



Lab 6: 网络层路由协议模拟与观察

Lab6 Part1: 分布式距离路由算法

一、实验原理

首先让我们回顾课堂上所讲的距离向量算法：

Bellman-Ford 方程 (动态规划)

$d_x(y) :=$ 从x到y的最小成本路径

$$d_x(y) = \min_v \{ c(x,v) + d_v(y) \}$$

\min 对于x的所有邻居v
到达邻居v的成本v
从邻居v到目的地y的成本

距离向量算法

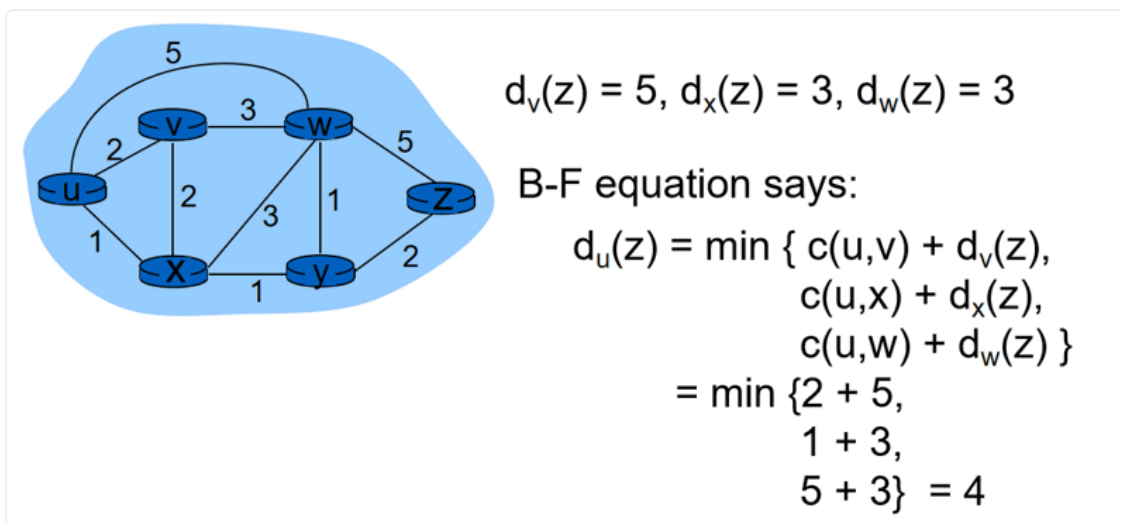
$D_x(y)$ 表示从x到y的最低开销估计值。节点x维护距离向量 $D_x = [D_x(y): y \in N]$ ，包含了x到网络中所有目的地y的开销估计值。对于节点x，其维护了到每个邻居的直接开销 $c(x,v)$ 和每个邻居的距离向量，即对每个邻居v，节点x保存 $D_v = [D_v(y): y \in N]$ 。

算法核心思路: 每个节点都将自己的距离向量估计值传递给邻居，定时或者距离向量有变化时，让对方去重新计算距离向量。当x从邻居收到距离向量时，采用B-F equation更新它自己的距离向量。

B-F equation: $D_x(y) \leftarrow \min_v \{ c(x,v) + D_v(y) \}$ for each node $y \in N$

