迈普通信技术股份有限公司

迈普设备实验指南

目录

[第一章 网络基础 5](#_Toc465170318)

[1.1. FTP&TFTP备份和恢复配置设备升级 5](#_Toc465170319)

[1.2. ARP获取和释放 9](#_Toc465170320)

[1.3. MAC地址表、ARP表及其路由表基础实验 11](#_Toc465170321)

[1.4. DHCP实验 20](#_Toc465170322)

[第二章 路由实验 23](#_Toc465170323)

[2.1. VLAN实验 23](#_Toc465170325)

[2.1.1. 不同vlan相互隔离 23](#_Toc465170326)

[2.1.2. 相同vlan互通 25](#_Toc465170327)

[2.1.3. Hybrid模式 28](#_Toc465170328)

[2.2. 路由基础 31](#_Toc465170329)

[2.2.1. 静态路由 31](#_Toc465170330)

[2.2.2. 静态浮动路由实现主备 35](#_Toc465170331)

[2.2.3. 静态浮动路由分流互备 40](#_Toc465170332)

[2.3. 有关RIP路由协议及其计时器问题讨论 45](#_Toc465170333)

[2.4. 分发列表、偏移列表实现Rip跳数选路 50](#_Toc465170334)

[2.5. 路由引入 55](#_Toc465170335)

[2.5.1. OSPF和rip 之间的路由重分发 55](#_Toc465170336)

[2.5.2. 静态、直连路由重分发进rip 59](#_Toc465170337)

[2.5.3. Route-map工具应用案例 62](#_Toc465170338)

[2.6. OSPF选路 66](#_Toc465170339)

[2.7. OSPF分流互备 74](#_Toc465170340)

[2.8. ACL 91](#_Toc465170341)

[2.8.1. Acl实现业务隔离 91](#_Toc465170342)

[2.8.2. 带时间域的IP扩展ACL 95](#_Toc465170343)

[2.9. NAT实验 99](#_Toc465170344)

[2.10. NAT端口映射 103](#_Toc465170345)

[2.11. 安全综合实验 106](#_Toc465170346)

[2.12. 分流备份综合实验 114](#_Toc465170347)

[2.13. BGP综合实验 129](#_Toc465170348)

[第三章 交换实验 143](#_Toc465170349)

[3.1. 交换端口启用ip+mac安全绑定功能 143](#_Toc465170351)

[3.2. 交换综合实验 147](#_Toc465170352)

[第四章 IPsec vpn 161](#_Toc465170353)

[4.1. 路由器之间建立IPsec 161](#_Toc465170355)

[4.1.1. 建立IPsec隧道 161](#_Toc465170356)

[4.1.2. 注入反向路由 165](#_Toc465170357)

[4.1.3. 重分发注入的反向静态路由 170](#_Toc465170358)

[4.1.4. IPsec隧道线路的备份 175](#_Toc465170359)

[4.1.5. IPsec 隧道分流备份 183](#_Toc465170360)

[4.2. L2tp+ipsec隧道 192](#_Toc465170361)

[第五章 无线实验部分 205](#_Toc465170362)

[5.1. AP静态注册 205](#_Toc465170366)

[5.2. AC集中转发 213](#_Toc465170367)

[5.3. 同一AP同时释放两个ssid 213](#_Toc465170368)

[5.4. AC管理多AP 217](#_Toc465170369)

[5.5. AP三层自动注册到AC 224](#_Toc465170370)

[5.6. WPA2方式加密认证 228](#_Toc465170371)

[5.7. 无线安全实验 233](#_Toc465170372)

[5.7.1. SSID隐藏 233](#_Toc465170373)

[5.7.2. mac地址白名单/黑名单 239](#_Toc465170374)

[5.7.3. 信道自动调整 245](#_Toc465170375)

[5.7.4. 不同AP二层漫游 248](#_Toc465170376)

[5.7.5. 无线用户带宽控制： 252](#_Toc465170377)

[5.7.6. 基于时间段控制AP在线时间 256](#_Toc465170378)

[第六章 语音实验部分 260](#_Toc465170379)

[6.1. 语音网关的基本配置操作 260](#_Toc465170380)

[6.2. 语音接口的基本参数操作 270](#_Toc465170381)

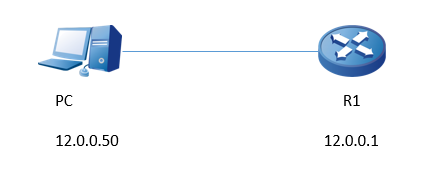
[6.3. 语音网关电话互通配置 282](#_Toc465170382)

[6.4. 通过SIP服务器实现电话互通 299](#_Toc465170383)

# 第一章 网络基础

## FTP&TFTP备份和恢复配置设备升级

实验拓扑结构：



实验要求：

通过FTP/TFTP协议实现系统备份、恢复和设备升级。

（1）将PC端作为客户端，R1作为服务器端。将R1配置备份到PC机上。

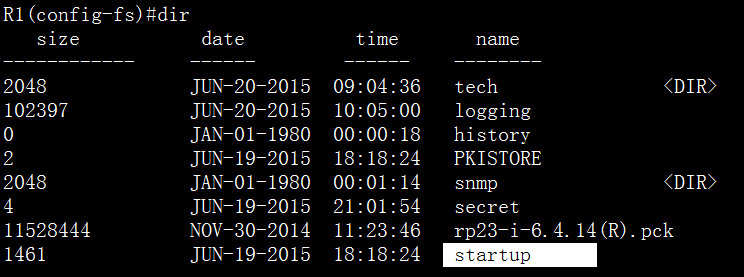
首先需要在PC机上安装一款3CD软件。

设置3CD：打开3CD软件，点击，点击**“FTP用户”**，设置“**用户名称”**，**点击“设置/改变用户口令**”设置用户密码，确定。**“用户目录”**为需要将配置文件保存的位置。点击**“保存用户”**，点击确定。点击按钮，当看到变为该红色字样，则3CD处于开启状态。3CD设置完成。

在R1上：进入文件系统视图

R1#**filesystem**

**R1(config-fs)#dir**



现在我们需要备份**startup**这个启动配置文件。使用该条命令实现。

R1(config-fs)#**copy startup-config ftp 12.0.0.50 ftptext ftptext backup1**

**\*\*//将startup-config 文件通过FTP协议拷贝到目标地址为12.0.0.50用户名为ftptext 密码为ftptext的主机上，且将该文件重新命名为backup1。**

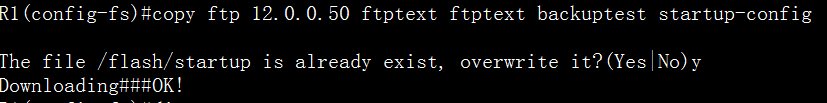
备份成功后可以在PC机上相应的保存位置看到这个备份后的文件。

（2）PC机作为服务器端，R1作为客户端，将刚备份的文件恢复到R1上。

3CD设置不变。在R1上使用该命令实现

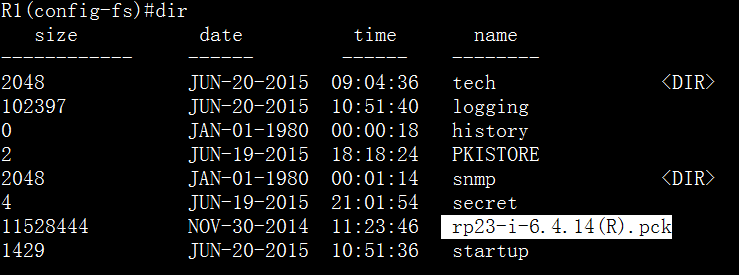
R1(config-fs)#copy ftp 12.0.0.50 ftptext ftptext backuptest startup-config

**\*\*//通过ftp将服务器12.0.0.50上用户名为ftptext,密码为ftptext下的backup1文件拷贝到startup-config中。**

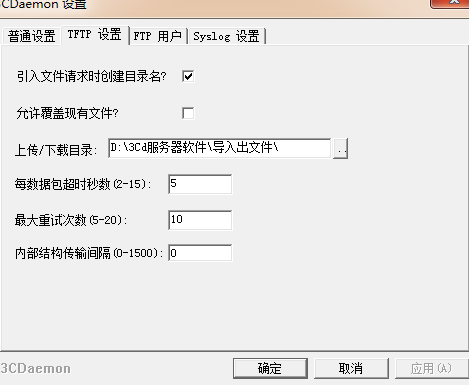


（3）通过TFTP备份、恢复系统软件。

首先将R1上的系统软件备份到PC上。



需要备份的文件为**rp23-i-6.4.14(R).pck**

3CD设置：

R1#filesystem

R1(config-fs)#**copy file-system rp23-i-6.4.14(R).pck tftp 12.0.0.50 tftptextrp23.pck**

**\*\*//将系统文件rp23-i-6.4.14(R).pck通过tftp协议备份到主机12.0.0.50上并命名为tftptextrp23.pck。**

备份成功后R1上会提示“**Total 11528444 bytes copying completed!”** 且在相应的保存位置可以看到这个文件。

（4）通过TFTP将设备系统恢复

3CD设置不变

在R1上使用该条命令：

**R1(config-fs)#copy tftp 12.0.0.50 tftptextrp23.pck file-system tftptextrp23.pck**

**\*\*//通过tftp协议将服务器12.0.0.50上的tftptextrp23.pck 恢复到文件系统中，并命名为tftptextrp23.pck**

升级成功后可以看到

R1(config-fs)#dir

size date time name

------------ ------ ------ --------

2048 JUN-20-2015 11:41:12 tech <DIR>

102397 JUN-20-2015 11:41:18 logging

0 JAN-01-1980 00:00:18 history

2 JUN-20-2015 11:39:40 PKISTORE

2048 JAN-01-1980 00:01:14 snmp <DIR>

4 JUN-20-2015 11:39:46 secret

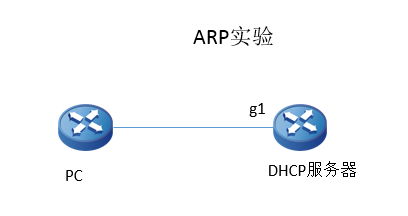
11528444 NOV-30-2014 11:23:46 **tftptextrp23.pck**

1462 JUN-20-2015 11:39:42 startup

版本名称已变为tftptextrp23.pck 。系统软件恢复成功。

## ARP获取和释放

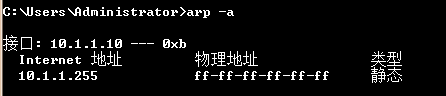
实验拓扑：



实验要求：清空ARP表项后通过ping工具，PC机能够自动发现服务器Mac地址。服务器能够自动发现PC机的Mac地址。

C:\Users\Administrator>arp –d //清空ARP表项

C:\Users\Administrator>arp –a //查看此时ARP表项



Ping 服务器g1接口



ARP –a 查看此时是否自动获取到服务器Mac地址

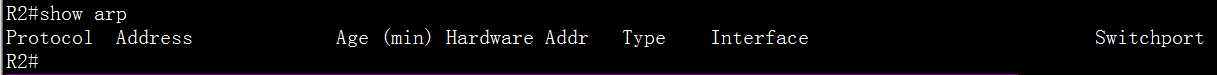


由图可知，PC机获取到服务器Mac地址。

在服务器上清空ARP表项：

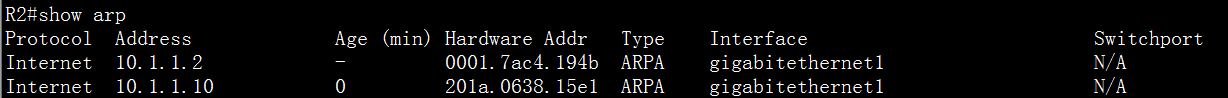
R2#clear arp-cache

R2#show arp



R2#ping 10.1.1.10

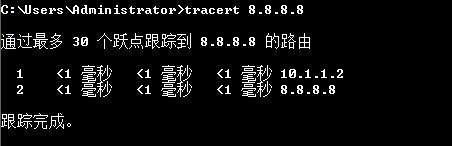
R2#show arp



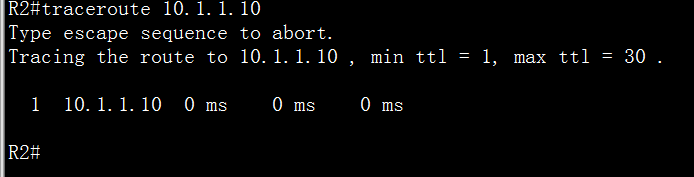
可以看到服务器获取到了PC机的Mac地址

Tracert命令用于PC端跟踪路由路径信息。服务器上另加一台路由器地址段设为8.8.8.0/24。

在PC机上tracert 8.8.8.8



Traceroute命令用于网络设备端跟踪路由路径信息。



## MAC地址表、ARP表及其路由表基础实验

实验拓扑图下：



实验要求：分别在每台设备上查看其mac地址表、arp表、及其路由表。理清各层设备转发特点及其依据。SW1/SW3为二层交换机。SW3为三层设备充当PC1和PC2的网关。PC2、PC3均用路由器模拟（PC设备有限）。

基本地址配置如下：

SW2(config)#vlan 10,20

SW2(config)#interface vlan10

SW2(config-if-vlan10)# ip address 192.168.1.254 255.255.255.0

SW2(config-if-vlan10)# exit

SW2(config)#interface vlan20

SW2(config-if-vlan20)# ip address 192.168.2.254 255.255.255.0

SW2(config-if-vlan20)# exit

SW2(config)#interface fastethernet0/0

SW2(config-if-fastethernet0/0)# switchport access vlan 10

SW2(config-if-fastethernet0/0)# exit

SW2(config)#interface fastethernet0/1

SW2(config-if-fastethernet0/1)# switchport access vlan 20

SW2(config-if-fastethernet0/1)# exit

SW2(config)#interface gigabitethernet2

SW2(config-if-gigabitethernet2)# ip address 10.0.0.2 255.255.255.252

SW2(config-if-gigabitethernet2)# exit

SW2(config)#ip route 0.0.0.0 0.0.0.0 10.0.0.1

R1(config)#interface gigabitethernet1

R1(config-if-gigabitethernet1)# ip address 192.168.3.254 255.255.255.0

R1(config-if-gigabitethernet1)# exit

R1(config)#interface gigabitethernet2

R1(config-if-gigabitethernet2)# ip address 10.0.0.1 255.255.255.252

R1(config-if-gigabitethernet2)# exit

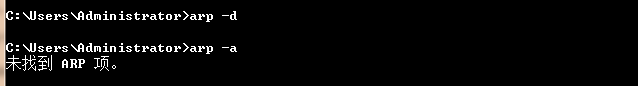
R1(config)#ip route 0.0.0.0 0.0.0.0 10.0.0.2

首先查看在没有数据流通过网络的情况下各个设备的Mac地址表、ARP表和路由表。

首先是PC1，我们先删除PC1上的ARP表缓存。通过命令arp -d删除其缓存信息，然后通过命令ARP -a 查看其ARP表项。

C:\Users\Administrator>arp -d

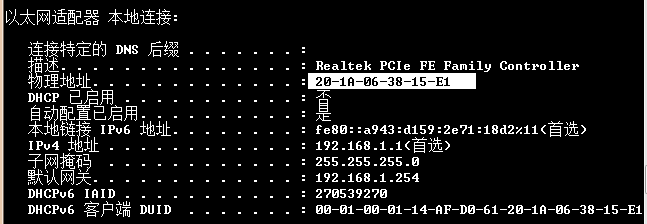
C:\Users\Administrator>arp –a



此时，ARP表项中无任何信息。

查看其Mac地址通过ipconfig/all命令可以看到PC1的网卡地址。

C:\Users\Administrator>ipconfig/all



显示其网卡地址为20-1A-06-38-15-E1.。

再看PC1上的路由表信息。Route print

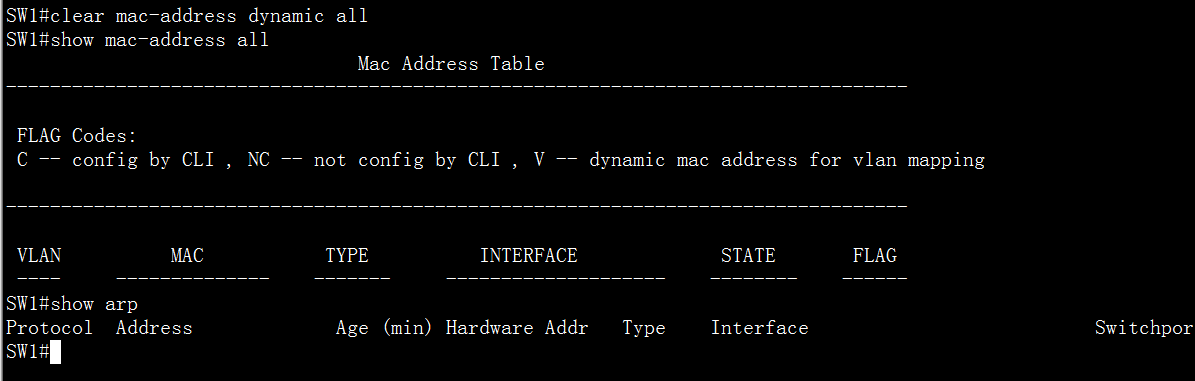
C:\Users\Administrator> route print



可以看到只配置了一条有效永久路由。默认路由的下一跳指向了网关192.168.1.254 。

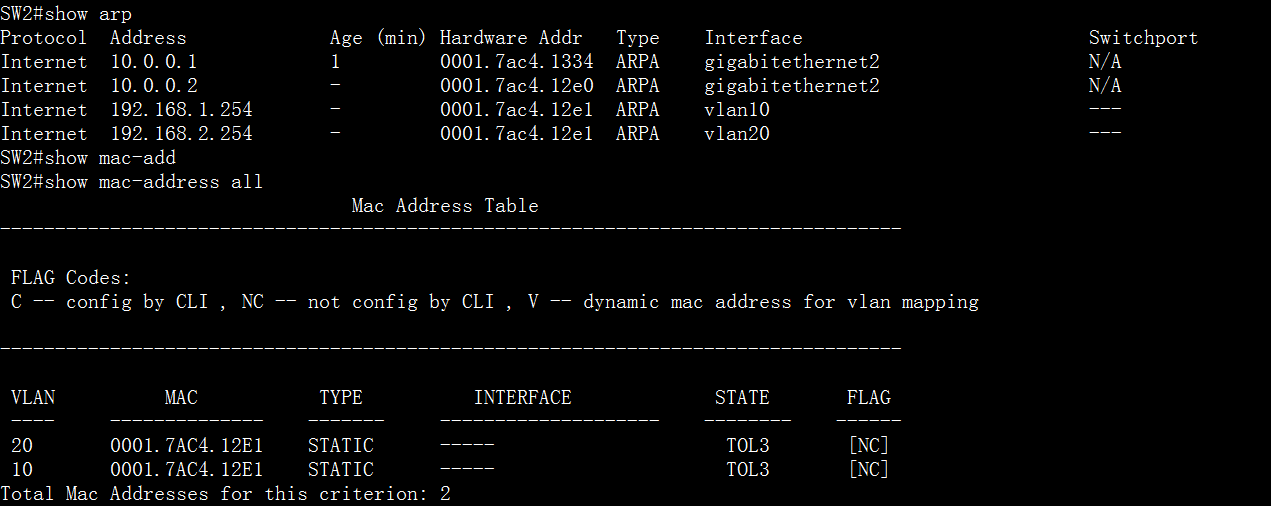
再分别查看SW1、SW2、SW3和R1的各个表项。

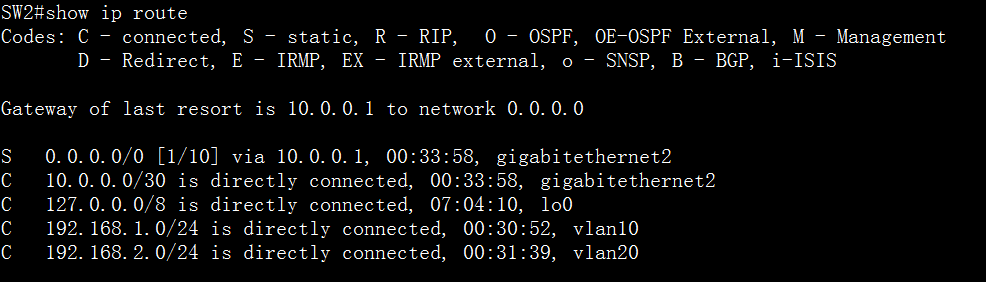
SW1：



没有数据通过网络时二层设备SW1的Mac地址表项为空。无ARP表项。二层设备无路由表。

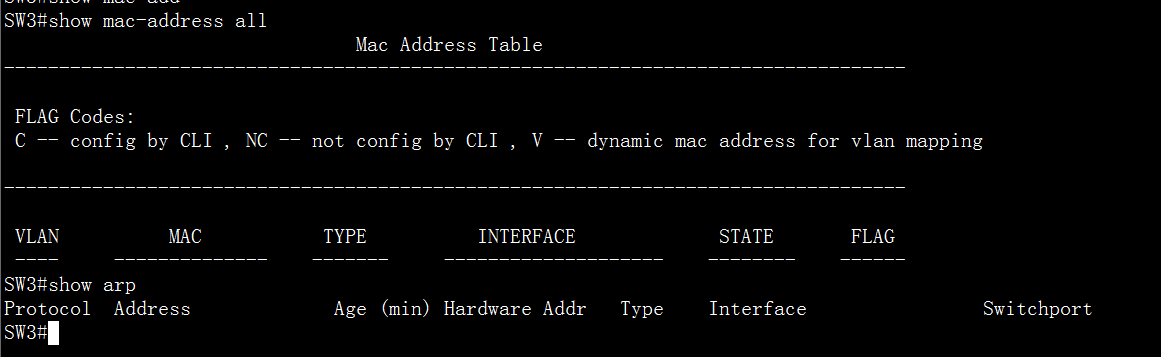
在三层设备SW2上看三张表。





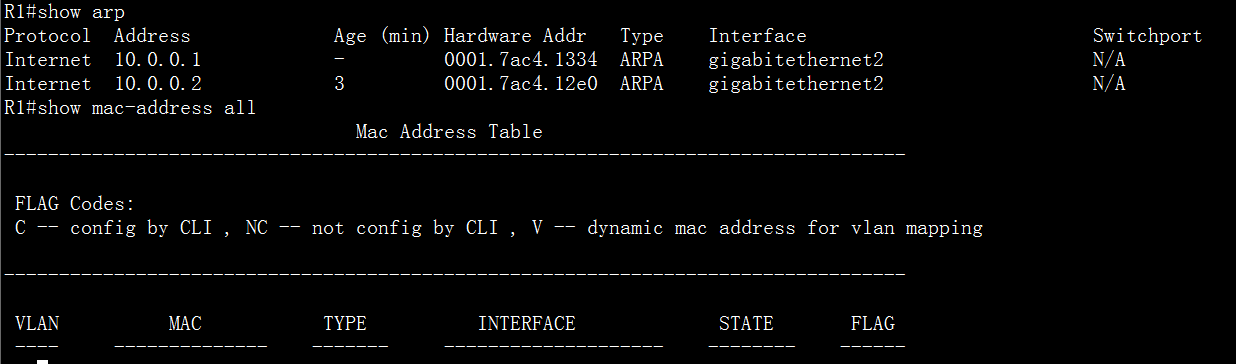
可以看到只有其相应本地接口的Mac地址和路由。其中ARP表项由于此时PC1和PC2相应接口均处于down状态，所以学习不到PC1和PC2的相关信息。但与对端R1所连接口处于up,所以ARP可以解析到R1的mac地址。

SW3：



没有任何流量通过，所以为空表

R1：



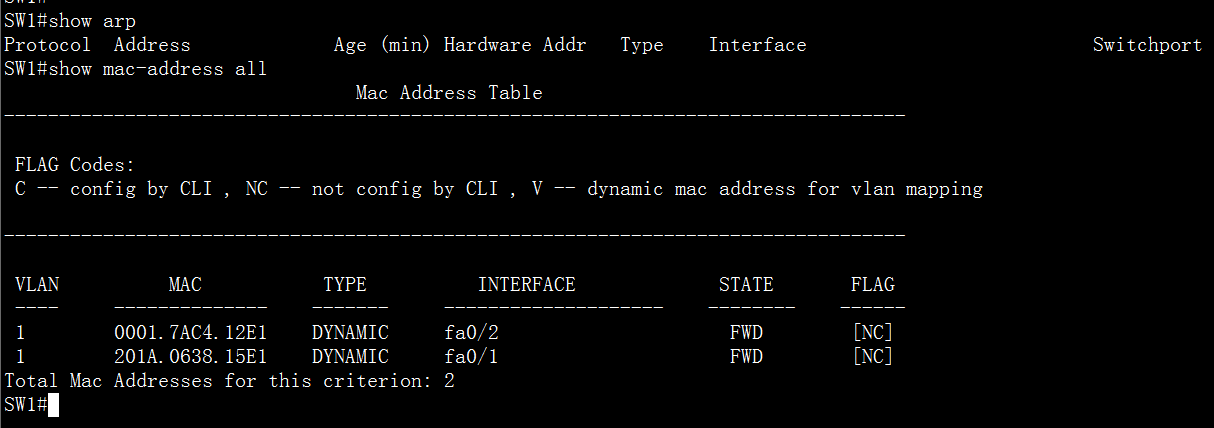
（这里需down掉R1的g1接口）与SW2的连接口up，ARP能解析到SW2的接口地址。

**现在我们在PC1上ping主机PC2**。然后观察相应设备的三张表。



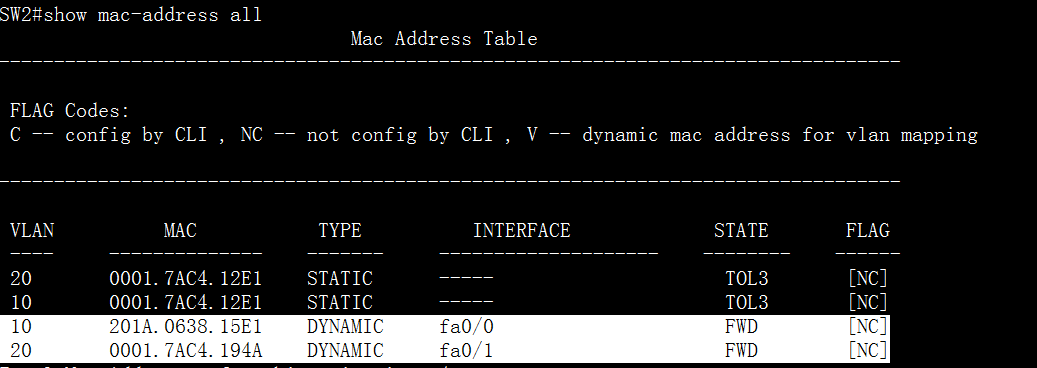
在PC1上ping 通PC2后，可以看到，主机将数据包交给网关后，网关由于默认启用了代理ARP，所以PC1学到了网关的 Mac地址00-01-7a-c4-12-e1。

SW1：

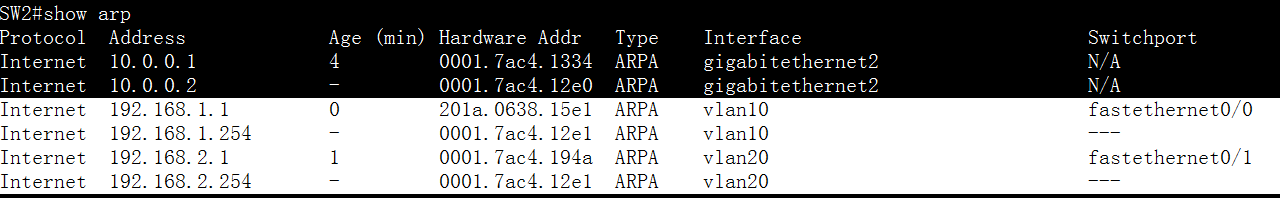


当数据包第一次从PC1经SW1的fa0/1口进入，交换机将PC1的Mac地址对应到fa0/1口，后续若有数据帧要发往PC1，则通过查找其Mac地址对应的接口发送数据帧。而PC2在给PC1回包的过程中，通过SW2发往SW1，再回到PC1，交换机SW1将学习到SW2对应的Mac地址，并将其添加到对应的Mac地址表中。二层设备转发依据的是Mac地址表，所以没有ARP信息和路由信息。

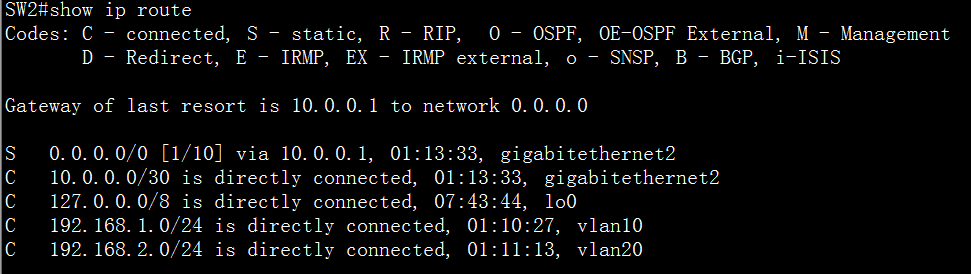
SW2：



这里的三层设备使用的是SVI接口，所以可以看到在SW2上也学习到了PC1的Mac地址信息和PC2的Mac地址信息，并分别对应到相应的接口。

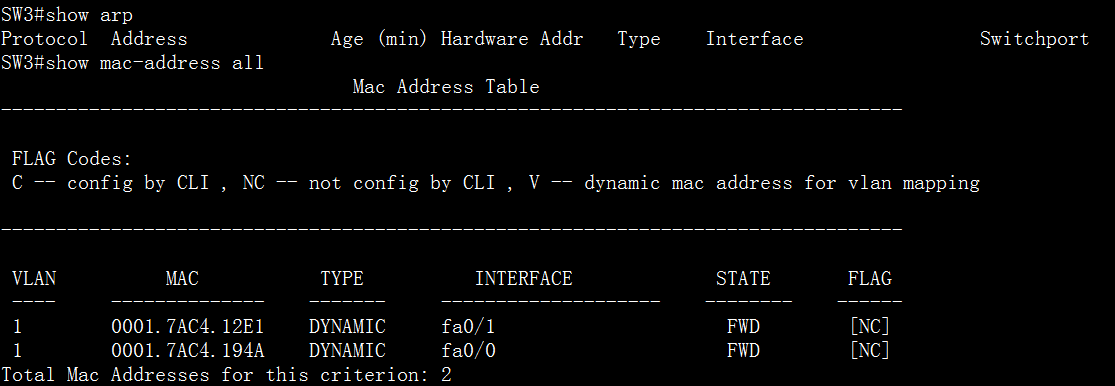


以上ARP表项表明了相应主机对应各自的Mac地址及其数据出接口。



三层设备的转发依靠的是查找路由表，路由器收到一个数据包时，首先会在其路由表中查找是否有相应的路由信息，如果没有，则直接丢弃该报文，如果有路由信息，则查看其相应的路由条目对应的接口，二层目的Mac地址的封装不变，然后通过接口发送出去。

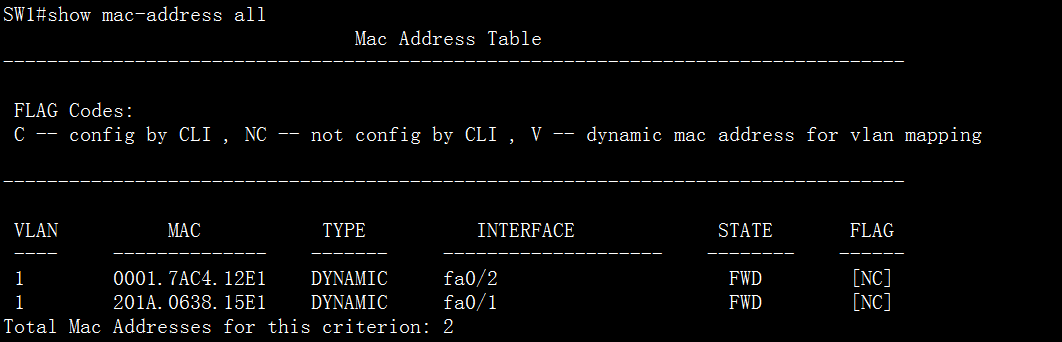
SW3：



交换机SW3学习到了PC2的Mac地址并对应到Fa0/0。学习到了三层设备SW2 Mac地址且对应到fa0/1。

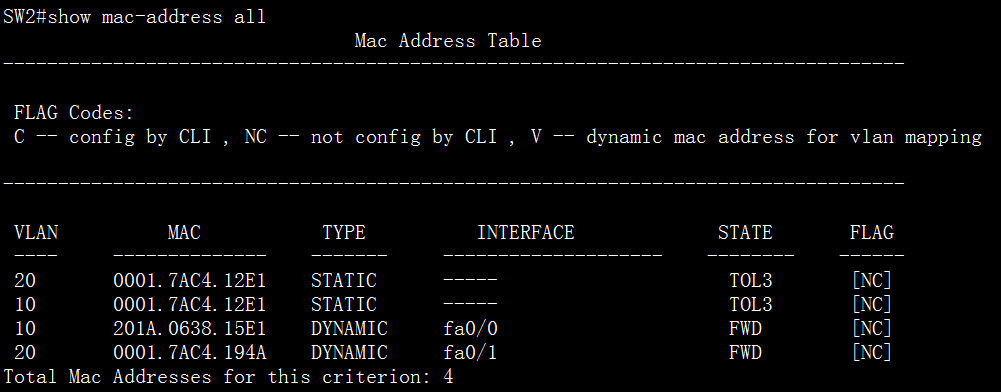
**再从PC1上ping PC3。**

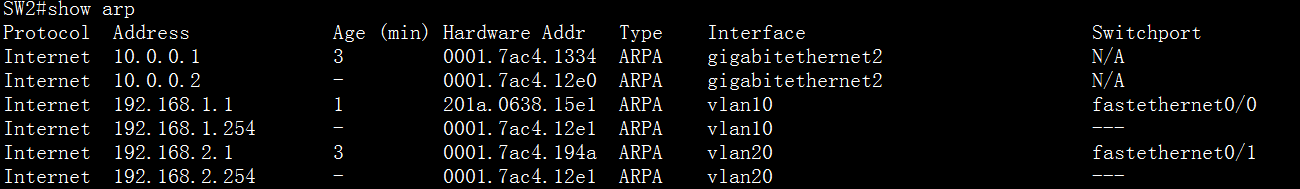
SW1：

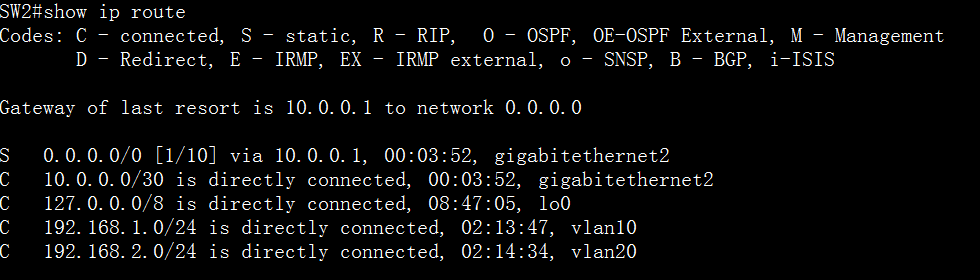


由于在PC1上设置的网关是SW2，即去往任何网段的路由都是首先交给网关进行ARP代理，因此，在SW1上的Mac地址表不变。

SW2：

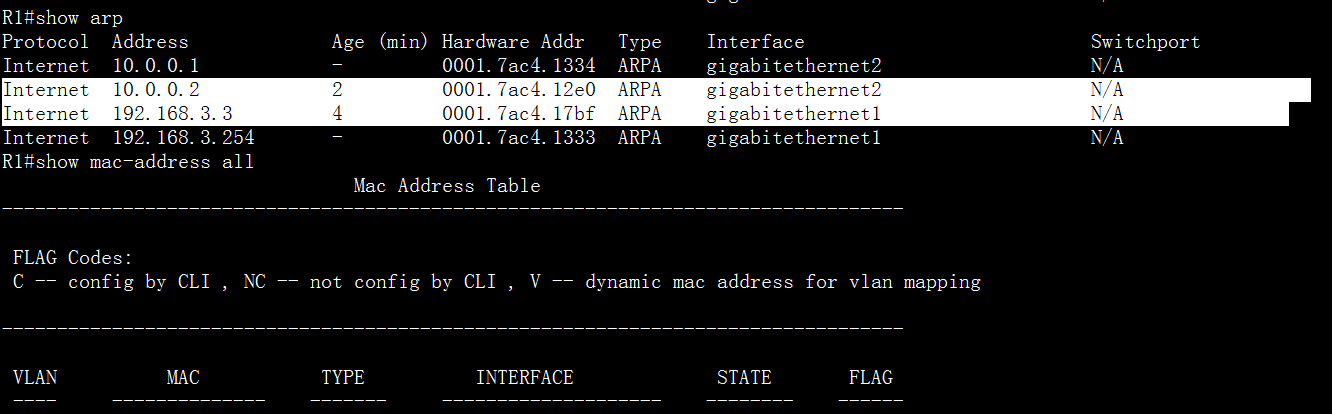


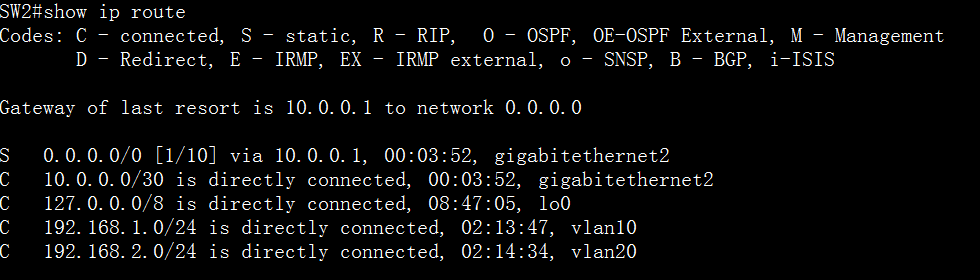




可以看到在SW2中的mac地址并没有关于PC3的有关信息，即三层设备间的转发是依靠路由表来进行。

R1：





可以看到，在路由器R1上的ARP表中有着PC3的ip地址对应的Mac地址，但其三层接口并不会学习有关mac地址信息。同样转发数据是通过路由表来进行。

## DHCP实验

实验拓扑如图

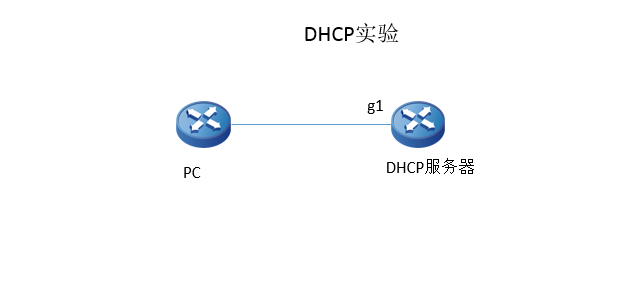


图 1

实验要求：

PC能够从DCHP服务器自动获取到ip地址

DHCP服务器配置如下：

R2#config terminal

R2(config)#interface g1

R2(config-if-gigabitethernet1)#ip address 10.1.1.2 24 //配置g1口ip地址

R2(config)#ip dhcp service //启动dhcp服务

R2(config)#ip dhcp pool p1 //设置dhcp地址池

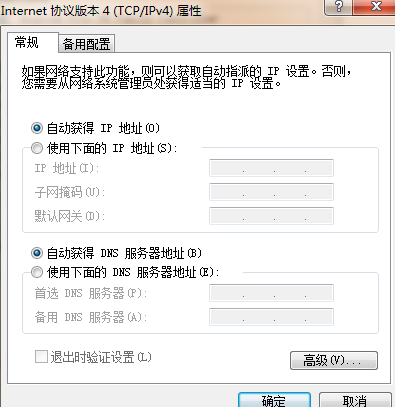
R2(dhcp-config)#range 10.1.1.10 10.1.1.100 255.255.255.0 **//设置dhcp池地址范围**

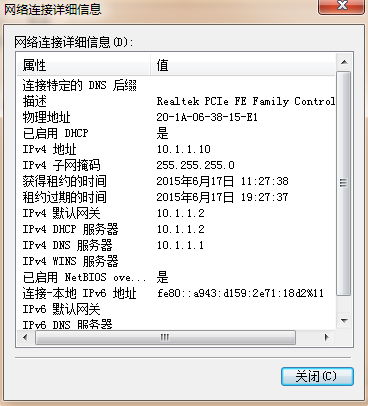
R2(dhcp-config)#default-router 10.1.1.254 //配置默认网关

R2(dhcp-config)#dns-server 10.1.1.1 //设置DNS服务器地址

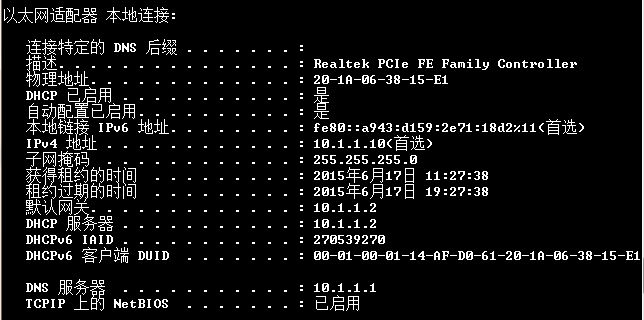
R2(dhcp-config)#lease 0 8 //设定dhcp地址租约期8小时

在PC端开启自动获取地址测试是否成功获取到了地址。





在PC机上打开cmd。输入ipconfig/all可以看到如下信息：



由上图可知PC端获取到了ip地址10.1.1.10。

# 第二章 路由实验



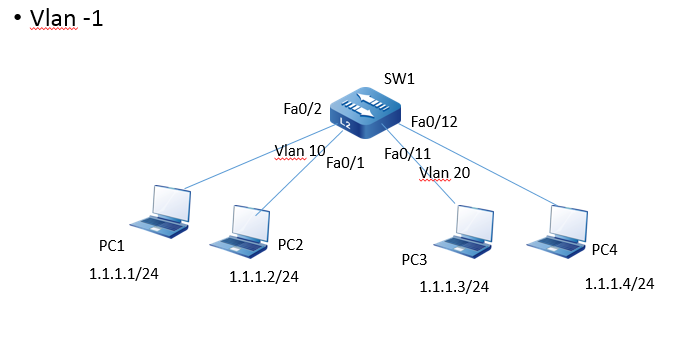
## VLAN实验

### 不同vlan相互隔离

实验目的：

验证不同vlan之间不能互访。

简单拓扑如下：



fa0/2和fa0/1属于vlan10。Fa0/11和fa0/12同属vlan20

相关配置如下：

S1(config)#vlan 10

S1(config-vlan10)#exit

S1(config)#vlan 20

S1(config-vlan20)#exit

S1(config)#interface fastethernet 0/1-0/2

S1(config-if-range)**#switchport mode access //设置接口模式为access**

S1(config-if-range)#**switchport access vlan 10 //将该接口划分进vlan10**

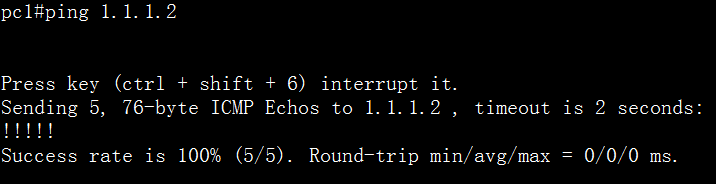
S1(config)#interface fastethernet 0/11-0/12

