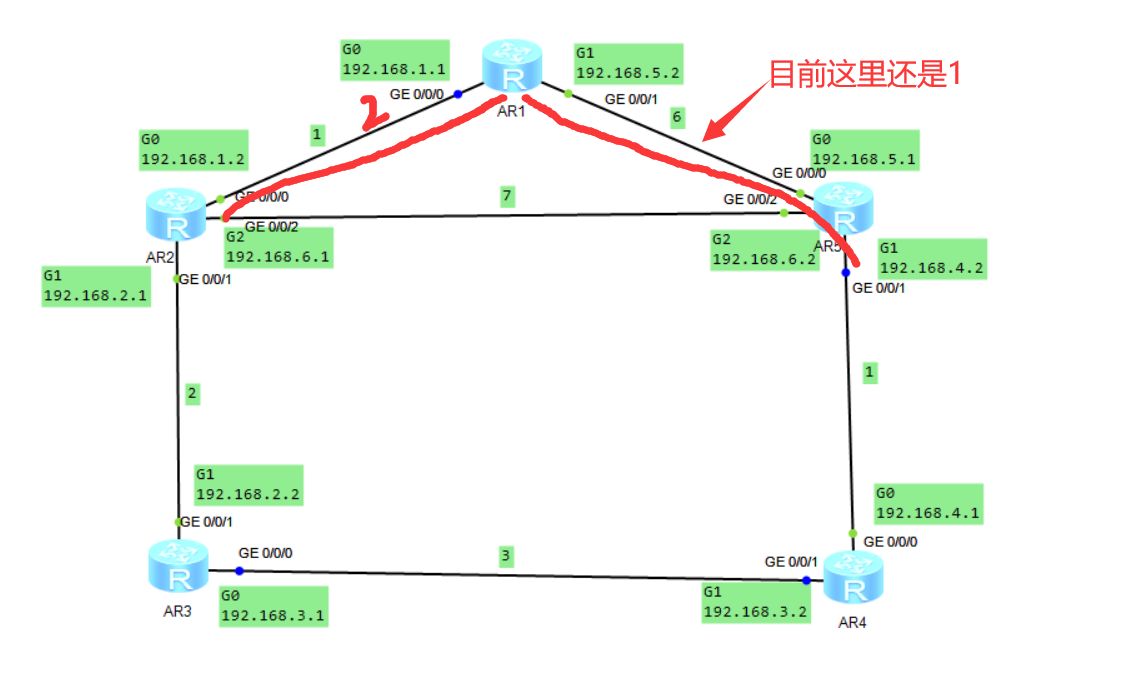


观察链路开销的改变造成的影响:

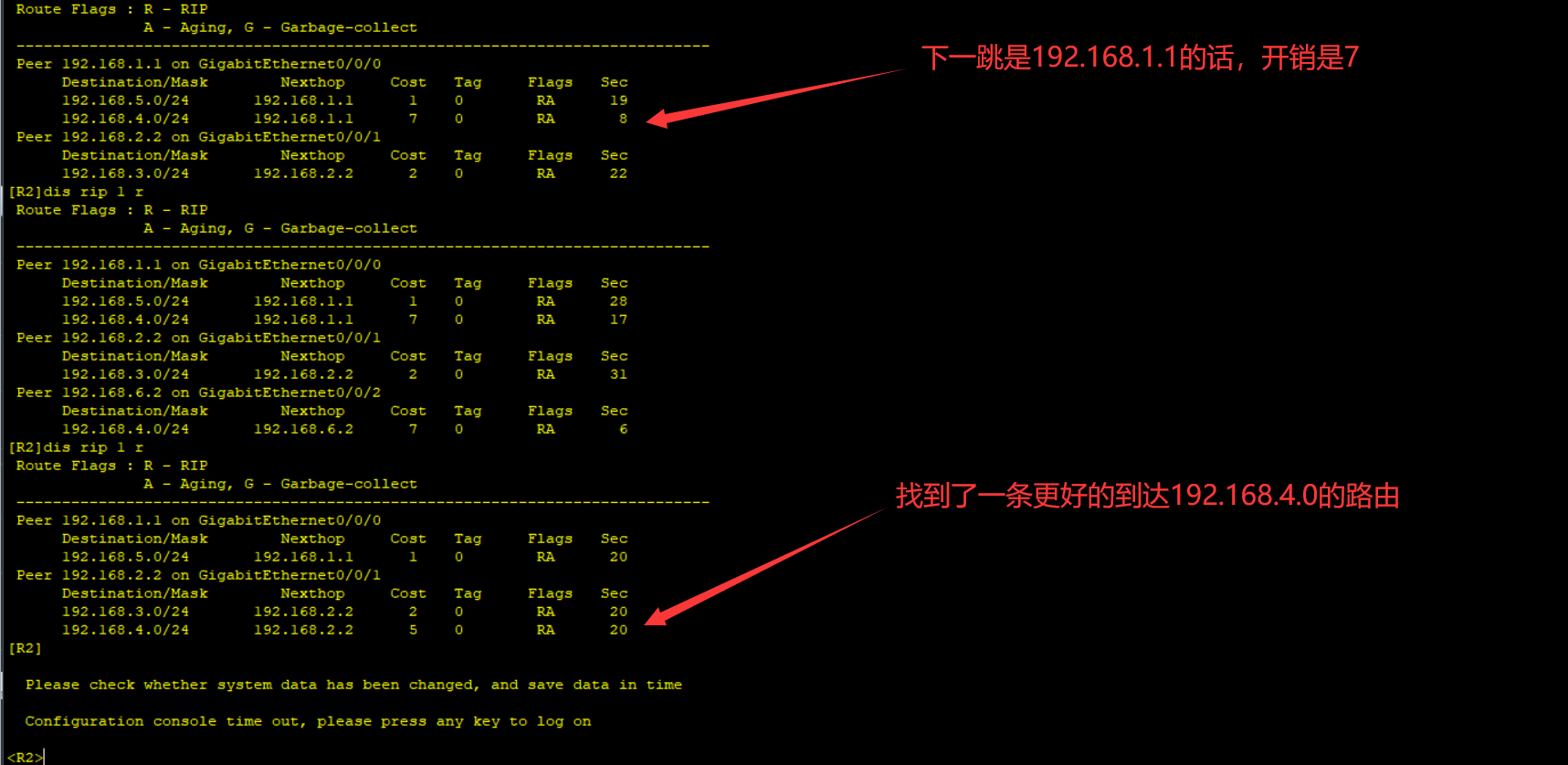


配置完最后一条链路的开销之前，观察路由器R2的路由表:





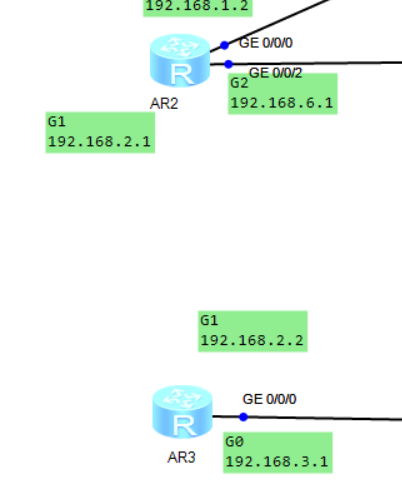
将最后一条链路的开销增大为6后:



## 任务2:在任务1基础上断开某段链路，观察并分析各节点路由的变化。

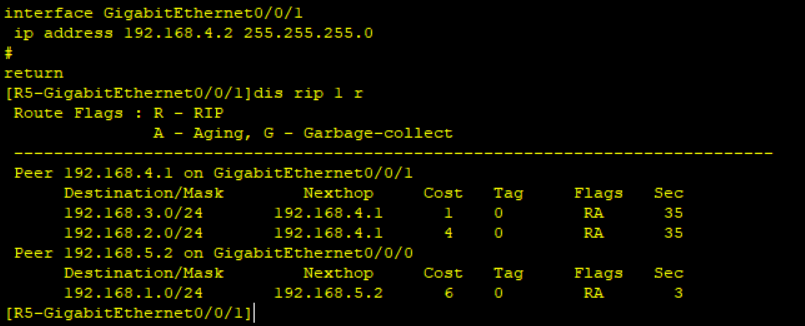
在任务1的网络收敛后，将AR2和AR3断开（坏消息！），分析该变化导致的重新收敛过程。

* 1. 将AR2和AR3断开



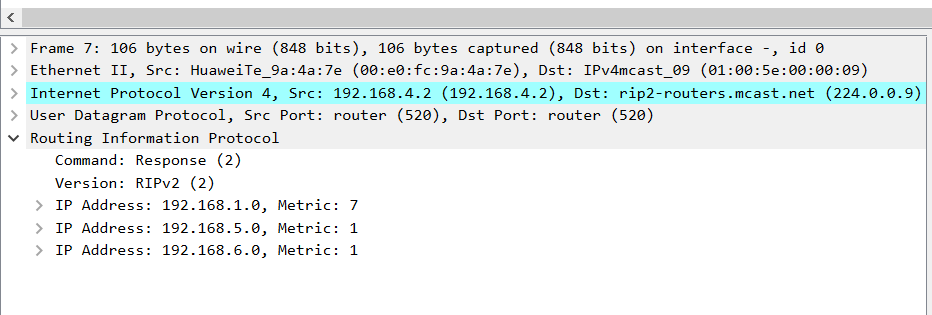
* 1. 分析该变化导致的重新收敛过程

断开之前，192.168.4.2的RIP路由表如下:



与拓扑图相符；

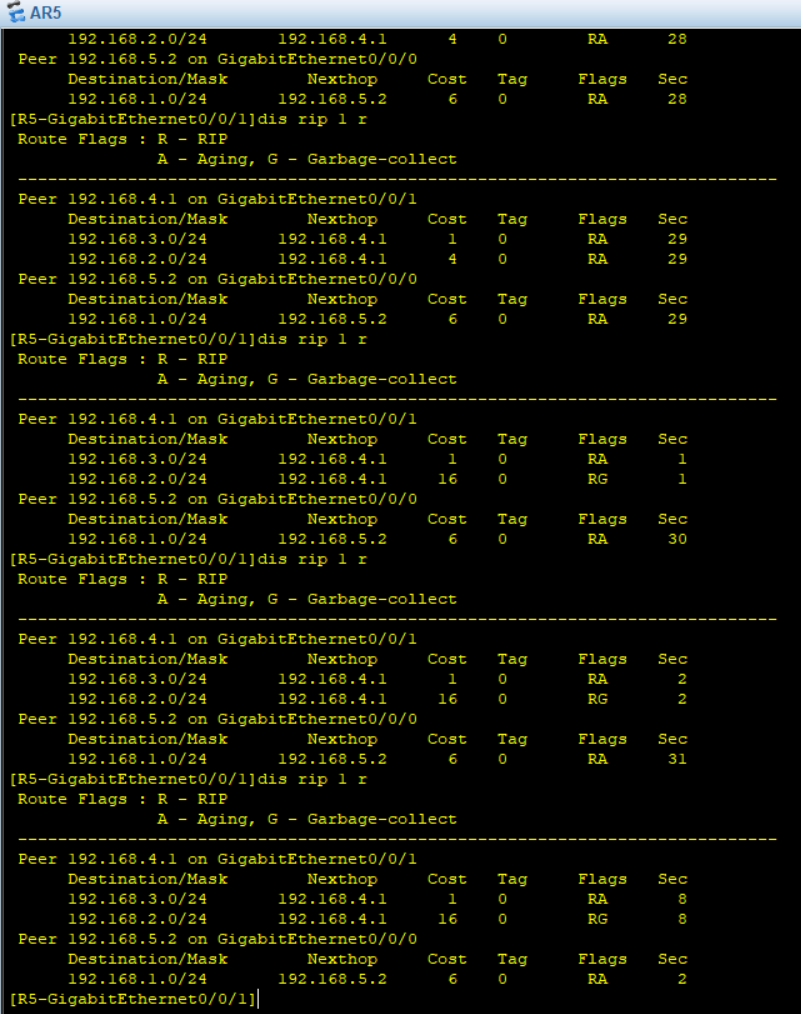
观察Wireshark抓到的数据包:



与网络拓扑相符;

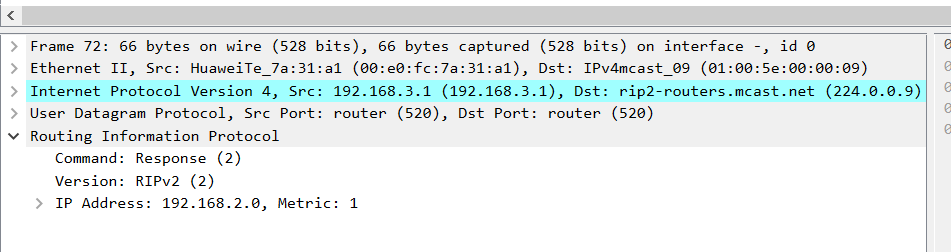
断开AR2和AR3的连接后,观察到如下结果:

观察路由器R5的路由表:



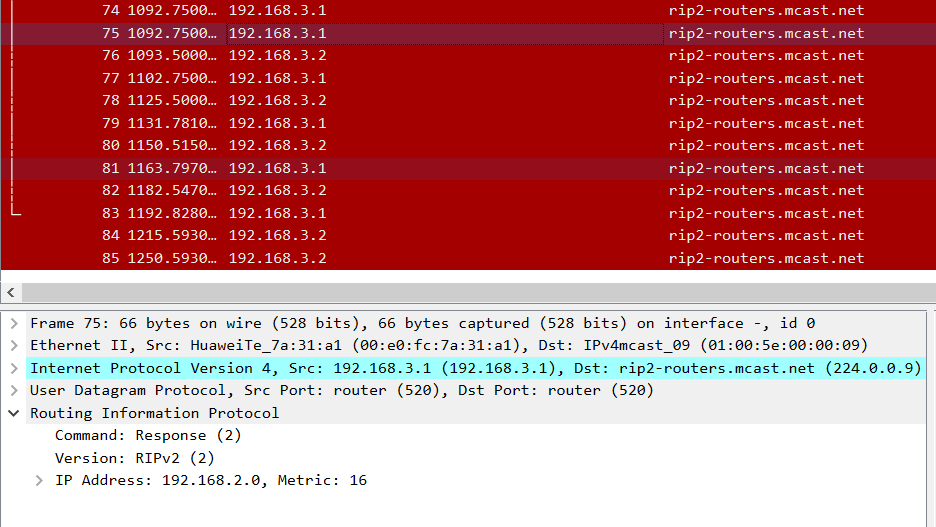
观察到“坏消息传得慢”的现象。

IP192.168.3.1到网段192.168.2.0原本metric为1:

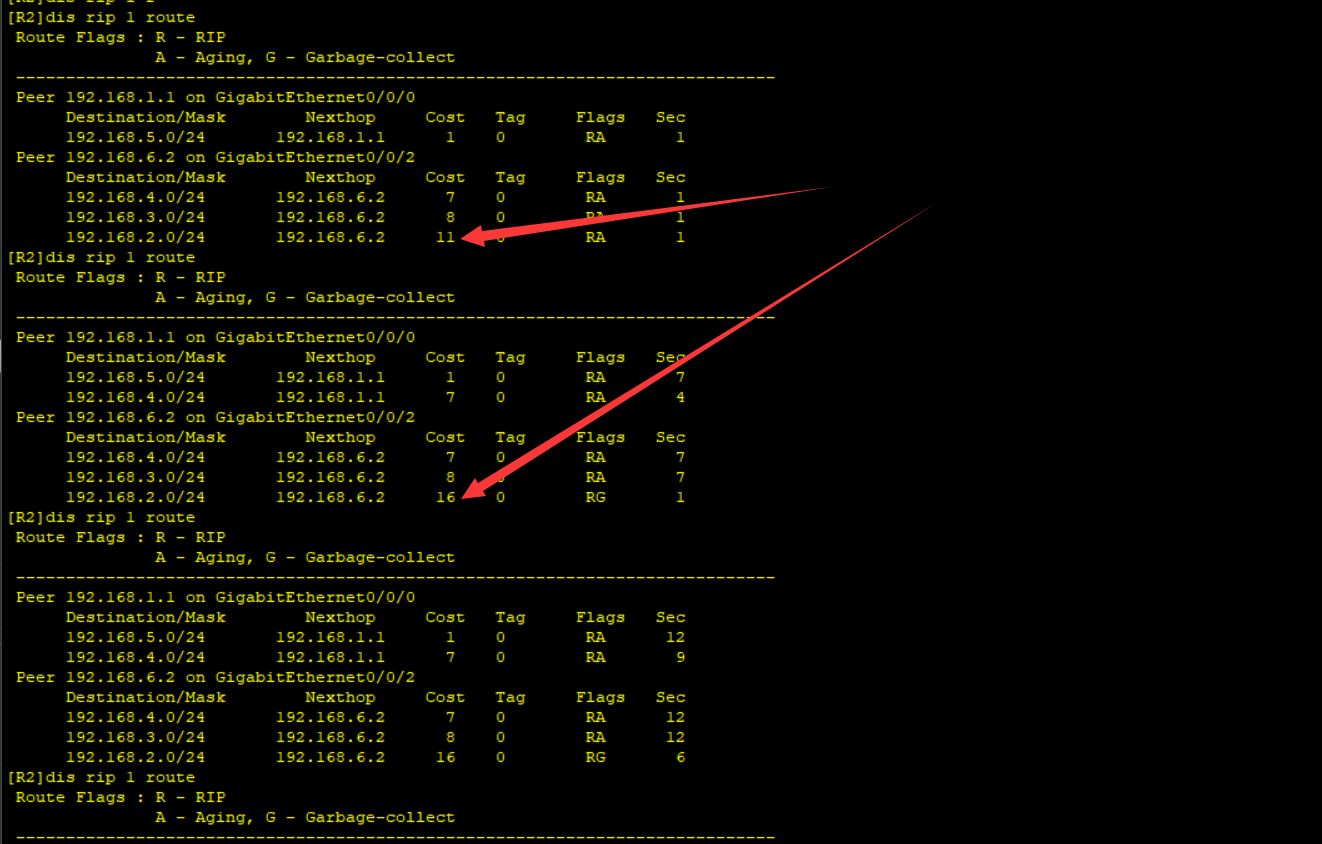


断开后，metric变为16:

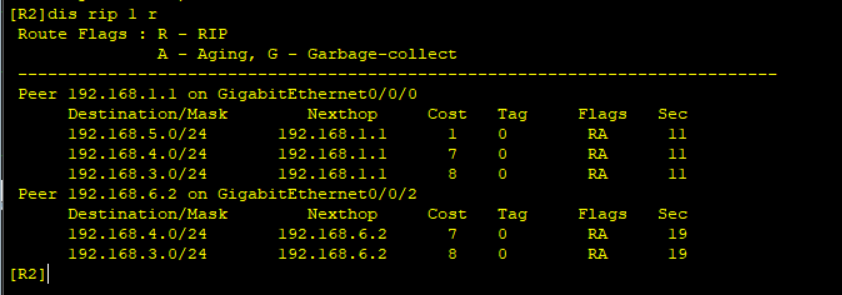




观察路由器R2的路由表:



可以看到,Cost并没有立刻变为16，这符合RIP“坏消息传得慢”的特点。

最后收敛，RIP路由发现网段192.168.2.0不可达。

## 任务3: 配置指定网络拓扑，分析OSPF路由协议收敛具体过程。

* 1. 在eNSP中搭建如下图拓扑，配置IP地址，配置OSPFv2。完成后观察链路是否连通。

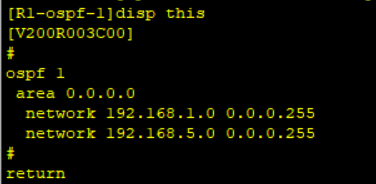


* + 1. 搭建网络拓扑

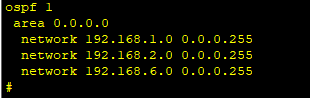
搭建网络拓扑与任务1操作相同，这里不再赘述。

* + 1. 配置OSPF动态路由

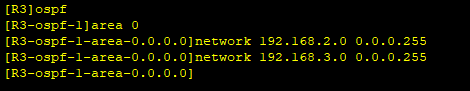
R1:



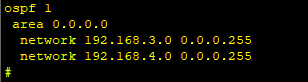
R2:



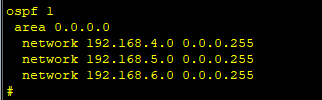
R3:



R4:

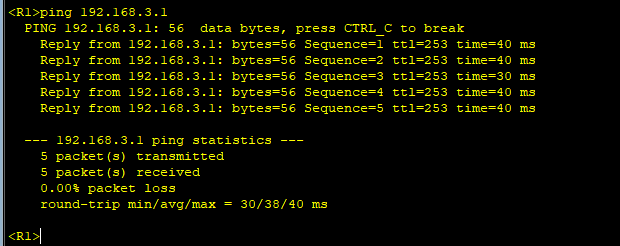


R5:

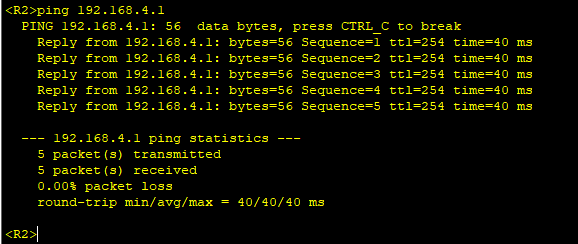


* + 1. 观察链路是否连通

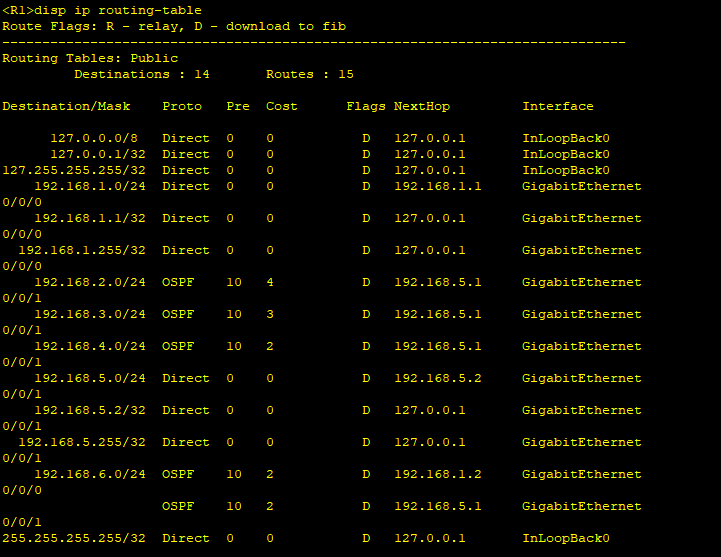
用路由器R1去ping 192.168.3.1，观察到可以ping通:



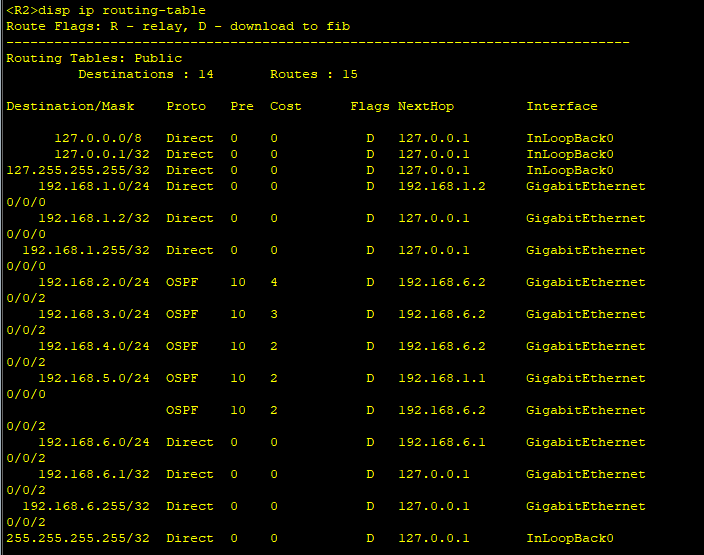
用路由器R2去ping 192.168.4.1，观察到可以ping通:



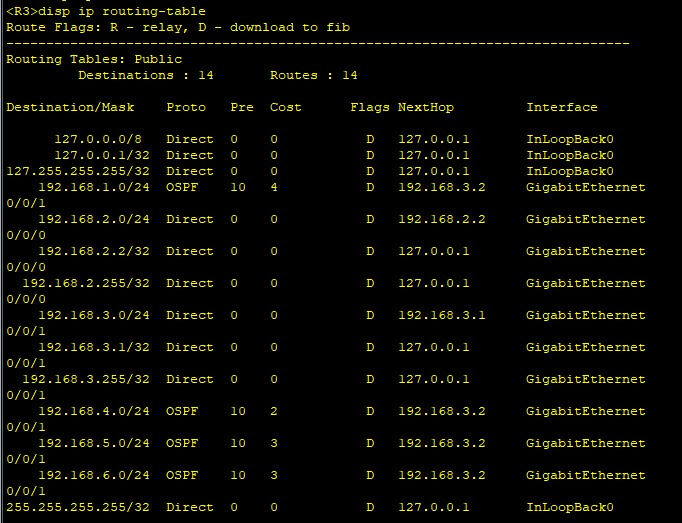
R1路由表截图:



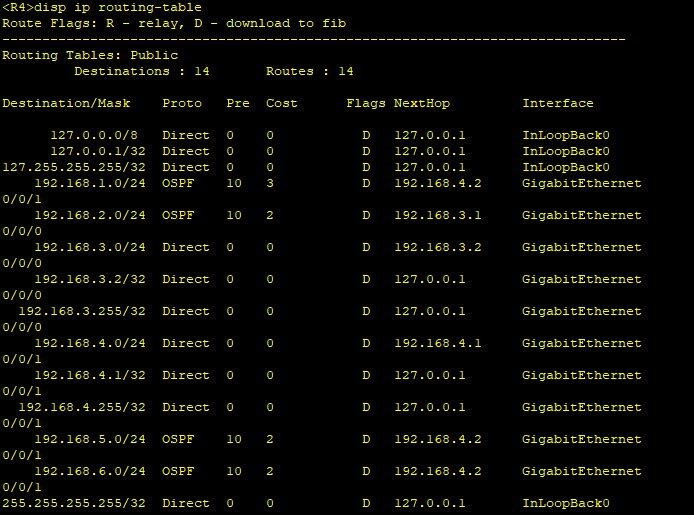
R2路由表截图:



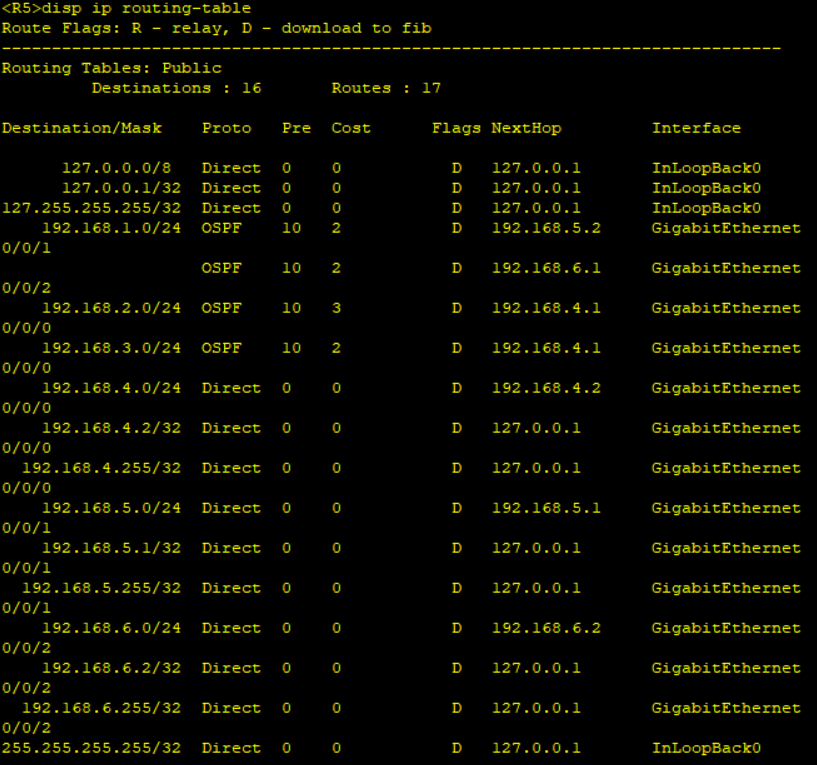
R3路由表截图:



R4路由表截图:

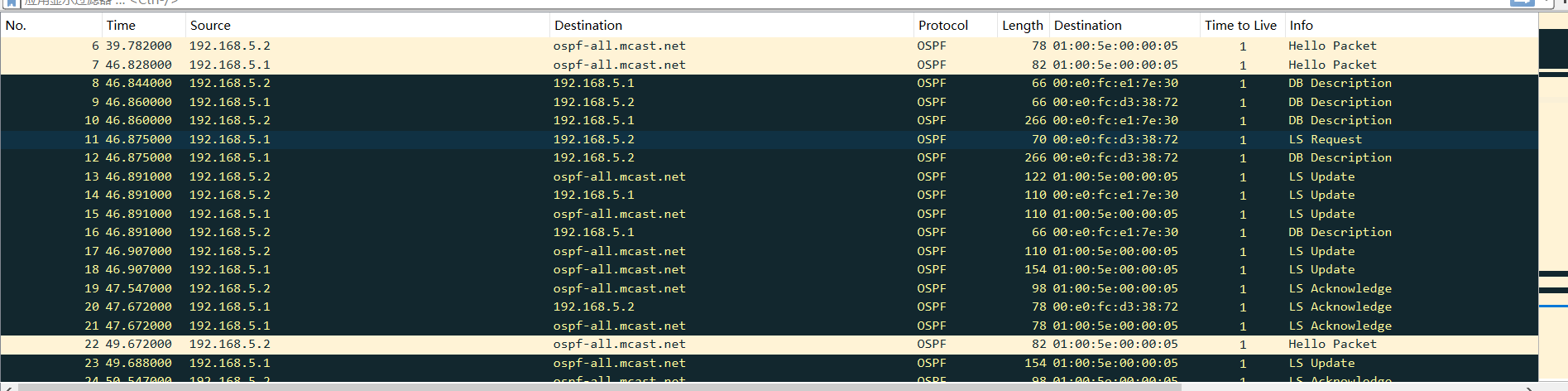


R5路由表截图:

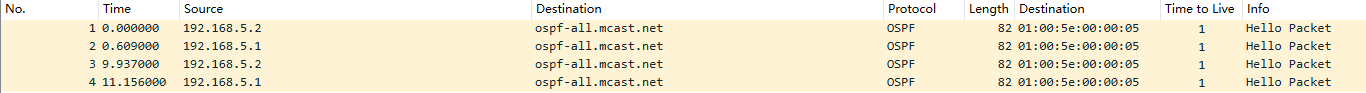


* 1. 通过Wireshark和各种路由器OSPF命令观察路由动态更新，将关键信息分析并截图。

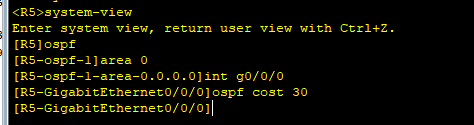
建立阶段，观察到DB Description报文:



OSPF向邻居结点发送的是Hello Packet:



当前情况下，经过R1发送出来的报文想要到达192.168.3.0网段的下一跳是192.168.5.1，使用命令增大链路192.168.4.1 <-> 192.168.4.2的开销:



Wireshark观察到OSPF的LS Update报文和LS Acknowledge报文:



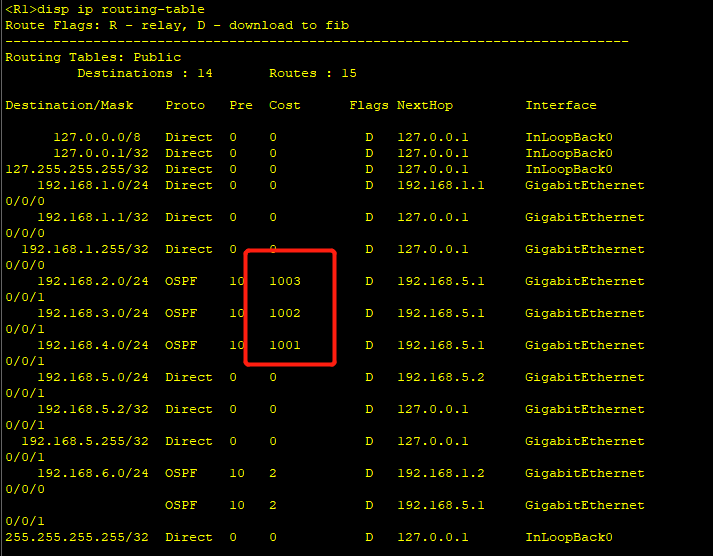
观察到路由器R1的路由表发生了开销的变化:



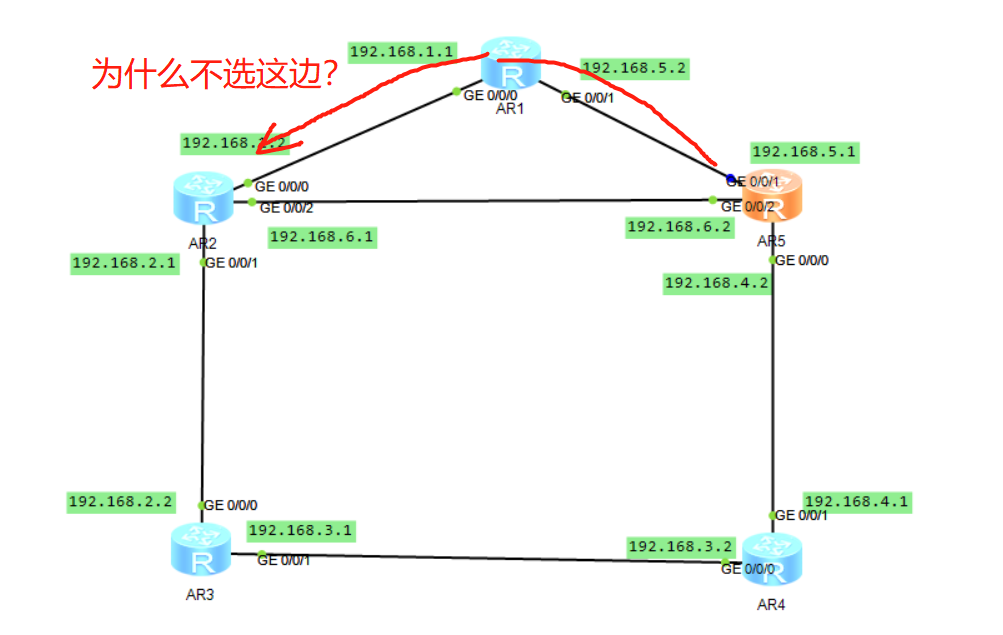
将该链路的开销增大到1000:



观察路由表:

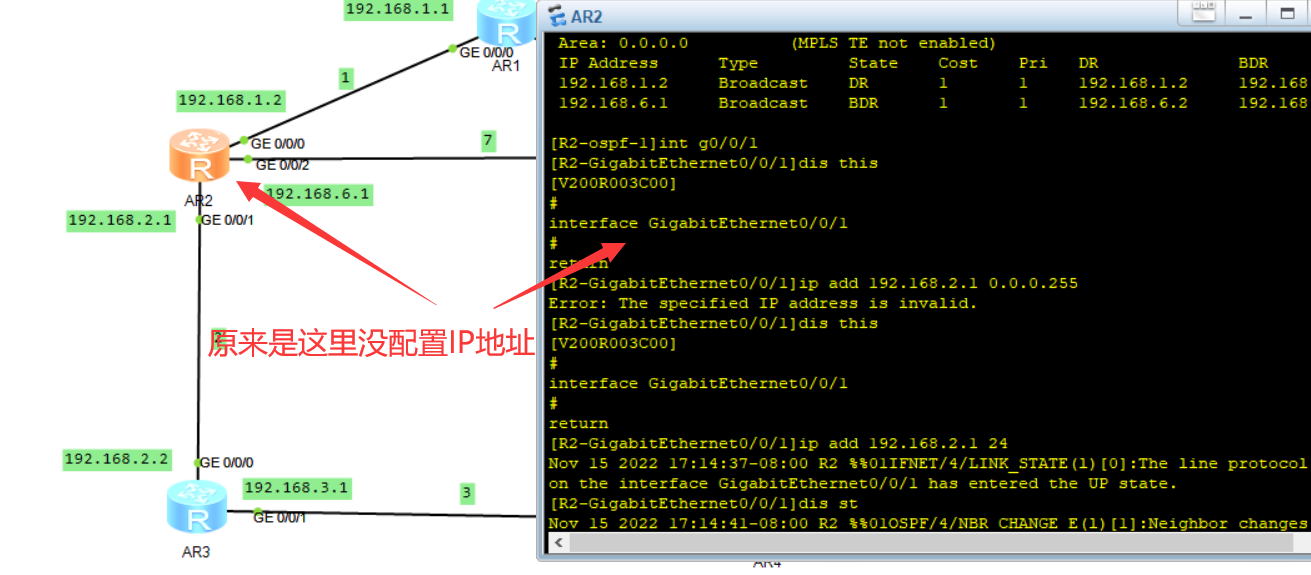


观察到R1的路由表仍然会选择192.168.5.1作为下一跳，我觉得很奇怪，它为什么会固执地选择192.168.5.1作为它的下一跳，而不选择192.168.1.2作为它的下一跳呢？

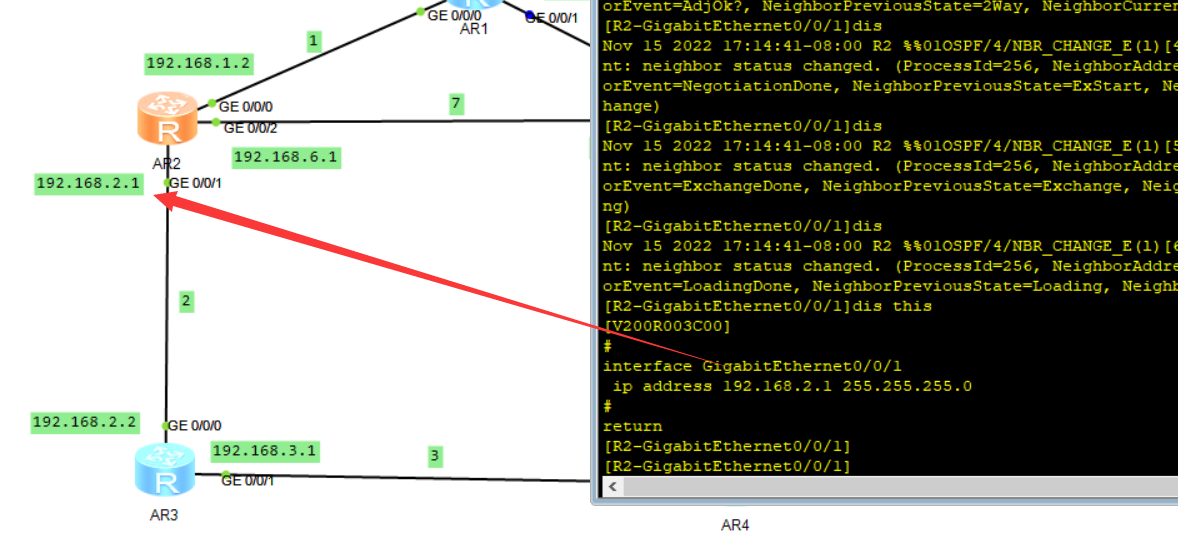


问题的解决:

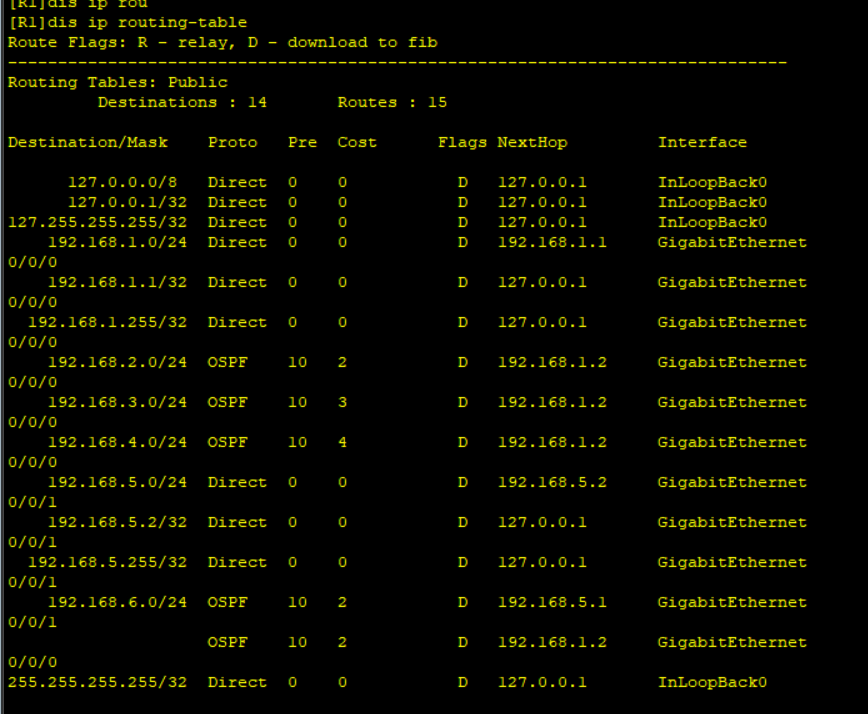
最后在老师的帮助下，发现原来是因为这个地方没有配置IP地址:



配置IP地址后:

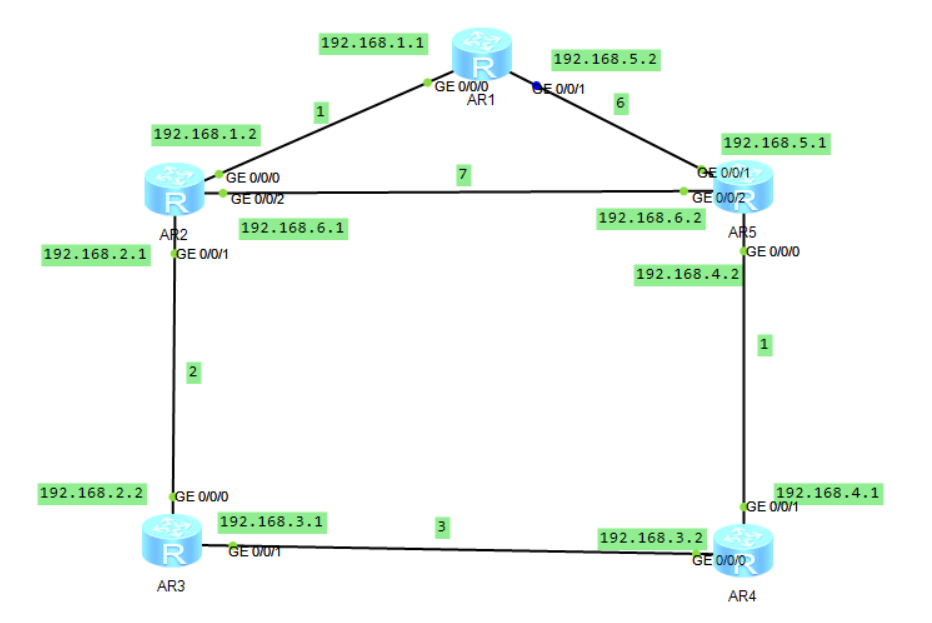


再观察:



发现路由表正确，注意这个时候我还没有改变其他链路上的Cost。

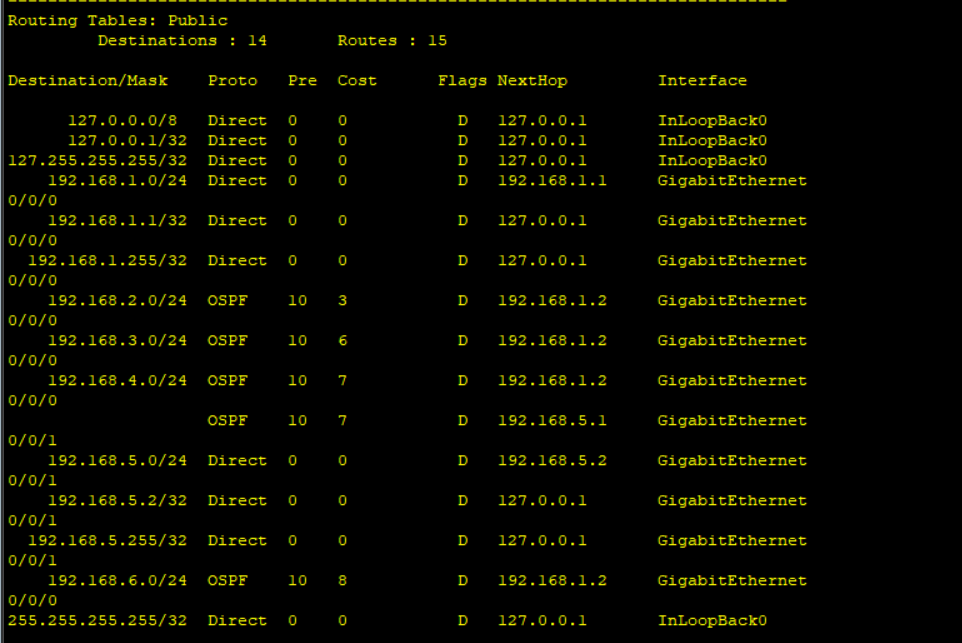
* 1. 将各链路开销配置成如图所示:



可以看到LS Update报文:



观察R1的路由表:



路由表与最短路径匹配，正确。

# 实验结果分析

对于RIP协议，本次实验完成得比较顺利，也能较好地解释Wireshark抓包得到的报文信息，可以观察到“坏消息传得慢”这样的现象。

但是对于OSPF协议，在做实验时，我发现了一个很奇怪的点，OSPF一开始选择的那条路径，无论我将这条链路的开销设置成多大，它始终会固执地选择这条路径。后来在老师的帮助下，找到了问题所在：原来是少配置了一个IP地址，将该IP地址配置上去后，问题就顺利解决了。

若想要观察到动态路由协议一开始建立时候的信息，就需要先抓包，再去设置动态路由协议，这是从助教老师那里学到的小Tips。

# 实验小结与感想

通过本次实验，我体会到了不同的动态路由算法RIP和OSPF的区别。RIP使用DV算法，OSPF使用LS算法。RIP更倾向于“人云亦云”，它无条件的相信自己邻居发过来的信息，也就是别人怎么说，我就怎么信；而OSPF则是“兼听则明”，它会得到整个网络的所有情况，至于究竟怎么走，由我自己计算来决定。