$D_x(y)$ 估计值最终收敛于实际的最小代价值 $d_x(y)$ 。

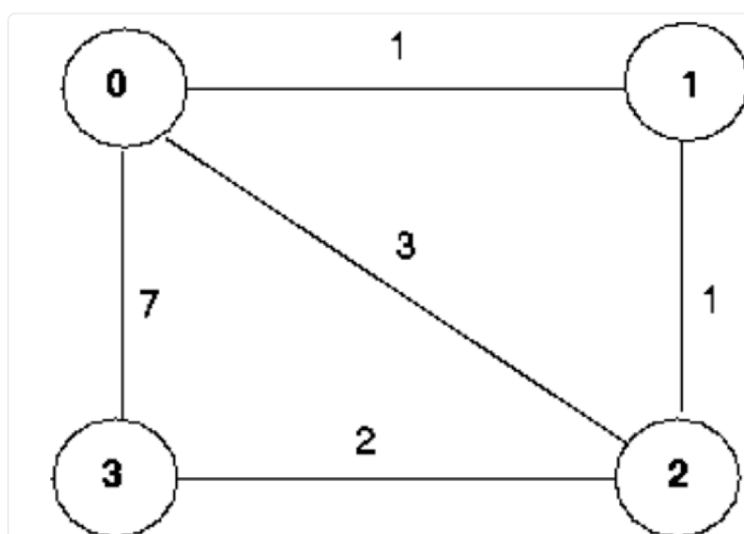
下面提供了一个具体的例子供同学们理解算法：



Bellman-Ford算法例子

二、实验任务

为此网络实现分布式异步距离向量路由算法：



网络拓扑与链路成本

1. 具体任务

对每个节点编程完成初始化与更新功能，这些功能将在为作业提供的模拟环境中异步执行。

以0号节点为例： `rtinit0()` 和 `rtupdate0()`

`rtinit0()`

- 在模拟开始将调用一次该函数，`rtinit0()` 无参数
- 它应当初始化节点0中的距离表，以反映出到达节点1、2和3的直接开销分别为1、3和7

- 它应向其直接连接的邻居（在本情况中为节点1、2和3）发送它到所有其他网络节点的最低开销路径的开销信息
- 通过调用函数`tolayer2()`，这种最低开销信息在一个路由选择更新分组中被发送给相邻节点

`rtupdate0(struct rtpkt* rcvdpkt)`

- 当节点0收到一个由其直接相连邻居之一发给它的路由选择分组时，调用该函数，参数`*rcvdpkt`是一个指向接收分组的指针
- `rtupdate0()`从其他节点*i*接收的路由选择更新分组中，包含了节点*i*到所有其他网络节点的当前最短路径开销值
- `rtupdate0()`使用这些收到的值来更新其自身的距离表
- 如果它自己到另外节点的最低开销由于此更新而发生改变的话，则节点0通过发送一个路由选择分组来通知其直接相连邻居这种最低开销的变化，仅有直接相连的节点才交换路由选择分组

2. 代码分析与软件接口

关键结构体：

```
struct distance_table
{
    int costs[4][4];
} dt0;
```

- `dt0`为节点0的距离表
- `cost[i][j]`为节点0通过邻居*j*到达节点*i*的当前开销估计值

```
extern struct rtpkt {
    int sourceid; /* 发送此分组的节点id*/
    int destid; /* 该分组的目的节点（必须是邻居） */
    int mincost[4]; /* 源到各节点的最小开销*/
};
```

Tips:

- `tolayer2(struct rtpkt pkt2send)` 传入参数是结构体，不是结构体的指针
- `printdt0(struct distance_table *dtptr)` 打印节点0的距离表

3. 网络环境仿真

- prog3.c中#define LINKCHANGES 设置为0
- rtinit0(), rtinit1(), rtinit2(), rtinit3(), rtupdate0(), rtupdate1(), rtupdate2(), rtupdate3() 将路由分组发送到传输介质中, 分组将会按序无丢失地达到目的地
- 只有直连的节点间可以通信, 时延为常数
- gcc prog3.c node0.c node1.c node2.c node3.c (C语言版本)
- 编译完成后, 运行代码时要求提供Tracing值
 - Tracing=1/2 打印模拟过程中的有用信息 (分组和计时器)
 - Tracing>2会输出所有调试信息

三、作业要求

- 为路由器分别编写node0.c, node1.c, node2.c, node3.c完成指定功能, 在其中一个.c文件中自定义的全局变量不可被其他.c文件访问
- 执行rtinit() 时, 输出 :
 - 调用时间(通过全局变量clocktime 获取)
 - 当前距离表的内容
 - 打印通告对象与信息
- 执行rtupdate()时, 输出 :
 - 调用时间(通过全局变量clocktime 获取)
 - 接收到的分组的发送方
 - 当前距离表的内容
 - 若距离表更新, 打印通告对象与信息; 若距离表未更新, 打印未更新