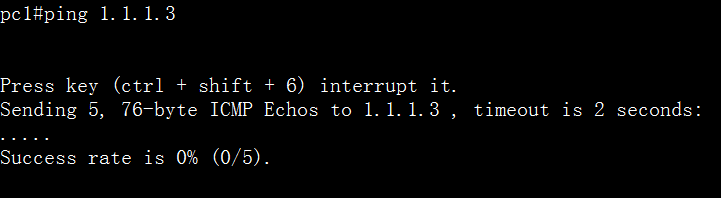
S1(config-if-range)#switchport mode access

S1(config-if-range)#**switchport access vlan 20 //将该接口划分进vlan20**

验证结果：

在PC1上ping PC2能通，PC1 ping PC3不能通

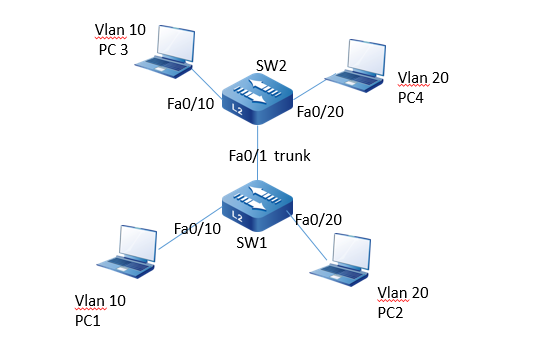




### 相同vlan互通

实验目的：验证相同vlan之间能互通，不同vlan之间不能互通。

网络拓扑如下：



PC1-PC4地址分别设为1.1.1.1、1.1.1.2、1.1.1.3、1.1.1.4

SW1配置如下：

S1(config)#vlan 10,20 //创建vlan 10,vlan 20

S1(config)#interface fastethernet 0/1

S1(config-if-fastethernet0/1)#switchport mode trunk //设置接口为trunk模式

S1(config-if-fastethernet0/1)#switchport trunk allowed vlan add 10,20

**//接口trunk模式允许vlan 10，vlan 20 通过。**

S1(config)#interface fa0/10

S1(config-if-fastethernet0/10)#switchport mode access

S1(config-if-fastethernet0/10)#switchport access vlan 10

//接口划分到vlan10

S1(config)#interface fa0/20

S1(config-if-fastethernet0/20)#switchport mode access

S1(config-if-fastethernet0/20)#switchport access vlan 20

SW2配置：

SW2(config)#vlan 10,20

SW2(config)#interface fa0/1

SW2(config-if-fastethernet0/1)#switchport mode trunk //设置接口为trunk模式

SW2(config-if-fastethernet0/1)#switchport trunk allowed vlan add 10,20

//trunk模式允许vlan10，vlan20通过

SW2(config)#interface fa0/10

SW2(config-if-fastethernet0/10)#switchport mode access

SW2(config-if-fastethernet0/10)#switchport access vlan 10

SW2(config)#interface fa0/20

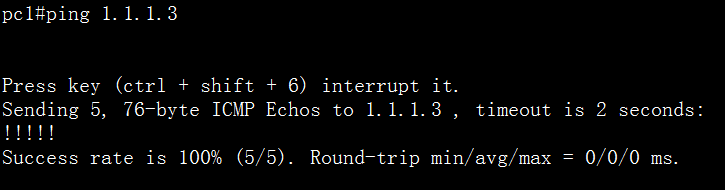
SW2(config-if-fastethernet0/20)#switchport mode access

SW2(config-if-fastethernet0/20)#switchport access vlan 20

PC1-PC4均由路由器模拟

在同一vlan10里PC1 ping PC3

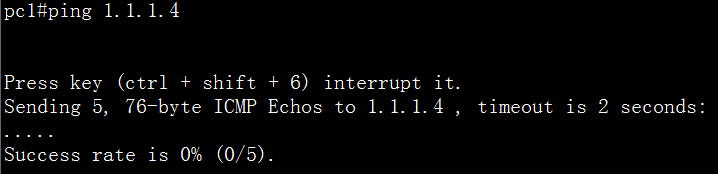
pc1#ping 1.1.1.3



可以看到，相同vlan之间能够互通

PC1 Ping PC4:

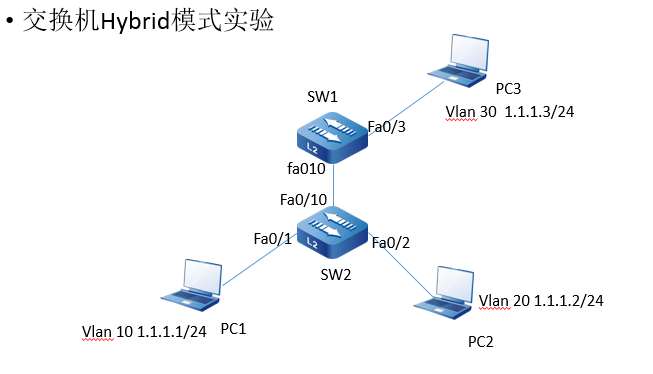
pc1#ping 1.1.1.4



可以看到一般情况下不同vlan之间一般情况下是不能直接相互通信的。

### Hybrid模式

实验拓扑如图：



实验要求：通过Hybrid端口模式实现不同vlan之间的互访和隔离。PC1和PC2之间不能互访，但PC1、PC2均能互PC3相互通信。地址规划如图所示。

（1）给各个PC机配上ip地址。略

（2）两台4120交换机配置如下

S1(config)#vlan 10,20,30 //创建vlan10,20,30

S1(config)#interface fa0/3

S1(config-if-fastethernet0/3)#switchport mode hybrid

**//设置端口模式为hybrid模式**

S1(config-if-fastethernet0/3)#***switchport hybrid pvid vlan 30***

**//设置端口pvid为vlan30**

S1(config-if-fastethernet0/3)#***switchport hybrid untagged vlan 10,20,30***

**\*\*//剥离vlan10，vlan20的标签后发送给PC，并允许vlan10,20,30通过**

S1(config-if-fastethernet0/3)#exit

S1(config)#interface fa0/10

S1(config-if-fastethernet0/10)#switchport mode hybrid

S1(config-if-fastethernet0/10)#**switchport hybrid tagged vlan 10,20,30**

**\*\*//允许vlan10,20,30通过**

SW2(config)#vlan 10,20,30

SW2(config)#interface fa0/1

SW2(config-if-fastethernet0/1)#switchport mode hybrid

SW2(config-if-fastethernet0/1)#***switchport hybrid pvid vlan 10***

//设置该端口的pvid为vlan10

SW2(config-if-fastethernet0/1)#**switchport hybrid untagged vlan 10**

**//允许vlan10通过，亦可用tagged vlan10命令。**

SW2(config-if-fastethernet0/1)#**switchport hybrid untagged vlan 30**

**//剥离vlan30标签后发送**

SW2(config-if-fastethernet0/1)#eixt

SW2(config)#interface fa0/2

SW2(config-if-fastethernet0/2)#switchport mode hybrid

SW2(config-if-fastethernet0/2)#switchport hybrid pvid vlan 20

SW2(config-if-fastethernet0/2)#switchport hybrid untagged vlan 20

SW2(config-if-fastethernet0/2)#switchport hybrid untagged vlan 30

SW2(config-if-fastethernet0/2)#exit

SW2(config)#interface fa0/10

SW2(config-if-fastethernet0/10)#switchport mode hybrid

SW2(config-if-fastethernet0/10)#switchport hybrid tagged vlan 10,20,30

（3）验证

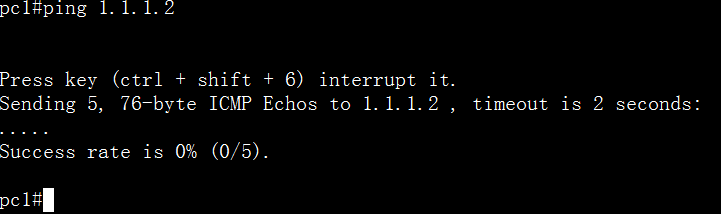
在PC3上分别ping PC1和PC2，看其能否相互通信。





PC3能够成功的与PC1和PC2相互通信。

在PC1上ping PC2

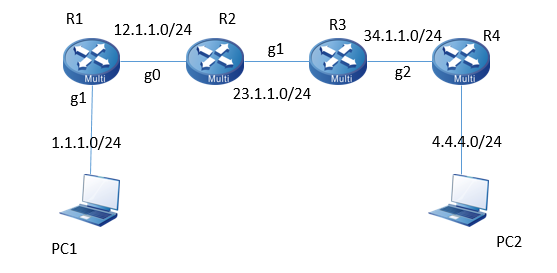


可以看到PC1和PC2不能相互通信。

实验总结：说明在Hybrid模式下，不同vlan之间能够实现灵活的相互访问和相互隔离。

## 路由基础

### 静态路由



实验要求：

配置静态路由，实现PC1能访问PC2。

注：在PC1上需设置默认网关。关闭防火墙。PC2在R4上使用环回接口模拟。

（1）各设备接口配置

R1(config)#interface gigabitethernet 0

R1(config-if-gigabitethernet0)#ip address 12.1.1.1 24

R1(config-if-gigabitethernet0)#exit

R1(config)#interface g1

R1(config-if-gigabitethernet1)#ip address 1.1.1.1 24

R1(config-if-gigabitethernet1)#exit

R2(config)#interface g0

R2(config-if-gigabitethernet0)#ip address 12.1.1.2 24

R2(config-if-gigabitethernet0)#exit

R2(config)#interface g1

R2(config-if-gigabitethernet1)#ip address 23.1.1.2 24

R2(config-if-gigabitethernet1)#exit

R3(config)#interface g1

R3(config-if-gigabitethernet1)#ip address 23.1.1.3 24

R3(config-if-gigabitethernet1)#exit

R3(config)#interface g2

R3(config-if-gigabitethernet2)#ip address 34.1.1.3 24

R3(config-if-gigabitethernet2)#exit

R4(config)#interface g2

R4(config-if-gigabitethernet2)#ip address 34.1.1.4 24

R4(config-if-gigabitethernet2)#exit

R4(config)#interface loopback0

R4(config-if-loopback0)#ip address 4.4.4.4 24

R4(config-if-loopback0)#exit

（2）配置静态路由

R1(config)#ip route 0.0.0.0 0.0.0.0 12.1.1.2 //根据本实验拓扑的特点，在R1上只需要配置一条指向12.1.1.2的缺省路由

R2(config)#ip route 1.1.1.0 255.255.255.0 12.1.1.1

R2(config)#ip route 34.1.1.0 255.255.255.0 23.1.1.3

R2(config)#ip route 4.4.4.0 255.255.255.0 23.1.1.3

R3(config)#ip route 1.1.1.0 255.255.255.0 23.1.1.2

R3(config)#ip route 12.1.1.0 255.255.255.0 23.1.1.2

R3(config)#ip route 4.4.4.0 255.255.255.0 34.1.1.4

R4(config)#ip route 0.0.0.0 0.0.0.0 34.1.1.3





可以看到PC端路由下一跳为1.1.1.1。

C:\Users\Administrator>ping 4.4.4.4

正在 Ping 4.4.4.4 具有 32 字节的数据:

来自 4.4.4.4 的回复: 字节=32 时间<1ms TTL=252

来自 4.4.4.4 的回复: 字节=32 时间<1ms TTL=252

来自 4.4.4.4 的回复: 字节=32 时间<1ms TTL=252

来自 4.4.4.4 的回复: 字节=32 时间<1ms TTL=252

4.4.4.4 的 Ping 统计信息:

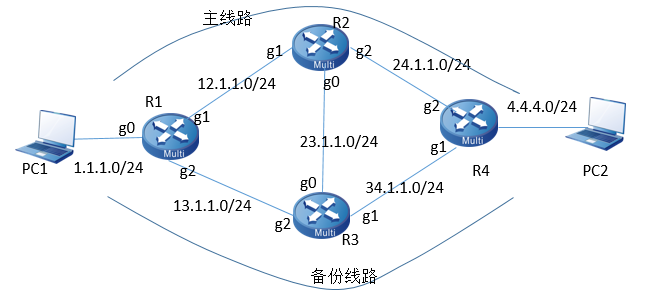
数据包: 已发送 = 4，已接收 = 4，丢失 = 0 (0% 丢失)，

往返行程的估计时间(以毫秒为单位):

最短 = 0ms，最长 = 0ms，平均 = 0ms

### 静态浮动路由实现主备

实验拓扑：



实验要求：PC1和PC2通过静态浮动路由实现主备互访。

（1）设备接口地址配置

R1(config)#interface g0

R1(config-if-gigabitethernet0)#ip address 1.1.1.1 24

R1(config-if-gigabitethernet0)#exit

R1(config)#interface g1

R1(config-if-gigabitethernet1)#ip address 12.1.1.1 24

R1(config-if-gigabitethernet1)#exit

R1(config)#interface g2

R1(config-if-gigabitethernet2)#ip address 13.1.1.1 24

R1(config-if-gigabitethernet2)#exit

R2(config)#interface g1

R2(config-if-gigabitethernet1)#ip address 12.1.1.2 24

R2(config-if-gigabitethernet1)#exit

R2(config)#interface g2

R2(config-if-gigabitethernet2)#ip address 24.1.1.2 24

R2(config-if-gigabitethernet2)#exit

R2(config)#interface g0

R2(config-if-gigabitethernet0)#ip address 23.1.1.2 24

R2(config-if-gigabitethernet0)#exit

R3(config)#interface g0

R3(config-if-gigabitethernet0)#ip address 23.1.1.3 24

R3(config-if-gigabitethernet0)#exit

R3(config)#interface g1

R3(config-if-gigabitethernet1)#ip address 34.1.1.3 24

R3(config-if-gigabitethernet1)#exit

R3(config)#interface g2

R3(config-if-gigabitethernet2)#ip address 13.1.1.3 24

R3(config-if-gigabitethernet2)#exit

R4(config)#interface g2

R4(config-if-gigabitethernet2)#ip address 24.1.1.4 24

R4(config-if-gigabitethernet2)#exit

R4(config)#interface g1

R4(config-if-gigabitethernet1)#ip address 34.1.1.4 24

R4(config-if-gigabitethernet1)#exit

R4(config)#interface loopback0

R4(config-if-loopback0)#ip address 4.4.4.4 24

R4(config-if-loopback0)#exit

（2）设置浮动静态路由实现路由备份。

R1(config)#ip route 4.4.4.0 255.255.255.0 12.1.1.2

//到达4.4.4.0网段的下一跳指向12.1.1.2

R1(config)#**ip route 4.4.4.0 255.255.255.0 13.1.1.3 5**

**\*\*//通过设置静态路由的AD值为5来实现配置浮动静态路由到达4.4.4.0网段的下一跳指向13.1.1.3**

R1(config)#ip route 23.1.1.0 255.255.255.0 12.1.1.2

R1(config)#ip route 24.1.1.0 255.255.255.0 12.1.1.2

R1(config)#ip route 34.1.1.0 255.255.255.0 12.1.1.2

R1(config)#ip route 34.1.1.0 255.255.255.0 13.1.1.3 5

R2(config)#ip route 1.1.1.0 255.255.255.0 12.1.1.1

R2(config)#ip route 1.1.1.0 255.255.255.0 23.1.1.3 5

R2(config)#ip route 4.4.4.0 255.255.255.0 24.1.1.4

R2(config)#ip route 4.4.4.0 255.255.255.0 23.1.1.3 5

R2(config)#ip route 13.1.1.0 255.255.255.0 23.1.1.3

R2(config)#ip route 34.1.1.0 255.255.255.0 23.1.1.3

R3(config)#ip route 1.1.1.0 255.255.255.0 13.1.1.1

R3(config)#ip route 4.4.4.0 255.255.255.0 34.1.1.4

R4(config)#ip route 1.1.1.0 255.255.255.0 24.1.1.2

R4(config)#ip route 1.1.1.0 255.255.255.0 34.1.1.3 5

R4(config)#ip route 12.1.1.0 255.255.255.0 24.1.1.2

R4(config)#ip route 13.1.1.0 255.255.255.0 24.1.1.2

R4(config)#ip route 13.1.1.0 255.255.255.0 34.1.1.3 5

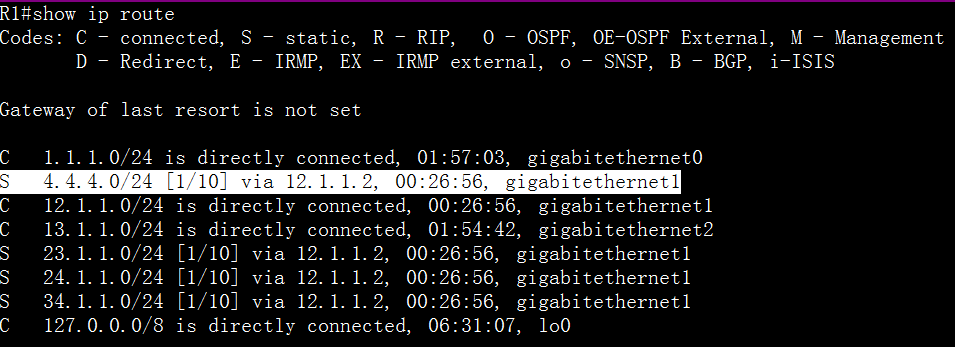
R4(config)#ip route 23.1.1.0 255.255.255.0 24.1.1.2

**验证：**

在PC1上ping 4.4.4.4



查看R1的路由表：



可以看到正常情况下从PC1到达PC2的路径只有一条下一跳为R2的12.1.1.2接口的路由。现在我们down掉主线路R1的g1接口。

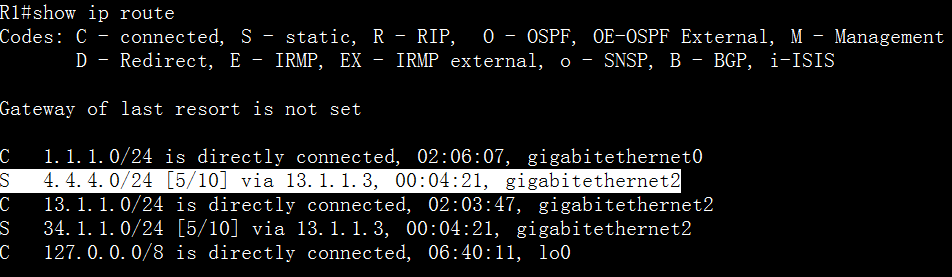


再次ping 4.4.4.4看能否通



仍然能够与PC2通讯。

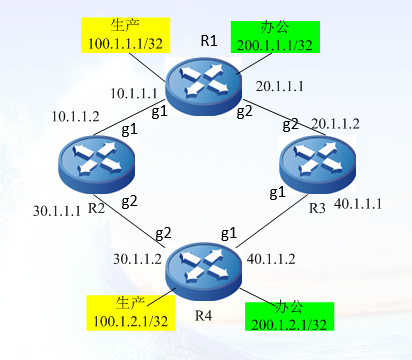
查看R1路由表，看PC1到达PC2的下一跳是否发生变化。



可以看到当R1的g1接口down掉以后，PC1到达PC2的下一跳路由切换到了备份路径下一跳13.1.1.3。

### 静态浮动路由分流互备

实验拓扑：



实验要求：

R1为中心，R4为网点，分别有生产和办公两个业务，要实现生产数据走左边，办公数据走右边。同时要两条线路要实现互备。使用静态浮动路由实现。

浮动静态路由：通过设置静态路由的管理距离值，实现当主路由down掉后，备份路由自动切换。

实验配置：

R1(config)#interface loop1

R1(config-if-loopback1)#ip address 100.1.1.1 32

R1(config-if-loopback1)#exit

R1(config)#interface loop2

R1(config-if-loopback2)#ip address 200.1.1.1 32

R1(config-if-loopback2)#exit

R1(config)#interface g1

R1(config-if-gigabitethernet1)#ip address 10.1.1.1 24

R1(config-if-gigabitethernet1)#exit

R1(config)#interface g2

R1(config-if-gigabitethernet2)#ip address 20.1.1.1 24

R1(config-if-gigabitethernet2)#exit

R1(config)#ip route 30.1.1.0 255.255.255.0 10.1.1.2

R1(config)#ip route 40.1.1.0 255.255.255.0 20.1.1.2

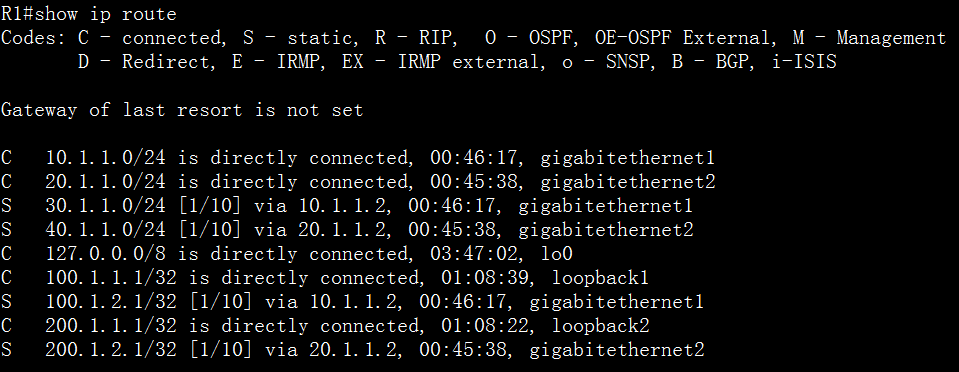
R1(config)#ip route 100.1.2.1 255.255.255.255 10.1.1.2

R1(config)#ip route 100.1.2.1 255.255.255.255 20.1.1.2 5 **//通过修改静态路由管理距离值实现静态浮动路由备份。**

R1(config)#ip route 200.1.2.1 255.255.255.255 20.1.1.2

R1(config)#ip route 200.1.2.1 255.255.255.255 10.1.1.2 5

R1路由表：



从R1的路由表中可以看到正常情况下，生产网段**100.1.0.0**和办公网段**200.1.0.0**均只有一条路由，且指向不同的下一跳。实现了分流。

R2#config t

R2(config)#interface g1

R2(config-if-gigabitethernet1)#ip address 10.1.1.2 24

R2(config-if-gigabitethernet1)#exit

R2(config)#interface g2

R2(config-if-gigabitethernet2)#ip address 30.1.1.1 24

R2(config-if-gigabitethernet2)#exit

R2(config)#ip route 100.1.2.1 255.255.255.255 30.1.1.2

R2(config)#ip route 100.1.1.1 255.255.255.255 10.1.1.1

R2(config)#ip route 200.1.2.1 255.255.255.255 30.1.1.2

R2(config)#ip route 200.1.1.1 255.255.255.255 10.1.1.1

R3(config)#interface g2

R3(config-if-gigabitethernet2)#ip address 20.1.1.2 24

R3(config-if-gigabitethernet2)#exit

R3(config)#interface g1

R3(config-if-gigabitethernet1)#ip address 40.1.1.1 24

R3(config-if-gigabitethernet1)#exit

R3(config)#ip route 200.1.2.1 255.255.255.255 40.1.1.2

R3(config)#ip route 200.1.1.1 255.255.255.255 20.1.1.1

R3(config)#ip route 100.1.1.1 255.255.255.255 20.1.1.1

R3(config)#ip route 100.1.2.1 255.255.255.255 40.1.1.2

R4(config)#interface loop1

R4(config-if-loopback1)#ip address 100.1.2.1 32

R4(config-if-loopback1)#exit

R4(config)#interface loop2

R4(config-if-loopback2)#ip address 200.1.2.1 32

R4(config-if-loopback2)#

R4(config-if-loopback2)#exit

R4(config)#interface g2

R4(config-if-gigabitethernet2)#ip address 30.1.1.2 24

R4(config-if-gigabitethernet2)#exit

R4(config)#interface g1

R4(config-if-gigabitethernet1)#ip address 40.1.1.2 24

R4(config-if-gigabitethernet1)#exit

R4(config)#ip route 100.1.1.1 255.255.255.255 30.1.1.1

R4(config)#ip route 100.1.1.1 255.255.255.255 40.1.1.1 5

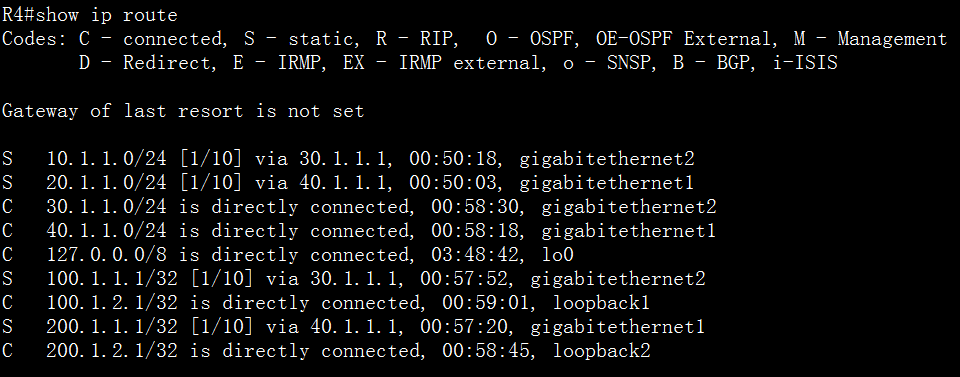
R4(config)#ip route 200.1.1.1 255.255.255.255 40.1.1.1

R4(config)#ip route 200.1.1.1 255.255.255.255 30.1.1.1 5

R4(config)#ip route 10.1.1.0 255.255.255.0 30.1.1.1

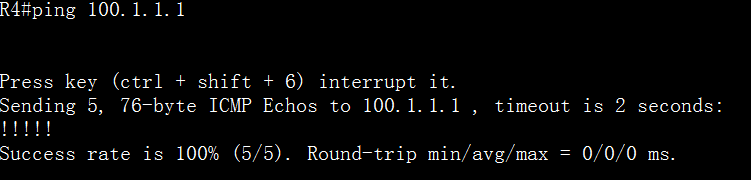
R4(config)#ip route 20.1.1.0 255.255.255.0 40.1.1.1

R4路由表：



可以看到正常情况下生产网10网段的路由R4的下一跳为30.1.1.1。而办公网20网段下一跳为40.1.1.1

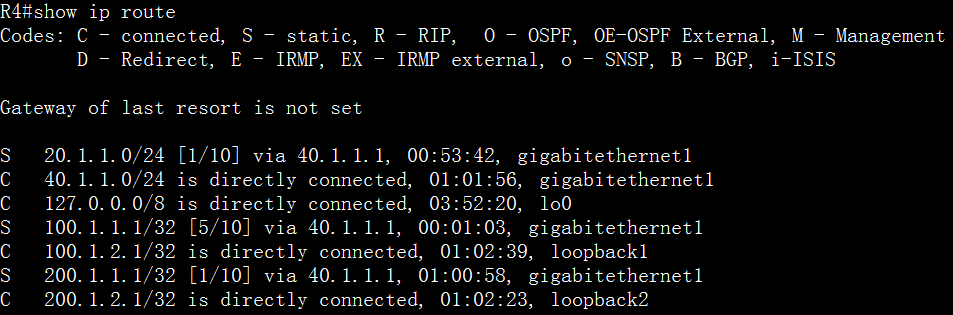
在R4上ping R1的环回口



现在down掉R4的g2口



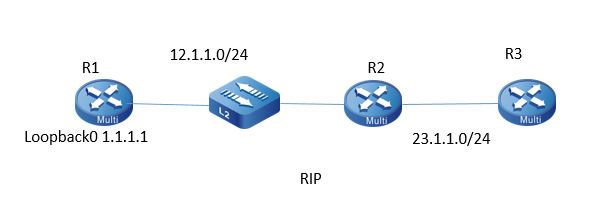
查看其路由表是否发生了变化



到达R1的路由100.1.1.1的出接口切换到了g1口，其下一跳切换到了R3的**40.1.1.1**接口。

## 有关RIP路由协议及其计时器问题讨论

实验拓扑如图：



Rip的四个计时器：

Update：更新计时器，指运行RIP协议的路由器向所有接口广播自己的全部路由表的时间间隔。默认30秒。

Holddown Timer：抑制计时器，路由器如果在相同的接口上收到某个路由条目的距离比原先收到的距离大，那么将启动一个抑制计时器。在抑制计时器的时间内该目的不可到达

Invalid Timer：无效计时器。路由器每收到一次路由条目就把无效计器置0，也就是说路由条目每隔无效计时器规定的时间内必须收到路由条目更新报文。如果没有收到相关条目的更新保文那么无效计时器超时，路由器就认为该目的不可达，并向所有接口广播不可达更新报文。无效计时器缺省是180秒。

Flush Timer：刷新计时器。

1. 如果在刷新时间内没有收到更新报文，那么该目的的路由条目将被刷掉也就是直接删除。
2. 如果在刷新时间内收到更新报文，那么该目的的路由条目的刷新计时器被刷新置0。

（1）以下是路由器rip协议基本配置。

R1(config)#interface g1

R1(config-if-gigabitethernet1)#ip address 12.1.1.1 24

R1(config-if-gigabitethernet1)#exit

R1(config)#interface loopback0

R1(config-if-loopback0)#ip address 1.1.1.1 24

R1(config-if-loopback0)#exit

R1(config)#**router rip** //启用rip协议进程

R1(config-rip)#no auto-summary //关闭自动汇总

R1(config-rip)#version 2 //设置rip版本号

R1(config-rip)#network 12.0.0.0 //宣告主类网络号

R1(config-rip)#network 1.0.0.0

R2(config)#interface g1

R2(config-if-gigabitethernet1)#ip address 12.1.1.2 24

R2(config-if-gigabitethernet1)#exit

R2(config)#interface g2

R2(config-if-gigabitethernet2)#ip address 23.1.1.2 24

R2(config-if-gigabitethernet2)#exit

R2(config)#router rip

R2(config-rip)#no auto-summary

R2(config-rip)#version 2

R2(config-rip)#network 12.0.0.0

R2(config-rip)#network 23.0.0.0

R5(config)#hostname R3

R3(config)#interface g2

R3(config-if-gigabitethernet2)#ip address 23.1.1.3 24

R3(config-if-gigabitethernet2)#exit

R3(config)#router rip

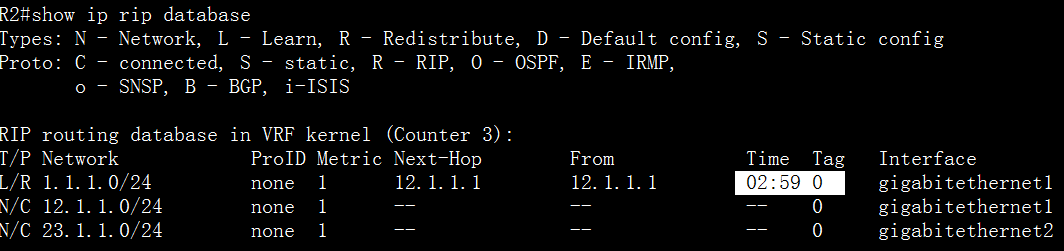
R3(config-rip)#no auto-summary

R3(config-rip)#version 2

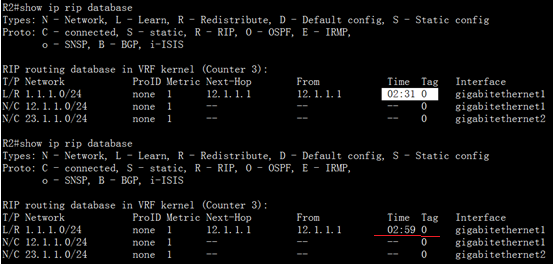
R3(config-rip)#network 23.0.0.0

（2）讨论验证

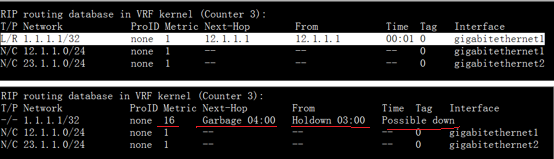
我们先看更新计时器更新时间：



通过使用show ip rip database这条命令，我们可以看到倒计时从2分59秒开始

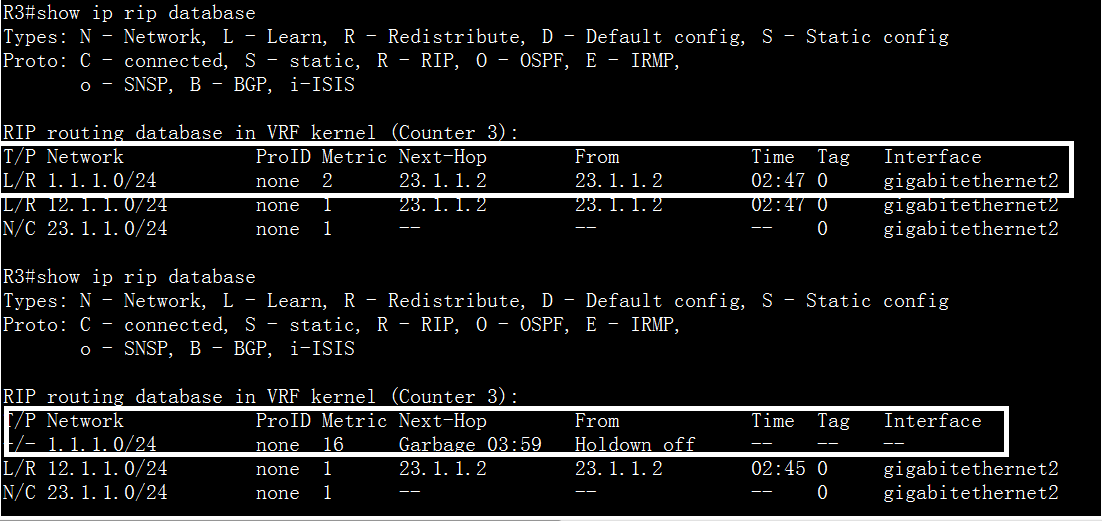


当倒计时到达2分31秒的时候，R2收到了来自R1的关于1.1.1.0/24路由条目的更新，此时，倒计时重置到3分开始下一轮倒计时。此间更新计时器历时28秒左右。此时我们通过立即拔掉R1的g1接口，来观察Invalid Timer。

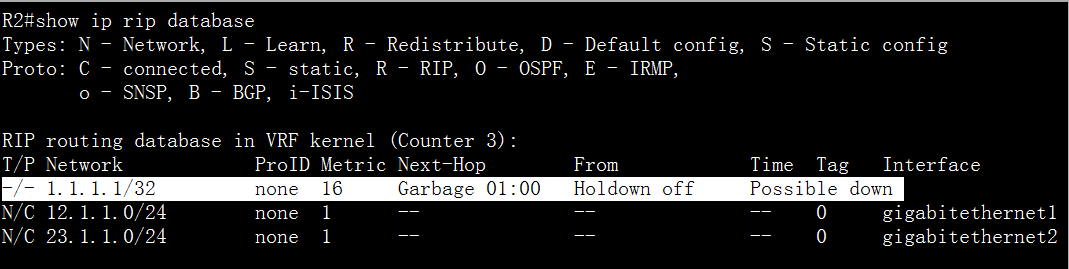


从上图可以看出，当立即拔掉R1的g1接口后，无效计时器和更新计时器同时启动，当倒计时从3分钟倒计到0时，此时R2还没有收到R1关于1.1.1.0/24的路由更新，此时无效计时器状态置为possible down，并且将其关于1.1.1.0/24的路由跳数置为16跳（不可达）。刷新计时器240秒（也称为垃圾计时器）和抑制计时器180秒立即同时启动。

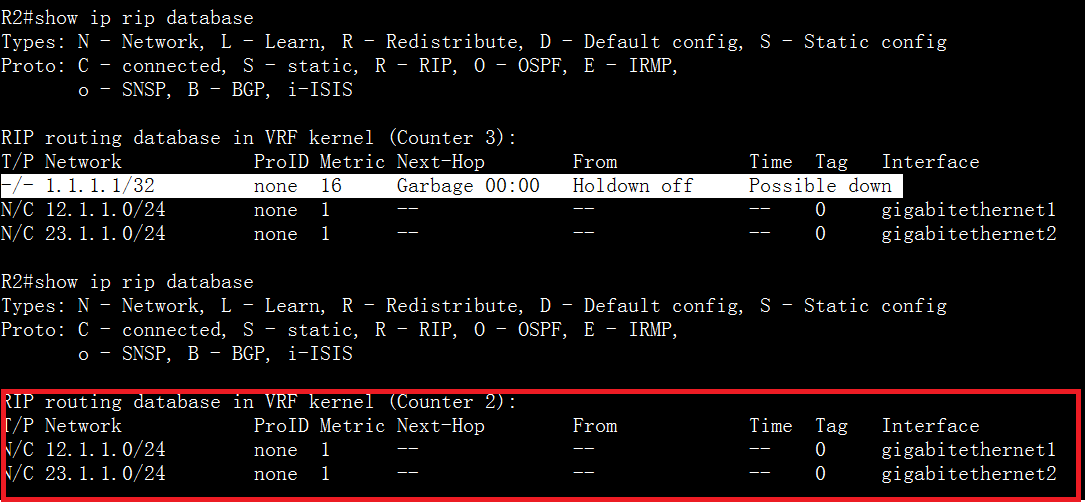
再看R3的关于1.1.1.0/24的更新情况。



我们可以看到，R3在R2的抑制计时器超时前，R2仍然会发送关于1.1.1.0/24的路由条目更新给R3。而只有当R2的抑制计时器超时后，R2才会向全网广播一条关于R1的1.1.1.0/24路由条目不可达的消息。此时R3收到关于1.1.1.0/24不可达的消息后，R3将该路由条目跳数置为16跳不可达。并同时启动刷新计时器240秒。



当R2抑制计时器倒计时为0后，R2仍然没有收到关于R1的 1.1.1.0/24的路由更新。抑制计时器状态关闭。路由仍然被标记为不可达。刷新计时器继续倒计时。

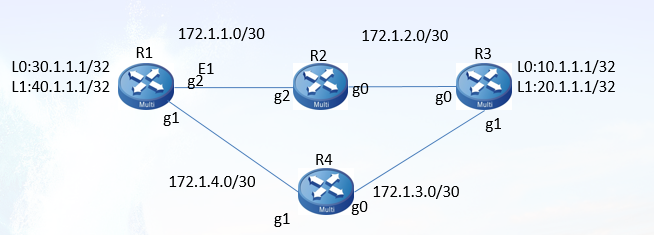


当刷新计时器倒计时到0时，R2仍然没有收到关于1.1.1.0/24的更新，此时，1.1.1.0/24路由条目从R2的数据库删除。

以上就是关于Rip协议计时器的讨论。

## 分发列表、偏移列表实现Rip跳数选路

实验拓扑如图所示：



拓扑如图所示，R1与R2通过E1链路互联，全网运行rip协议

实验要求：

1. R1通过E1线路只能学习到对端10网段，且路由跳数为5；

2. R1通过g1口只能学习到对端20网段，且路由跳数为6；

3. R3通过g0口只能学习到对端30网段，且路由跳数为5；

3. R3通过g1口只能学习到对端40网段，且路由跳数为6；

实验配置：

（1）各设备接口地址配置

R1：

R1(config)#interface loopback 0

R1(config-if-loopback0)#ip address 30.1.1.1 32

R1(config-if-loopback0)#exit

R1(config)#interface loopback 1

R1(config-if-loopback1)#ip address 40.1.1.1 32

R1(config)#interface gigabitethernet 2

R1(config-if-gigabitethernet2)#ip address 172.1.1.1 30

R1(config-if-gigabitethernet2)#exit

R1(config)#interface gigabitethernet 1

R1(config-if-gigabitethernet1)#ip address 172.1.4.1 30

R2：

R2(config)#interface gigabitethernet 2

R2(config-if-gigabitethernet2)#ip address 172.1.1.2 30

R2(config-if-gigabitethernet2)#exit

R2(config)#interface gigabitethernet 0

R2(config-if-gigabitethernet0)#ip address 172.1.2.2 30

R3：

R3(config)#interface loopback 0

R3(config-if-loopback0)#ip address 10.1.1.1 32

R3(config-if-loopback0)#exit

R3(config)#interface loopback 1

R3(config-if-loopback1)#ip address 20.1.1.1 32

R3(config-if-loopback1)#exit

R3(config)#interface gigabitethernet 0

R3(config-if-gigabitethernet0)#ip address 172.1.2.1 30

R3(config-if-gigabitethernet0)#exit

R3(config)#interface gigabitethernet 1

R3(config-if-gigabitethernet1)#ip address 172.1.3.1 30

R3(config-if-gigabitethernet1)#exit

R4：

R4(config)#interface gigabitethernet 1

R4(config-if-gigabitethernet1)#ip address 172.1.4.2 30

R4(config-if-gigabitethernet1)#exit

R4(config)#interface gigabitethernet 0

R4(config-if-gigabitethernet0)#ip address 172.1.3.2 30

R4(config-if-gigabitethernet0)#exit

**（2）使用分发列表实现路由过滤，偏移列表实现路由花费增加：**

R1配置：

R1(config)#router rip //开启rip进程

R1(config-rip)# version 2 //设置rip版本号

R1(config-rip)# network 30.0.0.0 //宣告主类网络号

R1(config-rip)# network 40.0.0.0

R1(config-rip)# network 172.1.0.0

R1(config-rip)# ***offset-list 2 in 4*** **//偏移列表调用ACL在in方向增加4跳**

R1(config-rip)# no auto-summary //关闭rip自动汇总功能

R1(config-rip)# exit

R1(config)#ip access-list standard 2

R1(config-std-nacl)# permit host 20.1.1.1

R1(config-std-nacl)# exit

R2配置：

R2(config)#ip access-list standard 1

R2(config-std-nacl)# deny host 20.1.1.1

R2(config-std-nacl)# permit any

R2(config-std-nacl)# exit

R2(config)#ip access-list standard 2

R2(config-std-nacl)# deny host 40.1.1.1

R2(config-std-nacl)# permit any

R2(config-std-nacl)# exit

R2(config)#ip access-list standard 3

R2(config-std-nacl)# permit host 10.1.1.1

R2(config-std-nacl)# exit

R2(config)#router rip

R2(config-rip)# version 2

R2(config-rip)# network 172.1.0.0

R2(config-rip)# ***offset-list 3 out 3***

**//偏移列表调用访问列表3 out方向增加3跳**

R2(config-rip)# ***distribute-list 2 out gigabitethernet0***

**//分发列表调用ACL在g0口的out方向**

R2(config-rip)# ***distribute-list 1 out gigabitethernet2***

**//分发列表调用ACL在g2口的out方向**

R2(config-rip)# no auto-summary

R3配置：

R3(config)#ip access-list standard 4

R3(config-std-nacl)# permit host 30.1.1.1

R3(config-std-nacl)# exit

R3(config)#router rip

R3(config-rip)# version 2

R3(config-rip)# network 10.0.0.0

R3(config-rip)# network 20.0.0.0

R3(config-rip)# network 172.1.0.0

R3(config-rip)# offset-list 4 in 3

R3(config-rip)# no auto-summary

R4配置：

R4(config)#ip access-list standard 1

R4(config-std-nacl)# deny host 10.1.1.1

R4(config-std-nacl)# permit any

R4(config-std-nacl)# exit

R4(config)#ip access-list standard 2

R4(config-std-nacl)# deny host 30.1.1.1

R4(config-std-nacl)# permit any

R4(config-std-nacl)# exit

R4(config)#ip access-list standard 3

R4(config-std-nacl)# permit host 40.1.1.1

R4(config-std-nacl)# exit

R4(config)# router rip

R4(config-rip)# version 2

R4(config-rip)# network 172.1.0.0

R4(config-rip)# offset-list 3 out 4

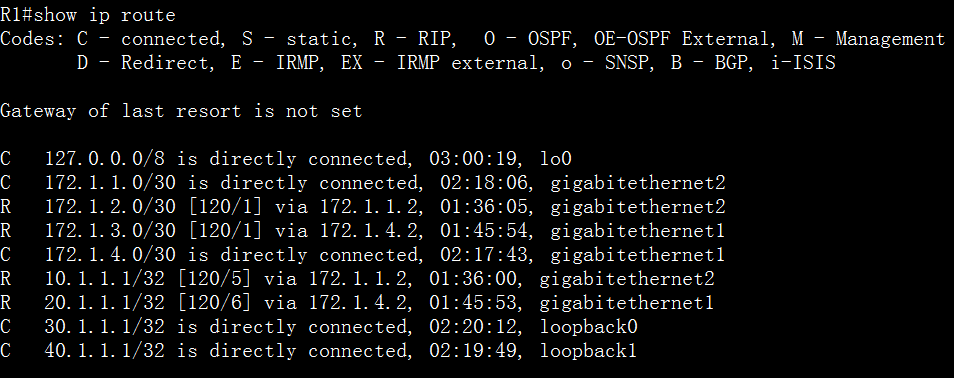
R4(config-rip)# ***distribute-list 2 out gigabitethernet0***