所需提交内容:

- 提交所有源文件 (C代码)
- 提交Trace=2的输出结果, output.pdf, 将每个节点最终的距离表高亮, 并给出每个节点到其他节点的最小开销值
- 按照实验要求完成实验, 提交实验报告, 并提供必要的截图

参考资料:

- https://media.pearsoncmg.com/aw/aw_kurose_network_3/labs/lab6/lab6.html
(C/JAVA code)

- <http://www.cse.psu.edu/~gxc27/teach/514/lab2/python/> (Python Code)

Lab6 Part2: RIP和OSPF路由协议观察

一、实验目的

- 回顾网络层数据平面IP地址、子网、掩码等概念与知识，加强对网络层控制平面基于距离向量和基于链路状态的路由协议的理解。
- 学会初步使用华为eNSP网络仿真模拟器，掌握网络拓扑搭建和配置IP等技能。
- 通过网络仿真平台，观察两种路由协议RIP和OSPF的报文，能够指出路由表中的表项是由哪些具体的报文产生的，并比较RIP和OSPF两种路由协议的异同点。

二、实验原理

首先让我们回顾课堂上所讲的RIP协议和OSPF协议：

RIP: Routing Information Protocol [RFC 1723]

- 经典的距离向量路由算法
- 原理部分可以参考Lab6 Part1的实验原理

OSPF: Open Shortest Path First [RFC 2328]

- 链路状态路由
- IS-IS protocol (ISO standard, not RFC standard) 本质上与OSPF相同
- 原理：Dijkstra算法，计算从一个节点(“源”)到所有其他节点的最小代价路径，算法的伪代码如下：

```
1 Initialization:
2   N' = {u}                                /* compute least cost path from u to all other r
3   for all nodes v
4     if v adjacent to u                    /* u initially knows direct-path-cost only to di
5       then D(v) = cu,v                    /* but may not be minimum cost! */
6     else D(v) = ∞
7
8 Loop
9   find w not in N' such that D(w) is a minimum
10  add w to N'
11  update D(v) for all v adjacent to w and not in N' :
12    D(v) = min ( D(v), D(w) + cw,v )
13  /* new least-path-cost to v is either old least-cost-path to v or knowr
14    least-cost-path to w plus direct-cost from w to v */
15 until all nodes in N'
```

下面提供一个实例供同学们理解：

Step	N'	^v D(v),p(v)	^w D(w),p(w)	^x D(x),p(x)	^y D(y),p(y)	^z D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					
5						

find *a* not in N' such that D(*a*) is a minimum
add *a* to N'
update D(*b*) for all *b* adjacent to *a* and not in N' :
 $D(b) = \min (D(b), D(a) + c_{a,b})$

链路状态算法

三、实验任务

本次实验将在华为eNSP网络仿真模拟器上进行。华为eNSP模拟器 (Enterprise Network Simulation Platform) 是华为官方推出的一款强大的图形化网络仿真工具平台，eNSP模拟器主要对企业网路由器、交换机等设备进行软件仿真，从而得以呈现真实设备部署实景，并且支持大型网络模拟，让使用者有机会在没有真实设备的情况下也能够开展实验测试、学习网络技术。

本次实验需要同学们在华为eNSP模拟器上搭建网络拓扑、配置子网、配置路由协议并在路由器端口进行抓包，观察RIP和OSPF协议的报文，分析两种协议的异同点。

1. 环境搭建与软件安装

所需要的安装包: https://pan.baidu.com/s/1XnNnxuqU_A01zNwL-daKYg

提取码: 4B45

安装步骤参考: https://blog.csdn.net/m0_74451345/article/details/135967195

需要windows环境, Mac电脑的同学请使用机房的电脑。

win11上安装virtual box可能会出现的问题及解决方案:

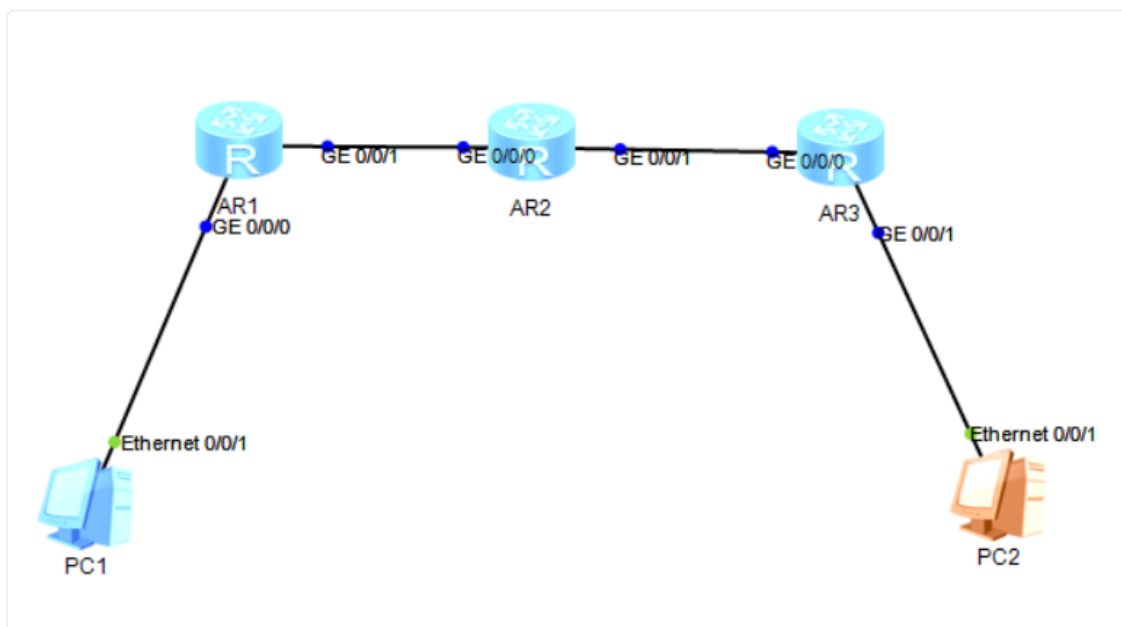
<https://blog.csdn.net/wenzhongxiang/article/details/133188587>

2. 实验任务

任务一: RIP协议报文观察

(1) 网络拓扑搭建

搭建好如下的网络拓扑:



网络拓扑

eNSP搭建网络拓扑采用拖拽式, 路由器选择 AR2220, 连线选择 Auto。

完成网络拓扑搭建后记得启动所有设备。

(2) IP配置

PC的基础配置：

PC1

基础配置

命令行

组播

UDP发包工具

串口

主机名:

MAC 地址:

54-89-98-ED-18-BC

IPv4 配置

☒ 静态

☐ DHCP

☐ 自动获取 DNS 服务器地址

IP 地址:

192 . 168 . 1 . 1

DNS1:

0 . 0 . 0 . 0

子网掩码:

255 . 255 . 255 . 0

DNS2:

0 . 0 . 0 . 0

网关:

192 . 168 . 1 . 254

IPv6 配置

☒ 静态

☐ DHCPv6

IPv6 地址:

::

前缀长度:

128

IPv6 网关:

::

应用

PC1配置

PC2

基础配置

命令行

组播

UDP发包工具

串口

主机名:

MAC 地址:

54-89-98-BA-76-60

IPv4 配置

☒ 静态

☐ DHCP

☐ 自动获取 DNS 服务器地址

IP 地址:

192 . 168 . 4 . 1

DNS1:

0 . 0 . 0 . 0

子网掩码:

255 . 255 . 255 . 0

DNS2:

0 . 0 . 0 . 0

网关:

192 . 168 . 4 . 254

IPv6 配置

☒ 静态

☐ DHCPv6

IPv6 地址:

::

前缀长度:

128

IPv6 网关:

::

应用

PC2配置

路由器配置：

AR1配置：


```

<Huawei>
Nov  5 2024 21:25:08-08:00 Huawei %%01IFPDT/4/IF_STATE(1)[0]:Interface GigabitEthernet0/0/1 has turned into UP state.
<Huawei>sys
<Huawei>system-view
Enter system view, return user view with Ctrl+Z.
[Huawei]int
[Huawei]interface G
[Huawei]interface GigabitEthernet 0/0/0
[Huawei-GigabitEthernet0/0/0]ip address 192.168.1.254 24
[Huawei-GigabitEthernet0/0/0]
Nov  5 2024 21:26:58-08:00 Huawei %%01IFNET/4/LINK_STATE(1)[1]:The line protocol IP on the interface GigabitEthernet0/0/0 has entered the UP state.
[Huawei-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
[Huawei-GigabitEthernet0/0/1]ip address 192.168.2.2 24
Nov  5 2024 21:27:19-08:00 Huawei %%01IFNET/4/LINK_STATE(1)[2]:The line protocol IP on the interface GigabitEthernet0/0/1 has entered the UP state.
[Huawei-GigabitEthernet0/0/1]

```

- interface GigabitEthernet 0/0/0 : 进入某个指定端口
- ip address 192.168.1.254 24: 配置端口IP地址, 最后 24 为掩码长度

AR2配置:

```

<Huawei>sys
<Huawei>system-view
Enter system view, return user view with Ctrl+Z.
[Huawei]int
[Huawei]interface G
[Huawei]interface GigabitEthernet 0/0/0
[Huawei-GigabitEthernet0/0/0]ip address 192.168.2.3 24
[Huawei-GigabitEthernet0/0/0]
Nov  5 2024 21:32:00-08:00 Huawei %%01IFNET/4/LINK_STATE(1)[0]:The line protocol IP on the interface GigabitEthernet0/0/0 has entered the UP state.
[Huawei-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
[Huawei-GigabitEthernet0/0/1]ip address 192.168.3.2 24
Nov  5 2024 21:32:14-08:00 Huawei %%01IFNET/4/LINK_STATE(1)[1]:The line protocol IP on the interface GigabitEthernet0/0/1 has entered the UP state.
[Huawei-GigabitEthernet0/0/1]

```

AR3配置:

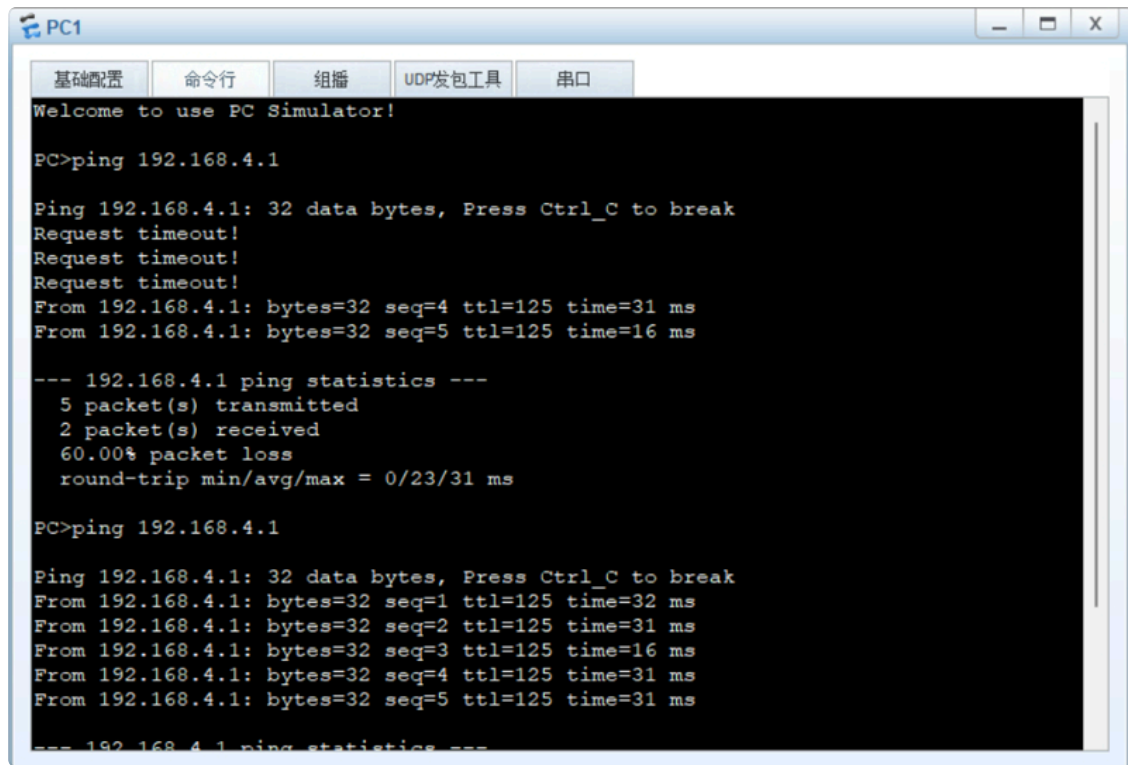
```
<Huawei>sys
<Huawei>system-view
Enter system view, return user view with Ctrl+Z.
[Huawei]int
[Huawei]interface G
[Huawei]interface GigabitEthernet 0/0/0
[Huawei-GigabitEthernet0/0/0]ip address 192.168.3.3 24
[Huawei-GigabitEthernet0/0/0]
Nov  5 2024 21:33:44-08:00 Huawei %%01IFNET/4/LINK_STATE(1)[0]:The line protocol
  IP on the interface GigabitEthernet0/0/0 has entered the UP state.
[Huawei-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
[Huawei-GigabitEthernet0/0/1]ip address 192.168.4.254 24
Nov  5 2024 21:34:00-08:00 Huawei %%01IFNET/4/LINK_STATE(1)[1]:The line protocol
  IP on the interface GigabitEthernet0/0/1 has entered the UP state.
[Huawei-GigabitEthernet0/0/1]
```