***//分发列表在g0口的出方向过滤acl 2的路由***

R4(config-rip)# distribute-list 1 out gigabitethernet1

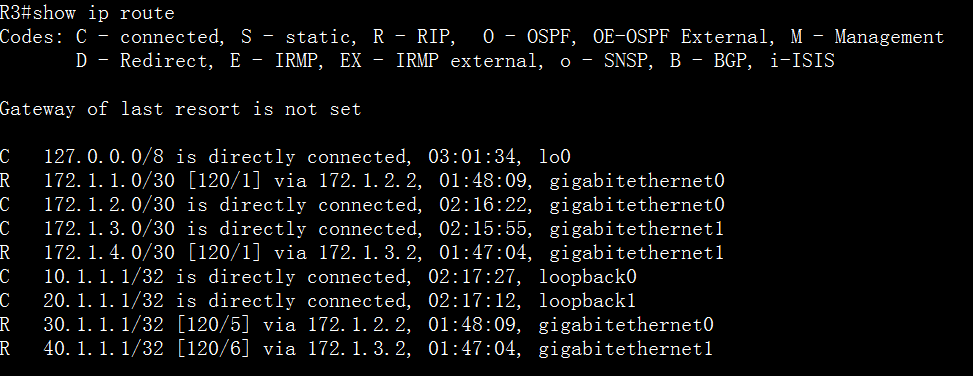
R4(config-rip)# no auto-summary

配置成功后R1路由表如图：



从上图中可以看到R1的g2口只学到了**10.0.0.0**网段的路由，且跳数为5。g1接口只学到了**20.0.0.0**网段的路由，且跳数为5

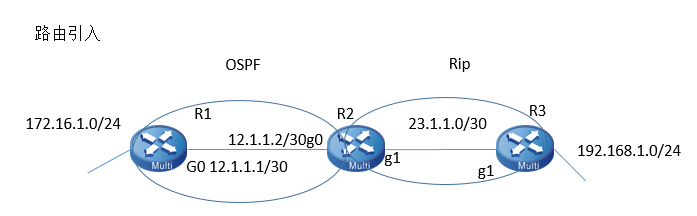
R3路由表如图所示：



同R1。g0接口只学到了**30.0.0.0**网段的路由，且跳数为5。而g1接口只学到了**40.0.0.0**网段的路由，且跳数为6。

## 路由引入

### OSPF和rip 之间的路由重分发



实验要求：

如图所示，R1和R2之间运行OSPF，R2和R3之间运行rip ，正常情况下R1和R3学习不到对方的路由，现在使用单点双向重分发实现全网互通。

实验配置：

R1配置

R1(config)#interface loo0

R1(config-if-loopback0)#ip address 172.16.1.1 24

R1(config-if-loopback0)#exit

R1(config)#interface g0

R1(config-if-gigabitethernet0)#ip address 12.1.1.1 30

R1(config-if-gigabitethernet0)#exit

R1(config)#router ospf 1

R1(config-ospf)#network 172.16.1.0 0.0.0.255 area 0

R1(config-ospf)#network 12.1.1.1 0.0.0.3 area 0

R2(config)#interface gigabitethernet 0

R2(config-if-gigabitethernet0)#ip address 12.1.1.2 30

R2(config-if-gigabitethernet0)#exit

R2(config)#interface g1

R2(config-if-gigabitethernet1)#ip address 23.1.1.2 30

R2(config-if-gigabitethernet1)#exit

R2(config)#router ospf 1

R2(config-ospf)#network 12.1.1.2 0.0.0.3 area 0

R2(config-ospf)#exit

R2(config)#router rip

R2(config-rip)#no auto-summary

R2(config-rip)#version 2

R2(config-rip)#network 23.0.0.0

R3(config)#interface g1

R3(config-if-gigabitethernet1)#ip address 23.1.1.1 30

R3(config-if-gigabitethernet1)#exit

R3(config)#interface loo0

R3(config-if-loopback0)#ip address 192.168.1.1 24

R3(config-if-loopback0)#exit

R3(config)#router rip

R3(config-rip)#no auto-summary

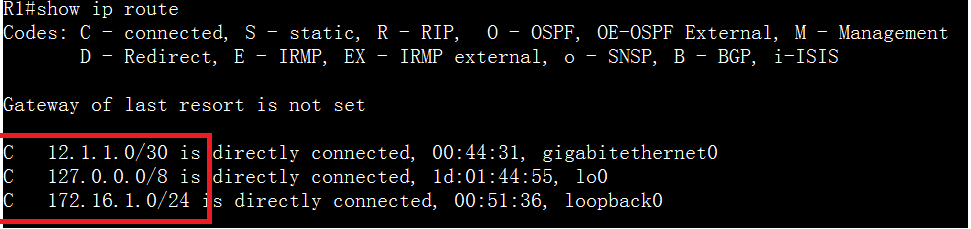
R3(config-rip)#version 2

R3(config-rip)#network 23.0.0.0

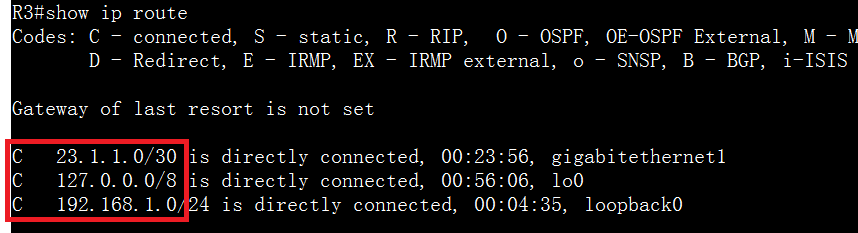
R3(config-rip)#network 192.168.1.0

重分发之前我们分别查看下R1和R3的路由表：

R1路由表：



R3路由表：



在R2上实现路由单点双向重分发。

R2(config)#router ospf 1

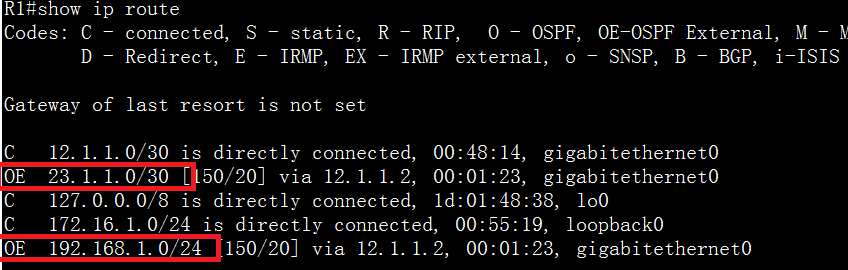
R2(config-ospf)#**redistribute rip //将rip重分发进OSPF**

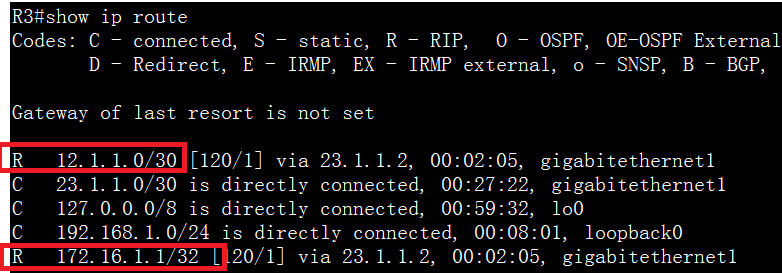
R2(config-ospf)#exit

R2(config)#router rip

R2(config-rip)#**redistribute ospf 1 //将OSPF重分发进rip**

在R1和R3上查看路由表的变化：

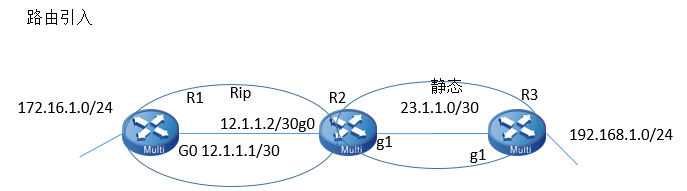




可以看到，R1和R3均学到了对方的路由。

### 静态、直连路由重分发进rip

实验拓扑：



实验要求：将静态路由和直连路由重分发进rip 协议。

相关配置如下：

R1(config)#interface loo0

R1(config-if-loopback0)#ip address 172.16.1.1 24

R1(config-if-loopback0)#exit

R1(config)#interface g0

R1(config-if-gigabitethernet0)#ip address 12.1.1.1 30

R1(config-if-gigabitethernet0)#exit

R1(config)#router rip

R1(config-rip)#no auto-summary

R1(config-rip)#version 2

R1(config-rip)#network 12.0.0.0

R2(config)#interface gigabitethernet 0

R2(config-if-gigabitethernet0)#ip address 12.1.1.2 30

R2(config-if-gigabitethernet0)#exit

R2(config)#interface g1

R2(config-if-gigabitethernet1)#ip address 23.1.1.2 30

R2(config-if-gigabitethernet1)#exit

R2(config)#router rip

R2(config-rip)#no auto-summary

R2(config-rip)#version 2

R2(config-rip)#network 12.0.0.0

R2(config-rip)#exit

R2(config)# ip route 192.168.1.0 255.255.255.0 23.1.1.1

R3(config)#interface g1

R3(config-if-gigabitethernet1)#ip address 23.1.1.1 30

R3(config-if-gigabitethernet1)#exit

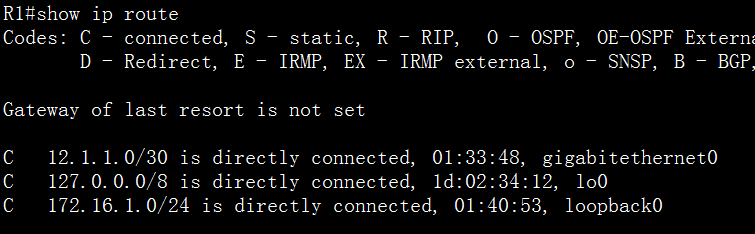
R3(config)#interface loo0

R3(config-if-loopback0)#ip address 192.168.1.1 24

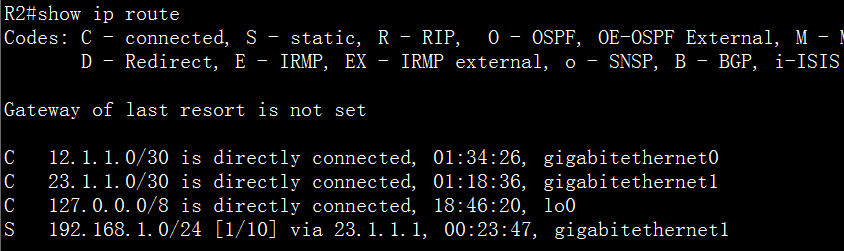
R3(config-if-loopback0)#exit

R3(config)#ip route 0.0.0.0 0.0.0.0 23.1.1.2

充分发前查看R1和R2的路由表：



R1没有23网段和192网段的路由



R2没有172网段的路由。

在R1上重分发直连路由，在R2上重分发静态和直连路由。

R1(config)#router rip

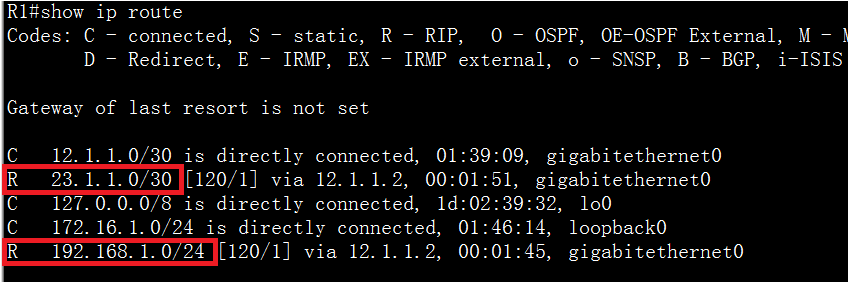
R1(config-rip)#**redistribute connected** //重分发直连路由

R2(config)#router rip

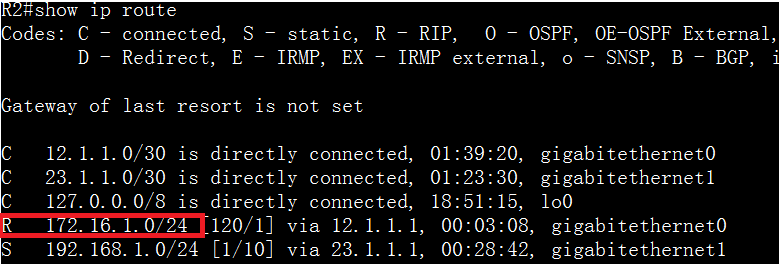
R2(config-rip)#redistribute connected

R2(config-rip)#**redistribute static** //重分发静态路由

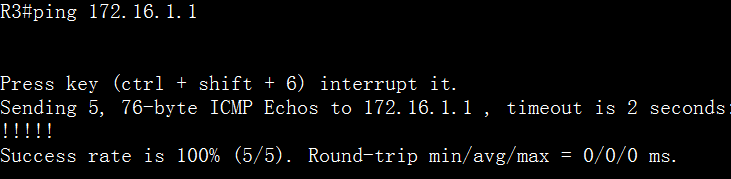
查看R1和R2的路由表：



R1学到了R3 192网段和23网段的路由

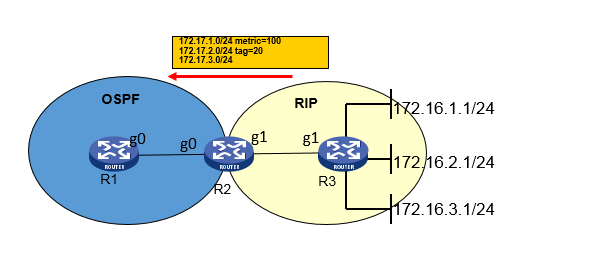


R2学到了R1的172网段路由。



### Route-map工具应用案例

实验拓扑如下：



实验要求：

在R2上将rip路由重分发进OSPF，其中172.16.1.0/24重分发时metric值要改为100，172.16.2.0/24网段重分发时要打上tag为20的标签。172.16.3.0/24重分发默认不变。

相关配置如下：

R3：

R3(config)#interface g1

R3(config-if-gigabitethernet1)#ip address 23.1.1.1 30

R3(config-if-gigabitethernet1)#exit

R3(config)#interface loop0

R3(config-if-loopback0)#ip address 172.16.1.1 24

R3(config-if-loopback0)#exit

R3(config)#interface loop1

R3(config-if-loopback1)#ip address 172.16.2.1 24

R3(config-if-loopback1)#exit

R3(config)#interface loop2

R3(config-if-loopback2)#ip address 172.16.3.1 24

R3(config-if-loopback2)#exit

R3(config)#router rip

R3(config-rip)#no auto-summary

R3(config-rip)#version 2

R3(config-rip)#network 23.0.0.0

R3(config-rip)#network 172.16.0.0

R2(config)#interface g1

R2(config-if-gigabitethernet1)#ip address 23.1.1.2 30

R2(config-if-gigabitethernet1)#exit

R2(config)#interface g0

R2(config-if-gigabitethernet0)#ip address 12.1.1.2 30

R2(config-if-gigabitethernet0)#exit

R2(config)#router rip

R2(config-rip)#no auto-summary

R2(config-rip)#version 2

R2(config-rip)#network 23.0.0.0

R2(config-rip)#exit

R2(config)#router ospf 1

R2(config-ospf)#network 12.1.1.2 0.0.0.3 area 0

R1(config)#interface g0

R1(config-if-gigabitethernet0)#ip address 12.1.1.1 30

R1(config-if-gigabitethernet0)#exit

R1(config)#router ospf 1

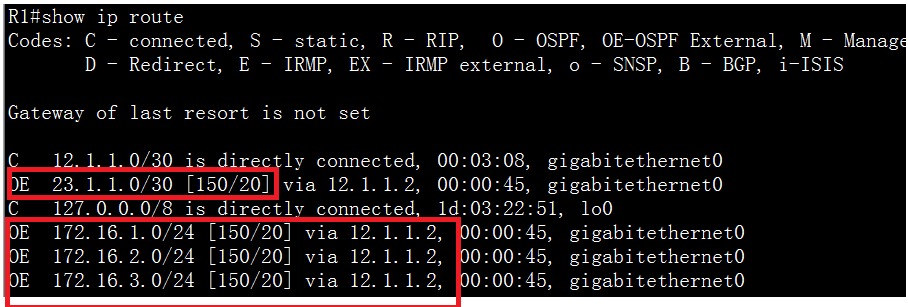
R1(config-ospf)#network 12.1.1.1 0.0.0.3 area 0

在R2上实现路由重分发。

R2(config)#router ospf 1

R2(config-ospf)#redistribute rip

在没有使用route-map策略之前先看R1的路由表。



现在在R2上使用route-map实现172网段重分发的不同实验要求。

R2(config)#ip access-list standard 1 //ACL1抓取路由172.16.1.0/24

R2(config-std-nacl)#permit 172.16.1.0 0.0.0.255

R2(config-std-nacl)#exit

R2(config)#ip access-list standard 2

R2(config-std-nacl)#permit 172.16.2.0 0.0.0.255

R2(config-std-nacl)#exit

R2(config)#ip access-list standard 3

R2(config-std-nacl)#permit 172.16.3.0 0.0.0.255

R2(config-std-nacl)#exit

R2(config)#route-map ccie permit 10

**//启用route-map命名为ccie,行为为permit，10为序列号**

R2(config-route-map)#match ip address 1

**//匹配acl 1**

R2(config-route-map)#**set metric 100 //给匹配的路由设置相关属性**

R2(config-route-map)#exit

R2(config)#route-map ccie permit 20

R2(config-route-map)#match ip address 2

R2(config-route-map)**#set tag 20 //给匹配的ACL 2路由条目打上tag 20**

R2(config-route-map)#exit

R2(config)#**route-map ccie permit 30 //增加一条空语句放行其他重分发路由**

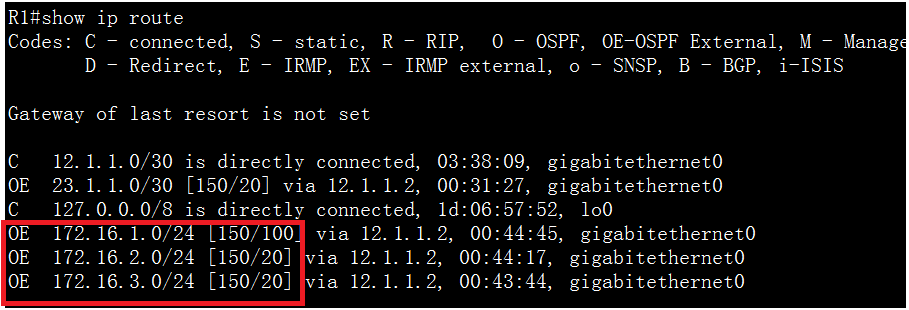
R2(config-route-map)#exit

R2(config)#router ospf 1

R2(config-ospf)#**redistribute rip route-map ccie //调用route-map ccie**

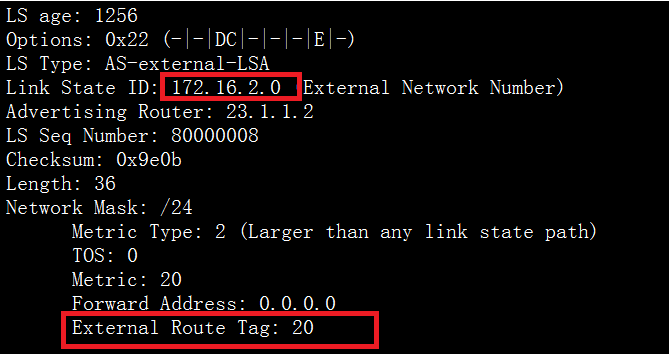
R2(config-ospf)#exit

查看R1的路由表：



可以看到172.16.1.0/24的路由metric值变为了100.且172.16.3.0/24路由正常重分发。

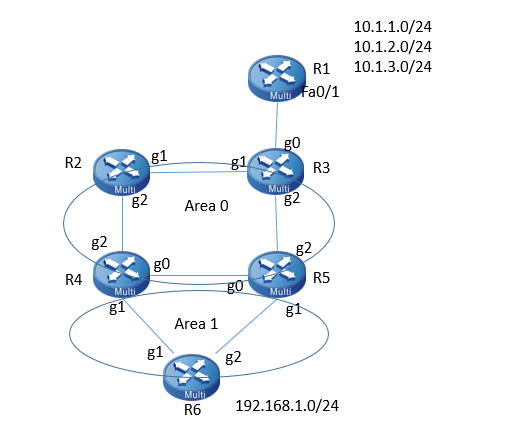
使用命令show ip ospf database external命令查看172.16.2.0/24是否打上了tag 20。



路由条目172.16.2.0/24打上了tag20 的标签。

## OSPF选路

实验拓扑如下图：



实验要求：

1>OSPF区域划分如图

2>R2和R3学到区域路由为一条汇总路由192.168.0.0/16

3>R3上有到R1三个网段的静态路由，重分发到OSPF后汇总为10.1.0.0/16

4>要求R6到10.1.0.0/16网段的路由优先选择从R4-R2-R3，来回路径一致。

5>从R5的路径为备份路径

（1）接口地址配置

R1(config)#interface loo0

R1(config-if-loopback0)#ip address 10.1.1.1 24

R1(config-if-loopback0)#exit

R1(config)#interface loop1

R1(config-if-loopback1)#ip address 10.1.2.1 24

R1(config-if-loopback1)#exit

R1(config)#interface loop2

R1(config-if-loopback2)#ip address 10.1.3.1 24

R1(config-if-loopback2)#exit

R1(config)#vlan 13

R1(config-vlan13)#exit

R1(config)#interface vlan13

R1(config-if-vlan13)#ip address 13.1.1.1 24

R1(config-if-vlan13)#exit

R1(config)#interface fa0/1

R1(config-if-fastethernet0/1)#switchport mode access

R1(config-if-fastethernet0/1)#switchport access vlan 13

R2(config)#interface g1

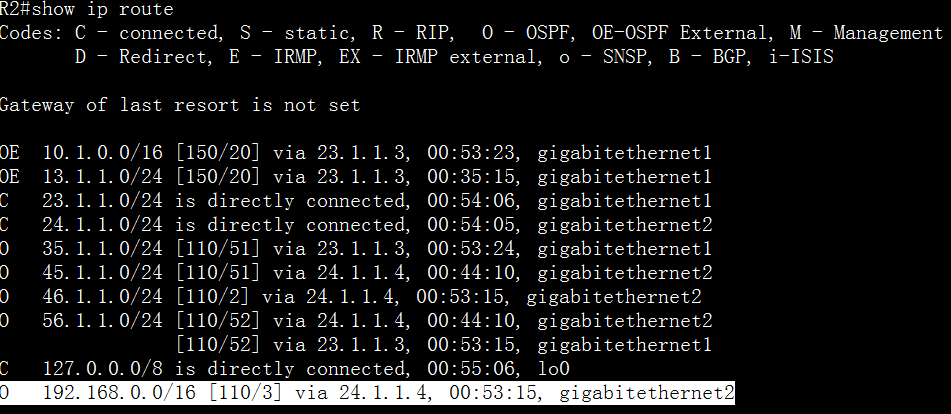
R2(config-if-gigabitethernet1)#ip address 23.1.1.2 24