(3) 路由协议配置

RIP路由协议配置，下面以AR1为例，AR2和AR3请同学自行完成：

```
<Huawei>sys
<Huawei>system-view
Enter system view, return user view with Ctrl+Z.
[Huawei]rip 1
[Huawei-rip-1]network 192.168.1.0
[Huawei-rip-1]network 192.168.2.0
[Huawei-rip-1]
```

- network：在RIP中通告自己所在的网络
- 测试完请使用ping指令测试：PC1 ping PC2



```
PC1
基础配置 命令行 组播 UDP发包工具 串口
Welcome to use PC Simulator!

PC>ping 192.168.4.1

Ping 192.168.4.1: 32 data bytes, Press Ctrl_C to break
Request timeout!
Request timeout!
Request timeout!
From 192.168.4.1: bytes=32 seq=4 ttl=125 time=31 ms
From 192.168.4.1: bytes=32 seq=5 ttl=125 time=16 ms

--- 192.168.4.1 ping statistics ---
    5 packet(s) transmitted
    2 packet(s) received
   60.00% packet loss
   round-trip min/avg/max = 0/23/31 ms

PC>ping 192.168.4.1

Ping 192.168.4.1: 32 data bytes, Press Ctrl_C to break
From 192.168.4.1: bytes=32 seq=1 ttl=125 time=32 ms
From 192.168.4.1: bytes=32 seq=2 ttl=125 time=31 ms
From 192.168.4.1: bytes=32 seq=3 ttl=125 time=16 ms
From 192.168.4.1: bytes=32 seq=4 ttl=125 time=31 ms
From 192.168.4.1: bytes=32 seq=5 ttl=125 time=31 ms

--- 192.168.4.1 ping statistics ---
```