R2(config-if-gigabitethernet1)#exit

R2(config)#interface g2

R2(config-if-gigabitethernet2)#ip address 24.1.1.2 24

R2(config-if-gigabitethernet2)#exit



由上图可知，R2学到192网段的路由为汇总路由192.168.0.0/16

R3(config)#interface g0

R3(config-if-gigabitethernet0)#ip address 13.1.1.3 24

R3(config-if-gigabitethernet0)#exit

R3(config)#interface g1

R3(config-if-gigabitethernet1)#ip address 23.1.1.3 24

R3(config-if-gigabitethernet1)#exit

R3(config)#interface g2

R3(config-if-gigabitethernet2)#ip address 35.1.1.3 24

R3(config-if-gigabitethernet2)#exit



R3学到192网段的路由为汇总路由192.168.0.0/16。R3上有到R1三个网段的静态路由，汇总为10.1.0.0/16后重分发到了OSPF

R4(config)#interface g0

R4(config-if-gigabitethernet0)#ip address 45.1.1.4 24

R4(config-if-gigabitethernet0)#exit

R4(config)#interface g1

R4(config-if-gigabitethernet1)#ip address 46.1.1.4 24

R4(config-if-gigabitethernet1)#exit

R4(config)#interface g2

R4(config-if-gigabitethernet2)#ip address 24.1.1.4 24

R5(config)#interface g0

R5(config-if-gigabitethernet0)#ip address 45.1.1.5 24

R5(config-if-gigabitethernet0)#exit

R5(config)#interface g1

R5(config-if-gigabitethernet1)#ip address 56.1.1.5 24

R5(config-if-gigabitethernet1)#exit

R5(config)#interface g2

R5(config-if-gigabitethernet2)#ip address 35.1.1.5 24

R6(config)#interface g1

R6(config-if-gigabitethernet1)#ip address 46.1.1.6 24

R6(config-if-gigabitethernet1)#exit

R6(config)#interface g2

R6(config-if-gigabitethernet2)#ip address 56.1.1.6 24

R6(config-if-gigabitethernet2)#exit

R6(config)#interface loop0

R6(config-if-loopback0)#ip address 192.168.1.6 24

**（2）OSPF配置**

R1配置：

R1(config)#ip route 0.0.0.0 0.0.0.0 13.1.1.3 //配置缺省路由指向R3

R2(config)#router ospf 1

R2(config-ospf)# network 23.1.1.0 0.0.0.255 area 0

R2(config-ospf)# network 24.1.1.0 0.0.0.255 area 0

R3(config)#ip route 10.1.1.0 255.255.255.0 13.1.1.1

R3(config)#ip route 10.1.2.0 255.255.255.0 13.1.1.1

R3(config)#ip route 10.1.3.0 255.255.255.0 13.1.1.1

R3(config)#router ospf 1

R3(config-ospf)# network 23.1.1.0 0.0.0.255 area 0

R3(config-ospf)# network 35.1.1.0 0.0.0.255 area 0

R3(config-ospf)# **redistribute connected //直连路由重分发进OSPF**

R3(config-ospf)# **redistribute static //重分发静态路由进OSPF**

R3(config-ospf)# **summary-address 10.1.0.0 255.255.0.0**

**//汇总外部静态路由**

R3(config-ospf)# exit

R3(config)#interface gigabitethernet2

R3(config-if-gigabitethernet2)# ***ip ospf cost 50***

***//调整OSPF cost值，让其路径优先选择g1口***

R3(config-if-gigabitethernet2)# exit

R4(config)#router ospf 1

R4(config-ospf)# ***area 1 range 192.168.0.0/16 //汇总OSPF域间路由***

R4(config-ospf)# network 24.1.1.0 0.0.0.255 area 0

R4(config-ospf)# network 45.1.1.0 0.0.0.255 area 0

R4(config-ospf)# network 46.1.1.0 0.0.0.255 area 1

R4(config-ospf)# exit

R4(config)#interface gigabitethernet0

R4(config-if-gigabitethernet0)# ***ip ospf cost 50***

***//调整OSPF cost值，让其OSPF路由10.1.0.0路径优先选择g2口***

R5(config)#router ospf 1

R5(config-ospf)# area 1 range 192.168.0.0/16 //汇总域间路由192.168.0.0

R5(config-ospf)# network 35.1.1.0 0.0.0.255 area 0

R5(config-ospf)# network 45.1.1.0 0.0.0.255 area 0

R5(config-ospf)# network 56.1.1.0 0.0.0.255 area 1

R6(config)#router ospf 1

R6(config-ospf)# network 46.1.1.0 0.0.0.255 area 1

R6(config-ospf)# network 56.1.1.0 0.0.0.255 area 1

R6(config-ospf)# network 192.168.1.0 0.0.0.255 area 1

R6(config-ospf)# exit

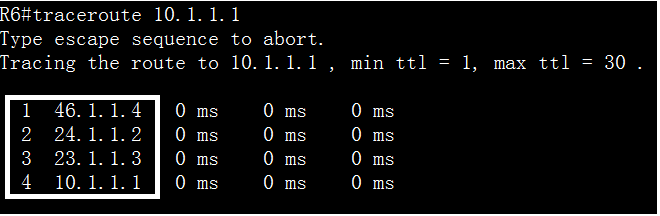
R6(config)#interface gigabitethernet2

R6(config-if-gigabitethernet2)# ***ip ospf cost 50***

***//增大OSPF cost值，让其路径优先选择g1接口***

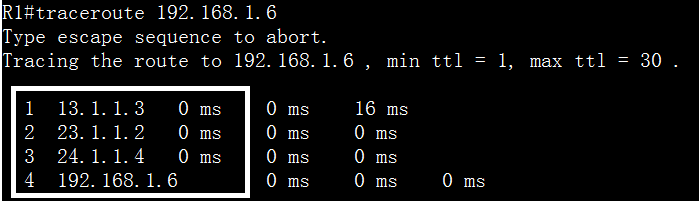
R6(config-if-gigabitethernet2)# exit

在R6上traceroute 10.1.0.0网段的路由，看其路由路径走向



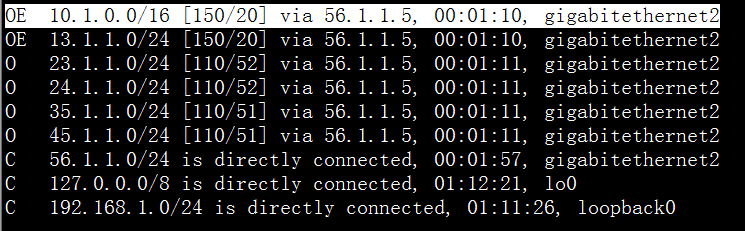
可以看到R6到达10.1.0.0/16网段的路径为R4-R2-R3-R1

再在R1上看回包：

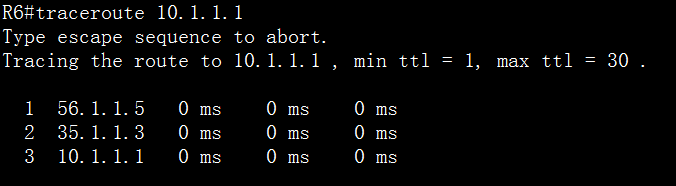


可以看到回包路径为R3-R2-R4-R6

现在我们down掉R6的g1口：



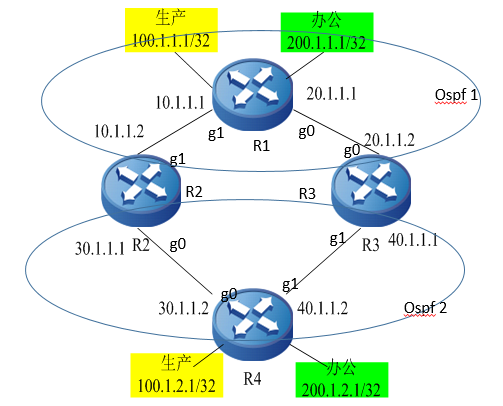
看到R6到达10.1.0.0/16网段的下一跳切换到了备份路径R5。



此时，R6到达10.1.0.0/16网段的路径为R5-R3-R1。

## OSPF分流互备

实验拓扑如下：



实验要求：

使用OSPF协议实现总部R1和分部R4的生产、办公网之间的分流互备。要求使用OSPF三种方法实现。

**方法一：不同OSPF进程间的重分发实现分流互备**

相关配置如下：

（1）基本ip地址配置：

R1(config)#interface loo0

R1(config-if-loopback0)#ip address 100.1.1.1 24

R1(config-if-loopback0)#exit

R1(config)#interface loo1

R1(config-if-loopback1)#ip address 200.1.1.1 24

R1(config-if-loopback1)#exit

R1(config)#interface g1

R1(config-if-gigabitethernet1)#ip address 10.1.1.1 30

R1(config-if-gigabitethernet1)#exit

R1(config)#interface g0

R1(config-if-gigabitethernet0)#ip address 20.1.1.1 30

R1(config-if-gigabitethernet0)#exit

R2(config)#interface g1

R2(config-if-gigabitethernet1)#ip address 10.1.1.2 30

R2(config-if-gigabitethernet1)#exit

R2(config)#interface g0

R2(config-if-gigabitethernet0)#ip address 30.1.1.1 30

R2(config-if-gigabitethernet0)#exit

R3(config)#interface g0

R3(config-if-gigabitethernet0)#ip address 20.1.1.2 30

R3(config-if-gigabitethernet0)#exit

R3(config)#interface g1

R3(config-if-gigabitethernet1)#ip address 40.1.1.1 30

R3(config-if-gigabitethernet1)#exit

R4(config)#interface loop0

R4(config-if-loopback0)#ip address 100.1.2.1 24

R4(config-if-loopback0)#exit

R4(config)#interface loo1

R4(config-if-loopback1)#ip address 200.1.2.1 24

R4(config-if-loopback1)#exit

R4(config)#interface g0

R4(config-if-gigabitethernet0)#ip address 30.1.1.2 30

R4(config-if-gigabitethernet0)#exit

R4(config)#interface g1

R4(config-if-gigabitethernet1)#ip address 40.1.1.2 30

R4(config-if-gigabitethernet1)#exit

（2）OSPF配置

R1(config)#router ospf **1** //启用OSPF进程1

R1(config-ospf)#network 100.1.1.0 0.0.0.255 area 0

R1(config-ospf)#network 200.1.1.0 0.0.0.255 area 0

R1(config-ospf)#network 10.1.1.0 0.0.0.255 area 0

R1(config-ospf)#network 20.1.1.0 0.0.0.255 area 0

R2(config)#router ospf 1

R2(config-ospf)#network 10.1.1.0 0.0.0.255 area 0

R2(config-ospf)#exit

R2(config)#router ospf **2 //启用OSPF进程2**

R2(config-ospf)#network 30.1.1.0 0.0.0.255 area 0

R3(config)#router ospf 1

R3(config-ospf)#network 20.1.1.0 0.0.0.255 area 0

R3(config-ospf)#exit

R3(config)#router ospf 2

R3(config-ospf)#network 40.1.1.0 0.0.0.255 area 0

R3(config-ospf)#exit

R4(config)#router ospf 2

R4(config-ospf)#network 30.1.1.0 0.0.0.255 area 0

R4(config-ospf)#network 40.1.1.0 0.0.0.255 area 0

R4(config-ospf)#network 100.1.2.0 0.0.0.255 area 0

R4(config-ospf)#network 200.1.2.0 0.0.0.255 area 0

**在R2和R3上重分发OSPF进程实现分流互备：**

将OSPF进程2的路由重分发进OSPF1

R2(config)#ip access-list standard 1

R2(config-std-nacl)#permit 200.1.2.1 0.0.0.0

R2(config-std-nacl)#exit

R2(config)#ip access-list standard 2

R2(config-std-nacl)#permit 100.1.2.1 0.0.0.0

R2(config-std-nacl)#exit

R2(config)#route-map 2-1 permit 10

R2(config-route-map)#match ip address 1

R2(config-route-map)#**set metric 21**

//将acl1匹配的路由重分发时增大其metric值

R2(config-route-map)#exit

R2(config)#route-map 2-1 permit 20

R2(config-route-map)#match ip address 2

R2(config-route-map)#**set metric 20**

R2(config-route-map)#exit

R2(config)#route-map 2-1 permit 30

R2(config-route-map)#exit

R2(config)#router ospf 1

R2(config-ospf)#**redistribute ospf 2 route-map 2-1**

R3(config)#ip access-list standard 1

R3(config-std-nacl)#permit 100.1.2.1 0.0.0.0

R3(config-std-nacl)#exit

R3(config)#ip access-list standard 2

R3(config-std-nacl)#permit 200.1.2.1 0.0.0.0

R3(config-std-nacl)#exit

R3(config)#route-map 2-1 permit 10

R3(config-route-map)#match ip address 1

R3(config-route-map)#**set metric 21**

//将acl1匹配的路由重分发时增大其metric值

R3(config-route-map)#exit

R3(config)#route-map 2-1 permit 20

R3(config-route-map)#match ip address 2

R3(config-route-map)#**set metric 20**

R3(config-route-map)#exit

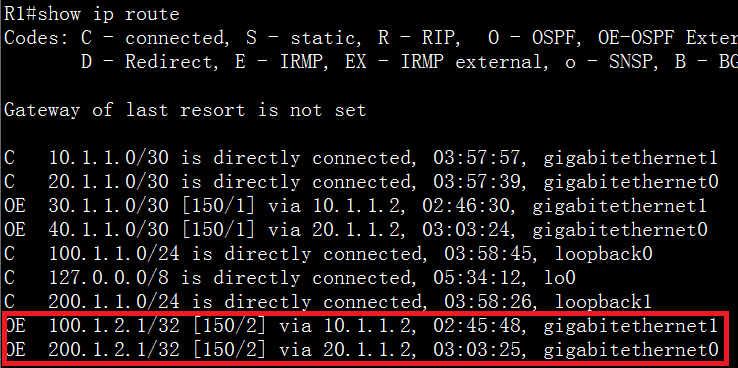
R3(config)#route-map 2-1 permit 30

R3(config-route-map)#exit

R3(config)#router ospf 1

R3(config-ospf)#**redistribute ospf 2 route-map 2-1**

重分发后我们查看R1的路由表：



在R1上实现了生产网段走左边，办公网段走右边。

现在将OSPF进程1的路由重分发进OSPF2。方法同上。

R2(config)#ip access-list standard 3

R2(config-std-nacl)#permit 200.1.1.1 0.0.0.0

R2(config-std-nacl)#exit

R2(config)#ip access-list standard 4

R2(config-std-nacl)#permit 100.1.1.1 0.0.0.0

R2(config-std-nacl)#exit

R2(config)#route-map 1-2 permit 10

R2(config-route-map)#match ip address 3

R2(config-route-map)#set metric 21

R2(config-route-map)#exit

R2(config)#route-map 1-2 permit 20

R2(config-route-map)#match ip address 4

R2(config-route-map)#set metric 20

R2(config-route-map)#exit

R2(config)#route-map 1-2 permit 30

R2(config-route-map)#exit

R2(config)#router ospf 2

R2(config-ospf)#redistribute ospf 1 route-map 1-2

R3(config)#ip access-list standard 3

R3(config-std-nacl)#permit 100.1.1.1 0.0.0.0

R3(config-std-nacl)#exit

R3(config)#ip access-list standard 4

R3(config-std-nacl)#permit 200.1.1.1 0.0.0.0

R3(config-std-nacl)#exit

R3(config)#route-map 1-2 permit 10

R3(config-route-map)#match

R3(config-route-map)#match ip address 3

R3(config-route-map)#set metric 21

R3(config-route-map)#exit

R3(config)#route-map 1-2 permit 20

R3(config-route-map)#match ip address 4

R3(config-route-map)#set metric 20

R3(config-route-map)#exit

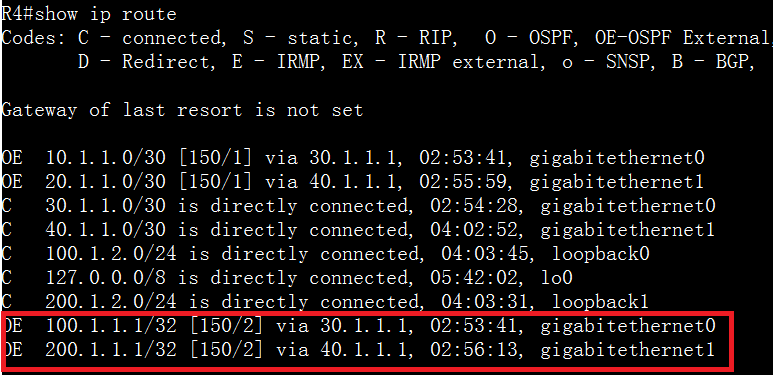
R3(config)#route-map 1-2 permit 30

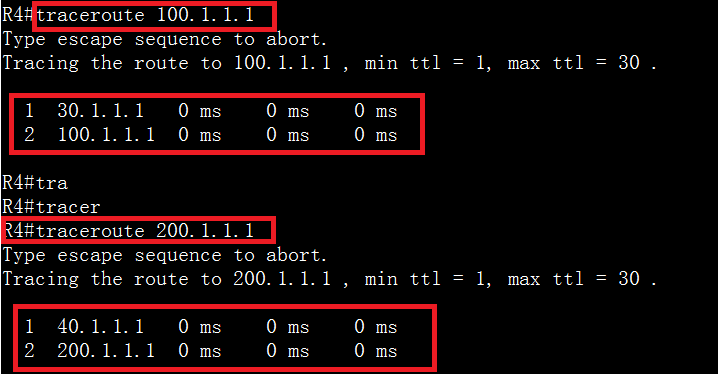
R3(config-route-map)#exit

R3(config)#router ospf 2

R3(config-ospf)#redistribute ospf 1 route-map 1-2

查看R4的路由表：



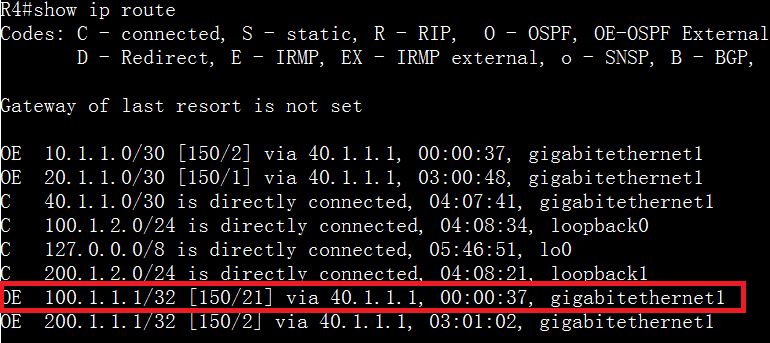


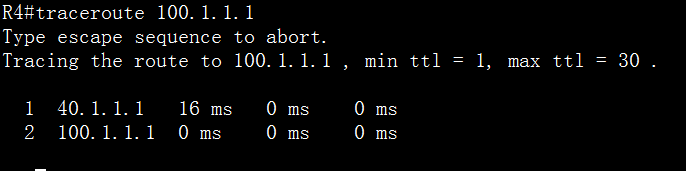
在R4上同样实现了生产办公分流。

现在验证线路是否相互备份。在R4上down掉g0接口：



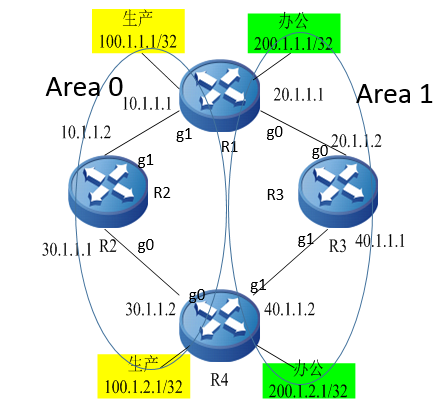
查看R4的路由表：





生产网段100.1.1.1成功切换到了R3上。其他网段分流备份验证同理。

**方法二：利用域内路由优先级高于域间路由优先级实现分流互备：**



Ip地址基本配置不变：

R1(config)#interface loo0

R1(config-if-loopback0)#ip address 100.1.1.1 24

R1(config-if-loopback0)#exit

R1(config)#interface loo1

R1(config-if-loopback1)#ip address 200.1.1.1 24

R1(config-if-loopback1)#exit

R1(config)#interface g1

R1(config-if-gigabitethernet1)#ip address 10.1.1.1 30

R1(config-if-gigabitethernet1)#exit

R1(config)#interface g0

R1(config-if-gigabitethernet0)#ip address 20.1.1.1 30

R1(config-if-gigabitethernet0)#exit

R2(config)#interface g1

R2(config-if-gigabitethernet1)#ip address 10.1.1.2 30

R2(config-if-gigabitethernet1)#exit

R2(config)#interface g0

R2(config-if-gigabitethernet0)#ip address 30.1.1.1 30

R2(config-if-gigabitethernet0)#exit

R3(config)#interface g0

R3(config-if-gigabitethernet0)#ip address 20.1.1.2 30

R3(config-if-gigabitethernet0)#exit

R3(config)#interface g1

R3(config-if-gigabitethernet1)#ip address 40.1.1.1 30

R3(config-if-gigabitethernet1)#exit

R4(config)#interface loop0

R4(config-if-loopback0)#ip address 100.1.2.1 24

R4(config-if-loopback0)#exit

R4(config)#interface loo1

R4(config-if-loopback1)#ip address 200.1.2.1 24

R4(config-if-loopback1)#exit

R4(config)#interface g0

R4(config-if-gigabitethernet0)#ip address 30.1.1.2 30

R4(config-if-gigabitethernet0)#exit

R4(config)#interface g1

R4(config-if-gigabitethernet1)#ip address 40.1.1.2 30

R4(config-if-gigabitethernet1)#exit

OSPF配置：

R1(config)#router ospf 1

R1(config-ospf)#network 100.1.1.1 0.0.0.0 area 0

R1(config-ospf)#network 10.1.1.1 0.0.0.0 area 0

R1(config-ospf)#network 20.1.1.1 0.0.0.0 area 1

R1(config-ospf)#network 200.1.1.1 0.0.0.0 area 1

R1(config-ospf)#exit

R2(config)#router ospf 1

R2(config-ospf)#network 10.1.1.2 0.0.0.0 area 0

R2(config-ospf)#network 30.1.1.1 0.0.0.0 area 0

R2(config-ospf)#exit

R3(config)#router ospf 1

R3(config-ospf)#network 20.1.1.2 0.0.0.0 area 1

R3(config-ospf)#network 40.1.1.1 0.0.0.0 area 1

R3(config-ospf)#exit

R4(config)#router ospf 1

R4(config-ospf)#network 30.1.1.2 0.0.0.255 area 0

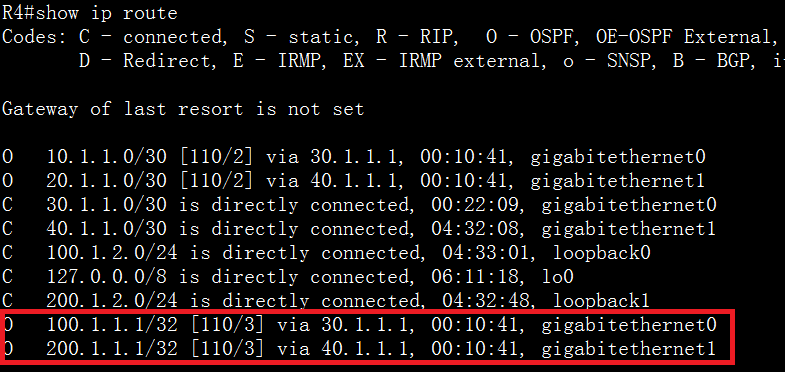
R4(config-ospf)#network 40.1.1.2 0.0.0.255 area 1

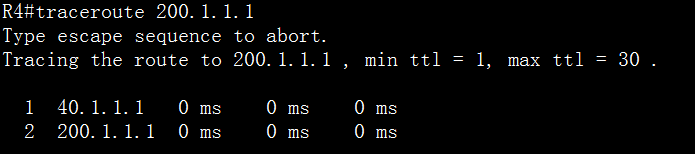
R4(config-ospf)#network 200.1.2.1 0.0.0.255 area 1

R4(config-ospf)#network 100.1.2.1 0.0.0.255 area 0

R4(config-ospf)#end

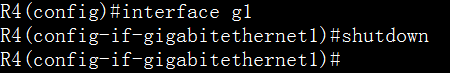
在R4上查看路由表：



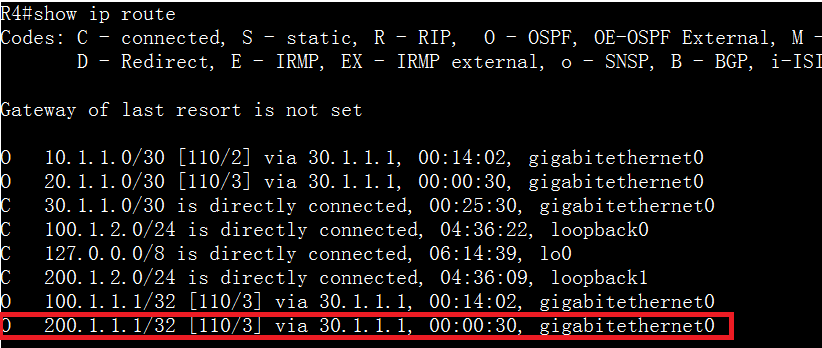


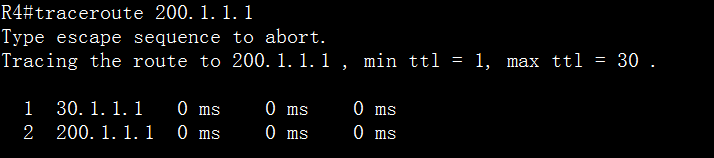
生产办公实现了分流，说明域内学到的路由优先级高于域间路由优先级。

现在down掉R4的g1接口：



查看R4路由表：

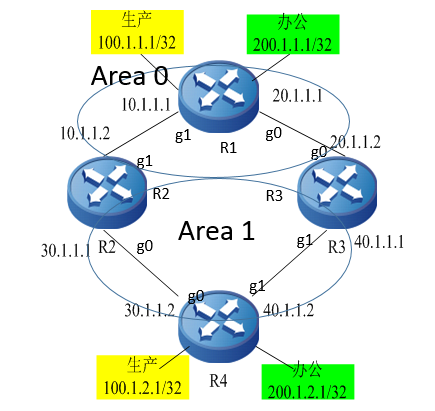




R4从g0接口学到了了200.1.1.1的路由。100网段的验证方法完全一样。

**方法三：利用通告的明细路由优先级高于汇总路由实现分流互备。**

实验拓扑不变：



基本接口ip地址配置不变（略）

OSPF配置：

R1(config)#router ospf 1

R1(config-ospf)#network 100.1.1.1 0.0.0.0 area 0

R1(config-ospf)#network 200.1.1.1 0.0.0.0 area 0

R1(config-ospf)#network 10.1.1.1 0.0.0.0 area 0

R1(config-ospf)#network 20.1.1.1 0.0.0.0 area 0

R2(config)#router ospf 1

R2(config-ospf)#network 10.1.1.2 0.0.0.0 area 0

R2(config-ospf)#network 30.1.1.1 0.0.0.0 area 1

R3(config)#router ospf 1

R3(config-ospf)#network 20.1.1.2 0.0.0.0 area 0

R3(config-ospf)#network 40.1.1.1 0.0.0.0 area 1

R3(config-ospf)#exit

R4(config)#router ospf 1

R4(config-ospf)#network 30.1.1.2 0.0.0.0 area 1

R4(config-ospf)#network 40.1.1.2 0.0.0.0 area 1

R4(config-ospf)#network 100.1.2.1 0.0.0.0 area 1

R4(config-ospf)#network 200.1.2.1 0.0.0.0 area 1

***在R2和R3上汇总路由。***

R2(config)#router ospf 1

R2(config-ospf)#area 1 range 200.1.0.0 255.255.0.0

R2(config-ospf)#area 0 range 200.1.0.0 255.255.0.0

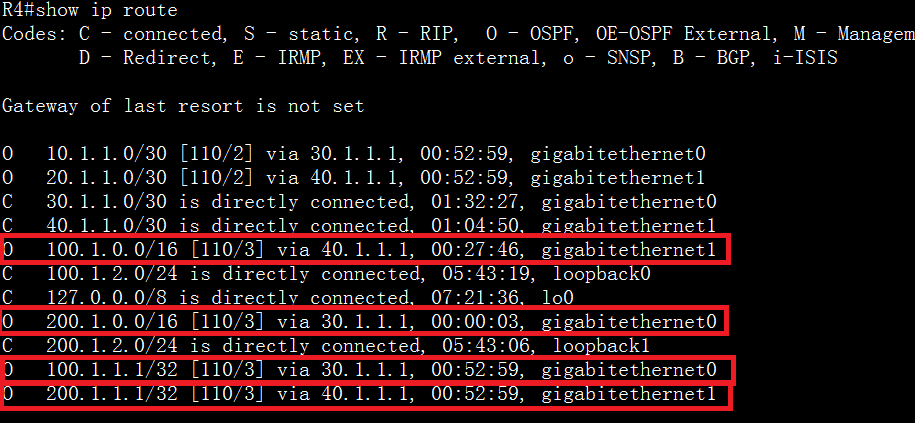
R2(config-ospf)#exit

R3(config)#router ospf 1

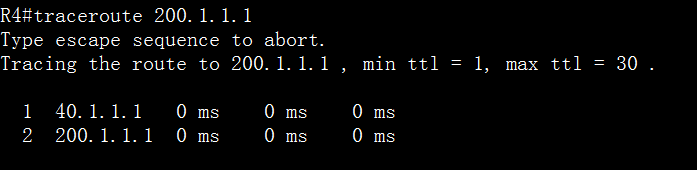
R3(config-ospf)#area 0 range 100.1.0.0 255.255.0.0

R3(config-ospf)#area 1 range 100.1.0.0 255.255.0.0

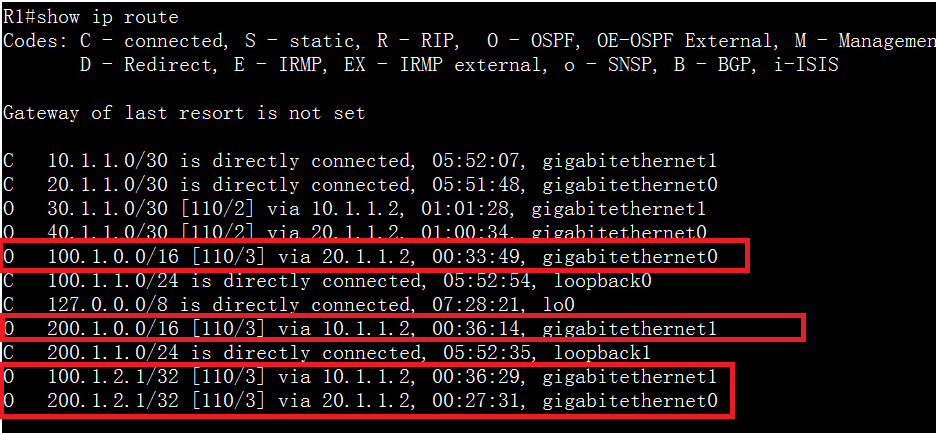
查看R4的路由表：



在R4上到达200.1.0.0网段的路径有两条，根据最长掩码匹配原则，路径将首选g1接口的路径。G0接口的路径则作为备份路径。到达100.1.0.0网段路径同理。



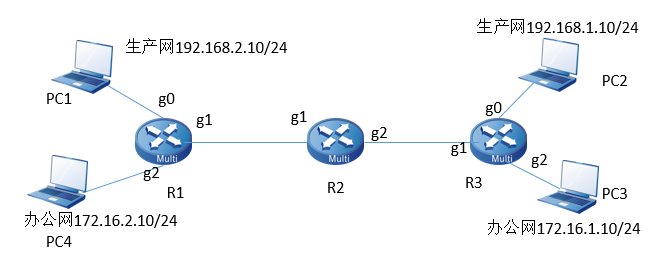
R1上路由表：



## ACL

### Acl实现业务隔离

实验拓扑如下：



实验要求：实现生产网之间的互访、生产网和办公网相互隔离。

基本配置如下：

（1）接口地址配置

R1(config)#interface g1

R1(config-if-gigabitethernet1)#ip address 12.1.1.1 24

R1(config-if-gigabitethernet1)#exit

R1(config)#interface g0

R1(config-if-gigabitethernet0)#ip address 192.168.2.1 24

R1(config-if-gigabitethernet0)#exit

R1(config)#interface g2

R1(config-if-gigabitethernet2)#ip address 172.16.2.1 24

R1(config-if-gigabitethernet2)#exit

R2(config)#interface g2

R2(config-if-gigabitethernet2)#ip address 23.1.1.2 24

R2(config-if-gigabitethernet2)#exit

R2(config)#interface g1

R2(config-if-gigabitethernet1)#ip address 12.1.1.2 24

R2(config-if-gigabitethernet1)#exit

R3(config)#interface g1

R3(config-if-gigabitethernet1)#ip address 23.1.1.3 24

R3(config-if-gigabitethernet1)#exit

R3(config)#interface g2

R3(config-if-gigabitethernet2)#ip address 172.16.1.3 24

R3(config-if-gigabitethernet2)#exit

R3(config)#interface g0

R3(config-if-gigabitethernet0)#ip address 192.168.1.3 24

R3(config-if-gigabitethernet0)#exit

（2）运行rip协议，配置ACL实现业务互访

R1(config)#router rip

R1(config-rip)# version 2

R1(config-rip)# network 12.0.0.0

R1(config-rip)# network 172.16.0.0

R1(config-rip)# network 192.168.2.0

R1(config-rip)# no auto-summary

R1(config-rip)# exit

R1(config)#ip access-list extended 1001 //配置扩展ACL 1001

R1(config-ext-nacl)# 10 deny ip 192.168.0.0 0.0.255.255 172.16.0.0 0.0.255.255

//拒绝192.168.0.0网段到172.16.0.0网段的访问

R1(config-ext-nacl)# 20 deny ip 172.16.0.0 0.0.255.255 192.168.0.0 0.0.255.255

//拒绝172.16.0.0网段到192.168.0.0网段的访问

R1(config-ext-nacl)# 30 permit ip any any

R1(config-ext-nacl)# exit

R1(config)#interface gigabitethernet0

R1(config-if-gigabitethernet0)# ip access-group 1001 out //在接口g0出方向调用ACL

R1(config-if-gigabitethernet0)# exit

R1(config)#interface gigabitethernet2

R1(config-if-gigabitethernet2)# ip address 172.16.2.1 255.255.255.0

R1(config-if-gigabitethernet2)# ip access-group 1001 out

R1(config-if-gigabitethernet2)# exit

R2(config)#router rip

R2(config-rip)# version 2

R2(config-rip)# network 12.0.0.0

R2(config-rip)# network 23.0.0.0

R2(config-rip)# no auto-summary

R3(config)#router rip

R3(config-rip)# version 2

R3(config-rip)# network 23.0.0.0

R3(config-rip)# network 172.16.0.0

R3(config-rip)# network 192.168.1.0

R3(config-rip)# no auto-summary

R3(config-rip)# exit

R3(config)#ip access-list extended 1001

R3(config-ext-nacl)# 10 deny ip 192.168.0.0 0.0.255.255 172.16.0.0 0.0.255.255

R3(config-ext-nacl)# 20 deny ip 172.16.0.0 0.0.255.255 192.168.0.0 0.0.255.255

R3(config-ext-nacl)# 30 permit ip any any

R3(config-ext-nacl)# exit

R3(config)#interface gigabitethernet0

R3(config-if-gigabitethernet0)# ip address 192.168.1.3 255.255.255.0

R3(config-if-gigabitethernet0)# ip access-group 1001 out

R3(config-if-gigabitethernet0)# exit

R3(config)#interface gigabitethernet2

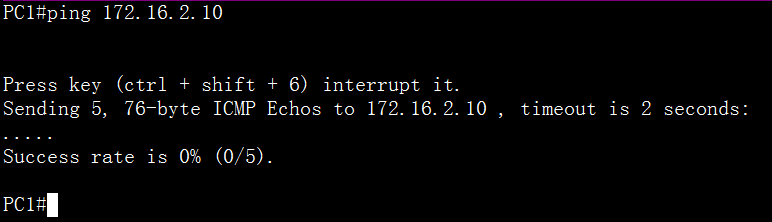
R3(config-if-gigabitethernet2)# ip address 172.16.1.3 255.255.255.0

R3(config-if-gigabitethernet2)# ip access-group 1001 out

R3(config-if-gigabitethernet2)# exit

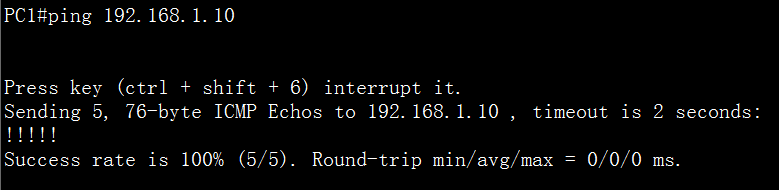
验证：

在PC1上ping PC4

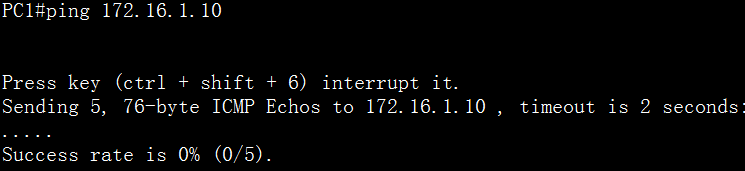


由上图可知，PC1不能访问PC4

PC1 ping PC2

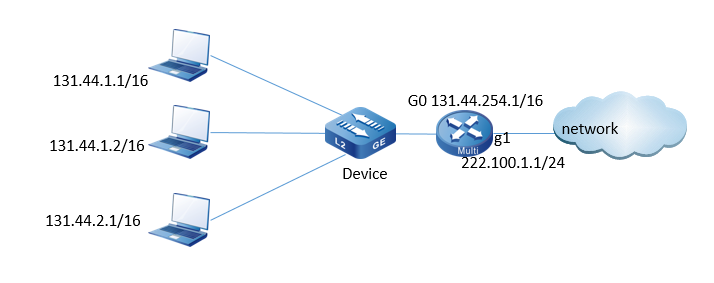


PC1能够与PC2互访。



PC1和PC3不能互访。

### 带时间域的IP扩展ACL



实验要求：PC1的IP地址为131.44.1.1/16，PC2的IP地址为131.44.1.2/16，PC3的IP地址为131.44.2.1/16。

在Device上配置acl规则，允许PC2能通过FTP访问IP Network。

在Device上配置ACL规则，允许PC1在指定时间范围内访问IP Network。

在Device上配置ACL规则，阻止131.44.2.0/24网段的PC访问IP Network。

实验配置：

PC1(config)#interface g0

PC1(config-if-gigabitethernet0)#ip add 131.44.1.1 16

PC1(config-if-gigabitethernet0)#exit

PC1(config)#ip route 0.0.0.0 0.0.0.0 131.44.254.1

PC2(config)#interface g0

PC2(config-if-gigabitethernet0)#ip add 131.44.1.2 16

PC2(config-if-gigabitethernet0)#exit

PC2(config)#ip route 0.0.0.0 0.0.0.0 131.44.254.1

PC3(config)#interface g0

PC3(config-if-gigabitethernet0)#ip add 131.44.2.1 16

PC3(config-if-gigabitethernet0)#exit

PC3(config)#ip route 0.0.0.0 0.0.0.0 131.44.254.1

device(config)#interface g0

device(config-if-gigabitethernet0)#ip add 131.44.254.1 16

device(config-if-gigabitethernet0)#exit

device(config)#interface g1

device(config-if-gigabitethernet1)#ip add 222.100.1.1 24

device(config-if-gigabitethernet1)#exit

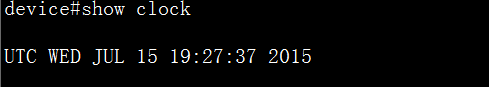
Device(config)#time-range time-range-work

Device(config-time-range)#periodic daily 8:00 to 18:00

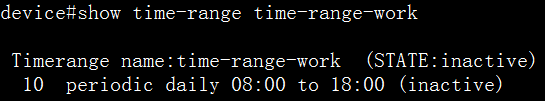
//配置时间域time-range-work，时间域的范围为每天8：00到18：00

Device(config-time-range)#exit

#在Device上查看当前系统时间。



在Device上查看定义的时间域time-range-work信息。



配置ACL列表1001。

device(config)#ip access-list extended 1001

device(config-ext-nacl)#deny ip 131.44.2.0 0.0.0.255 any

//阻止网段131.44.2.0/24访问IP Network

device(config-ext-nacl)#permit tcp host 131.44.1.2 any eq ftp

//允许PC2能通过FTP访问IP Network。

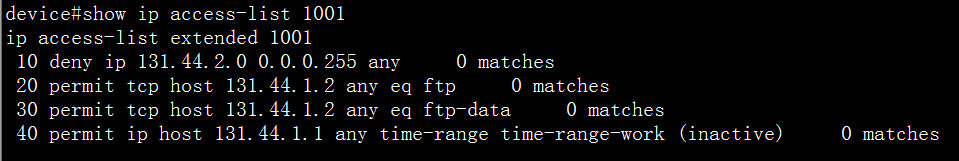
device(config-ext-nacl)#permit tcp host 131.44.1.2 any eq ftp-data

device(config-ext-nacl)#permit ip host 131.44.1.1 any time-range time-range-work

//允许PC1在定义的时间域time-range-work范围内访问IP Network

device(config-ext-nacl)#exit

在Device上查看acl列表1001的信息。



将已配置的acl 1001应用于接口gigabitethernet1的出方向。

device(config)#interface g1

device(config-if-gigabitethernet1)#ip access-group 1001 out

device(config-if-gigabitethernet1)#exit

**在Device上查看ACL应用于接口的信息**

Device#show access-lists interface gigabitethernet 1

-----------Interface-----Bind-----Instance--------------

Interface----------------Direction----AclType----AclName

gigabitethernet1         OUT          IP         1001

detail:

ip access-list extended 1001

 10 deny ip 131.44.2.0 0.0.0.255 any

 20 permit tcp host 131.44.1.2 any eq ftp

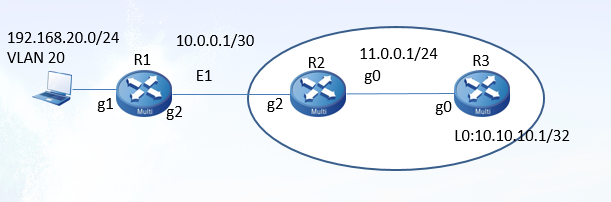
 30 permit tcp host 131.44.1.2 any eq ftp-data

 40 permit ip host 131.44.1.1 any time-range time-range-work (active)

**检验结果**：PC1在每天的8：00到18：00能访问IP Network；PC2能通过FTP访问IP Network；PC3不能访问IP Network。

## NAT实验

网络拓扑如图：



实验要求：

R1与R2通过128K的E1链路互联，R2与R3之间运行OSPF路由协议，R2与R1之间使用静态路由。

要求：

1. R1上需要将PC所在地址段转换为E1接口的地址

2. PC1可以ping通R3的loopback0接口；

R1配置：

router#config t

router(config)#hostname R1

R1(config)#interface gigabitethernet 1

R1(config-if-gigabitethernet1)#ip address 192.168.20.254 24

//R1连接PC端接口地址

R1(config)#interface gigabitethernet 2

R1(config-if-gigabitethernet2)#ip address 10.0.0.1 30 //R1连接R2接口地址

R1(config)#ip access-list standard 1 //配置标准访问列表

R1(config-std-nacl)#permit 192.168.20.0 0.0.0.255 //匹配PC地址段

R1(config-std-nacl)#exit

R1(config)#ip nat inside source list 1 interface gigabitethernet 2 overload

//将ACL匹配的地址段转换为E1接口地址。

R1(config)#interface gigabitethernet 1

R1(config-if-gigabitethernet1)#ip nat inside //设置g1口为inside

R1(config-if-gigabitethernet1)#exit

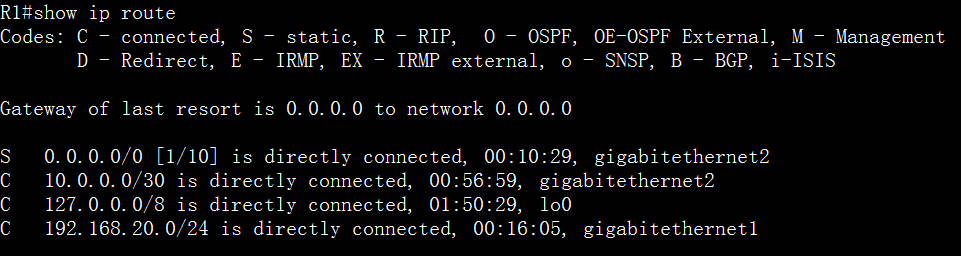
R1(config)#interface gigabitethernet 2

R1(config-if-gigabitethernet2)#ip nat outside //设置g2口为outside

R1(config-if-gigabitethernet2)#exit

R1(config)#ip route 0.0.0.0 0.0.0.0 gigabitethernet2 //配置静态路由指向g2口

R1路由表如下图所示：



R2配置：

R2(config)#interface gigabitethernet 2

R2(config-if-gigabitethernet2)#ip address 10.0.0.2 30

R2(config)#interface gigabitethernet 0

R2(config-if-gigabitethernet0)#ip address 11.0.0.1 24

R2(config-if-gigabitethernet0)#exit

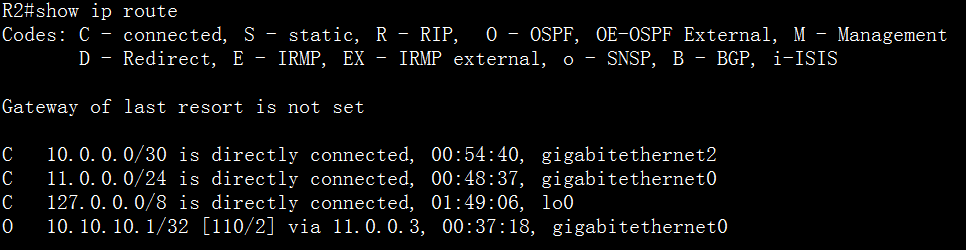
R2(config)#router ospf 1 //开启OSPF进程

R2(config-ospf)#router-id 2.2.2.2 //设置OSPF路由器ID

R2(config-ospf)#network 11.0.0.1 0.0.0.255 area 0 //将该网段宣告进OSPF

R2(config-ospf)# redistribute connected //将直连路由重分发进OSPF

R2路由表如图所示：



R3配置：

R3(config)#interface g0

R3(config-if-gigabitethernet0)#ip address 11.0.0.3 24

R3(config-if-gigabitethernet0)#exit

R3(config)#interface loopback 0 //创建环回口

R3(config-if-loopback0)#ip address 10.10.10.1 32

R3(config)#router ospf 1

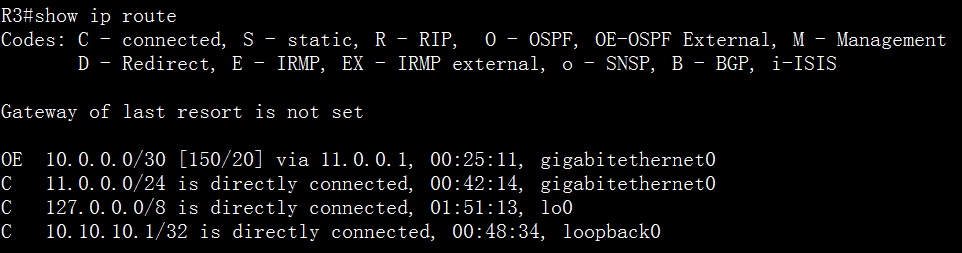
R3(config-ospf)# router-id 3.3.3.3

R3(config-ospf)#network 11.0.0.3 0.0.0.255 area 0

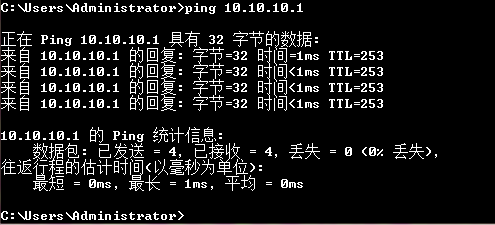
R3(config-ospf)#network 10.10.10.1 0.0.0.0 area 0

R3(config-ospf)#exit

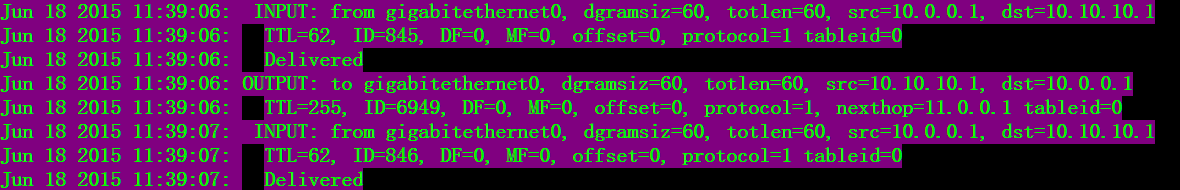
R3路由表如下图



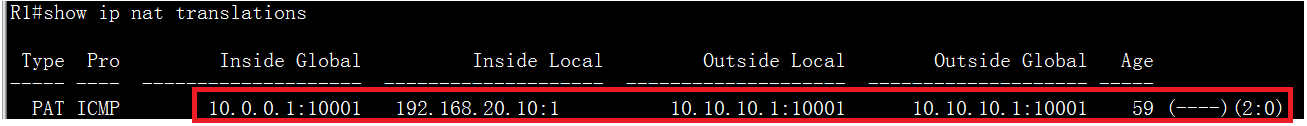
在PC机上ping 10.10.10.1 ,在R3上开启debug信息



R3#debug ip packet



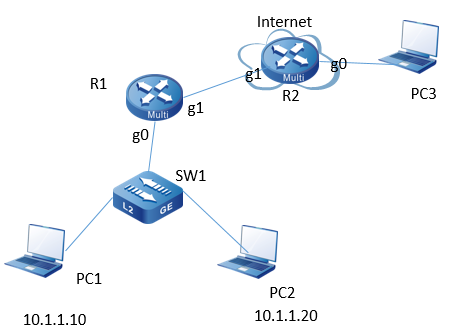
在R1上通过show ip nat translation查看地址转换情况：



可以看到PC1在ping R3上的loopback0地址时，R1将PC源地址转换为了E1口10.0.0.1。

## NAT端口映射

网络拓扑如图



实验要求：

1. PC1能通过R1 NAT后上外网访问PC3，PC2不能访问PC3
2. PC3能访问用4120模拟的web server

实验配置：

接口地址配置：

R1(config)#interfac g0

R1(config-if-gigabitethernet0)#ip address 10.1.1.1 24

R1(config-if-gigabitethernet0)#exit

R1(config)#interface g1

R1(config-if-gigabitethernet1)#ip address 12.1.1.1 24

R1(config-if-gigabitethernet1)#exit

R2(config)#interface g1

R2(config-if-gigabitethernet1)#ip address 12.1.1.2 24

R2(config-if-gigabitethernet1)#exit

R2(config)#interface g0

R2(config-if-gigabitethernet0)#ip address 23.1.1.2 24

R2(config-if-gigabitethernet0)#exit

NAT配置：

R1(config)#ip access-list standard 1

R1(config-std-nacl)# permit host 10.1.1.10

R1(config-std-nacl)# deny host 10.1.1.20

R1(config-std-nacl)# permit any

R1(config-std-nacl)# exit

R1(config)# ip nat inside source list 1 interface gigabitethernet1 overload

//将ACL匹配的地址通过g1口进行端口转换。

R1(config)#ip nat inside source static tcp 10.1.1.50 80 12.1.1.1 80

//将http server的地址通过80端口映射出去

R1(config)# ip route 0.0.0.0 0.0.0.0 12.1.1.2

在4120上开启http服务：

SW1(config)#ip http server

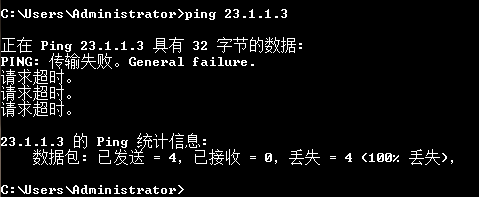
SW1(config)#interface vlan1

SW1(config-if-vlan1)# ip address 10.1.1.50 255.255.255.0 //给SW1一个对外的http地址

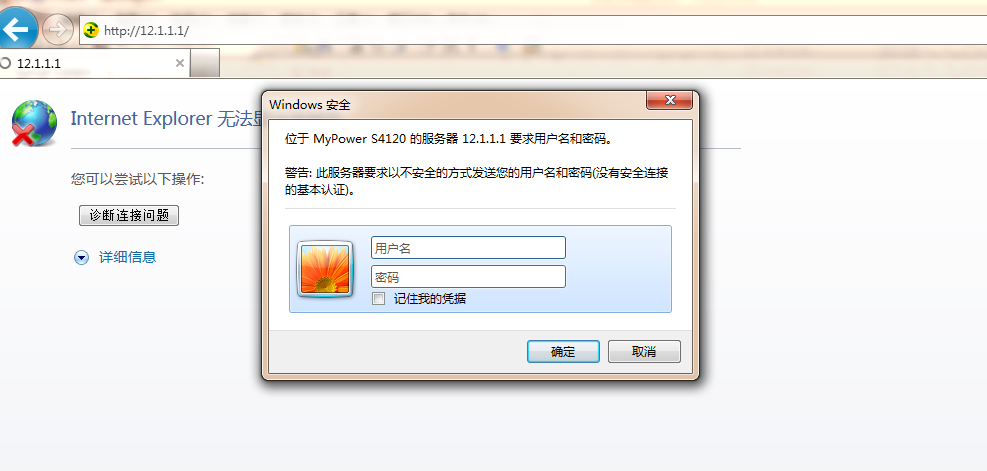
SW1(config)#ip route 0.0.0.0 0.0.0.0 10.1.1.1 //配置一条默认路由



PC1能上外网与PC3互通。



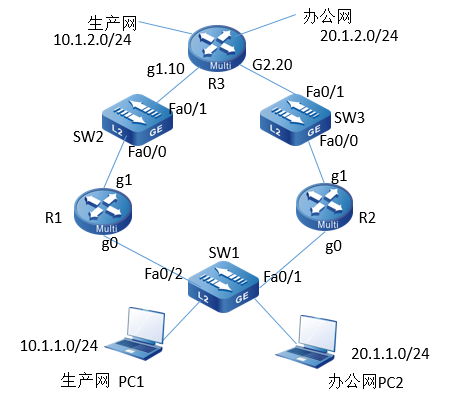
PC2不能上外网，不能与PC3互通。



在PC3上登陆12.1.1.1，R1将会映射到SW的http服务。

## 安全综合实验

实验拓扑如下所示：



实验要求：

1>通过VRRP实现生产网走左边，办公网走右边。且互为备份。

2>VRRP需跟踪路由器上行链路，一旦发生链路故障，需切换到备份链路。

3>PC1和PC2通过DHCP获取ip地址。

4>生产网和办公网不能互访。

（1）接口地址配置：

R1(config)#interface gigabitethernet1

R1(config-if-gigabitethernet1)# ip address 13.1.1.1 255.255.255.0

R2(config)#interface gigabitethernet1

R2(config-if-gigabitethernet1)# ip address 23.1.1.2 255.255.255.0

R3(config)#interface loopback1

R3(config-if-loopback1)# ip address 10.1.2.3 255.255.255.0

R3(config-if-loopback1)# exit

R3(config)#interface loopback2

R3(config-if-loopback2)# ip address 20.1.2.3 255.255.255.0

R3(config-if-loopback2)# exit

（2）VRRP分流互备及其路由配置

R1(config)#vlan 10,20 //创建vlan10,20

R1(config)#interface gigabitethernet0.10 //创建子接口

R1(config-if-gigabitethernet0.10)# ip address 10.1.1.2 255.255.255.0

R1(config-if-gigabitethernet0.10)# **encapsulation dot1q 10**

**//子接口封装802.1q，并对应vlan10**

R1(config-if-gigabitethernet0.10)# **vrrp 10 ip 10.1.1.1**

**//配置vrrp 10 ip 地址**

R1(config-if-gigabitethernet0.10)# **vrrp 10 priority 150**

**//设置vrrp优先级为150**

R1(config-if-gigabitethernet0.10)# ***vrrp 10 track gigabitethernet1 60***

***\*\*//追踪g1接口，当g1接口down掉后，其vrrp优先级自动降低60***

R1(config-if-gigabitethernet0.10)#vrrp 10 preempt //开启vrrp抢占功能

R1(config-if-gigabitethernet0.10)#exit

R1(config)#interface gigabitethernet0.20

R1(config-if-gigabitethernet0.20)#ip address 20.1.1.2 255.255.255.0

R1(config-if-gigabitethernet0.20)#encapsulation dot1q 20

R1(config-if-gigabitethernet0.20)#vrrp 20 ip 20.1.1.1

R1(config-if-gigabitethernet0.20)#vrrp 20 preempt

R1(config-if-gigabitethernet0.20)#exit

R1(config)#interface gigabitethernet1

R1(config-if-gigabitethernet1)#**keepalive gateway 13.1.1.3**

**//保活网关功能，能够检测上联网段是否down掉**

R1(config-if-gigabitethernet1)#exit

R1(config)#ip route 0.0.0.0 0.0.0.0 gigabitethernet1

R2(config)#vlan 10,20

R2(config)#interface gigabitethernet0.20

R2(config-if-gigabitethernet0.20)# ip address 20.1.1.3 255.255.255.0

R2(config-if-gigabitethernet0.20)# encapsulation dot1q 20

R2(config-if-gigabitethernet0.20)# vrrp 20 ip 20.1.1.1

R2(config-if-gigabitethernet0.20)# vrrp 20 priority 150

R2(config-if-gigabitethernet0.20)# vrrp 20 track gigabitethernet1 60

R2(config-if-gigabitethernet0.20)#vrrp 20 preempt

R2(config-if-gigabitethernet0.20)#exit

R2(config)#interface gigabitethernet0.10

R2(config-if-gigabitethernet0.10)# ip address 10.1.1.3 255.255.255.0

R2(config-if-gigabitethernet0.10)# encapsulation dot1q 10

R2(config-if-gigabitethernet0.10)# vrrp 10 ip 10.1.1.1

R2(config-if-gigabitethernet0.10)#vrrp 10 preempt

R2(config)#interface gigabitethernet1

R2(config-if-gigabitethernet1)# keepalive gateway 23.1.1.3

R2(config-if-gigabitethernet1)# exit

R2(config)#ip route 0.0.0.0 0.0.0.0 gigabitethernet1

R3(config)#interface loopback1

R3(config-if-loopback1)# ip address 10.1.2.3 255.255.255.0

R3(config-if-loopback1)# exit

R3(config)#interface loopback2

R3(config-if-loopback2)# ip address 20.1.2.3 255.255.255.0

R3(config-if-loopback2)# exit

R3(config)#interface gigabitethernet1.10

**//此处使用子接口是因为下联可能有多个网点。**

R3(config-if-gigabitethernet1.10)# ip address 13.1.1.3 255.255.255.0

R3(config-if-gigabitethernet1.10)# keepalive gateway 13.1.1.1

R3(config-if-gigabitethernet1.10)# encapsulation dot1q 10

R3(config-if-gigabitethernet1.10)# exit

R3(config)#interface gigabitethernet2.20

R3(config-if-gigabitethernet2.20)# ip address 23.1.1.3 255.255.255.0

R3(config-if-gigabitethernet2.20)# keepalive gateway 23.1.1.2

R3(config-if-gigabitethernet2.20)# encapsulation dot1q 20

R3(config-if-gigabitethernet2.20)# exit

R3(config)#ip route 10.1.1.0 255.255.255.0 gigabitethernet1.10

R3(config)#ip route 10.1.1.0 255.255.255.0 gigabitethernet2.20 5

R3(config)#ip route 20.1.1.0 255.255.255.0 gigabitethernet2.20

R3(config)#ip route 20.1.1.0 255.255.255.0 gigabitethernet1.10 5

R3(config)# ip access-list extended 1001

**//使用扩展访问列表，阻止生产和办公网段互访。**

R3(config-ext-nacl)#deny ip 10.1.0.0 0.0.255.255 20.1.0.0 0.0.255.255

R3(config-ext-nacl)#deny ip 20.1.0.0 0.0.255.255 10.1.0.0 0.0.255.255

R3(config-ext-nacl)#permit ip any any

R3(config-ext-nacl)#exit

R3(config)#interface gigabitethernet1.10

R3(config-if-gigabitethernet1.10)#ip access-group 1001 in

**//接口in方向调用ACL 1001**

R3(config-if-gigabitethernet1.10)#exit

R3(config)#interface gigabitethernet2.20

R3(config-if-gigabitethernet1.10)#ip access-group 1001 in

（3）交换机配置及其DHCP配置

R1(config)#ip dhcp pool p10 //配置DHCP地址池p10

R1(dhcp-config)# range 10.1.1.10 10.1.1.100 255.255.255.0

**//供分配的地址段为10.1.1.10到10.1.1.100**

R1(dhcp-config)# default-router 10.1.1.1

R1(dhcp-config)# exit

R2(config)#ip dhcp pool p20

R2(dhcp-config)# range 20.1.1.10 20.1.1.20 255.255.255.0

R2(dhcp-config)# default-router 20.1.1.1

R2(dhcp-config)# exit

SW1(config)#interface fastethernet0/1

SW1(config-if-fastethernet0/1)# switchport mode trunk

SW1(config-if-fastethernet0/1)# switchport trunk allowed vlan all

SW1(config-if-fastethernet0/1)# exit

SW1(config)#interface fastethernet0/2

SW1(config-if-fastethernet0/2)# switchport mode trunk

SW1(config-if-fastethernet0/2)# switchport trunk allowed vlan all

SW1(config-if-fastethernet0/2)# exit

SW1(config)#vlan 10,20

SW1(config)#interface fa0/10-0/16

SW1(config-if-range)#switchport access

SW1(config-if-range)#switchport access vlan 10

SW1(config-if-range)#exit

SW1(config)#interface fa0/17-0/24

SW1(config-if-range)#switchport access

SW1(config-if-range)#switchport access vlan 20

SW1(config-if-range)#exit

SW2(config)#vlan 10

SW2(config)#interface fastethernet0/0

SW2(config-if-fastethernet0/0)# switchport access vlan 10

SW2(config-if-fastethernet0/0)# exit

SW2(config)#interface fastethernet0/1

SW2(config-if-fastethernet0/1)# switchport mode trunk

SW2(config-if-fastethernet0/1)# switchport trunk allowed vlan all

SW2(config-if-fastethernet0/1)# exit

SW2(config)#interface vlan10

SW2(config-if-vlan10)# ip address 13.1.1.50 255.255.255.0

SW2(config-if-vlan10)# exit

SW3(config)#vlan 20

SW3(config)#interface fastethernet0/0

SW3(config-if-fastethernet0/0)# switchport access vlan 20

SW3(config-if-fastethernet0/0)# exit

SW3(config)#interface fastethernet0/1

SW3(config-if-fastethernet0/1)# switchport mode trunk

SW3(config-if-fastethernet0/1)# switchport trunk allowed vlan all

SW3(config-if-fastethernet0/1)# exit

SW3(config)#interface vlan20

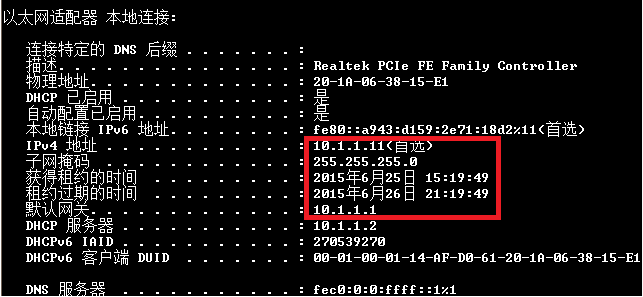
SW3(config-if-vlan20)# ip address 23.1.1.50 255.255.255.0