PC1 ping PC2

(4) 查看路由器内路由表

```

<Huawei>sys
<Huawei>system-view
Enter system view, return user view with Ctrl+Z.
[Huawei]display ip routing-table
Route Flags: R - relay, D - download to fib
-----
Routing Tables: Public
      Destinations : 12          Routes : 12

Destination/Mask    Proto    Pre  Cost           Flags NextHop         Interface
-----
      127.0.0.0/8     Direct   0     0                D    127.0.0.1        InLoopBack0
      127.0.0.1/32     Direct   0     0                D    127.0.0.1        InLoopBack0
127.255.255.255/32   Direct   0     0                D    127.0.0.1        InLoopBack0
      192.168.1.0/24   Direct   0     0                D    192.168.1.254    GigabitEthernet
0/0/0
      192.168.1.254/32 Direct   0     0                D    127.0.0.1        GigabitEthernet
0/0/0
      192.168.1.255/32 Direct   0     0                D    127.0.0.1        GigabitEthernet
0/0/0
      192.168.2.0/24   Direct   0     0                D    192.168.2.2      GigabitEthernet
0/0/1
      192.168.2.2/32   Direct   0     0                D    127.0.0.1        GigabitEthernet
0/0/1
      192.168.2.255/32 Direct   0     0                D    127.0.0.1        GigabitEthernet
0/0/1
      192.168.3.0/24   RIP      100    1                D    192.168.2.3      GigabitEthernet
0/0/1
      192.168.4.0/24   RIP      100    2                D    192.168.2.3      GigabitEthernet
0/0/1
255.255.255.255/32   Direct   0     0                D    127.0.0.1        InLoopBack0

[Huawei]

```

(5) 端口抓包

右键点击交换机选择“数据抓包”

3 934.718000	192.168.2.2	255.255.255.255	RIPv1	66	1	0x0000 (0)	Response
4 934.718000	192.168.2.2	255.255.255.255	RIPv1	66	1	0x0001 (1)	Request
5 968.718000	192.168.2.2	255.255.255.255	RIPv1	66	1	0x0002 (2)	Response
6 978.750000	192.168.2.2	255.255.255.255	RIPv1	86	1	0x0005 (5)	Response
7 981.750000	192.168.2.2	255.255.255.255	RIPv1	66	1	0x0006 (6)	Response
8 1005.750000	192.168.2.2	255.255.255.255	RIPv1	86	1	0x0008 (8)	Response
9 1038.765000	192.168.2.2	255.255.255.255	RIPv1	86	1	0x000b (11)	Response
10 1045.750000	192.168.2.1	255.255.255.255	RIPv1	66	1	0x0002 (2)	Request
11 1045.765000	192.168.2.1	255.255.255.255	RIPv1	86	1	0x0003 (3)	Response
12 1046.750000	192.168.2.2	255.255.255.255	RIPv1	86	1	0x000c (12)	Response
13 1048.750000	192.168.2.1	255.255.255.255	RIPv1	66	1	0x0006 (6)	Response
14 1072.765000	192.168.2.2	255.255.255.255	RIPv1	86	1	0x000f (15)	Response
15 1074.781000	192.168.2.1	255.255.255.255	RIPv1	86	1	0x0008 (8)	Response