SW3(config-if-vlan20)# exit

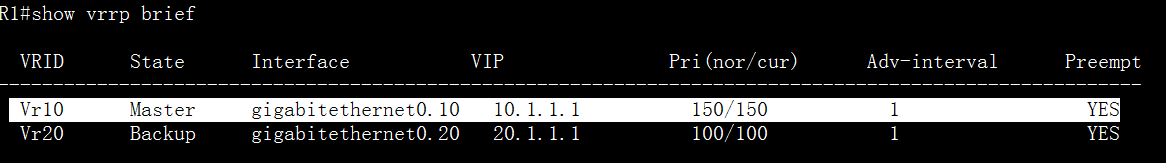
（4）验证

**DHCP验证：**

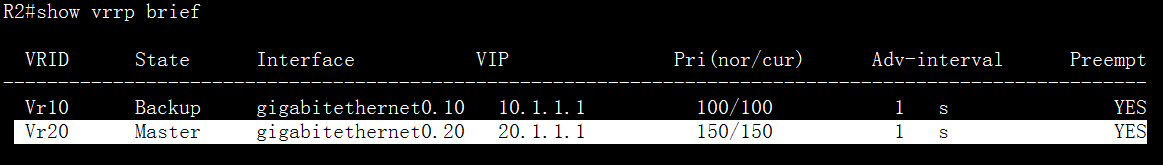
在PC1上设置ip地址自动获取。然后打开PC1的Cmd 。输入ipconfig/all。可以看到PC1获取到了10.1.1.11地址。



VRRP验证：

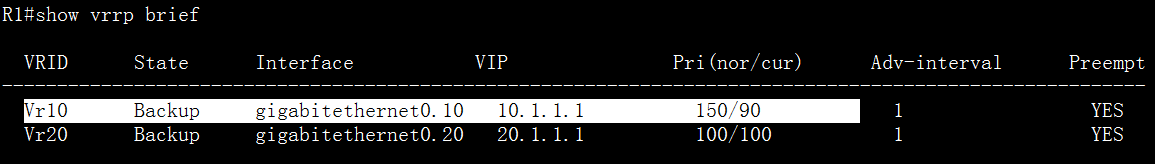


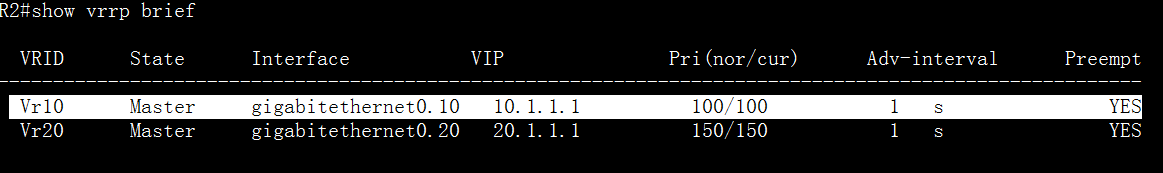
对于10.1.1.0/24网段来说，正常情况下该网段的master是R1。



对于20.1.1.0/24网段来说，正常情况下该网段的master是R2。

现在down掉R1的上联链路。可以看到，对于网段10.1.1.0/24。R1切换到了backup状态。且优先级降为了90。备份路由器R2则成为了master。

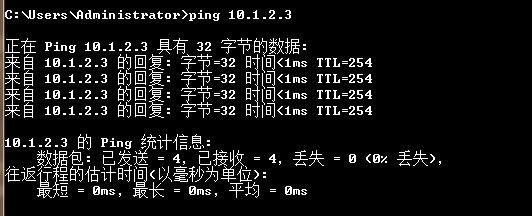




验证生产网路径：



上图显示生产网路径正常情况下为R1-R3。然后down掉R1的上行链路。





线路已切换到了备份路径R2-R3。

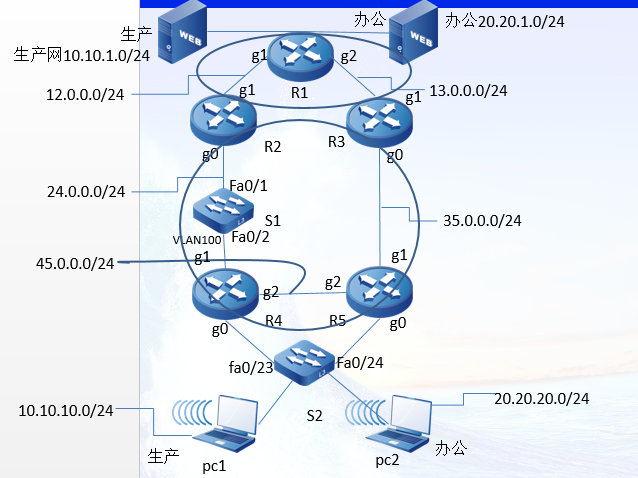
验证生产办公网之间不能互访。恢复上面R1 down掉的链路。



生产网PC1不能访问办公网。

## 分流备份综合实验

实验拓扑如下所示：



实验要求：

某单位网络拓扑如上，根据需求搭建模拟环境实现如下要求。

1、自行规划拓扑中所有设备IP地址，在拓扑上标出接口以及IP地址。

2、R1、R2、R3模拟分行网络运行ospf动态路由协议；R2、R3、R4、R5、运行RIP协议。

3、S1为二层交换机，模拟运营商MSTP线路，连接R4的接口为access口属于VLAN 100；S1连接R2接口为trunk口。

4、PC1为生产PC，网关正常时在R4上，当R2和S1之间线路中断时，PC1的网关在R5上，PC2为办公PC，正常时在网关R5上，当R3和R5之间线路中断后，切换到R4上。

4、R4、R5模拟网点接入路由器，要求实现分流互备。即PC1访问生产服务器走左边，PC2访问办公服务器走右边，当任意一边线路出现故障时，另外一边线路作为备份。

（1）各设备的地址规划如图

各接口地址配置如下：

**R1接口配置**

R1(config)#interface loopback0

R1(config-if-loopback0)# ip address 1.1.1.1 255.255.255.255

R1(config-if-loopback0)# exit

R1(config)#interface loopback10

R1(config-if-loopback10)# ip address 10.10.1.1 255.255.255.255

R1(config-if-loopback10)# exit

R1(config)#interface loopback20

R1(config-if-loopback20)# ip address 20.20.1.1 255.255.255.255

R1(config-if-loopback20)# exit

R1(config)#interface gigabitethernet1

R1(config-if-gigabitethernet1)# ip address 12.0.0.1 255.255.255.0

R1(config-if-gigabitethernet1)# exit

R1(config)#interface gigabitethernet2

R1(config-if-gigabitethernet2)# ip address 13.0.0.1 255.255.255.0

R1(config-if-gigabitethernet2)# exit

**R2接口配置**

R2(config)#interface loopback0

R2(config-if-loopback0)# ip address 2.2.2.2 255.255.255.255

R2(config-if-loopback0)# exit

R2(config)#interface gigabitethernet0.100 **//启用子接口**

R2(config-if-gigabitethernet0.100)# ip address 24.0.0.2 255.255.255.0

R2(config-if-gigabitethernet0.100)# **encapsulation dot1q 100**

//封装dot1q协议，关联到vlan100

R2(config-if-gigabitethernet0.100)# exit

R2(config)#interface gigabitethernet1

R2(config-if-gigabitethernet1)# ip address 12.0.0.2 255.255.255.0

R2(config-if-gigabitethernet1)# exit

**R3接口配置**

R3(config)#interface loopback0

R3(config-if-loopback0)# ip address 3.3.3.3 255.255.255.255

R3(config-if-loopback0)# exit

R3(config)#interface gigabitethernet0

R3(config-if-gigabitethernet0)# ip address 35.0.0.3 255.255.255.0

R3(config-if-gigabitethernet0)# exit

R3(config)#interface gigabitethernet1

R3(config-if-gigabitethernet1)# ip address 13.0.0.3 255.255.255.0

R3(config-if-gigabitethernet1)# exit

**R4接口配置**

R4(config)#interface loopback0

R4(config-if-loopback0)# ip address 4.4.4.4 255.255.255.255

R4(config-if-loopback0)# exit

R4(config)#interface gigabitethernet1

R4(config-if-gigabitethernet1)# ip address 24.0.0.4 255.255.255.0

R4(config-if-gigabitethernet1)#exit

R4(config)#interface gigabitethernet2

R4(config-if-gigabitethernet2)# ip address 45.0.0.4 255.255.255.0

R4(config-if-gigabitethernet2)# exit

R5接口配置

R5(config)#interface loopback0

R5(config-if-loopback0)# ip address 5.5.5.5 255.255.255.255

R5(config-if-loopback0)# exit

R5(config)#interface gigabitethernet1

R5(config-if-gigabitethernet1)# ip address 35.0.0.5 255.255.255.0

R5(config-if-gigabitethernet1)#exit

R5(config)#interface gigabitethernet2

R5(config-if-gigabitethernet2)# ip address 45.0.0.5 255.255.255.0

R5(config-if-gigabitethernet2)# exit

（2）OSPF、RIP协议配置

R1(config)#router ospf 1

R1(config-ospf)# router-id 1.1.1.1 //设置router-id

R1(config-ospf)# network 1.1.1.1 0.0.0.0 area 0

R1(config-ospf)# network 10.10.1.0 0.0.0.255 area 0

R1(config-ospf)# network 12.0.0.0 0.0.0.255 area 0

R1(config-ospf)# network 13.0.0.0 0.0.0.255 area 0

R1(config-ospf)# network 20.20.1.0 0.0.0.255 area 0

R2(config)#router rip

R2(config-rip)# version 2

R2(config-rip)# network 24.0.0.0

R2(config-rip)# **redistribute ospf 1 //将OSPF 1重分发进rip**

R2(config-rip)# no auto-summary

R2(config-rip)# exit

R2(config)#router ospf 1

R2(config-ospf)# router-id 2.2.2.2

R2(config-ospf)# network 2.2.2.2 0.0.0.0 area 0

R2(config-ospf)# network 12.0.0.0 0.0.0.255 area 0

R2(config-ospf)# **redistribute rip route-map B**

**//将rip重分发进OSPF 1 并调用route-map B**

R2(config-ospf)# exit

R3(config)#router rip

R3(config-rip)# version 2

R3(config-rip)# network 35.0.0.0

R3(config-rip)# **redistribute ospf 1 //将OSPF重分发进rip**

R3(config-rip)# no auto-summary

R3(config-rip)# exit

R3(config)#router ospf 1

R3(config-ospf)# router-id 3.3.3.3

R3(config-ospf)# network 3.3.3.3 0.0.0.0 area 0

R3(config-ospf)# network 13.0.0.0 0.0.0.255 area 0

R3(config-ospf)# r**edistribute rip route-map A**

**//将rip重分发进OSPF并调用route-map A**

R4(config)#router rip

R4(config-rip)# version 2

R4(config-rip)# network 4.0.0.0

R4(config-rip)# network 10.0.0.0

R4(config-rip)# network 20.0.0.0

R4(config-rip)# network 24.0.0.0

R4(config-rip)# network 45.0.0.0

R5(config)#router rip

R5(config-rip)# version 2

R5(config-rip)# network 5.0.0.0

R5(config-rip)# network 10.0.0.0

R5(config-rip)# network 20.0.0.0

R5(config-rip)# network 35.0.0.0

R5(config-rip)# network 45.0.0.0

（3）前缀列表、route-map工具分流

R2(config)#***ip prefix-list B seq 5 permit 20.20.0.0/16 ge 24 le 24***

**//使用前缀列表，名称为B，序列号为5，抓路由前缀20.20.0.0**

R2(config)#route-map B permit 10 //设置route-map

R2(config-route-map)# **match ip address prefix-list B //匹配前缀列表B**

R2(config-route-map)# set metric 25 //设置metric值为25

R2(config-route-map)# exit

R2(config)#**route-map B permit 20 //空route-map表示对其余做默认处理**

R2(config-route-map)# exit

R3(config)#ip prefix-list A seq 5 permit 10.10.0.0/16 ge 24 le 24

R3(config)#route-map A permit 10

R3(config-route-map)# match ip address prefix-list A

R3(config-route-map)# set metric 25

R3(config-route-map)# exit

R3(config)#route-map A permit 20

R3(config-route-map)# exit

（4）VRRP协议实现分流互备

R4(config)#ip access-list standard A //ACL抓流量

R4(config-std-nacl)# 10 permit 10.10.0.0 0.0.255.255

R4(config-std-nacl)# exit

R4(config)#ip access-list standard B

R4(config-std-nacl)# 10 permit 20.20.0.0 0.0.255.255

R4(config-std-nacl)# exit

R4(config)#interface gigabitethernet0.10 \*\*//启用g0口子接口0.10

R4(config-if-gigabitethernet0.10)# ip address 10.10.10.2 255.255.255.0 //子接口设置地址

R4(config-if-gigabitethernet0.10)# **encapsulation dot1q 10**

**\*\*//子接口封装dot1q，对应vlan10**

R4(config-if-gigabitethernet0.10)# ip access-group A in

R4(config-if-gigabitethernet0.10)# ip access-group A out

R4(config-if-gigabitethernet0.10)# **vrrp 10 ip 10.10.10.1**

**\*\*//在子接口启用vrrp协议，并设置其地址**

R4(config-if-gigabitethernet0.10)#**vrrp 10 priority 150**

**\*\*//设置vrrp优先级为150，使其成为10.10.10.0网段的主路由器**

R4(config-if-gigabitethernet0.10)#vrrp 10 preempt //启用vrrp抢占模式

R4(config-if-gigabitethernet0.10)# **vrrp 10 track gigabitethernet1 60**

**\*\*//监控上联g1接口，当g1口down掉后，其vrrp优先级自动降低60**

R4(config-if-gigabitethernet0.10)# exit

R4(config)#interface gigabitethernet0.20

\*\*//启用子接口0.20

R4(config-if-gigabitethernet0.20)#ip address 20.20.20.3 255.255.255.0

R4(config-if-gigabitethernet0.20)# encapsulation dot1q 20

\*\*//封装dot1q，对应vlan20

R4(config-if-gigabitethernet0.20)# ip access-group B in

R4(config-if-gigabitethernet0.20)# ip access-group B out

R4(config-if-gigabitethernet0.20)# vrrp 20 ip 20.20.20.1

\*\*//对于20.20.20.0网段，R4充当备份路由器

R4(config-if-gigabitethernet0.20)#vrrp 20 preempt

R4(config)#interface gigabitethernet1

R4(config-if-gigabitethernet1)# **keepalive gateway 24.0.0.2**

**\*\*//跟踪网关接口24.0.0.2，检测网关接口是否down**

R4(config-if-gigabitethernet1)# exit

R4(config)#router rip

R4(config-rip)# **offset-list B in 3 gigabitethernet0.20**

**//对于ACL B的路由，在子接口0.20的入方向上增加3跳**

R4(config-rip)# offset-list B in 5 gigabitethernet1

R4(config-rip)# offset-list B out 5 gigabitethernet1

R4(config-rip)# no auto-summary

**验证：**

在R4上查看VRRP对于网段10.10.10.0，R4是否成为主路由器

R4#show vrrp brief

VRID State Interface VIP Pri(nor/cur) Adv-interval Preempt

--------------------------------------------------------------------------------------------------------

Vr10  **Master**  gigabitethernet0.10 **10.10.10.1** 150/150 1 YES

Vr20  **Backup**  gigabitethernet0.20  **20.20.20.1** 100/100 1 YES

-------------------------------------------------------------------------------------------------------

可以看到对于vrrp 10/10.10.10.0 R4为Master ,vrrp20 R4为Backup。在R5上同理。

R5(config)#ip access-list standard A

R5(config-std-nacl)# 10 permit 10.10.0.0 0.0.255.255

R5(config-std-nacl)# exit

R5(config)#ip access-list standard B

R5(config-std-nacl)# 10 permit 20.20.0.0 0.0.255.255

R5(config-std-nacl)# exit

R5(config)#interface gigabitethernet0.10

R5(config-if-gigabitethernet0.10)# ip address 10.10.10.3 255.255.255.0

R5(config-if-gigabitethernet0.10)# encapsulation dot1q 10

R5(config-if-gigabitethernet0.10)# ip access-group A in

R5(config-if-gigabitethernet0.10)# ip access-group A out

R5(config-if-gigabitethernet0.10)# vrrp 10 ip 10.10.10.1

\*\*//对于网段10.10.10.0 R5充当备用网关

R5(config-if-gigabitethernet0.10)#vrrp 10 preempt

R5(config-if-gigabitethernet0.10)# exit

R5(config)#interface gigabitethernet0.20

R5(config-if-gigabitethernet0.20)#ip address 20.20.20.2 255.255.255.0

R5(config-if-gigabitethernet0.20)# encapsulation dot1q 20

R5(config-if-gigabitethernet0.20)# ip access-group B in

R5(config-if-gigabitethernet0.20)# ip access-group B out

R5(config-if-gigabitethernet0.20)# vrrp 20 ip 20.20.20.1

R5(config-if-gigabitethernet0.20)#**vrrp 20 priority 150**

**\*\*//对于网段20.20.20.0 ，通过设置vrrp优先级为150，使其成为主网关master**

R5(config-if-gigabitethernet0.20)# **vrrp 20 track gigabitethernet1 60**

**\*\*//监控g1接口，当g1口down掉以后，vrrp值自动降低60**

R5(config-if-gigabitethernet0.20)#vrrp 20 preempt

R5(config-if-gigabitethernet0.20)# exit

R5(config)#interface gigabitethernet1

R5(config-if-gigabitethernet1)# ip address 35.0.0.5 255.255.255.0

R5(config-if-gigabitethernet1)# keepalive gateway 35.0.0.3

R5(config-if-gigabitethernet1)# exit

R5(config)#router rip

R5(config-rip)# passive-interface gigabitethernet0.10

R5(config-rip)# passive-interface gigabitethernet0.20

R5(config-rip)# offset-list A in 3 gigabitethernet0.10

R5(config-rip)# offset-list A in 5 gigabitethernet1

R5(config-rip)# offset-list A out 5 gigabitethernet1

R5(config-rip)# no auto-summary

**验证：**

查看R5上vrrp状态。

R5#show vrrp brief

VRID State Interface VIP Pri(nor/cur) Adv-interval Preempt

--------------------------------------------------------------------------------------------------------

Vr10 **Backup** gigabitethernet0.10 **10.10.10.1**  100/100 1 YES

Vr20 **Master**  gigabitethernet0.20 **20.20.20.1** 150/150 1 YES

对于Vrrp10 R5为Backup，而对于Vrrp20 R5为Master。

交换机配置：

SW1配置：

S1(config)#vlan 100

S1(config)#interface fastethernet0/1

S1(config-if-fastethernet0/1)# switchport mode trunk //设置端口模式为Trunk

S1(config-if-fastethernet0/1)# switchport trunk allowed vlan all

//允许所有vlan通过

S1(config-if-fastethernet0/1)# exit

S1(config)#interface fastethernet0/2

S1(config-if-fastethernet0/2)#**switchport access vlan 100 //端口划分到vlan100**

S1(config-if-fastethernet0/2)# exit

SW2配置：

SW2(config)#vlan 10,20

SW2(config)#interface fa0/1-0/10 //批量划分接口fa0/1到fa0/10进入vlan10

SW2(config-if-range)#switchport access vlan 10

SW2(config)#interface fastethernet 0/11-0/20

//批量划分接口fa0/11到fa0/20进入vlan20

SW2(config-if-range)#switchport access vlan 20

SW2(config-if-range)#exit

SW2(config)#interface fastethernet0/23

SW2(config-if-fastethernet0/23)# switchport mode trunk //设置端口模式为Trunk

SW2(config-if-fastethernet0/23)# switchport trunk allowed vlan add 1,10,20

//允许vlan1，vlan10，vlan20 通过

SW2(config-if-fastethernet0/23)# exit

SW2(config)#interface fastethernet0/24

SW2(config-if-fastethernet0/24)# switchport mode trunk

SW2(config-if-fastethernet0/24)# switchport trunk allowed vlan add 1,10,20

SW2(config-if-fastethernet0/24)# exit

**验证结果：**

查看R1路由表

R1#show ip route

Codes: C - connected, S - static, R - RIP, O - OSPF, OE-OSPF External, M - Management

D - Redirect, E - IRMP, EX - IRMP external, o - SNSP, B - BGP, i-ISIS

**OE 10.10.10.0/24 [150/20] via 12.0.0.2, 00:05:38, gigabitethernet1**

**OE 20.20.20.0/24 [150/20] via 13.0.0.3, 00:01:39, gigabitethernet2**

OE 24.0.0.0/24 [150/20] via 12.0.0.2, 00:05:38, gigabitethernet1

以上显示的部分路由表中的黑色粗字体表明，服务器端生产、办公流量实现了分流。

在PC2上Ping 20.20.1.1是否能通。

C:\Users\Administrator>**ping 20.20.1.1**

正在 Ping 20.20.1.1 具有 32 字节的数据:

来自 20.20.1.1 的回复: 字节=32 时间<1ms TTL=253

来自 20.20.1.1 的回复: 字节=32 时间<1ms TTL=253

来自 20.20.1.1 的回复: 字节=32 时间<1ms TTL=253

来自 20.20.1.1 的回复: 字节=32 时间<1ms TTL=253

20.20.1.1 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 = 0ms，最长 = 0ms，平均 = 0ms

**在PC2上测试是到达20.20.1.1.的路径**

C:\Users\Administrator>**tracert 20.20.1.1**

通过最多 30 个跃点跟踪到 20.20.1.1 的路由

**1 <1 毫秒 <1 毫秒 <1 毫秒 20.20.20.2**

**2 <1 毫秒 <1 毫秒 <1 毫秒 35.0.0.3**

**3 <1 毫秒 <1 毫秒 <1 毫秒 20.20.1.1**

跟踪完成。

查看R5路由表。

表明办公网段正常情况下线路为R5-R3-R1。

现在我们down掉R5的上联接口，看路径是否切换到左边。

R5(config)#interface g1

R5(config-if-gigabitethernet1)#**shutdown**

**R5#show vrrp brief**

VRID State Interface VIP Pri(nor/cur) Adv-interval Preempt

--------------------------------------------------------------------------------------------------------

Vr10 Backup gigabitethernet0.10 10.10.10.1 100/100 1 YES

Vr20  **Backup**  gigabitethernet0.20 20.20.20.1 150/90 1 YES

-------------------------------------------------------------------------------------------------------

可以看到对于**vrrp20**，R5成为了**Backup**。

再看R4的vrrp状态：

R4#show vrrp brief

VRID State Interface VIP Pri(nor/cur) Adv-interval Preempt

--------------------------------------------------------------------------------------------------------

Vr10 Master gigabitethernet0.10 10.10.10.1 150/150 1 YES

**Vr20 Master** gigabitethernet0.20 20.20.20.1 100/100 1 YES

-------------------------------------------------------------------------------------------------------

可以看到vrrp20已经将R4切换为了**Master。**

**在PC2上再次ping 20.20.1.1**

C:\Users\Administrator>ping 20.20.1.1

正在 Ping 20.20.1.1 具有 32 字节的数据:

来自 20.20.1.1 的回复: 字节=32 时间<1ms TTL=253

来自 20.20.1.1 的回复: 字节=32 时间=1ms TTL=253

来自 20.20.1.1 的回复: 字节=32 时间<1ms TTL=253

来自 20.20.1.1 的回复: 字节=32 时间<1ms TTL=253

20.20.1.1 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 = 0ms，最长 = 1ms，平均 = 0ms

**在PC2上再次跟踪到达20.20.1.1的路径**。

C:\Users\Administrator>tracert 20.20.1.1

通过最多 30 个跃点跟踪到 20.20.1.1 的路由

1 <1 毫秒 <1 毫秒 <1 毫秒 **20.20.20.3**

2 \* \* \* 请求超时。

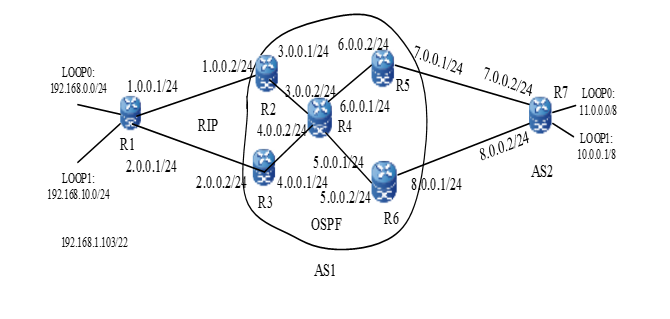
3 <1 毫秒 <1 毫秒 <1 毫秒 **20.20.1.1**

跟踪完成。

可以看到PC2已经成功将路径切换到了R4上。PC1的验证跟PC2完全一样。

## BGP综合实验

实验拓扑如下：



实验要求：

**环境说明：**

所有路由器、交换机、PC都采用以太连接，IP地址分配都已经在图中标出。

2、R1/R2/R3间采用rip路由协议，R2、R3、R4、R5、R6间采用ospf路由协议通告接口地址和loopback地址，通过loopback地址建立IBGP其中R4为RR，其他为RR客户端，R7分别同R5和R6建立EBGP。

3、在R2、R3上，在RIP和BGP之间互相重分发路由（注意互相重分发不要形成环路）

4、R2、R3、R4、R5、R6 使用的loopback地址分别为2.2.2.2、3.3.3.3、4.4.4.4、5.5.5.5、6.6.6.6。

**实现要求：**

1、在R7通过EBGP通告路由10.0.0.0/8、11.0.0.0/8给R5和R6；R1通过RIP通告路由192.168.0.0/24和192.168.10.0/24给R2、R3。（通过Loopback口模拟）

2、在R1－R7合适的设备上，通过一定的策略控制，使：

R1上Loopback0口访问10.0.0.0/8路径为：R1－>R2－>R4－>R5－>R7

R1上Loopback1访问11.0.0.0/8路径为：R1－>R3－>R4－>R6－>R7

R7上Loopback0访问192.168.0.0/24路径为：R7－>R5－>R4－>R2－>R1

R7上Loopback1访问192.168.10.0/24路径为：R7－>R6－>R4－>R3－>R1

并且R1到R2、或R1到R3一条线路中断，能切换到另一条线路上；R5到R7、或R6到R7一条线路中断，也能切换到另一条线路上。

3、路由192.168.0.0/24和192.168.10.0/24在各台BGP路由器传递中都带了团体属性， 192.168.0.0/24团体属性：1:100，192.168.10.0/24团体属性1:200。