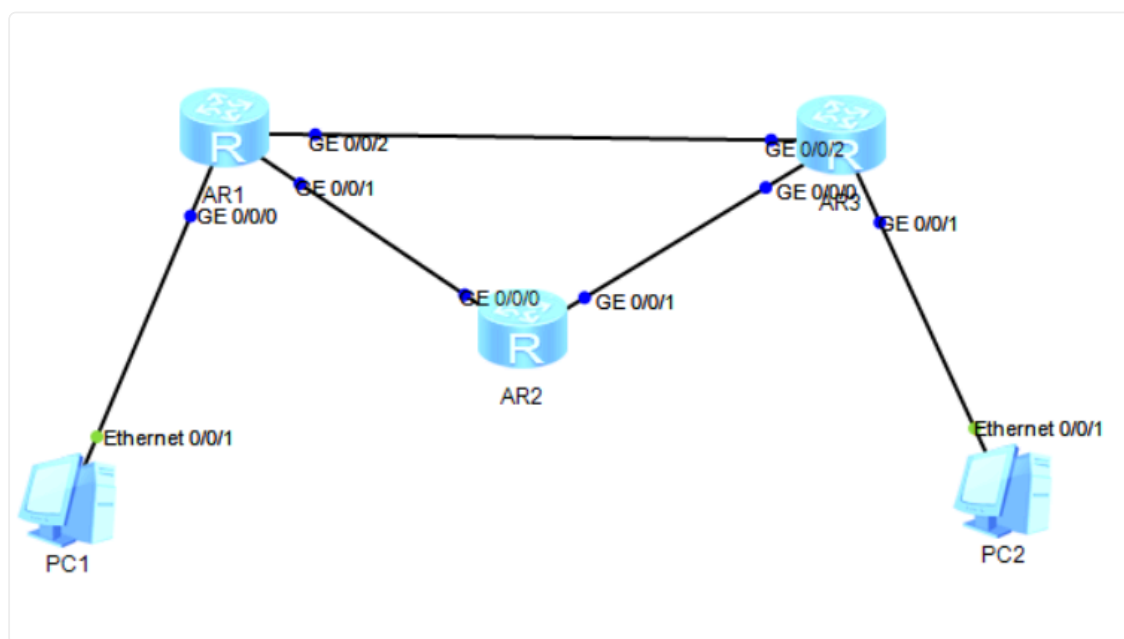
端口抓包

part1: 完成以下任务并回答问题

1. 按照要求搭建拓扑，完成相关PC与路由器的配置，配置成功 PC1 应能 ping PC2，截图。并在图中指出网络中共有几个子网，每个子网的IP及掩码是什么？
2. 查看各个路由器的中的IP路由表，截图并用红色的框标识出RIP协议产生的路由表项。
3. 在各个路由器端口进行抓包，观察RIP协议报文，截图举例说明报文各字段的含义。并说明路由器中的IP路由表中的RIP路由表项是如何由RIP协议报文中的距离向量计算得出的，举一个例子即可。
4. RIP协议中，距离向量的传递是在邻居之间，还是距离向量会直接传播到全网？传递的距离向量是完整的距离向量吗？截图举例说明？

(6) 变化网络拓扑

按照下图修改网络拓扑：



新网络拓扑

part2: 完成以下任务并回答问题

1. 按照要求搭建新的拓扑，完成相关路由器的配置（AR1和AR3的新端口的配置、RIP路由协议的配置）。并在图中指出网络中共有几个子网，每个子网的IP及掩码是什么？
2. 观察各个路由器的IP路由表变化，截图AR1和AR3的IP路由表；并在各个路由器端口进行抓包，说明在改变网络拓扑后RIP报文中距离向量的变化以及该变化如何影响IP路由表，截图并举例说明。

任务二：OSPF协议报文观察

网络拓扑搭建（使用旧的网络拓扑）、IP配置、路由表查看、端口抓包与任务一相同，此处不再赘述。

OSPF路由协议配置

```
<Huawei>sys
<Huawei>system-view
Enter system view, return user view with Ctrl+Z.
[Huawei]ospf 1
[Huawei-ospf-1]area 0
[Huawei-ospf-1-area-0.0.0.0]network 192.168.1.0 0.0.0.255
[Huawei-ospf-1-area-0.0.0.0]network 192.168.2.0 0.0.0.255
[Huawei-ospf-1-area-0.0.0.0]
```

完成以下任务并回答问题：

1. 按照要求搭建拓扑，完成相关PC与路由器的配置，配置成功PC1应能ping PC2，截图。
2. 查看各个路由器的中的IP路由表，截图并用红色的框标识出OSPF协议产生的路由表项。
3. 在各个路由器端口进行抓包，观察OSPF协议中LS Update Packet，截图举例说明Router-LSA和Network-LSA各自的作用与区别。并说明路由器中的IP路由表中的OSPF路由表项是如何由OSPF协议报文中的LSA计算得出的，截图并说明。参考资料：OSPF报文格式解析 <https://blog.51cto.com/xxy12345/2518384>
4. OSPF协议中的LSA信息传递是在邻居间交换还是在全网传递，截图并说明。

Previous

Lab 5: 网络层数据平面观察实验

Next

附录A: Wireshark 教程

Last updated 33 minutes ago