实验配置：

R1(config)#ip access-list standard 100

R1(config-std-nacl)# 10 permit 11.0.0.0 0.255.255.255

R1(config-std-nacl)# exit

R1(config)#ip access-list standard 101

R1(config-std-nacl)# 10 permit 10.0.0.0 0.255.255.255

R1(config-std-nacl)# exit

R1(config)#interface loopback0

R1(config-if-loopback0)# ip address 192.168.10.1 255.255.255.0

R1(config-if-loopback0)# ip ospf network point-to-point

R1(config-if-loopback0)# exit

R1(config)#interface loopback1

R1(config-if-loopback1)# ip address 192.168.0.1 255.255.255.0

R1(config-if-loopback1)# ip ospf network point-to-point

R1(config-if-loopback1)# exit

R1(config)#interface gigabitethernet0

R1(config-if-gigabitethernet0)# ip address 1.0.0.1 255.255.255.0

R1(config-if-gigabitethernet0)# exit

R1(config)#interface gigabitethernet1

R1(config-if-gigabitethernet1)# ip address 2.0.0.1 255.255.255.0

R1(config-if-gigabitethernet1)# exit

R1(config)#router rip

R1(config-rip)# version 2

R1(config-rip)# network 1.0.0.0

R1(config-rip)# network 2.0.0.0

R1(config-rip)# redistribute connected //重分发直连路由进rip

R1(config-rip)# no auto-summary

R1(config-rip)# exit

**R2配置：**

R2(config)#ip access-list standard 100

R2(config-std-nacl)# 10 permit 192.168.0.0 0.0.3.255

R2(config-std-nacl)# exit

R2(config)#ip access-list standard 101

R2(config-std-nacl)# 10 permit 192.168.10.0 0.0.0.255

R2(config-std-nacl)# exit

R2(config)#ip access-list standard 200

R2(config-std-nacl)# 10 permit 11.0.0.0 0.255.255.255

R2(config-std-nacl)# exit

R2(config)#ip access-list standard 300

R2(config-std-nacl)# 10 permit 192.0.0.0 0.255.255.255

R2(config-std-nacl)# exit

R2(config)#ip access-list standard 100

R2(config-std-nacl)# 10 permit 192.168.0.0 0.0.3.255

R2(config-std-nacl)# exit

R2(config)#ip access-list standard 101

R2(config-std-nacl)# 10 permit 192.168.10.0 0.0.0.255

R2(config-std-nacl)# exit

R2(config)#ip access-list standard 200

R2(config-std-nacl)# 10 permit 11.0.0.0 0.255.255.255

R2(config-std-nacl)# exit

R2(config)#ip access-list standard 300

R2(config-std-nacl)# 10 permit 192.0.0.0 0.255.255.255

R2(config-std-nacl)# exit

R2(config)#interface loopback0

R2(config-if-loopback0)# ip address 2.2.2.2 255.255.255.255

R2(config-if-loopback0)# exit

R2(config)#interface gigabitethernet0

R2(config-if-gigabitethernet0)# ip address 1.0.0.2 255.255.255.0

R2(config-if-gigabitethernet0)# exit

R2(config)#interface gigabitethernet1

R2(config-if-gigabitethernet1)# ip address 3.0.0.1 255.255.255.0

R2(config-if-gigabitethernet1)# exit

R2(config)#router rip

R2(config-rip)# version 2

R2(config-rip)# network 1.0.0.0

R2(config-rip)# offset-list 200 out 5 gigabitethernet0

//偏移列表在g0接口in方向调用acl200，增加5跳

R2(config-rip)# redistribute bgp 1

R2(config-rip)# distribute-list 300 in gigabitethernet0

//分发列表调用acl300 在g0接口in方向

R2(config-rip)# no auto-summary

R2(config-rip)# exit

R2(config)#router ospf 1

R2(config-ospf)# network 2.2.2.2 0.0.0.0 area 0

R2(config-ospf)# network 3.0.0.1 0.0.0.0 area 0

R2(config-ospf)# exit

R2(config)#router bgp 1

R2(config-bgp)# no auto-summary

R2(config-bgp)# no synchronization //关闭BGP的同步规则

R2(config-bgp)# redistribute rip route-map comm //rip重分发进BGP

R2(config-bgp)# neighbor 4.4.4.4 remote-as 1 //指定建立BGP邻居及其AS号

R2(config-bgp)# neighbor 4.4.4.4 update-source loopback0 //指定更新源地址

R2(config-bgp)# neighbor 4.4.4.4 next-hop-self //指定更新BGP路由下一跳

R2(config-bgp)# neighbor 4.4.4.4 send-community //发送团体属性

R2(config-bgp)# neighbor 4.4.4.4 route-map comm out

R2(config-bgp)# exit

R2(config)#route-map comm permit 10

R2(config-route-map)# match ip address 100

R2(config-route-map)# set community 1:100

//对acl100抓取的路由设置团体属性

R2(config-route-map)# set metric 50

R2(config-route-map)# exit

R2(config)#route-map comm permit 20

R2(config-route-map)# match ip address 101

R2(config-route-map)# set community 1:200 //设置团体属性

R2(config-route-map)# set metric 100

R2(config-route-map)# exit

**R3配置：**

R3(config)#ip flow enable

R3(config)#ip access-list standard 100

R3(config-std-nacl)# 10 permit 192.168.0.0 0.0.3.255

R3(config-std-nacl)# exit

R3(config)#ip access-list standard 101

R3(config-std-nacl)# 10 permit 192.168.10.0 0.0.0.255

R3(config-std-nacl)# exit

R3(config)#ip access-list standard 200

R3(config-std-nacl)# 10 permit 10.0.0.0 0.255.255.255

R3(config-std-nacl)# exit

R3(config)#ip access-list standard 300

R3(config-std-nacl)# 10 permit 192.0.0.0 0.255.255.255

R3(config-std-nacl)# exit

R3(config)#interface loopback0

R3(config-if-loopback0)# ip address 3.3.3.3 255.255.255.255

R3(config-if-loopback0)# exit

R3(config)#interface gigabitethernet0

R3(config-if-gigabitethernet0)# ip address 4.0.0.1 255.255.255.0

R3(config-if-gigabitethernet0)# exit

R3(config)#interface gigabitethernet1

R3(config-if-gigabitethernet1)# ip address 2.0.0.2 255.255.255.0

R3(config-if-gigabitethernet1)# exit

R3(config)#router rip

R3(config-rip)# version 2

R3(config-rip)# network 2.0.0.0

R3(config-rip)# offset-list 200 out 5 gigabitethernet1

R3(config-rip)# redistribute bgp 1

R3(config-rip)# distribute-list 300 in gigabitethernet1

R3(config-rip)# no auto-summary

R3(config-rip)# exit

R3(config)#router ospf 1

R3(config-ospf)# router-id 3.3.3.3

R3(config-ospf)# network 3.3.3.3 0.0.0.0 area 0

R3(config-ospf)# network 4.0.0.1 0.0.0.0 area 0

R3(config-ospf)# exit

R3(config)#router bgp 1

R3(config-bgp)# no auto-summary

R3(config-bgp)# no synchronization

R3(config-bgp)# redistribute rip route-map comm

R3(config-bgp)# neighbor 4.4.4.4 remote-as 1

R3(config-bgp)# neighbor 4.4.4.4 update-source loopback0

R3(config-bgp)# neighbor 4.4.4.4 next-hop-self

R3(config-bgp)# neighbor 4.4.4.4 send-community

R3(config-bgp)# exit

R3(config)#route-map comm permit 10

R3(config-route-map)# match ip address 100

R3(config-route-map)# set community 1:100

R3(config-route-map)# set metric 100

R3(config-route-map)# exit

R3(config)#route-map comm permit 20

R3(config-route-map)# match ip address 101

R3(config-route-map)# set community 1:100

R3(config-route-map)# set metric 50

R3(config-route-map)# exit

**R4配置：**

R4(config)#vlan 3-4

R4(config)#interface fastethernet0/4

R4(config-if-fastethernet0/4)# switchport access vlan 3

R4(config-if-fastethernet0/4)# exit

R4(config)#interface fastethernet0/6

R4(config-if-fastethernet0/6)# switchport access vlan 4

R4(config-if-fastethernet0/6)# exit

R4(config)#interface loopback0

R4(config-if-loopback0)# ip address 4.4.4.4 255.255.255.255

R4(config-if-loopback0)# exit

R4(config)#interface gigabitethernet0

R4(config-if-gigabitethernet0)# ip address 4.0.0.2 255.255.255.0

R4(config-if-gigabitethernet0)# exit

R4(config)#interface gigabitethernet1

R4(config-if-gigabitethernet1)# ip address 3.0.0.2 255.255.255.0

R4(config-if-gigabitethernet1)# exit

R4(config)#interface vlan3

R4(config-if-vlan3)# ip address 6.0.0.1 255.255.255.0

R4(config-if-vlan3)# exit

R4(config)#interface vlan4

R4(config-if-vlan4)# ip address 5.0.0.1 255.255.255.0

R4(config-if-vlan4)# exit

R4(config)#router ospf 1

R4(config-ospf)# network 3.0.0.2 0.0.0.0 area 0

R4(config-ospf)# network 4.0.0.2 0.0.0.0 area 0

R4(config-ospf)# network 4.4.4.4 0.0.0.0 area 0

R4(config-ospf)# network 5.0.0.1 0.0.0.0 area 0

R4(config-ospf)# network 6.0.0.1 0.0.0.0 area 0

R4(config-ospf)# exit

R4(config)#router bgp 1

R4(config-bgp)# no auto-summary

R4(config-bgp)# no synchronization

R4(config-bgp)**# neighbor a peer-group //配置BGP对等体组**

R4(config-bgp)# neighbor a remote-as 1

R4(config-bgp)# neighbor a update-source loopback0

R4(config-bgp)# neighbor a route-reflector-client

R4(config-bgp)# neighbor a next-hop-self

R4(config-bgp)# neighbor a send-community

R4(config-bgp)# **neighbor 2.2.2.2 peer-group a //调用BGP对等体组**

R4(config-bgp)# neighbor 3.3.3.3 peer-group a

R4(config-bgp)# neighbor 5.5.5.5 peer-group a

R4(config-bgp)# neighbor 6.6.6.6 peer-group a

R4(config-bgp)# exit

**R5配置：**

R5(config)#ip access-list standard 100

R5(config-std-nacl)# 10 permit 10.0.0.0 0.255.255.255

R5(config-std-nacl)# exit

R5(config)#ip access-list standard 101

R5(config-std-nacl)# 10 permit 11.0.0.0 0.255.255.255

R5(config-std-nacl)# exit

R5(config)#vlan 3,7

R5(config)#interface fastethernet0/2

R5(config-if-fastethernet0/2)# switchport access vlan 7

R5(config-if-fastethernet0/2)#exit

R5(config)#interfa fa0/4

R5(config-if-fastethernet0/4)#switchport access vlan 3

R5(config-if-fastethernet0/4)#exit

R5(config)#interface loopback0

R5(config-if-loopback0)# ip address 5.5.5.5 255.255.255.255

R5(config-if-loopback0)# exit

R5(config)#interface vlan3

R5(config-if-vlan3)# ip address 6.0.0.2 255.255.255.0

R5(config-if-vlan3)# exit

R5(config)#interface vlan7

R5(config-if-vlan7)# ip address 7.0.0.1 255.255.255.0

R5(config-if-vlan7)# exit

R5(config)#router ospf 1

R5(config-ospf)# router-id 5.5.5.5

R5(config-ospf)# network 5.5.5.5 0.0.0.0 area 0

R5(config-ospf)# network 6.0.0.2 0.0.0.0 area 0

R5(config-ospf)# exit

R5(config)#router bgp 1

R5(config-bgp)# no auto-summary

R5(config-bgp)# no synchronization

R5(config-bgp)# network 7.0.0.0 255.255.255.0

R5(config-bgp)# neighbor 4.4.4.4 remote-as 1

R5(config-bgp)# neighbor 4.4.4.4 update-source loopback0

R5(config-bgp)# neighbor 4.4.4.4 next-hop-self

R5(config-bgp)# neighbor 4.4.4.4 send-community

R5(config-bgp)# neighbor 7.0.0.2 remote-as 2

R5(config-bgp)# neighbor 7.0.0.2 next-hop-self

R5(config-bgp)# neighbor 7.0.0.2 send-community

R5(config-bgp)# neighbor 7.0.0.2 route-map lp in

R5(config-bgp)# exit

R5(config)#route-map lp permit 10

R5(config-route-map)# match ip address 100

R5(config-route-map)# **set local-preference 200 //设置BGP路由本地优先级**

R5(config-route-map)# exit

R5(config)#route-map lp permit 20

R5(config-route-map)# match ip address 101

R5(config-route-map)# set local-preference 100

R5(config-route-map)# exit

R5(config)#route-map lp permit 30

R5(config-route-map)# exit

**R6配置：**

R6(config)#ip access-list standard 100

R6(config-std-nacl)# 10 permit 10.0.0.0 0.255.255.255

R6(config-std-nacl)# exit

R6(config)#ip access-list standard 101

R6(config-std-nacl)# 10 permit 11.0.0.0 0.255.255.255

R6(config-std-nacl)# exit

R6(config)#vlan 4,8

R6(config)#interface fastethernet0/4

R6(config-if-fastethernet0/4)# switchport access vlan 8

R6(config-if-fastethernet0/4)# exit

R6(config)#interface fastethernet0/6

R6(config-if-fastethernet0/6)# switchport access vlan 4

R6(config-if-fastethernet0/6)# exit

R6(config)#interface loopback0

R6(config-if-loopback0)# ip address 6.6.6.6 255.255.255.255

R6(config-if-loopback0)# exit

R6(config)#interface vlan4

R6(config-if-vlan4)# ip address 5.0.0.2 255.255.255.0

R6(config-if-vlan4)# exit

R6(config)#interface vlan8

R6(config-if-vlan8)# ip address 8.0.0.1 255.255.255.0

R6(config-if-vlan8)# exit

R6(config)#router ospf 1

R6(config-ospf)# network 5.0.0.2 0.0.0.0 area 0

R6(config-ospf)# network 6.6.6.6 0.0.0.0 area 0

R6(config-ospf)# exit

R6(config)#router bgp 1

R6(config-bgp)# no auto-summary

R6(config-bgp)# no synchronization

R6(config-bgp)# network 8.0.0.0 255.0.0.0

R6(config-bgp)# neighbor 4.4.4.4 remote-as 1

R6(config-bgp)# neighbor 4.4.4.4 update-source loopback0

R6(config-bgp)# neighbor 4.4.4.4 next-hop-self

R6(config-bgp)# neighbor 4.4.4.4 send-community

R6(config-bgp)# neighbor 8.0.0.2 remote-as 2

R6(config-bgp)# neighbor 8.0.0.2 next-hop-self

R6(config-bgp)# neighbor 8.0.0.2 send-community

R6(config-bgp)# neighbor 8.0.0.2 route-map lp in

R6(config-bgp)# exit

R6(config)#route-map lp permit 10

R6(config-route-map)# match ip address 101

R6(config-route-map)# set local-preference 200

R6(config-route-map)# exit

R6(config)#route-map lp permit 20

R6(config-route-map)# match ip address 100

R6(config-route-map)# set local-preference 100

R6(config-route-map)# exit

R6(config)#route-map lp permit 30

R6(config-route-map)# exit

**R7配置：**

R7(config)#ip access-list standard 100

R7(config-std-nacl)# 10 permit 192.168.0.0 0.0.3.255

R7(config-std-nacl)# exit

R7(config)#ip access-list standard 101

R7(config-std-nacl)# 10 permit 192.168.10.0 0.0.0.255

R7(config-std-nacl)# exit

R7(config)#vlan 7-8

R7(config)#interface fastethernet0/2

R7(config-if-fastethernet0/2)# switchport access vlan 7

R7(config-if-fastethernet0/2)# exit

R7(config)#interface fastethernet0/4

R7(config-if-fastethernet0/4)# switchport access vlan 8

R7(config-if-fastethernet0/4)# exit

R7(config)#interface loopback0

R7(config-if-loopback0)# ip address 11.0.0.1 255.0.0.0

R7(config-if-loopback0)# exit

R7(config)#interface loopback1

R7(config-if-loopback1)# ip address 10.0.0.1 255.0.0.0

R7(config-if-loopback1)# exit

R7(config)#interface vlan7

R7(config-if-vlan7)# ip address 7.0.0.2 255.255.255.0

R7(config-if-vlan7)# exit

R7(config)#interface vlan8

R7(config-if-vlan8)# ip address 8.0.0.2 255.255.255.0

R7(config-if-vlan8)# exit

R7(config)#router bgp 2

R7(config-bgp)# no auto-summary

R7(config-bgp)# no synchronization

R7(config-bgp)# network 10.0.0.0 255.0.0.0

R7(config-bgp)# network 11.0.0.0 255.0.0.0

R7(config-bgp)# neighbor b peer-group

R7(config-bgp)# neighbor b remote-as 1

R7(config-bgp)# neighbor b send-community

R7(config-bgp)# neighbor 7.0.0.1 peer-group b

R7(config-bgp)# neighbor 7.0.0.1 route-map lp in

R7(config-bgp)# neighbor 8.0.0.1 peer-group b

R7(config-bgp)# neighbor 8.0.0.1 route-map lp2 in

R7(config-bgp)# exit

R7(config)#route-map lp permit 10

R7(config-route-map)# match ip address 100

R7(config-route-map)# set local-preference 200

R7(config-route-map)# exit

R7(config)#route-map lp permit 20

R7(config-route-map)# match ip address 101

R7(config-route-map)# set local-preference 100

R7(config-route-map)# exit

R7(config)#route-map lp2 permit 10

R7(config-route-map)# match ip address 100

R7(config-route-map)# set local-preference 100

R7(config-route-map)# exit

R7(config)#route-map lp2 permit 20

R7(config-route-map)# match ip address 101

R7(config-route-map)# set local-preference 200

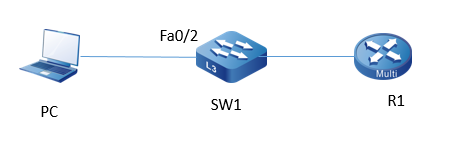
R7(config-route-map)# exit

# 第三章 交换实验



## 交换端口启用ip+mac安全绑定功能

实验拓扑如下：



实验要求：交换机打开端口安全功能，要求做IP+MAC绑定。必须为正确的ip和MAC对应才能ping通路由器。通过修改PC 的ip来验证。

SW1基本配置：

sw1(config)#interface vlan1

sw1(config-if-vlan1)# ip address 192.168.1.2 255.255.255.0

sw1(config-if-vlan1)# exit

sw1(config)#interface fastethernet0/2

sw1(config-if-fastethernet0/2)# **port-security enable //启用端口安全功能**

sw1(config-if-fastethernet0/2)# **port-security permit mac-address 201A.0638.15E1 ip-address 192.168.1.10 //绑定PC的mac地址和ip地址**

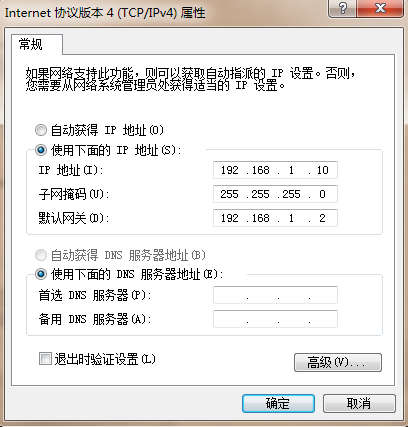
sw1(config-if-fastethernet0/2)# exit

R1(config)#interface gigabitethernet0

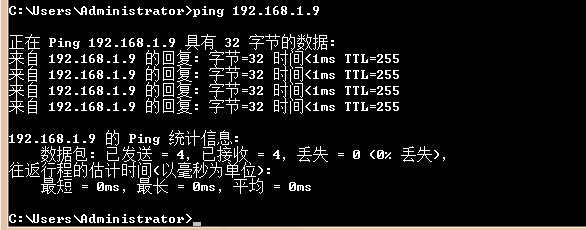
R1(config-if-gigabitethernet0)# ip address 192.168.1.9 255.255.255.0

R1(config-if-gigabitethernet0)# exit

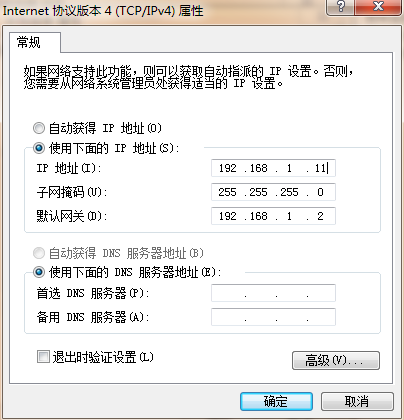
PC机设置：

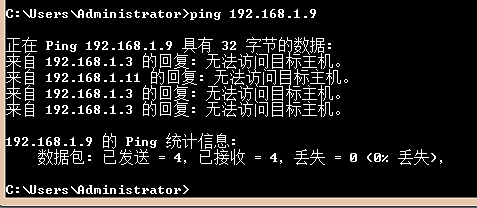


在PC上ping R1的地址



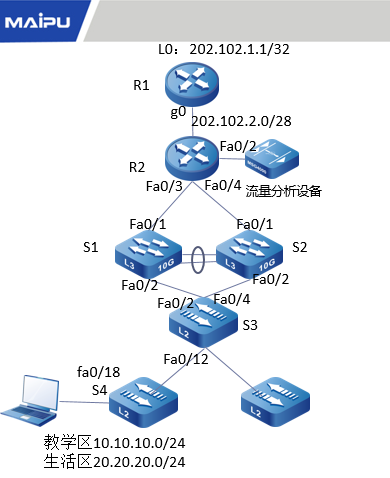
现在改变PC的地址，看能否ping通R1





当更改了PC的IP地址后，由于交换机开启了端口绑定功能，所以PC此时不能与R1相互通信。

## 交换综合实验



实验要求：

1>搭建如图的园区网络环境

2>R1模拟公网路由器，配置公网地址和环回地址，不配置任何路由

3>R2为园区网出口路由器，通过NAT实现内部网络访问外部网络，同时作为内部网络的DHCP服务器提供IP地址租赁服务。

4>R2通过交换端口连接核心交换机，便于通过端口镜像进行流量分析。

5>S1与S2为核心交换机，通过链路汇聚互联，作为内部网络的冗余网关（VRRP)

6>S3为汇聚交换机，汇聚接入层交换设备，与核心交换机之间启用MSTP规避环路实现VLAN间的负载分流（配合VRRP)

7>在S4开启端口安全。使用mac地址绑定方式。

8>全网路由可达，但教学区不能访问外网，且教学区与生活区不能互访。

实验基本配置如下：

1. R1模拟公网：

R1(config)#interface loopback0

R1(config-if-loopback0)# ip address 202.102.1.1 255.255.255.255

R1(config-if-loopback0)# exit

R1(config)#interface gigabitethernet0

R1(config-if-gigabitethernet0)# ip address 202.102.2.1 255.255.255.240

R1(config-if-gigabitethernet0)# exit

1. R2上配置NAT提供生活区上公网，DHCP为内网分配地址：

R2(config)#interface gigabitethernet0

R2(config-if-gigabitethernet0)# ip address 202.102.2.2 255.255.255.240

R2(config-if-gigabitethernet0)# ip nat outside

R2(config-if-gigabitethernet0)# exit

R2(config)#vlan 100,200

R2(config)#interface vlan100

R2(config-if-vlan100)# ip address 12.1.1.2 255.255.255.0

R2(config-if-vlan100)# ip nat inside

R2(config-if-vlan100)# exit

R2(config)#interface vlan200

R2(config-if-vlan200)# ip address 22.1.1.2 255.255.255.0

R2(config-if-vlan200)# ip nat inside

R2(config-if-vlan200)# exit

R2(config)#ip access-list standard 1

R2(config-std-nacl)# permit 20.20.20.0 0.0.0.255

R2(config-std-nacl)# exit

R2(config)#ip nat inside source list 1 interface gigabitethernet0 overload

**//匹配acl 1，只允许生活区上外网**

R2(config)#ip route 0.0.0.0 0.0.0.0 202.102.2.1

R2(config)#ip dhcp pool p1

R2(dhcp-config)# range 10.10.10.10 10.10.10.200 255.255.255.0

R2(dhcp-config)# default-router 10.10.10.254

R2(dhcp-config)# exit

R2(config)#ip dhcp pool p2

R2(dhcp-config)# range 20.20.20.10 20.20.20.200 255.255.255.0

R2(dhcp-config)# default-router 20.20.20.254

R2(dhcp-config)# exit

（3）R2通过交换端口连接核心交换机，便于通过端口镜像进行流量分析

R2(config)#***monitor session 1 source interface fa0/3-0/4 both***

***//设置监控会话1，监控流量源接口为fa0/3和fa0/4,监控出入流量***

R2(config)#***monitor session 1 destination interface fastethernet0/2***

***//设置监控目的端口为fa0/2***

（4）S1和S2上配置vrrp协议，实现网关冗余。S1和S2进行链路汇聚。

S1(config)#vlan 10,20,100

S1(config)#interface vlan100

S1(config-if-vlan100)# ip address 12.1.1.1 255.255.255.0 //与R1接口地址

S1(config-if-vlan100)# keepalive gateway 12.1.1.2 //保活网关功能

S1(config-if-vlan100)# exit

S1(config)#interface fastethernet0/1

S1(config-if-fastethernet0/1)#switchport access vlan 100

S1(config-if-fastethernet0/1)# exit

S1(config)#interface fastethernet0/2

S1(config-if-fastethernet0/2)# switchport mode trunk

S1(config-if-fastethernet0/2)# switchport trunk allowed vlan all

S1(config-if-fastethernet0/2)# switchport trunk pvid vlan 1

S1(config-if-fastethernet0/2)# exit

**链路汇聚配置：**

S1(config)#**link-aggregation 1 mode lacp //链路汇聚组1，模式为自动汇聚**

S1(config)#link-aggregation 1 //配置汇聚组

S1(config-link-aggregation1)# switchport mode trunk

S1(config-link-aggregation1)# switchport trunk allowed vlan all

S1(config-link-aggregation1)# switchport trunk pvid vlan 1

S1(config-link-aggregation1)# exit

S1(config)#interface fastethernet0/3

S1(config-if-fastethernet0/3)# **link-aggregation 1 active** //交换口加入汇聚组1

S1(config-if-fastethernet0/3)# exit

S1(config)#interface fastethernet0/4

S1(config-if-fastethernet0/4)# link-aggregation 1 active

S1(config-if-fastethernet0/4)# exit

**Vrrp配置：**

S1(config)#ip access-list extended 1001

S1(config-ext-nacl)# **deny ip 10.10.10.0 0.0.0.255 20.20.20.0 0.0.0.255**

**//防止教学区和生活区互访**

S1(config-ext-nacl)# permit ip any any

S1(config-ext-nacl)# exit

S1(config)#ip access-list extended 1002

S1(config-ext-nacl)# deny ip 20.20.20.0 0.0.0.255 10.10.10.0 0.0.0.255

S1(config-ext-nacl)# permit ip any any

S1(config-ext-nacl)# exit

S1(config)#interface vlan10

S1(config-if-vlan10)# ip address 10.10.10.1 255.255.255.0

S1(config-if-vlan10)# ip access-group 1001 in //in方向调用扩展acl1001

S1(config-if-vlan10)# vrrp 10 ip 10.10.10.254 //配置虚拟ip地址作为网关

S1(config-if-vlan10)# vrrp 10 priority 150 //增大优先级成为master

S1(config-if-vlan10)# ***vrrp 10 track vlan100 60***

***//vrrp 10检测vlan100状态，若vlan100 down，则vrrp10优先级自动减少60***

S1(config-if-vlan10)#**vrrp 10 preempt //开启vrrp抢占模式**

S1(config-if-vlan10)#exit

S1(config)#interface vlan20

S1(config-if-vlan20)# ip address 20.20.20.1 255.255.255.0

S1(config-if-vlan20)# ip access-group 1002 in

S1(config-if-vlan20)# vrrp 20 ip 20.20.20.254

S1(config-if-vlan20)#vrrp 20 preempt

S1(config-if-vlan20)#exit

S1(config)#**ip dhcp-server 12.1.1.2 //开启dhcp中继服务**

**S2链路汇聚配置：**

S2(config)#vlan 10,20,200

S2(config)#interface fastethernet0/1

S2(config-if-fastethernet0/1)# switchport access vlan 200

S2(config-if-fastethernet0/1)# exit

S2(config)#interface fastethernet0/2

S2(config-if-fastethernet0/2)# switchport mode trunk

S2(config-if-fastethernet0/2)# switchport trunk allowed vlan all

S2(config-if-fastethernet0/2)# switchport trunk pvid vlan 1

S2(config-if-fastethernet0/2)# exit

S2(config)#interface vlan200

S2(config-if-vlan200)# ip address 22.1.1.1 255.255.255.0

S2(config-if-vlan200)# keepalive gateway 22.1.1.2

S2(config-if-vlan200)# exit

S2(config)#**link-aggregation 1 mode lacp**

S2(config)#interface fastethernet0/3

S2(config-if-fastethernet0/3)# link-aggregation 1 active

S2(config-if-fastethernet0/3)# exit

S2(config)#interface fastethernet0/4

S2(config-if-fastethernet0/4)# link-aggregation 1 active

S2(config-if-fastethernet0/4)# exit

S2(config)#link-aggregation 1

S2(config-link-aggregation1)# switchport mode trunk

S2(config-link-aggregation1)# switchport trunk allowed vlan all

S2(config-link-aggregation1)# switchport trunk pvid vlan 1

S2(config-link-aggregation1)# exit

***Vrrp 配置：***

S2(config)#ip access-list extended 1001

S2(config-ext-nacl)# 10 deny ip 20.20.20.0 0.0.0.255 10.10.10.0 0.0.0.255

S2(config-ext-nacl)# 20 permit ip any any

S2(config-ext-nacl)# exit

S2(config)#ip access-list extended 1002

S2(config-ext-nacl)# 10 deny ip 10.10.10.0 0.0.0.255 20.20.20.0 0.0.0.255

S2(config-ext-nacl)# 20 permit ip any any

S2(config-ext-nacl)# exit

S2(config)#interface vlan10

S2(config-if-vlan10)# ip address 10.10.10.2 255.255.255.0

S2(config-if-vlan10)# ip access-group 1002 in

S2(config-if-vlan10)# vrrp 10 ip 10.10.10.254

S2(config-if-vlan10)#vrrp 10 preempt

S2(config-if-vlan10)#exit

S2(config)#interface vlan20

S2(config-if-vlan20)# ip address 20.20.20.2 255.255.255.0

S2(config-if-vlan20)# ip access-group 1001 in

S2(config-if-vlan20)# vrrp 20 ip 20.20.20.254

S2(config-if-vlan20)# vrrp 20 priority 150

S2(config-if-vlan20)# vrrp 20 track vlan200 60

S2(config-if-vlan20)#vrrp 20 preempt

S2(config-if-vlan20)#exit

（5）S1、S2、S3之间配置mstp

S1(config)#**spanning-tree mst configuration**

S1(config-mst-region)# region-name mene //配置区域名

S1(config-mst-region)# revision-level 1 //设定修正号

S1(config-mst-region)# instance 1 vlan 10 //关联实例到vlan

S1(config-mst-region)# instance 2 vlan 20

S1(config-mst-region)#***active configuration pending //激活配置***

S1(config-mst-region)#exit

S1(config)#**spanning-tree mst instance 1 priority 0**

**//设定实例1优先级,优先级越小越可能成为master。**

S1(config)#**spanning-tree mst instance 2 priority 8192**

S1(config)#spanning-tree enable //使能mstp

**S2:**

S2(config)#**spanning-tree mst configuration**

S2(config-mst)# region-name mene

S2(config-mst)# revision-level 1

S2(config-mst)# instance 1 vlan 10

S2(config-mst)# instance 2 vlan 20

S2(config-mst)#active configuration pending

S2(config-mst)#exit

S2(config)#spanning-tree mst instance 1 priority 8192

S2(config)#spanning-tree mst instance 2 priority 0

S2(config)#spanning-tree enable

**S3:**

S3(config)#vlan 10,20

S3(config)#spanning-tree mst configuration

S3(config-mst-region)# region-name mene

S3(config-mst-region)# revision-level 1

S3(config-mst-region)# instance 1 vlan 10

S3(config-mst-region)# instance 2 vlan 20

S3(config-mst-region)#active configuration pending

S3(config-mst-region)#exit

S3(config)#spanning-tree mst instance 1 priority 8192

S3(config)#spanning-tree mst instance 2 priority 8192

S3(config)#spanning-tree enable

S3(config)#interface fa0/0-0/23

S3(config-if-range)#switchport mode trunk

S3(config-if-range)#switchport trunk allowed vlan all

（6）在R2、S1、S2上启用rip路由协议：

R2(config)#ip access-list standard 2

R2(config-std-nacl)# permit 10.10.10.0 0.0.0.255

R2(config-std-nacl)# exit

R2(config)#ip access-list standard 1

R2(config-std-nacl)# permit 20.20.20.0 0.0.0.255

R2(config-std-nacl)# exit

R2(config)#router rip

R2(config-rip)# version 2

R2(config-rip)# network 12.0.0.0

R2(config-rip)# network 22.0.0.0

R2(config-rip)# offset-list 1 in 3 vlan100

R2(config-rip)# offset-list 2 in 3 vlan200

R2(config-rip)# **default-information originate //下放一条默认路由到S1和S2**

R2(config-rip)# no auto-summary

R2(config-rip)# exit

R2(config)#interface fastethernet0/3

R2(config-if-fastethernet0/3)#switchport access vlan 100

R2(config-if-fastethernet0/3)# exit

R2(config)#interface fastethernet0/4

R2(config-if-fastethernet0/4)# switchport access vlan 200

S1(config)#router rip

S1(config-rip)# version 2

S1(config-rip)# network 10.0.0.0

S1(config-rip)# network 12.0.0.0

S1(config-rip)# network 20.0.0.0

S1(config-rip)# passive-interface vlan10

S1(config-rip)# passive-interface vlan20

S1(config-rip)# no auto-summary

S2(config)#router rip

S2(config-rip)# version 2

S2(config-rip)# network 10.0.0.0

S2(config-rip)# network 20.0.0.0

S2(config-rip)# network 22.0.0.0

S2(config-rip)# passive-interface vlan10

S2(config-rip)# passive-interface vlan20

S2(config-rip)# no auto-summary

S2(config-rip)# exit

S2(config)#ip dhcp-server 22.1.1.2 //设置dhcp中继地址为22.1.1.2

（7）在S4上开启端口安全：

S4(config)#vlan 10,20

S4(config)#interface fastethernet0/18

S4(config-if-fastethernet0/18)# switchport access vlan 10

S4(config-if-fastethernet0/18)#**port-security enable**

S4(config-if-fastethernet0/18)#**port-security permit mac-address 201A.0638.15E1**

//开启端口安全，绑定PC的mac地址

S4(config-if-fastethernet0/18)# exit

S4(config)#interface fastethernet0/12

S4(config-if-fastethernet0/12)# switchport mode trunk

S4(config-if-fastethernet0/12)# switchport trunk allowed vlan all

S4(config-if-fastethernet0/12)# switchport trunk pvid vlan 1

S4(config-if-fastethernet0/12)# exit

S4(config)#interface fastethernet0/13

S4(config-if-fastethernet0/13)# switchport mode trunk

S4(config-if-fastethernet0/13)# switchport trunk allowed vlan all

S4(config-if-fastethernet0/13)# switchport trunk pvid vlan 1

S4(config-if-fastethernet0/13)# exit

S4(config)#interface fa0/1-0/11

S4(config-if-range)#switchport access vlan 20

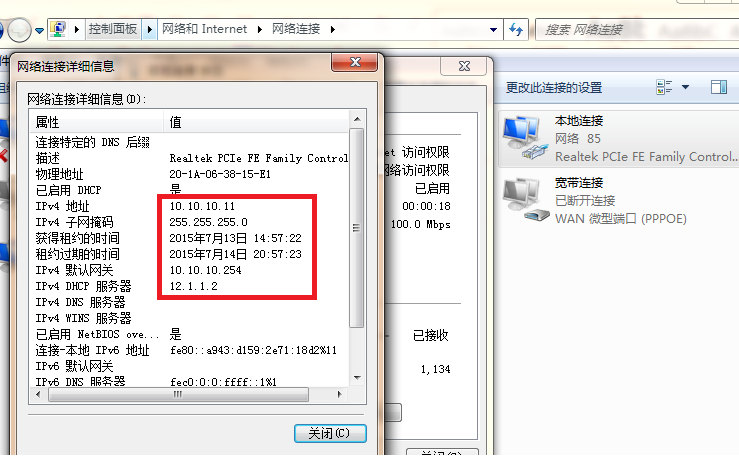
S4(config-if-range)#exit

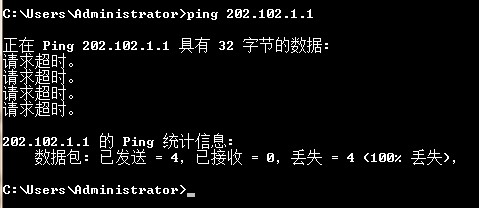
S4(config)#interface fa0/14-0/24

S4(config-if-range)#switchport access vlan 10

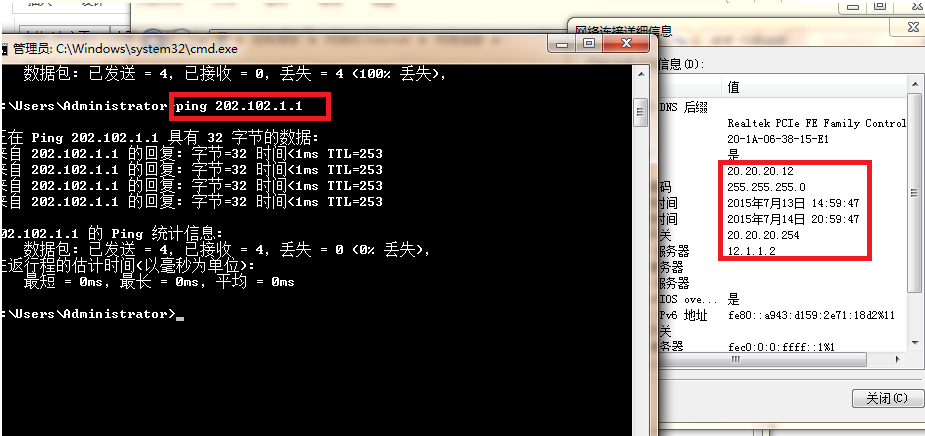
**（8）验证：**

在PC上验证是否能自动获取ip地址



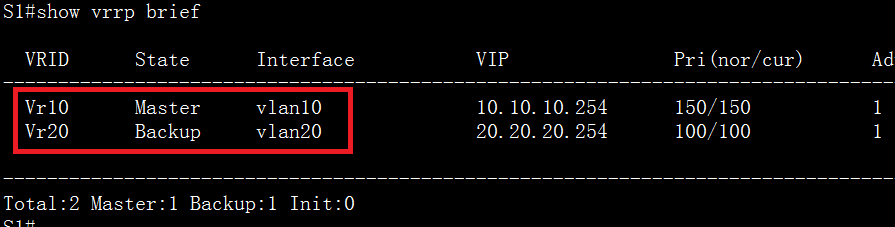


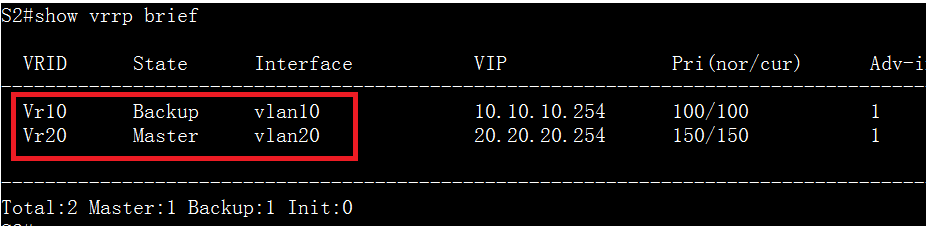
教学区不能上外网。切换PC连接端口到vlan20。



生活区能上外网。

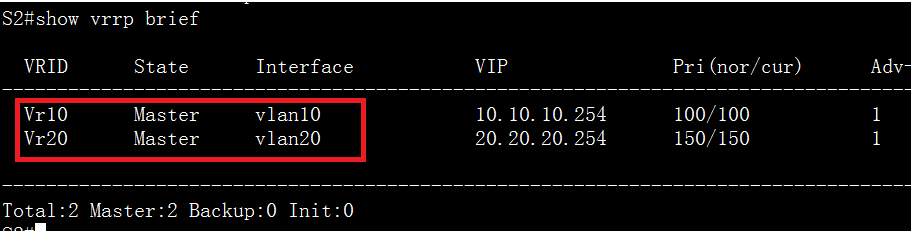
在S1和S2上查看vrrp：





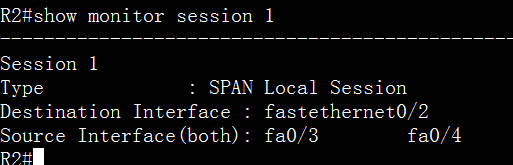
现在拔掉S1上的fa0/1线路，看vrrp10的master是否会切换到S2上：





Vrrp10已经切换到了S2上。

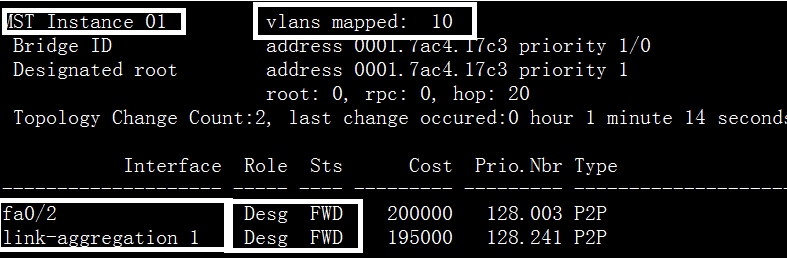
验证流量监控：

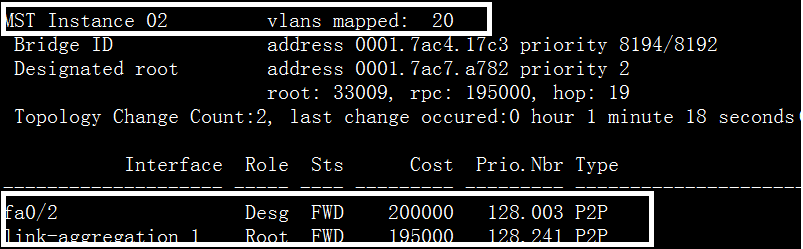


镜像会话已建立。

验证mstp：

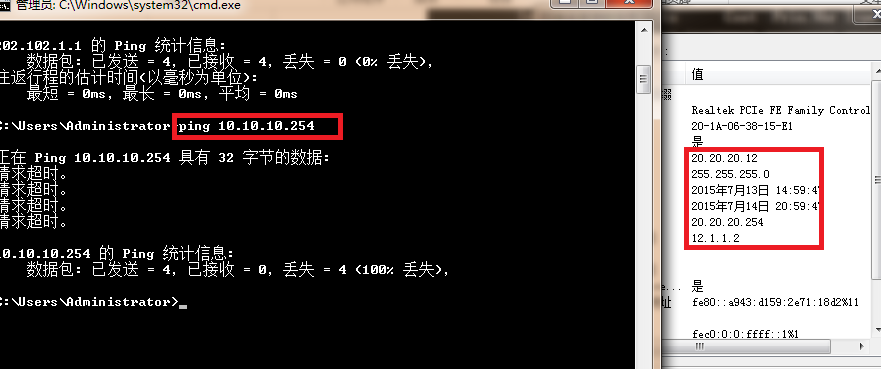
S1#show spanning-tree mst





对于vlan 10，S1的两个端口均为指定端口。对于vlan20，S1的fa0/2为指定端口，汇聚端口为根端口。在S2上的mstp验证一样。

验证教学区和生活区不能互访：



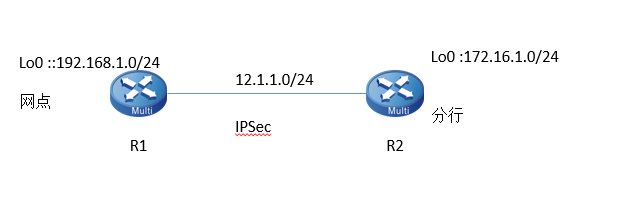
# 第四章 IPsec vpn



## 路由器之间建立IPsec

### 建立IPsec隧道

实验拓扑：



实验要求：

在R1和R2之间建立IPsec隧道，R1的私网地址段能Ping通R2的私网地址段（R1和R2互指默认路由）。

R1配置：

R1(config)#interface loopback0

R1(config-if-loopback0)# ip address 192.168.1.1 255.255.255.0

R1(config-if-loopback0)#exit

R1(config)#interface gigabitethernet0

R1(config-if-gigabitethernet0)# ip address 12.1.1.1 255.255.255.0

R1(config-if-gigabitethernet0)#exit

R1(config)#**crypto ike key maipu any //设置Ike 密钥为maipu**

R1(config)#**crypto ike proposal 1 //设置Ike 协商算法**

R1(config-ike-prop)# **encryption 3des //Ike加密采用3des**

R1(config-ike-prop)# **integrity md5 //Ike完整性保护采用md5**

R1(config-ike-prop)# exit

R1(config)#**crypto ipsec proposal 1 //设置IPsec协商算法**

R1(config-ipsec-prop)# ***esp 3des md5***

***//IPsec封装安全负载采用3des加密，md5完整性保护***

R1(config-ipsec-prop)# exit

R1(config)#**crypto tunnel t1 //建立隧道并命名为t1**

R1(config-tunnel)# **local address 12.1.1.1 //指定建立隧道的本地地址**

R1(config-tunnel)# **peer any //指定隧道的对端为any**

R1(config-tunnel)#**set Ike proposal 1 //隧道ike协商调用ike 1**

R1(config-tunnel)#**set IPsec proposal 1 //隧道IPsec协商调用IPsec 1**

R1(config-tunnel)# exit

R1(config)#**crypto policy p1 //设置隧道的策略**

R1(config-policy)#**flow 192.168.1.0 255.255.255.0 172.16.1.0 255.255.255.0 ip tunnel t1 //指定隧道保护数据流**

R1(config-policy)#exit

R1(config)#ip route 0.0.0.0 0.0.0.0 12.1.1.2

分行R2配置：

R2(config)#interface loopback0

R2(config-if-loopback0)# ip address 172.16.1.2 255.255.255.0

R2(config-if-loopback0)# exit

R2(config)#interface gigabitethernet0

R2(config-if-gigabitethernet0)# ip address 12.1.1.2 255.255.255.0

R2(config-if-gigabitethernet0)# exit

R2(config)#crypto ike key maipu address 12.1.1.1

R2(config)#crypto ike proposal 1

R2(config-ike-prop)# encryption 3des

R2(config-ike-prop)# integrity md5

R2(config-ike-prop)# exit

R2(config)#crypto ipsec proposal 1

R2(config-ipsec-prop)# esp 3des md5

R2(config-ipsec-prop)# exit

R2(config)#crypto tunnel t1

R2(config-tunnel)# local address 12.1.1.2

R2(config-tunnel)# peer address 12.1.1.1

R2(config-tunnel)# set ike proposal 1

R2(config-tunnel)# set ipsec proposal 1

R2(config-tunnel)# set auto-up

R2(config-tunnel)# exit

R2(config)#crypto policy p1

R2(config-policy)# flow 172.16.1.0 255.255.255.0 192.168.1.0 255.255.255.0 ip tunnel t1

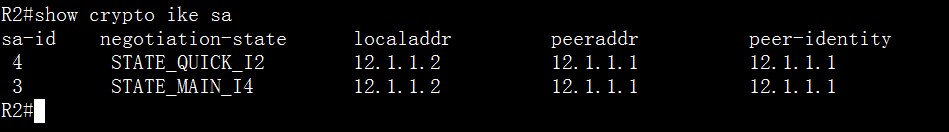
R2(config-policy)# exit

R2(config)#ip route 0.0.0.0 0.0.0.0 12.1.1.1

在R2上查看隧道是否建立：

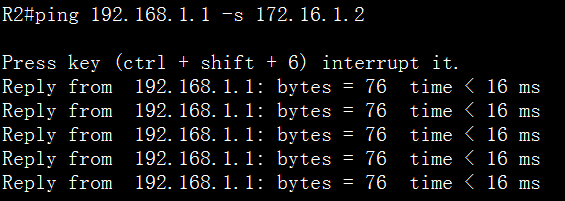
通过命令show crypto ike sa

Show crypto ipsec sa





隧道协商成功且IPsec建立成功。



### 注入反向路由

实验拓扑同实验一不变

实验要求：R1和R2之间建立IPsec，R1设置默认路由，R2上不指定默认路由，R2上使用反向路由注入。（使用debug crypto ike normal 查看ike协商过程）

R1配置同实验一不变

R2配置如下：

R2(config)#interface loopback0

R2(config-if-loopback0)# ip address 172.16.1.2 255.255.255.0

R2(config-if-loopback0)# exit

R2(config)#interface gigabitethernet0

R2(config-if-gigabitethernet0)# ip address 12.1.1.2 255.255.255.0

R2(config-if-gigabitethernet0)# exit

R2(config)#crypto ike key maipu address 12.1.1.1

R2(config)#crypto ike proposal 1

R2(config-ike-prop)# encryption 3des

R2(config-ike-prop)# integrity md5

R2(config-ike-prop)# exit

R2(config)#crypto ipsec proposal 1

R2(config-ipsec-prop)# esp 3des md5

R2(config-ipsec-prop)# exit

R2(config)#crypto tunnel t1

R2(config-tunnel)# local address 12.1.1.2

R2(config-tunnel)# peer address 12.1.1.1

R2(config-tunnel)# set ike proposal 1

R2(config-tunnel)# set ipsec proposal 1

R2(config-tunnel)# ***set auto-up //由分行主动发起隧道建立***

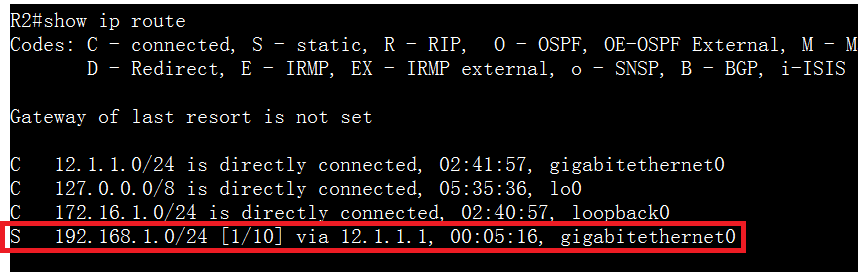
R2(config-tunnel)# exit

R2(config)#crypto policy p1

R2(config-policy)# flow 172.16.1.0 255.255.255.0 192.168.1.0 255.255.255.0 ip tunnel t1

R2(config-policy)# **set reverse-route //设置反向路由注入**

查看其路由表:



这里通过反向路由注入在R2上生成了一条与保护流匹配的静态路由。

可以通过debug crypto ike normal 命令来查看整个协商过程。

R2#debug crypto ike normal

%DBG\_NORMAL enabled

R2#clear crypto sa 清除IPsec sa 重新触发隧道建立。

R2#

Jul 9 2015 17:04:18: %IKE-DBG:

Jul 9 2015 17:04:18: %IKE-DBG: new conf request received

Jul 9 2015 17:04:18: %IKE-DBG: CONF: parse the message for clear all sa

Jul 9 2015 17:04:18: %IKE-DBG: delete states by flow p1

Jul 9 2015 17:04:18: %IKE-DBG: delete states by ftn p1: t1: 0

Jul 9 2015 17:04:18: %IKE-DBG: delete state #12 of tunnel "t1", notify\_peer: 1

Jul 9 2015 17:04:18: %IKE-DBG: delete phase2 state #12 of flow "p1" of tunnel "t1"

Jul 9 2015 17:04:18: %IKE-IPSECSA\_DEL\_P-5: The IPSEC SA of policy p1(protocal ip, src 172.16.1.0/24:any, dst 192.168.1.0/24:any) of tunnel t1 deleted.

Jul 9 2015 17:04:18: %IKE-DBG: Send the ipsec sa deleted notify to SNMP server

Jul 9 2015 17:04:18: %IKE-DBG: delete route (192.168.1.0/24 -> 12.1.1.1) for flow 'p1' success

Jul 9 2015 17:04:18: %IKE-DBG: delete phase1 states by tunnel t1

Jul 9 2015 17:04:18: %IKE-DBG: delete state #11 of tunnel "t1", notify\_peer: 1

Jul 9 2015 17:04:18: %IKE-DBG: delete phase1 state #11 of tunnel "t1"

Jul 9 2015 17:04:18: %IKE-IKESA\_DEL\_P-5: IKE SA between 12.1.1.2(12.1.1.2) and 12.1.1.1(12.1.1.1) for permanent tunnel t1 deleted.

Jul 9 2015 17:04:18: %IKE-DBG: delete pending by tunnel t1

Jul 9 2015 17:04:18: %IKE-DBG: delete states by flow p1

Jul 9 2015 17:04:18: %IKE-DBG: delete states by ftn p1: t1: 0

Jul 9 2015 17:04:18: %IKE-DBG: delete phase1 states by tunnel t1

Jul 9 2015 17:04:18: %IKE-DBG: ***create\_ike\_addr\_sockets: address: 12.1.1.2, vrfindex: 0, ifflags: c008063***

Jul 9 2015 17:04:18: %IKE-DBG: initiate\_ft\_node for flow "p1" of tunnel "t1"

Jul 9 2015 17:04:18: %IKE-DBG: ***initiate flow "p1" of tunnel "t1"***

Jul 9 2015 17:04:18: %IKE-DBG: creating state object #13 at 0x1009b568

Jul 9 2015 17:04:18: %IKE-DBG: pending Quick Mode with 12.1.1.1 "t1"

Jul 9 2015 17:04:18: %IKE-DBG: ***initiating Main Mode***

Jul 9 2015 17:04:18: %IKE-DBG: received Vendor ID payload [RFC 3947] method set to=109

Jul 9 2015 17:04:18: %IKE-DBG: received Vendor ID payload [Dead Peer Detection]

Jul 9 2015 17:04:18: %IKE-DBG: received Vendor ID payload [Maipu VRC REG Version 1.0]

Jul 9 2015 17:04:18: %IKE-DBG: received Vendor ID payload [Maipu PM3 VRC Binding 1.0]

Jul 9 2015 17:04:18: %IKE-DBG: ***instantiate local id '(none)' with '12.1.1.2'***

Jul 9 2015 17:04:18: %IKE-DBG: enabling possible NAT-traversal with method 3

Jul 9 2015 17:04:18: %IKE-DBG: transition from state STATE\_MAIN\_I1 to state STATE\_MAIN\_I2

Jul 9 2015 17:04:18: %IKE-DBG: NAT-Traversal: Result using 3: no NAT detected

Jul 9 2015 17:04:18: %IKE-DBG: ***transition from state STATE\_MAIN\_I2 to state STATE\_MAIN\_I3***

Jul 9 2015 17:04:18: %IKE-DBG: ***Main mode peer ID is ID\_IPV4\_ADDR: '12.1.1.1'***

Jul 9 2015 17:04:18: %IKE-DBG: ***instantiate peer id '(none)' with '12.1.1.1'***

Jul 9 2015 17:04:18: %IKE-IKESA\_EST\_P-5: ***IKE SA between 12.1.1.2(12.1.1.2) and 12.1.1.1(12.1.1.1) for permanent tunnel t1 established***.

Jul 9 2015 17:04:18: %IKE-DBG: Dead Peer Detection (RFC3706) enabled for state #13

Jul 9 2015 17:04:18: %IKE-DBG: transition from state STATE\_MAIN\_I3 to state STATE\_MAIN\_I4

Jul 9 2015 17:04:18: %IKE-DBG: ***ISAKMP SA established***

Jul 9 2015 17:04:18: %IKE-DBG: unpending Quick Mode with 12.1.1.1 "t1", local is initiator

Jul 9 2015 17:04:18: %IKE-DBG: duplicating state object #13

Jul 9 2015 17:04:18: %IKE-DBG: creating state object #14 at 0x10099548

Jul 9 2015 17:04:18: %IKE-DBG: initiating Quick Mode

Jul 9 2015 17:04:18: %IKE-DBG: kernel\_alg\_esp\_enc\_default\_key\_len(): transid=3, keylen=24

Jul 9 2015 17:04:18: %IKE-DBG: kernel\_alg\_auth\_ok(): success: auth=1

Jul 9 2015 17:04:18: %IKE-DBG: our client is subnet 172.16.1.0/24

Jul 9 2015 17:04:18: %IKE-DBG: our client protocol/port is 0/0

Jul 9 2015 17:04:18: %IKE-DBG: peer client is subnet 192.168.1.0/24

Jul 9 2015 17:04:18: %IKE-DBG: peer client protocol/port is 0/0

Jul 9 2015 17:04:18: %IKE-DBG: kernel\_alg\_auth\_keylen(): auth = 1, keylen = 16

Jul 9 2015 17:04:18: %IKE-DBG: KERN : success to setup the esp inbound ipsec sa of state 14

Jul 9 2015 17:04:18: %IKE-DBG: KERN : success to setup the esp outbound ipsec sa of state 14

Jul 9 2015 17:04:18: %IKE-DBG: KERN : success to setup the relationship between inbound and outbound sas

Jul 9 2015 17:04:18: %IKE-IPSECSA\_INST\_P-5: ***The ESP SA(inbound) for policy p1(protocal ip, src 172.16.1.0/24:any, dst 192.168.1.0/24:any) of tunnel t1 installed.***

Jul 9 2015 17:04:18: %IKE-IPSECSA\_INST\_P-5: The ESP SA(outbound) for policy p1(protocal ip, src 172.16.1.0/24:any, dst 192.168.1.0/24:any) of tunnel t1 installed.

Jul 9 2015 17:04:18: %IKE-DBG: Send the ipsec sa established notify to SNMP server

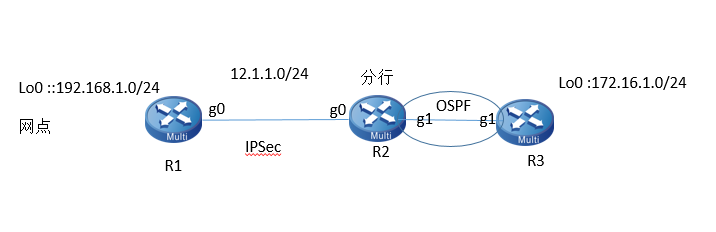
Jul 9 2015 17:04:18: %IKE-DBG: install route (192.168.1.0/24 -> 12.1.1.1) for flow 'p1' success

Jul 9 2015 17:04:18: %IKE-DBG: transition from state STATE\_QUICK\_I1 to state STATE\_QUICK\_I2

Jul 9 2015 17:04:18: %IKE-DBG: sent QI2, IPsec SA established

### 重分发注入的反向静态路由

实验拓扑：



实验要求：

在网点和分行之间建立IPsec实现R1和R3下的私网互通。在R1上指定默认路由。R2上使用反向路由注入，并将注入的静态路由引入OSPF。

R1配置：

R1(config)#interface loopback0

R1(config-if-loopback0)# ip address 192.168.1.1 255.255.255.0

R1(config-if-loopback0)#exit

R1(config)#interface gigabitethernet0

R1(config-if-gigabitethernet0)# ip address 12.1.1.1 255.255.255.0

R1(config-if-gigabitethernet0)#exit

R1(config)#**crypto ike key maipu any //设置Ike 密钥为maipu**

R1(config)#**crypto ike proposal 1 //设置Ike 协商算法**

R1(config-ike-prop)# **encryption 3des //Ike加密采用3des**

R1(config-ike-prop)# **integrity md5 //Ike完整性保护采用md5**

R1(config-ike-prop)# exit

R1(config)#**crypto ipsec proposal 1 //设置IPsec协商算法**

R1(config-ipsec-prop)# ***esp 3des md5***

***//IPsec封装安全负载采用3des加密，md5完整性保护***

R1(config-ipsec-prop)# exit

R1(config)#**crypto tunnel t1 //建立隧道并命名为t1**

R1(config-tunnel)# **local address 12.1.1.1 //指定建立隧道的本地地址**

R1(config-tunnel)# **peer any //指定隧道的对端为any**

R1(config-tunnel)#**set Ike proposal 1 //隧道ike协商调用ike 1**

R1(config-tunnel)#**set IPsec proposal 1 //隧道IPsec协商调用IPsec 1**

R1(config-tunnel)# exit

R1(config)#**crypto policy p1 //设置隧道的策略**

R1(config-policy)#**flow 192.168.1.0 255.255.255.0 172.16.1.0 255.255.255.0 ip tunnel t1 //指定隧道保护数据流**

R1(config-policy)#exit

R1(config)#ip route 0.0.0.0 0.0.0.0 12.1.1.2

R2配置：

R2(config)#interface gigabitethernet1

R2(config-if-gigabitethernet1)# ip address 10.1.1.2 255.255.255.0

R2(config-if-gigabitethernet1)#exit

R2(config)#interface gigabitethernet0

R2(config-if-gigabitethernet0)# ip address 12.1.1.2 255.255.255.0

R2(config-if-gigabitethernet0)# exit

R2(config)#crypto ike key maipu address 12.1.1.1

R2(config)#crypto ike proposal 1

R2(config-ike-prop)# encryption 3des

R2(config-ike-prop)# integrity md5

R2(config-ike-prop)# exit

R2(config)#crypto ipsec proposal 1

R2(config-ipsec-prop)# esp 3des md5

R2(config-ipsec-prop)# exit

R2(config)#crypto tunnel t1

R2(config-tunnel)# local address 12.1.1.2

R2(config-tunnel)# peer address 12.1.1.1

R2(config-tunnel)# set ike proposal 1

R2(config-tunnel)# set ipsec proposal 1

R2(config-tunnel)# set auto-up

R2(config-tunnel)# exit

R2(config)#crypto policy p1

R2(config-policy)# flow 172.16.1.0 255.255.255.0 192.168.1.0 255.255.255.0 ip tunnel t1

R2(config-policy)# **set reverse-route //设置反向路由注入**

R2(config-policy)#exit

R2(config)#router ospf 1

R2(config-ospf)# network 10.1.1.0 0.0.0.255 area 0

R2(config-ospf)# exit

R3配置：

R3(config)#interface g1

R3(config-if-gigabitethernet1)#ip add 10.1.1.3 24

R3(config-if-gigabitethernet1)#exit

R3(config)#router ospf 1

R3(config-ospf)#network 10.1.1.3 0.0.0.255 area 0

R3(config-ospf)#network 172.16.1.0 0.0.0.255 area 0

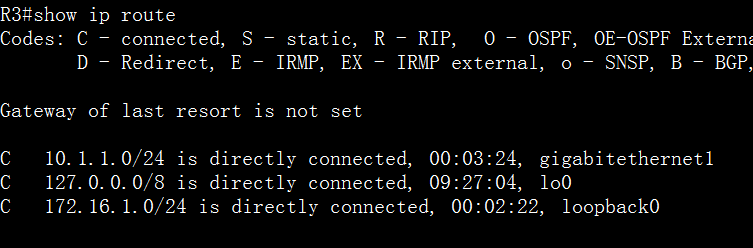
R3(config-ospf)#exit

R3(config)#interface loo0

R3(config-if-loopback0)#ip add 172.16.1.3 24

R3(config-if-loopback0)#exit

重分发静态路由前R3的路由表

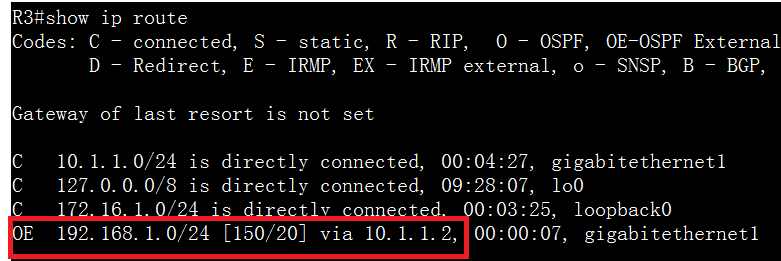


现在在R2上重分发通过反向注入的静态路由：

R2(config)#router ospf 1

R2(config-ospf)#redistribute static //重分发静态路由

重分发后R3学习到了R1的192网点的路由



R3#ping

Protocol [ip]:

**Target IP address or hostname: 192.168.1.1**

Repeat count [5]:

Datagram size [76]:

Timeout in seconds [2]:

**Extended commands [no]: y**

**Source address or interface: 172.16.1.3**

Type of service [0]:

Set DF bit in IP header? [no]:

Validate reply data? [no]:

Data pattern [abcd]:

Loose, Strict, Record, Timestamp, Verbose[none]:

Sweep range of sizes [no]:

Press key (ctrl + shift + 6) interrupt it.

Sending 5, 76-byte ICMP Echos to 192.168.1.1 , timeout is 2 seconds:

!!!!!

Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms.

**R1#ping 172.16.1.3 -s 192.168.1.1**

Press key (ctrl + shift + 6) interrupt it.

Reply from 172.16.1.3: bytes = 76 time < 16 ms

Reply from 172.16.1.3: bytes = 76 time < 16 ms

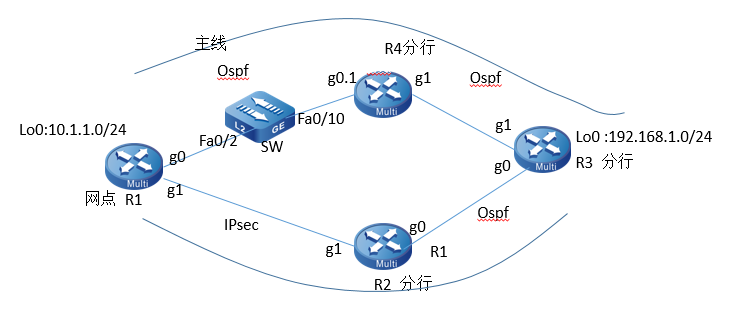
Reply from 172.16.1.3: bytes = 76 time < 16 ms

Reply from 172.16.1.3: bytes = 76 time < 16 ms

Reply from 172.16.1.3: bytes = 76 time < 16 ms

### IPsec隧道线路的备份

实验拓扑：



实验要求：

网点和分行的私网要求能相互通信，正常情况下走上面的主线，当主线故障后走下面的IPsec备份线路。

接口地址配置：

R1(config)#interface g0

R1(config-if-gigabitethernet0)#ip add 14.1.1.1 24

R1(config-if-gigabitethernet0)#exit

R1(config-if-gigabitethernet0)#***keepalive gateway 14.1.1.4***

***//跟踪网关，检测是否up***

R1(config-if-gigabitethernet0)#***backup interface gigabitethernet1***

***//将g1接口设为备份口***

R1(config)#interface loo0

R1(config-if-loopback0)#ip address 10.1.1.1 24

R1(config-if-loopback0)#ip ospf network point-to-point

//将环回口掩码设置为24位宣告

R1(config-if-loopback0)#exit

R1(config)#interface g1

R1(config-if-gigabitethernet1)#ip address 12.1.1.1 24

R1(config-if-gigabitethernet1)#exit

R2(config)#interface g1

R2(config-if-gigabitethernet1)#ip address 12.1.1.2 24

R2(config-if-gigabitethernet1)#exit

R2(config)#interface g0

R2(config-if-gigabitethernet0)#ip address 23.1.1.2 24

R2(config-if-gigabitethernet0)#exit

R3(config)#interface g0

R3(config-if-gigabitethernet0)#ip address 23.1.1.3 24

R3(config-if-gigabitethernet0)#exit

R3(config)#interface g1

R3(config-if-gigabitethernet1)#ip address 34.1.1.3 24

R3(config-if-gigabitethernet1)#exit

R3(config)#interface loo0

R3(config-if-loopback0)#ip address 192.168.1.3 24

R3(config-if-loopback0)#ip ospf network point-to-point

//将环回口掩码设置为24位宣告

R3(config-if-loopback0)#exit

sw1(config)#vlan 4

sw1(config-vlan4)#exit

sw1(config)#interface fa0/10

sw1(config-if-fastethernet0/10)#switchport mode trunk

sw1(config-if-fastethernet0/10)#switchport trunk allowed vlan all

sw1(config-if-fastethernet0/10)#exit

sw1(config)#interface fa0/2

sw1(config-if-fastethernet0/2)#switchport access vlan 4

R4(config)#interface g1

R4(config-if-gigabitethernet1)#ip address 34.1.1.4 24

R4(config-if-gigabitethernet1)#exit

R4(config)#interface g0.1

R4(config-if-gigabitethernet0.1)#encapsulation dot1q 4

R4(config-if-gigabitethernet0.1)#ip address 14.1.1.4 24

R4(config-if-gigabitethernet0.1)# keepalive gateway 14.1.1.1

R4(config-if-gigabitethernet0.1)#exit

**OSPF配置：**

R1(config)#router ospf 1

R1(config-ospf)#network 14.1.1.1 0.0.0.255 area 0

R1(config-ospf)#network 10.1.1.0 0.0.0.255 area 0

R1(config-ospf)#exit

R2(config)#router ospf 1

R2(config-ospf)#network 23.1.1.2 0.0.0.255 area 0

R2(config-ospf)#redistribute static //将反向注入的静态路由重分发到OSPF

R2(config-ospf)#exit

R3(config)#router ospf 1

R3(config-ospf)#network 23.1.1.3 0.0.0.255 area 0

R3(config-ospf)#network 34.1.1.3 0.0.0.255 area 0

R3(config-ospf)#network 192.168.1.0 0.0.0.255 area 0

R3(config-ospf)#exit

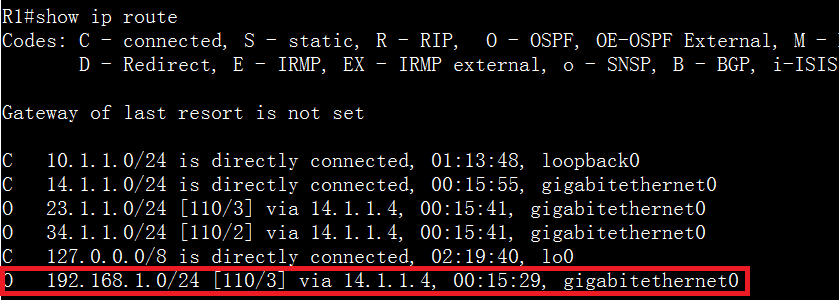
R4(config)#router ospf 1

R4(config-ospf)#network 34.1.1.4 0.0.0.255 area 0

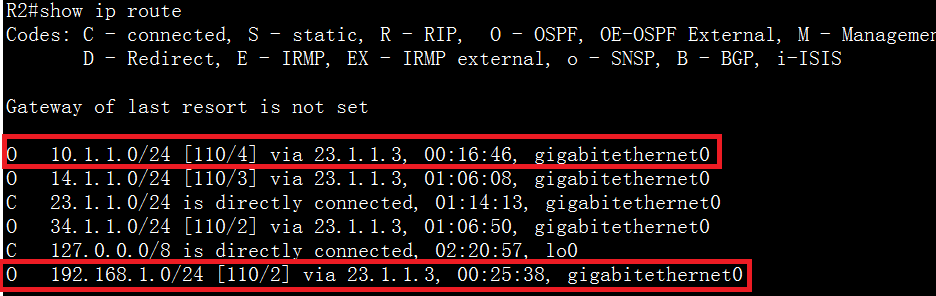
R4(config-ospf)#network 14.1.1.4 0.0.0.255 area 0

R4(config-ospf)#exit

查看R1的路由表



R2的路由表：



在R1和R2之间建立隧道：

R1(config)#crypto ike key maipu any

R1(config)#crypto ike proposal 1

R1(config-ike-prop)#integrity md5

R1(config-ike-prop)#encryption 3des

R1(config-ike-prop)#exit

R1(config)#crypto ipsec proposal 1

R1(config-ipsec-prop)#esp 3des md5

R1(config-ipsec-prop)#exit

R1(config)#crypto tunnel t1

R1(config-tunnel)#peer any

R1(config-tunnel)#local address 12.1.1.1

R1(config-tunnel)#set ike proposal 1

R1(config-tunnel)#set ipsec proposal 1

R1(config-tunnel)#exit

R1(config)#crypto policy p1

R1(config-policy)#**flow 10.1.1.0 255.255.255.0 192.168.1.0 255.255.255.0 ip tunnel t1 bypass //参数bypass表示隧道不存在时忽略该条策略**

R1(config-policy)# **set reverse-route //设置反向路由注入**

R2(config)#crypto ike key maipu address 12.1.1.1

R2(config)#crypto ike proposal 1

R2(config-ike-prop)#encryption 3des

R2(config-ike-prop)#integrity md5

R2(config-ike-prop)#exit

R2(config)#crypto ipsec proposal 1

R2(config-ipsec-prop)#esp 3des md5

R2(config-ipsec-prop)#exit

R2(config)#crypto tunnel t1

R2(config-tunnel)#peer address 12.1.1.1

R2(config-tunnel)#local address 12.1.1.2

R2(config-tunnel)#set ike proposal 1

R2(config-tunnel)#set ipsec proposal 1

R2(config-tunnel)#**set auto-up //分行自动发起建立隧道**

R2(config-tunnel)#exit

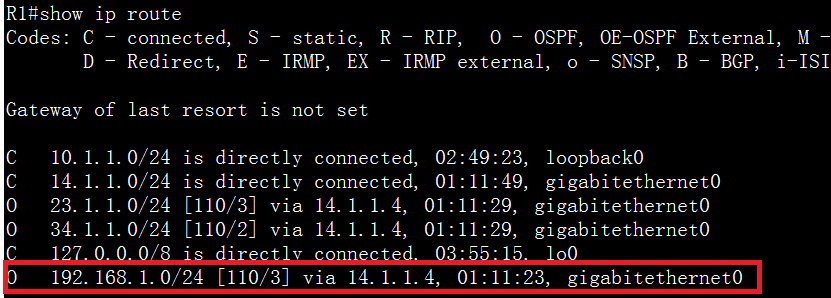
R2(config)#crypto policy p1

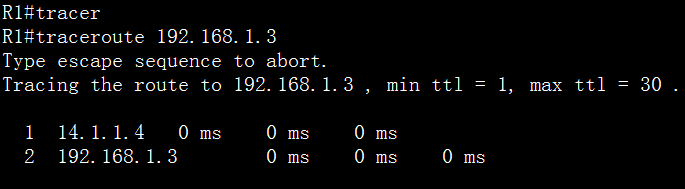
R2(config-policy)#flow 192.168.1.0 255.255.255.0 10.1.1.0 255.255.255.0 ip tunnel t1 bypass

R2(config-policy)#set reverse-route

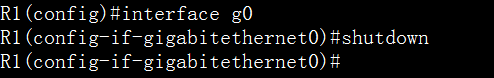
R2(config-policy)#exit

由于g0接口设置的备份接口为g1，正常情况下隧道是拆除状态。查看R1的路由表：

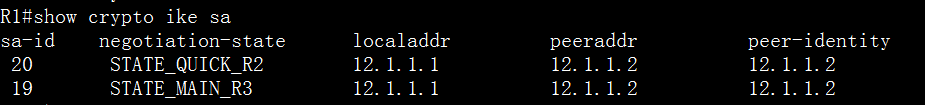


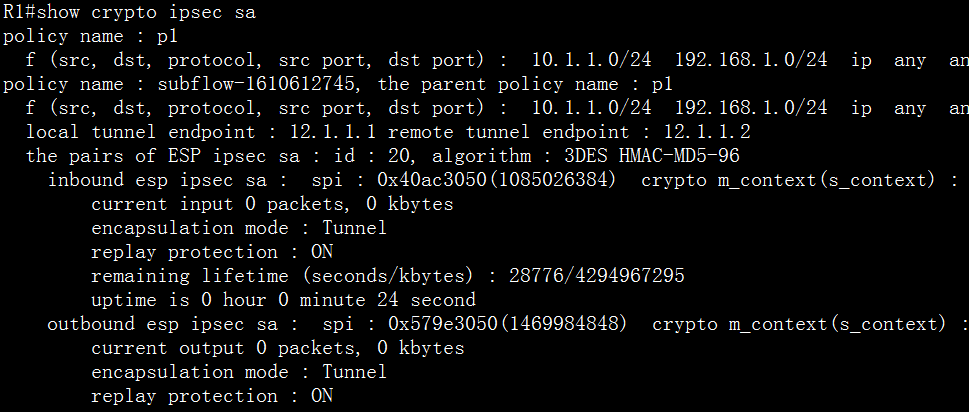


现在将R1的g0接口down掉。

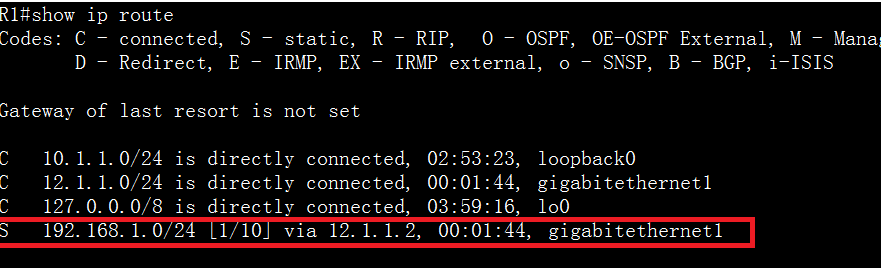


查看隧道是否起来：



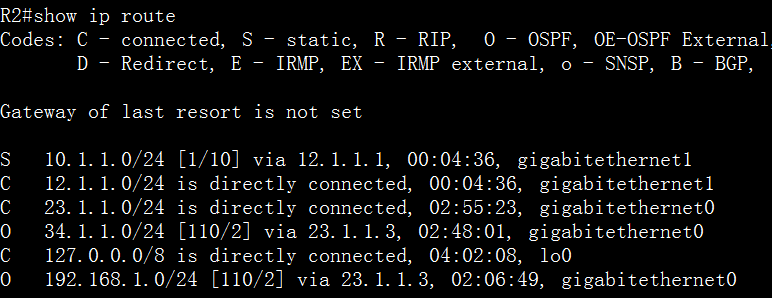


隧道已经建立。现在查看R1的路由表：

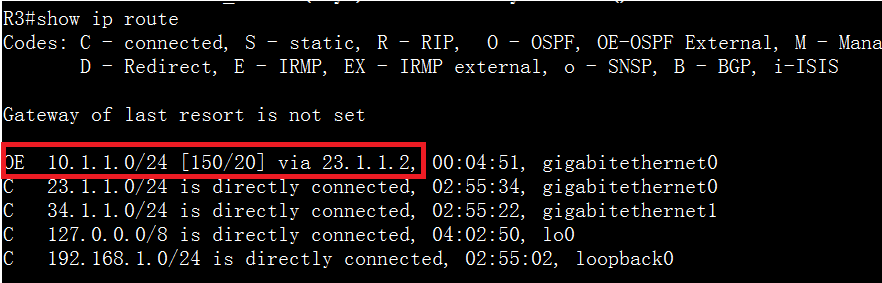


到达分行192.168.1.0网段的路由已经切换到了备份线路。

查看R2的路由表

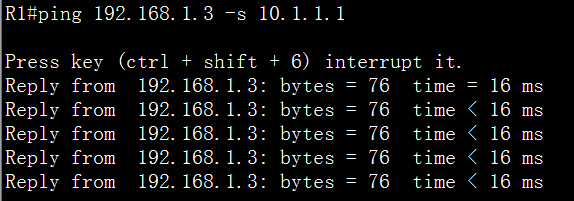


R3的路由表：



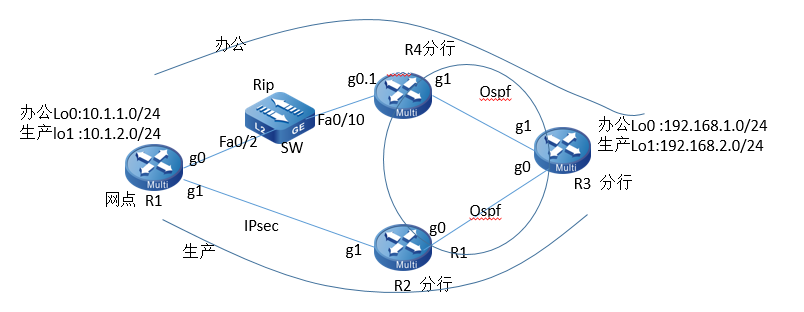
R3学到了一条10.1.1.0的外部路由。

在R1上ping分行的192.168.1.3



### IPsec 隧道分流备份

实验拓扑：



实验要求：办公网和生产网分别走上下两条线路，并相互备份。发生故障后能各自切换到备份线路。

R1(config)#interface g0

R1(config-if-gigabitethernet0)#ip add 14.1.1.1 24

R1(config-if-gigabitethernet0)#exit

R1(config-if-gigabitethernet0)#***keepalive gateway 14.1.1.4***

***//跟踪网关，检测是否up***

R1(config-if-gigabitethernet0)#exit

R1(config)#interface loo0

R1(config-if-loopback0)#ip address 10.1.1.1 24

R1(config-if-loopback0)#ip ospf network point-to-point

//将环回口掩码设置为24位宣告

R1(config-if-loopback0)#exit

R1(config)#interface loo1

R1(config-if-loopback1)#ip address 10.1.2.1 24

R1(config-if-loopback1)#ip ospf network point-to-point

R1(config-if-loopback1)#exit

R1(config)#interface g1

R1(config-if-gigabitethernet1)#ip address 12.1.1.1 24

R1(config-if-gigabitethernet1)#exit

R2(config)#interface g1

R2(config-if-gigabitethernet1)#ip address 12.1.1.2 24

R2(config-if-gigabitethernet1)#exit

R2(config)#interface g0

R2(config-if-gigabitethernet0)#ip address 23.1.1.2 24

R2(config-if-gigabitethernet0)#exit

R3(config)#interface g0

R3(config-if-gigabitethernet0)#ip address 23.1.1.3 24

R3(config-if-gigabitethernet0)#exit

R3(config)#interface g1

R3(config-if-gigabitethernet1)#ip address 34.1.1.3 24

R3(config-if-gigabitethernet1)#exit

R3(config)#interface loo0

R3(config-if-loopback0)#ip address 192.168.1.3 24

R3(config-if-loopback0)#ip ospf network point-to-point

//将环回口掩码设置为24位宣告

R3(config-if-loopback0)#exit

R3(config)#interface loo1

R3(config-if-loopback1)#ip address 192.168.2.3 24

R3(config-if-loopback1)#ip ospf network point-to-point

R3(config-if-loopback1)#exit

sw1(config)#vlan 4

sw1(config-vlan4)#exit

sw1(config)#interface fa0/10

sw1(config-if-fastethernet0/10)#switchport mode trunk

sw1(config-if-fastethernet0/10)#switchport trunk allowed vlan all

sw1(config-if-fastethernet0/10)#exit

sw1(config)#interface fa0/2

sw1(config-if-fastethernet0/2)#switchport access vlan 4

R4(config)#interface g1

R4(config-if-gigabitethernet1)#ip address 34.1.1.4 24

R4(config-if-gigabitethernet1)#exit

R4(config)#interface g0.1

R4(config-if-gigabitethernet0.1)#encapsulation dot1q 4

R4(config-if-gigabitethernet0.1)#ip address 14.1.1.4 24

R4(config-if-gigabitethernet0.1)# keepalive gateway 14.1.1.1

R4(config-if-gigabitethernet0.1)#exit

**Rip 、OSPF配置：**

R1(config)#router rip

R1(config-rip)#no auto-summary

R1(config-rip)#version 2

R1(config-rip)#network 14.0.0.0

R1(config-rip)#network 10.0.0.0

R2(config)#router ospf 1

R2(config-ospf)#network 23.1.1.2 0.0.0.255 area 0

R2(config-ospf)#exit

R3(config)#router ospf 1

R3(config-ospf)#network 23.1.1.3 0.0.0.255 area 0

R3(config-ospf)#network 34.1.1.3 0.0.0.255 area 0

R3(config-ospf)#network 192.168.1.0 0.0.0.255 area 0

R3(config-ospf)#network 192.168.2.0 0.0.0.255 area 0

R3(config-ospf)#exit

R4(config)#router ospf 1

R4(config-ospf)#network 34.1.1.4 0.0.0.255 area 0

R4(config-ospf)#exit

R4(config)#router rip

R4(config-rip)#no auto-summary

R4(config-rip)#version 2

R4(config-rip)#network 14.0.0.0

R4(config-rip)#exit

**隧道配置：**

R1：

R1(config)#**crypto ike key maipu any //网点可以与任何分行建立IPsec**

R1(config)#crypto ike proposal 1

R1(config-ike-prop)#integrity md5

R1(config-ike-prop)#encryption 3des

R1(config-ike-prop)#exit

R1(config)#crypto ipsec proposal 1

R1(config-ipsec-prop)#esp 3des md5

R1(config-ipsec-prop)#exit

R1(config)#crypto tunnel t1

R1(config-tunnel)#peer any

R1(config-tunnel)#local address 12.1.1.1

R1(config-tunnel)#set ike proposal 1

R1(config-tunnel)#set ipsec proposal 1

R1(config-tunnel)#exit

R1(config)#crypto policy p1

R1(config-policy)#**flow 10.1.1.0 255.255.255.0 192.168.1.0 255.255.255.0 ip tunnel t1 bypass //参数bypass表示隧道不存在时忽略该条策略**

R1(config-policy)#**set reverse-route 121**

**//设置注入的静态路由管理值为121，起到静态路由备份的作用**

R1(config)#crypto policy p2

R1(config-policy)#**flow 10.1.2.0 255.255.255.0 192.168.2.0 255.255.255.0 ip tunnel t1 //设置生产网段保护流**

R1(config-policy)# **set backup gigabitethernet0 //设置生产网段备份接口为g0**

R1(config-policy)# **set reverse-route**

R2(config)#crypto ike key maipu address 12.1.1.1

R2(config)#crypto ike proposal 1

R2(config-ike-prop)#encryption 3des

R2(config-ike-prop)#integrity md5

R2(config-ike-prop)#exit

R2(config)#crypto ipsec proposal 1

R2(config-ipsec-prop)#esp 3des md5

R2(config-ipsec-prop)#exit

R2(config)#crypto tunnel t1

R2(config-tunnel)#peer address 12.1.1.1

R2(config-tunnel)#local address 12.1.1.2

R2(config-tunnel)#set ike proposal 1

R2(config-tunnel)#set ipsec proposal 1

R2(config-tunnel)#**set auto-up //分行自动发起建立隧道**

R2(config-tunnel)#exit

R2(config)#crypto policy p1

R2(config-policy)#**flow 192.168.1.0 255.255.255.0 10.1.1.0 255.255.255.0 ip tunnel t1 bypass**

R2(config-policy)#**set reverse-route**

R2(config-policy)#exit

R2(config)#crypto policy p2

R2(config-policy)# **flow 192.168.2.0 255.255.255.0 10.1.2.0 255.255.255.0 ip tunnel t1 bypass**

R2(config-policy)# **set reverse-route**

R2(config-policy)# exit

**在R4上双向重分发路由：**

R4(config)#router ospf 1

R4(config-ospf)#redistribute rip

R4(config-ospf)#exit

R4(config)#router rip

R4(config-rip)#redistribute ospf 1

R4(config-rip)#exit

在R2上将静态路由重分发进OSPF 1，并使用route-map实现分流互备。

R2(config)#ip access-list standard 1

R2(config-std-nacl)# permit 10.1.1.0 0.0.0.255

R2(config-std-nacl)# exit

R2(config)#ip access-list standard 2

R2(config-std-nacl)#permit 10.1.2.0 0.0.0.255

R2(config-std-nacl)# exit

R2(config)#route-map bc permit 10

R2(config-route-map)# match ip address 1

R2(config-route-map)# ***set metric 21 //充分发进入OSPF的均为外部路由，所以这里通过将metric值增大来实现选路***

R2(config-route-map)# exit

R2(config)#route-map bc permit 20

R2(config-route-map)# match ip address 2

R2(config-route-map)# ***set metric 10***

***//同理，这里通过将metric值减小来实现选路***

R2(config-route-map)# exit

R2(config)#route-map bc permit 30

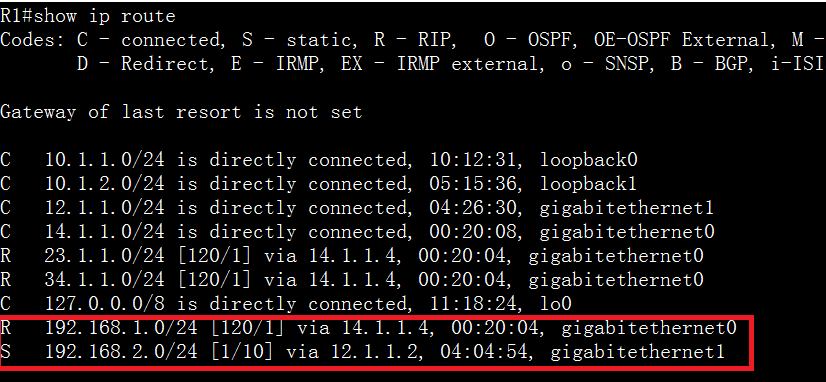
R2(config-route-map)# exit

R2(config)#router ospf 1

R2(config-ospf)# **redistribute static route-map bc //重分发调用route-map bc**

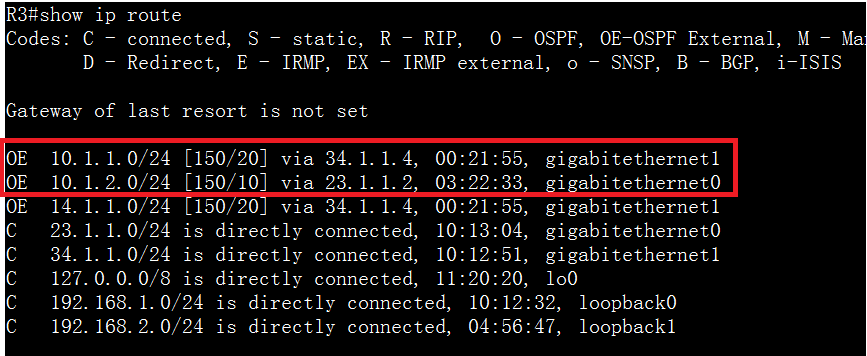
R2(config-ospf)# exit

在R1上查看正常情况下的路由表：

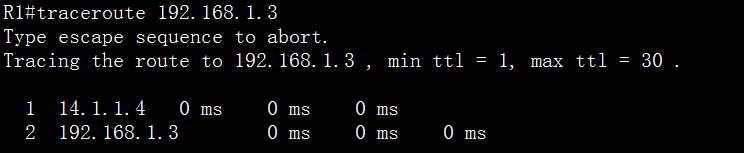


可以看到正常情况下办公网和生产网分别走上下不同路径。

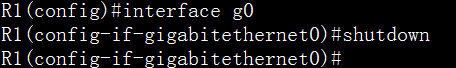
R3路由表：

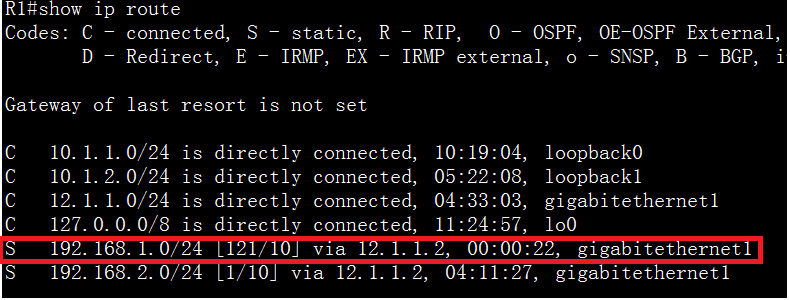


跟踪办公生产路径信息：

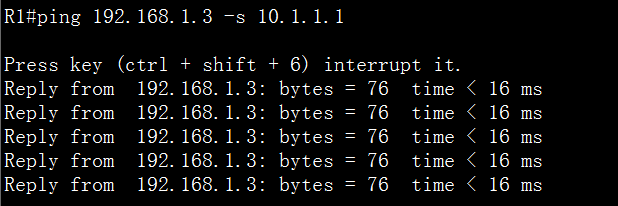


现在在R1上通过down掉g0接口后查看路由表的变化：

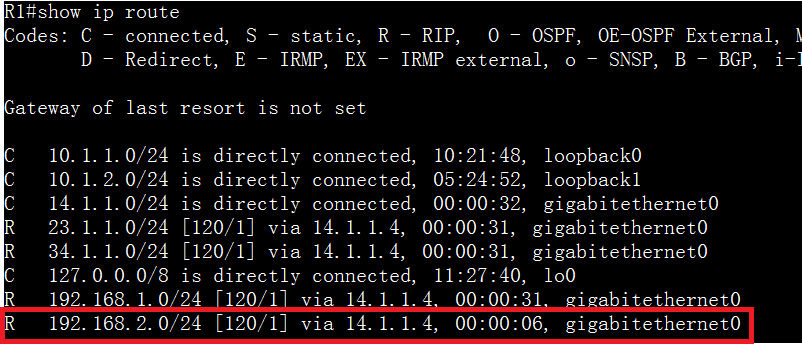


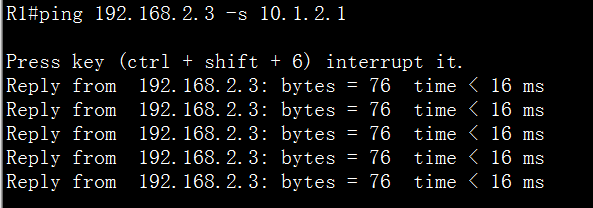


路径切换到了g1接口。



开启g0接口，拔掉g0接口：

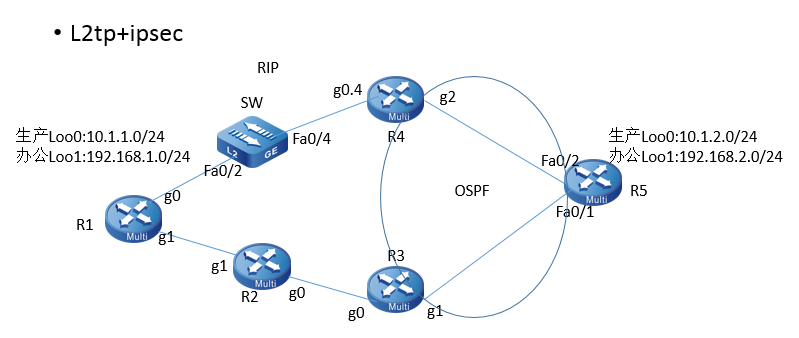




当拔掉g0接口后，生产网段切换到了g0口线路，且能与分行相互通信。

## L2tp+ipsec隧道

实验拓扑：



实验要求：（拨号实验LAC需要升级相关版本）

如图所示，在R1和R3之间建立l2tp隧道拨号（采用本地认证），拨号成功后建立IPsec隧道。使生产网走上面的线路，办公网走下面的线路，并相互备份。

首先在R1和R3上建立拨号：

R1(config)#dialer-list 1 protocol ip permit //设置拨号列表

R1(config)#interface dialer0 //设置拨号接口

R1(config-if-dialer0)# encapsulation ppp //接口封装ppp

R1(config-if-dialer0)# ppp pap sent-username maipu@maipu.com password 0 maipu

//发送本地用户名和密码进行认证

R1(config-if-dialer0)# ip address negotiated

//拨号口地址采用自动协商

R1(config-if-dialer0)# dialer in-band

R1(config-if-dialer0)# dialer pool 1

R1(config-if-dialer0)# dialer-group 1

R1(config-if-dialer0)#exit

R1(config)#interface gigabitethernet1

R1(config-if-gigabitethernet1)# pppoe-client dial-pool-number 1

//接口下启用拨号客户端使用拨号列表

R1(config-if-gigabitethernet1)# exit

R1(config)#ip route 0.0.0.0 0.0.0.0 dialer0 //默认路由指向拨号口

LAC配置：

lac(config)#interface gigabitethernet0

lac(config-if-gigabitethernet0)# ip address 23.1.1.2 255.255.255.0

lac(config-if-gigabitethernet0)# exit

lac(config)#**interface virtual-template0 //配置虚模板接口**

lac(config-if-virtual-template0)# **encapsulation ppp //虚模板使用ppp封装**

lac(config-if-virtual-template0)# **ppp authentication pap //认证采用pap认证**

lac(config-if-virtual-template0)# exit

lac(config)#interface gigabitethernet1

lac(config-if-gigabitethernet1)# **pppoe enable //接口下使能PPPOE拨号**

lac(config-if-gigabitethernet1)# exit

lac(config)#**vpdn enable //使能vpdn组**

lac(config)#**vpdn-group pppoe //配置vpdn组**

lac(config-vpdn)# accept-dialin

lac(config-vpdn-acc-in)# protocol pppoe

lac(config-vpdn-acc-in)# virtual-template 0

lac(config-vpdn-acc-in)# exit

lac(config-vpdn)# exit

lac(config)#vpdn-group lac

lac(config-vpdn)# **request-dialin**

**//让LAC接收远程接入客户的PPP会话，并尝试通过隧道将它们延伸到指定的LNS**

lac(config-vpdn-req-in)#**protocol l2tp //指定建立隧道时使用VPDN协议为L2TP**

lac(config-vpdn-req-in)#**domain maipu.com //该命令使得域名为maipu.com的远程接入用户的 PPP会话将通过隧道被延伸到VPDN组中指定的LNS**

lac(config-vpdn-req-in)#exit

lac(config-vpdn)**# initiate-to ip 23.1.1.3 //指定了隧道另一端的LNS的IP地址**

lac(config-vpdn)# **local name lac //配置本地主机名**

lac(config-vpdn)# **l2tp tunnel password 0 maipu**

**//指定了LNS和LAC验证对方时使用的共享密钥**

lac(config-vpdn)# exit

R3(config)#**user maipu@maipu.com password 0 maipu**

**//远程客户端的用户名和密码**

R3(config)#**ip local pool l2tp 172.16.1.10 172.16.1.100**

**//为远程客户分配IP的地址池**

R3(config)#interface loopback0

R3(config-if-loopback0)# ip address 3.3.3.3 255.255.255.0

R3(config-if-loopback0)# exit

R3(config)#interface virtual-template0

R3(config-if-virtual-template0)# encapsulation ppp

R3(config-if-virtual-template0)# **peer default ip address pool l2tp**

**//给远程客户端从地址池中分配IP地址**

R3(config-if-virtual-template0)# ppp authentication pap

R3(config-if-virtual-template0)# **ip unnumbered loopback0**

**//虚模板接口借用loo0的地址**

R3(config-if-virtual-template0)# exit

R3(config)#interface gigabitethernet0

R3(config-if-gigabitethernet0)# ip address 23.1.1.3 255.255.255.0

R3(config-if-gigabitethernet0)# exit

R3(config)#vpdn enable

R3(config)#vpdn-group lns

R3(config-vpdn)# accept-dialin //**该命令指定LNS接收来自LAC的隧道和会话**

R3(config-vpdn-acc-in)#protocol l2tp

R3(config-vpdn-acc-in)#virtual-template 0

**//该命令导致任何通过隧道来自LAC的入站PPP会话都将由虚拟接入接口端接，虚拟接入接口复制了虚拟模板接口0的配置。**

R3(config-vpdn-acc-in)#exit

R3(config-vpdn)# terminate-from hostname lac

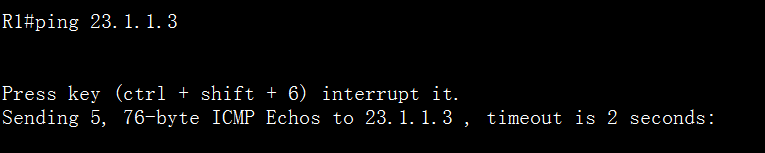
R3(config-vpdn)# **l2tp tunnel password 0 maipu**

**//指定了LNS和LAC验证对方时使用的共享密钥**

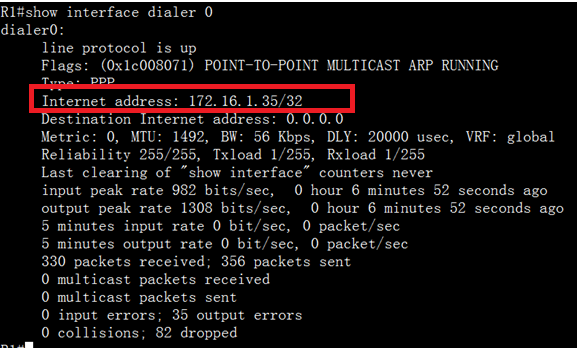
R3(config-vpdn)# exit

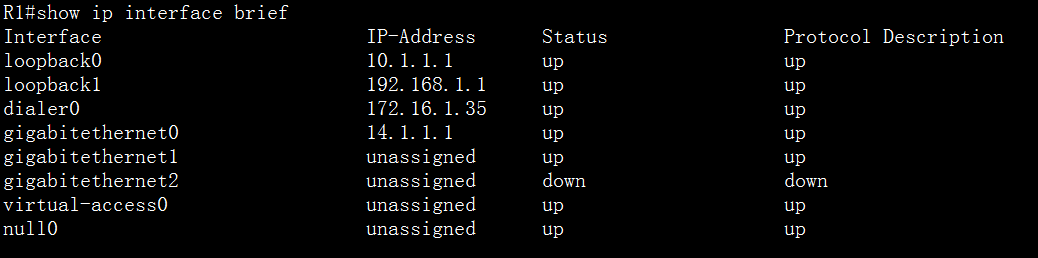
R3(config)#ip route 0.0.0.0 0.0.0.0 gigabitethernet1

配置完后在R1上ping触发拨号：

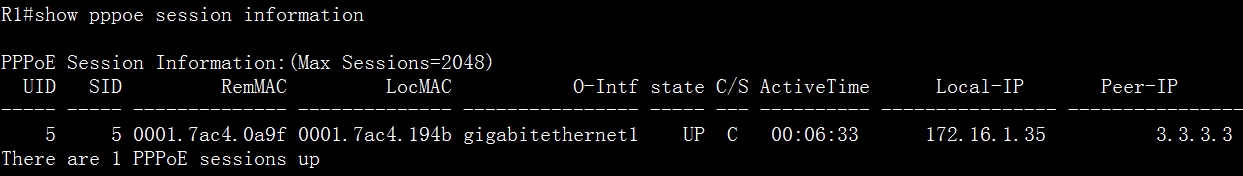


拨号成功前无法ping通。拨号成功后可以查看拨号口是否获取到了地址：





查看PPPOE会话：



可以看到拨号已成功且获取到了ip地址。

**建立了l2tp隧道后，在此基础上建立IPsec：**

R1(config)#interface loopback0

R1(config-if-loopback0)# ip address 10.1.1.1 255.255.255.0

R1(config-if-loopback0)# ip ospf network point-to-point

R1(config-if-loopback0)# exit

R1(config)#interface loopback1

R1(config-if-loopback1)# ip address 192.168.1.1 255.255.255.0

R1(config-if-loopback1)# ip ospf network point-to-point

R1(config-if-loopback1)# exit

R1(config)#interface gigabitethernet0

R1(config-if-gigabitethernet0)# ip address 14.1.1.1 255.255.255.0

R1(config-if-gigabitethernet0)# exit

R1(config)#crypto ike key maipu address 23.1.1.3

R1(config)#crypto ike proposal 1

R1(config-ike-prop)# encryption 3des

R1(config-ike-prop)# integrity md5

R1(config-ike-prop)# exit

R1(config)#crypto ipsec proposal 1

R1(config-ipsec-prop)# esp 3des md5

R1(config-ipsec-prop)# exit

R1(config)#crypto tunnel t1

R1(config-tunnel)# local interface dialer0

R1(config-tunnel)# peer address 23.1.1.3

R1(config-tunnel)# set ike proposal 1

R1(config-tunnel)# set ipsec proposal 1

R1(config-tunnel)# set auto-up

R1(config-tunnel)# exit

R1(config)#crypto policy p1

R1(config-policy)# flow 192.168.1.0 255.255.255.0 192.168.2.0 255.255.255.0 ip tunnel t1 bypass

R1(config-policy)# set reverse-route

R1(config-policy)# exit

R1(config)#crypto policy p2

R1(config-policy)# flow 10.1.1.0 255.255.255.0 10.1.2.0 255.255.255.0 ip tunnel t1 bypass

R1(config-policy)# set reverse-route 121

R1(config-policy)# exit

R1(config)#router rip

R1(config-rip)# version 2

R1(config-rip)# network 10.0.0.0

R1(config-rip)# network 14.0.0.0

R1(config-rip)# network 192.168.1.0

R1(config-rip)# no auto-summary

**R3配置**：

R3(config)#ip access-list standard 1

R3(config-std-nacl)# 10 permit 10.1.1.0 0.0.0.255

R3(config-std-nacl)# exit

R3(config)#ip access-list standard 2

R3(config-std-nacl)# 10 permit 192.168.1.0 0.0.0.255

R3(config-std-nacl)# exit

R3(config)#interface gigabitethernet1

R3(config-if-gigabitethernet1)# ip address 35.1.1.3 255.255.255.0

R3(config-if-gigabitethernet1)# exit

R3(config)#crypto ike key maipu any

R3(config)#crypto ike proposal 1

R3(config-ike-prop)# encryption 3des

R3(config-ike-prop)# integrity md5

R3(config-ike-prop)# exit

R3(config)#crypto ipsec proposal 1

R3(config-ipsec-prop)# esp 3des md5

R3(config-ipsec-prop)# exit

R3(config)#crypto tunnel t1

R3(config-tunnel)# local address 23.1.1.3

R3(config-tunnel)# peer any

R3(config-tunnel)# set ike proposal 1

R3(config-tunnel)# set ipsec proposal 1

R3(config-tunnel)# exit

R3(config)#crypto policy p1

R3(config-policy)# flow 192.168.2.0 255.255.255.0 192.168.1.0 255.255.255.0 ip tunnel t1 bypass

R3(config-policy)# set reverse-route

R3(config-policy)# exit

R3(config)#crypto policy p2

R3(config-policy)# flow 10.1.2.0 255.255.255.0 10.1.1.0 255.255.255.0 ip tunnel t1 bypass

R3(config-policy)# set reverse-route

R3(config-policy)# exit

R3(config)#route-map r3-r5 permit 10

R3(config-route-map)# match ip address 1

R3(config-route-map)# set metric 21

R3(config-route-map)# exit

R3(config)#route-map r3-r5 permit 20

R3(config-route-map)# match ip address 2

R3(config-route-map)# set metric 19

R3(config-route-map)# exit

R3(config)#route-map r3-r5 permit 30

R3(config-route-map)# exit

R3(config)#router ospf 1

R3(config-ospf)# network 35.1.1.0 0.0.0.255 area 0

%OSPF There is already the same network statement

R3(config-ospf)# redistribute static route-map r3-r5

**R4配置：**

R4(config)#vlan 4

R4(config-vlan4)# exit

R4(config)#interface gigabitethernet0.4

R4(config-if-gigabitethernet0.4)# ip address 14.1.1.4 255.255.255.0

R4(config-if-gigabitethernet0.4)# encapsulation dot1q 4

R4(config-if-gigabitethernet0.4)# exit

R4(config)#interface gigabitethernet2

R4(config-if-gigabitethernet2)# ip address 45.1.1.4 255.255.255.0

R4(config-if-gigabitethernet2)# exit

R4(config)#router rip

R4(config-rip)# version 2

R4(config-rip)# network 14.0.0.0

R4(config-rip)# redistribute ospf 1

R4(config-rip)# no auto-summary

R4(config-rip)# exit

R4(config)#router ospf 1

R4(config-ospf)# network 45.1.1.0 0.0.0.255 area 0

R4(config-ospf)# redistribute rip

R4(config-ospf)# exit

**R5配置：**

R5(config)#vlan 35,45

R5(config)#interface fastethernet0/1

R5(config-if-fastethernet0/1)# switchport access vlan 35

R5(config-if-fastethernet0/1)# exit

R5(config)#interface fastethernet0/2

R5(config-if-fastethernet0/2)# switchport access vlan 45

R5(config-if-fastethernet0/2)# exit

R5(config)#interface loopback0

R5(config-if-loopback0)# ip address 10.1.2.5 255.255.255.0

R5(config-if-loopback0)# ip ospf network point-to-point

R5(config-if-loopback0)# exit

R5(config)#interface loopback1

R5(config-if-loopback1)# ip address 192.168.2.5 255.255.255.0

R5(config-if-loopback1)# ip ospf network point-to-point

R5(config-if-loopback1)# exit

R5(config)#interface vlan35

R5(config-if-vlan35)# ip address 35.1.1.5 255.255.255.0

R5(config-if-vlan35)# exit

R5(config)#interface vlan45

R5(config-if-vlan45)# ip address 45.1.1.5 255.255.255.0

R5(config-if-vlan45)# exit

R5(config)#router ospf 1

R5(config-ospf)# network 10.1.2.0 0.0.0.255 area 0

R5(config-ospf)# network 35.1.1.0 0.0.0.255 area 0

R5(config-ospf)# network 45.1.1.0 0.0.0.255 area 0

R5(config-ospf)# network 192.168.2.0 0.0.0.255 area 0

R5(config-ospf)# exit

**SW配置：**

sw1(config)#vlan 4

sw1(config-vlan4)# exit

sw1(config)#interface fastethernet0/2

sw1(config-if-fastethernet0/2)# switchport access vlan 4

sw1(config-if-fastethernet0/2)# exit

sw1(config)#interface fastethernet0/4

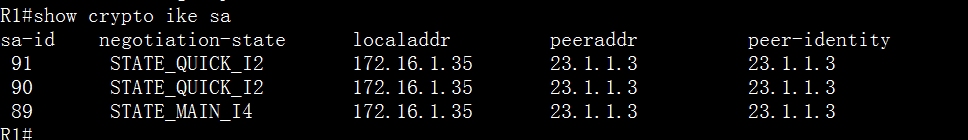
sw1(config-if-fastethernet0/4)# switchport mode trunk

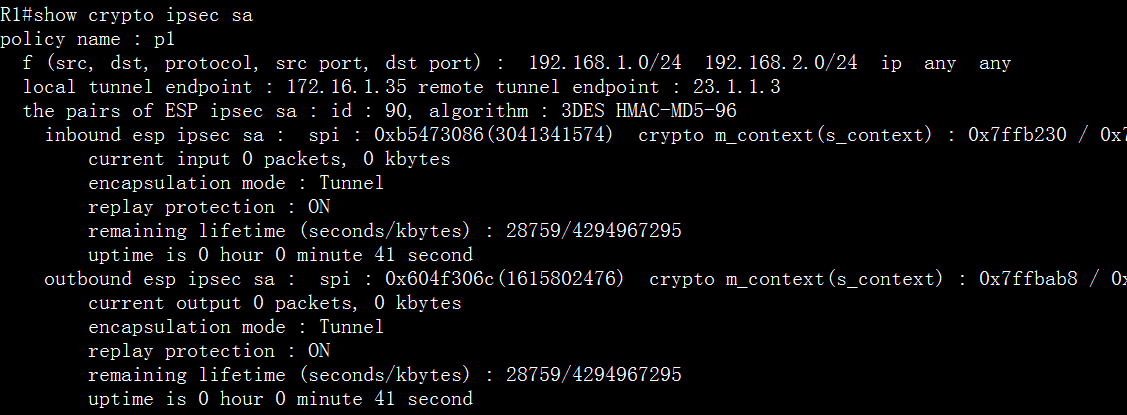
sw1(config-if-fastethernet0/4)# switchport trunk allowed vlan all

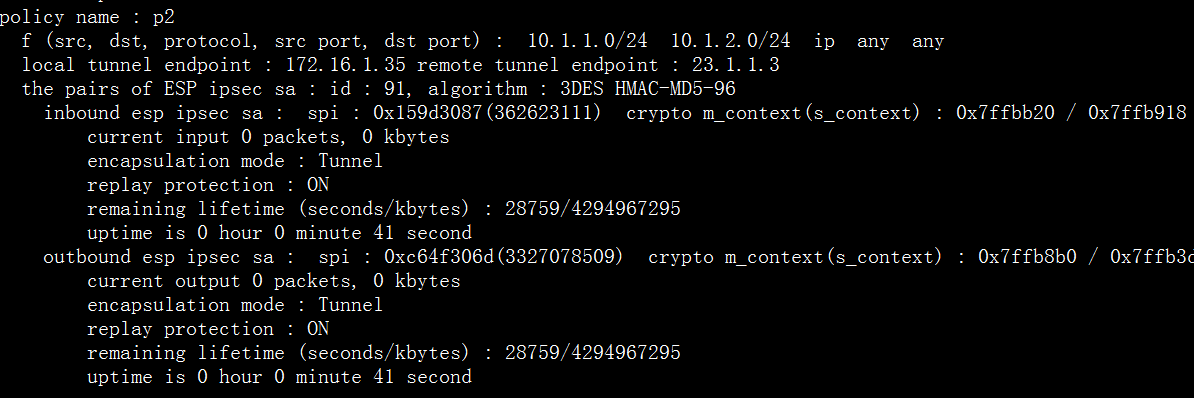
sw1(config-if-fastethernet0/4)# switchport trunk pvid vlan 1

sw1(config-if-fastethernet0/4)# exit

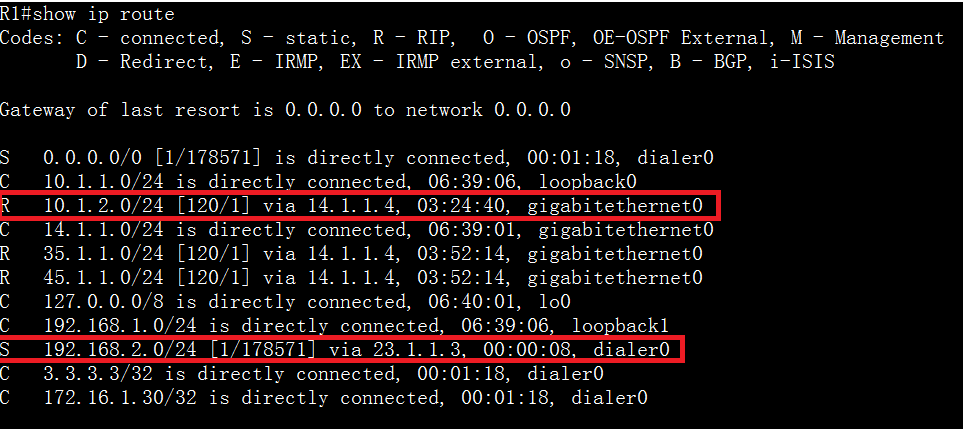
配置完成后查看隧道状态：



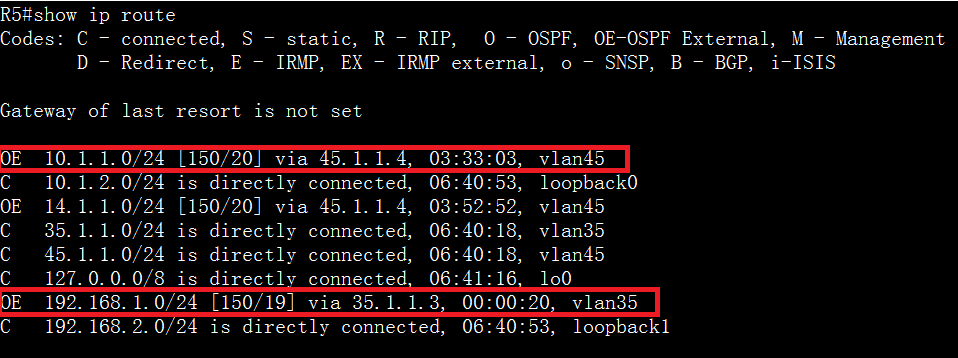


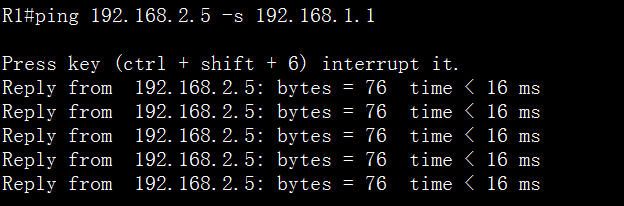


隧道已建立。查看R1的路由表：生产网段走上面，办公网段走下面。

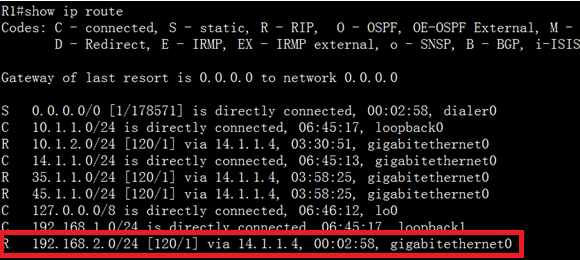


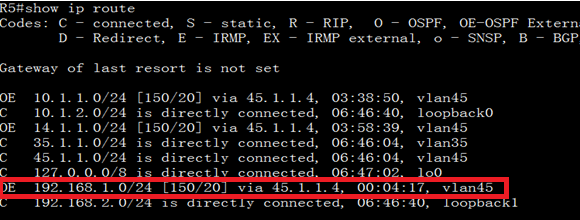
R5路由表：

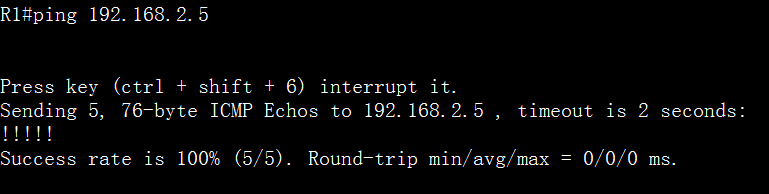




现在断掉R1的g1拨号口，查看路由表：办公网段切换到了g0口





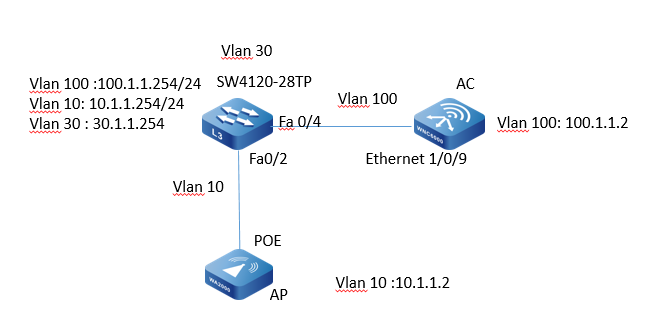


# 第五章 无线实验部分



## AP静态注册

实验拓扑如下：



实验要求：

1. 所有网关在SW4120上。
2. AP静态注册上AC。
3. 释放ssid为:mp123
4. PC自动获取ip地址（30网段）
5. 本地转发。

实验配置：

AP基本配置：

当AP通过CRT连接时，波特率设置为115200。初始登陆用户名和密码为admin，admin。

MAIPU-WLAN-AP#**factory-reset //将AP恢复出厂设置**

MAIPU-WLAN-AP# **set management static-ip 10.1.1.2**

//设置AP的静态ip地址为10.1.1.2

MAIPU-WLAN-AP# **set management static-mask 255.255.255.0**

//设置网络地址掩码

MAIPU-WLAN-AP# **set static-ip-route gateway 10.1.1.254**

//设置AP网关地址，该地址为交换机与AP连接的接口地址

MAIPU-WLAN-AP# **set managed-ap switch-address-1 100.1.1.2**

//设置用于AP发现AC的地址

MAIPU-WLAN-AP# **set management dhcp-status down**

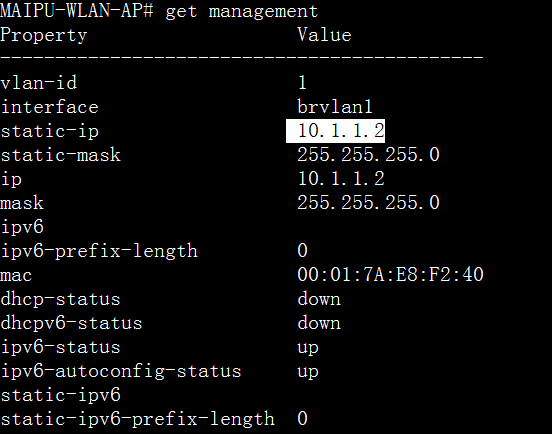
//关闭dhcp地址自动获取功能。

MAIPU-WLAN-AP# save-running

//保存配置

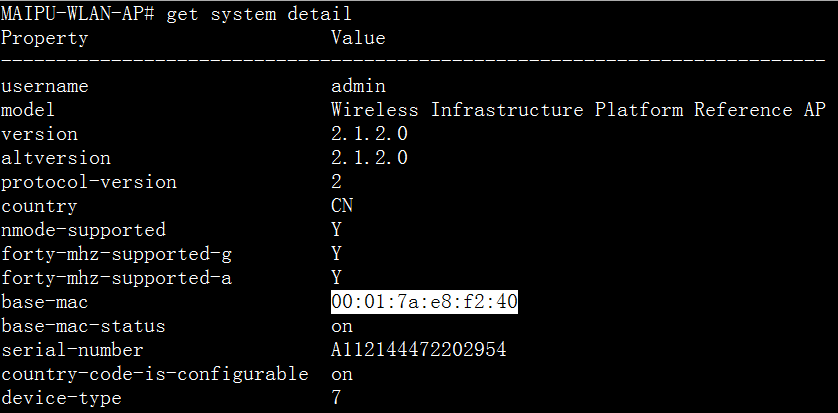
查看配置命令：

MAIPU-WLAN-AP# **get management**



查看系统参数：

MAIPU-WLAN-AP# **get system detail**



记下AP的硬件地址base-mac和设备类型device-type参数。

**SW4120配置：**

SW2(config)#vlan 10,30,100

SW2(config)#interface fastethernet0/2

SW2(config-if-fastethernet0/2)# switchport mode trunk

SW2(config-if-fastethernet0/2)# switchport trunk allowed vlan all

SW2(config-if-fastethernet0/2)# ***switchport trunk pvid vlan 10***

**//此处端口模式为trunk，且pvid为vlan10，是因为pc获取的地址和AP的ip地址不是在同一vlan，且均需要和AC通信。**

SW2(config-if-fastethernet0/2)# exit

SW2(config)#interface fastethernet0/4

SW2(config-if-fastethernet0/4)# switchport mode trunk

SW2(config-if-fastethernet0/4)# switchport trunk allowed vlan all

SW2(config-if-fastethernet0/4)# exit

SW2(config)#interface vlan10

SW2(config-if-vlan10)# ip address 10.1.1.254 255.255.255.0

//该vlan10 ip地址作为AP的网关

SW2(config-if-vlan10)# exit

SW2(config)#interface vlan30

SW2(config-if-vlan30)# ip address 30.1.1.254 255.255.255.0

//该vlan30 IP地址作为客户端网关。

SW2(config-if-vlan30)# exit

SW2(config)#interface vlan100

SW2(config-if-vlan100)# ip address 100.1.1.254 255.255.255.0

//vlan100 IP地址作为AC网关。

SW2(config-if-vlan100)# exit

SW2(config)#ip dhcp pool p1 //配置用于分配给客户端的ip地址池。

SW2(dhcp-config)# range 30.1.1.1 30.1.1.253 255.255.255.0

SW2(dhcp-config)# default-router 30.1.1.254

SW2(dhcp-config)# exit

**AC6000配置：**

WNC6000-1000-AC(v2)#config t

WNC6000-1000-AC(v2)(config)#no interface vlan1

WNC6000-1000-AC(v2)(config)#vlan 100

WNC6000-1000-AC(v2)(config-vlan100)#exit

WNC6000-1000-AC(v2)(config)#interface vlan 100

WNC6000-1000-AC(v2)(config-if-vlan100)#ip address 100.1.1.2 255.255.255.0

//配置AC管理vlan ip地址。

WNC6000-1000-AC(v2)(config-if-vlan100)#exit

WNC6000-1000-AC(v2)(config)#interface ethernet 1/0/9

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)#switchport mode trunk

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)#switchport trunk allowed vlan all

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)#exit

WNC6000-1000-AC(v2)(config)#ip route 0.0.0.0 0.0.0.0 100.1.1.254

**//配置AC缺省路由。**

WNC6000-1000-AC(v2)(config)#wireless

WNC6000-1000-AC(v2)(config-wireless)#enable

WNC6000-1000-AC(v2)(config-wireless)#**no auto-ip-assign**

**//关闭AC自动IP地址分配。**

WNC6000-1000-AC(v2)(config-wireless)#static-ip 100.1.1.2

WNC6000-1000-AC(v2)(config-wireless)#**ap database 00-01-7a-e8-f2-40**

**//将AP硬件地址添加到AC数据库。默认对应profile1**

WNC6000-1000-AC(v2)(config-ap)#exit

WNC6000-1000-AC(v2)(config-wireless)#ap profile 1

WNC6000-1000-AC(v2)(config-ap-profile)#**hwtype 7 //添加AP硬件类型**

WNC6000-1000-AC(v2)(config-ap-profile)#exit

WNC6000-1000-AC(v2)(config-wireless)#network 1

WNC6000-1000-AC(v2)(config-network)#**ssid mp123**

**//配置ssid为mp123**

WNC6000-1000-AC(v2)(config-network)#**vlan 30**

**//下发到用户的IP地址段属于vlan30**

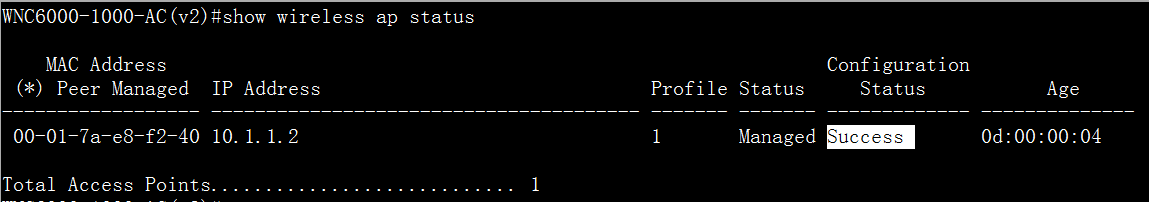
WNC6000-1000-AC(v2)(config-network)#end

WNC6000-1000-AC(v2)#**wireless ap profile apply 1**

//下发配置给AP。

结果验证：

在AC上查看AP状态: WNC6000-1000-AC(v2)#show wireless ap status

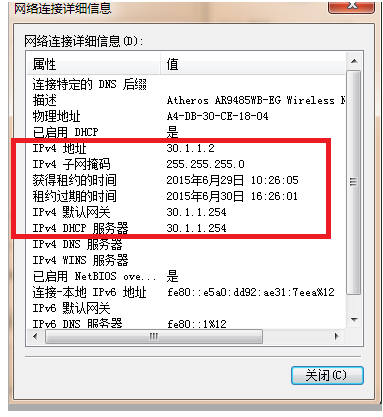


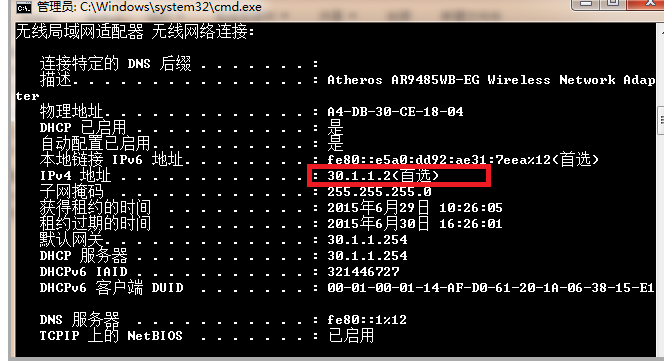
可以看到AP已成功注册到AC上，且配置已下发成功。

我们在PC机上打开无线网络可以看到：



Ssid名为mp123的网络出现在列表中，连接上该网络。





PC机已获取到30网段的地址及其网关信息。

在PC机上ping AC地址看其能否与AC通信。



## AC集中转发

在上面的实验基础上统一由AC集中转发用户数据流量。

AC上需要增加的配置：

WNC6000-1000-AC(v2)(config)#wireless

WNC6000-1000-AC(v2)(config-wireless)#l2tunnel vlan-list add1,30

//配置vlan30用户为集中转发。

WNC6000-1000-AC(v2)(config-wireless)#no enable

WNC6000-1000-AC(v2)(config-wireless)#enable

## 同一AP同时释放两个ssid

实验拓扑同实验一，在实验一的配置基础上实现一颗AP释放两个SSID。且SSID1：vlan30使用本地转发，SSID2:vlan40使用集中转发。SSID2网段为40.1.1.0/24。

基础配置：

**SW4120需要增加的配置**：

SW2(config)#vlan 40

SW2(config-vlan40)#exit

SW2(config)#interface vlan40

SW2(config-if-vlan40)#ip address 40.1.1.254 24

SW2(config-if-vlan40)#exit

SW2(config)#ip dhcp pool p2

SW2(dhcp-config)#range 40.1.1.1 40.1.1.253 255.255.255.0

SW2(dhcp-config)#default-router 40.1.1.254

SW2(dhcp-config)#exit

**AC上需要增加的配置：**

WNC6000-1000-AC(v2)(config)#wireless

WNC6000-1000-AC(v2)(config-wireless)#ap profile 1

WNC6000-1000-AC(v2)(config-ap-profile)#radio 1

WNC6000-1000-AC(v2)(config-ap-profile-radio)#vap 1

WNC6000-1000-AC(v2)(config-ap-profile-vap)#enable **//使能vap1**

WNC6000-1000-AC(v2)(config-ap-profile-vap)#exit

WNC6000-1000-AC(v2)(config-ap-profile-radio)#exit

WNC6000-1000-AC(v2)(config-ap-profile)#exit

WNC6000-1000-AC(v2)(config-wireless)#network 2

WNC6000-1000-AC(v2)(config-network)#**ssid mp321**

**//设置network2 ssid为mp321**

WNC6000-1000-AC(v2)(config-network)#vlan 40

//ssid为mp321，分配给用户的网段为40网段。

WNC6000-1000-AC(v2)(config-network)#exit

WNC6000-1000-AC(v2)(config-wireless)#no l2tunnel vlan-list

WNC6000-1000-AC(v2)(config-wireless)#**l2tunnel vlan-list add 1,40**

**//设置ssid为mp321的用户采用集中转发。**

WNC6000-1000-AC(v2)(config-wireless)#exit

WNC6000-1000-AC(v2)(config)#exit

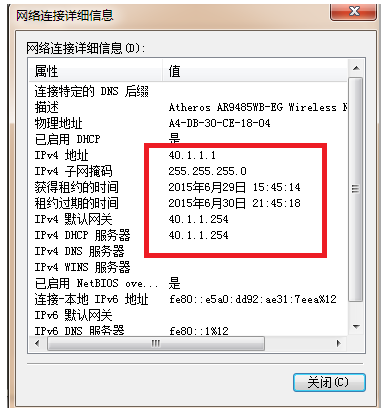
WNC6000-1000-AC(v2)#wireless ap profile apply 1

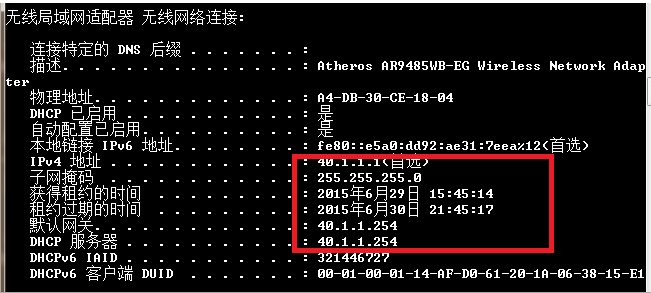
//将配置重新下发到AP。

验证：

当AC下发配置到AP成功后，打开PC机的无线网络当搜索到如下图所示ssid后，连接上mp321。







上图表明PC机获取到了Vlan40 网段的ip地址及其网关信息。

## AC管理多AP

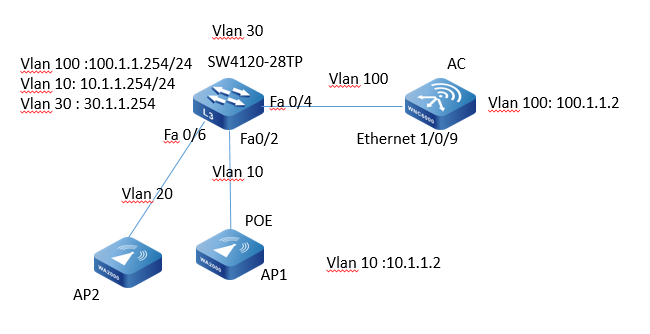
实验要求：

在实验一的拓扑基础上另外增加一颗AP，并由AC统一管理。

AP-1释放的ssid为vlan30，采用本地转发。

AP-2释放的ssid为vlan40，采用集中转发。

实验拓扑：



基本配置：

AP1配置不变：

AP2配置：

MAIPU-WLAN-AP# **set management static-ip 20.1.1.2**

//设置AP的静态ip地址为20.1.1.2

MAIPU-WLAN-AP# **set management static-mask 255.255.255.0**

//设置网络地址掩码

MAIPU-WLAN-AP# **set static-ip-route gateway 20.1.1.254**

//设置AP网关地址，该地址为交换机与AP连接的接口地址

MAIPU-WLAN-AP# **set managed-ap switch-address-1 100.1.1.2**

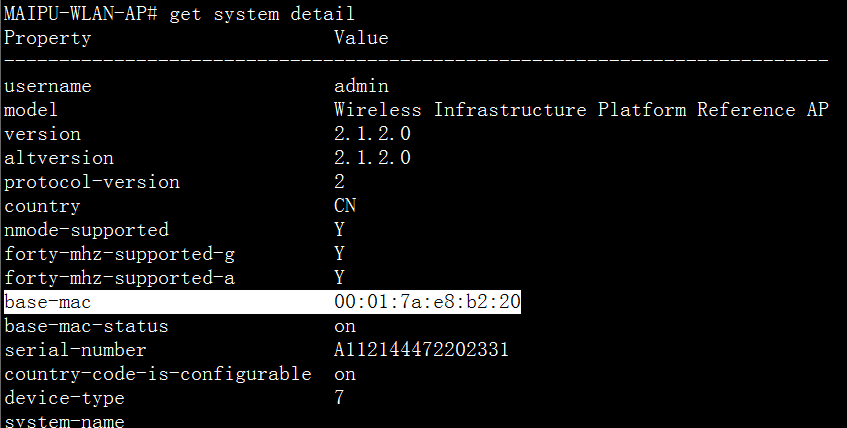
//设置用于AP发现AC的地址

MAIPU-WLAN-AP# **set management dhcp-status down**

//关闭dhcp地址自动获取功能。

MAIPU-WLAN-AP# save-running

查看AP系统信息：



仍然记录下该AP的硬件地址信息。

交换机上的配置：

SW2(config)#vlan 10,30,20,100

SW2(config)#interface fastethernet0/2

SW2(config-if-fastethernet0/2)# switchport mode trunk

SW2(config-if-fastethernet0/2)# switchport trunk allowed vlan all

SW2(config-if-fastethernet0/2)# ***switchport trunk pvid vlan 10***

**//此处端口模式为trunk，且pvid为vlan10，是因为pc获取的地址和AP的ip地址不是在同一vlan，且均需要和AC通信。**

SW2(config-if-fastethernet0/2)# exit

SW2(config)#interface fastethernet0/4

SW2(config-if-fastethernet0/4)# switchport mode trunk

SW2(config-if-fastethernet0/4)# switchport trunk allowed vlan all

SW2(config-if-fastethernet0/4)# exit

SW2(config)#interface vlan10

SW2(config-if-vlan10)# ip address 10.1.1.254 255.255.255.0

//该vlan10 ip地址作为AP的网关

SW2(config-if-vlan10)# exit

SW2(config)#interface vlan30

SW2(config-if-vlan30)# ip address 30.1.1.254 255.255.255.0

//该vlan30 IP地址作为客户端网关。

SW2(config-if-vlan30)# exit

SW2(config)#interface vlan100

SW2(config-if-vlan100)# ip address 100.1.1.254 255.255.255.0

//vlan100 IP地址作为AC网关。

SW2(config-if-vlan100)# exit

SW2(config)#interface vlan20

SW2(config-if-vlan20)#ip address 20.1.1.254 255.255.255.0

SW2(config-if-vlan20)#exit

SW2(config)#ip dhcp pool p1 //配置用于分配给客户端的ip地址池。

SW2(dhcp-config)# range 30.1.1.1 30.1.1.253 255.255.255.0

SW2(dhcp-config)# default-router 30.1.1.254

SW2(dhcp-config)# exit

SW2(config)#ip dhcp pool p2

SW2(dhcp-config)# range 40.1.1.1 40.1.1.253 255.255.255.0

SW2(dhcp-config)# default-router 40.1.1.254

SW2(dhcp-config)# exit

SW2(config)#interface fa0/6

SW2(config-if-fastethernet0/6)#switchport mode trunk

SW2(config-if-fastethernet0/6)#switchport trunk allowed vlan all

SW2(config-if-fastethernet0/6)#switchport trunk pvid vlan 20

AC配置：

WNC6000-1000-AC(v2)#config t

WNC6000-1000-AC(v2)(config)#no interface vlan1

WNC6000-1000-AC(v2)(config)#vlan 100

WNC6000-1000-AC(v2)(config-vlan100)#exit

WNC6000-1000-AC(v2)(config)#interface vlan 100

WNC6000-1000-AC(v2)(config-if-vlan100)#ip address 100.1.1.2 255.255.255.0

//配置AC管理vlan ip地址。

WNC6000-1000-AC(v2)(config-if-vlan100)#exit

WNC6000-1000-AC(v2)(config)#interface ethernet 1/0/9

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)#switchport mode trunk

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)#switchport trunk allowed vlan all

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)#exit

WNC6000-1000-AC(v2)(config)#ip route 0.0.0.0 0.0.0.0 100.1.1.254

**//配置AC缺省路由。**

WNC6000-1000-AC(v2)(config)#wireless

WNC6000-1000-AC(v2)(config-wireless)#enable

WNC6000-1000-AC(v2)(config-wireless)#**no auto-ip-assign**

**//关闭AC自动IP地址分配。**

WNC6000-1000-AC(v2)(config-wireless)#static-ip 100.1.1.2

WNC6000-1000-AC(v2)(config-wireless)#**ap database 00-01-7a-e8-f2-40**

**//将AP1硬件地址添加到AC数据库。**

WNC6000-1000-AC(v2)(config-ap)#exit

WNC6000-1000-AC(v2)(config-wireless)#ap profile 1

WNC6000-1000-AC(v2)(config-ap-profile)#**hwtype 7 //添加AP1硬件类型**

WNC6000-1000-AC(v2)(config-ap-profile)#exit

WNC6000-1000-AC(v2)(config-wireless)#network 1

WNC6000-1000-AC(v2)(config-network)#**ssid mp123**

**//配置ssid为mp123**

WNC6000-1000-AC(v2)(config-network)#**vlan 30**

**//下发到AP1的用户的IP地址段属于vlan30**

WNC6000-1000-AC(v2)(config-network)#exit

WNC6000-1000-AC(v2)(config-wireless)#ap database 00-01-7a-e8-b2-20

**//添加AP2的硬件地址到数据库**

WNC6000-1000-AC(v2)(config-ap)#**profile 2**

**//将AP2与profile 2对应。**

WNC6000-1000-AC(v2)(config-ap)#exit

WNC6000-1000-AC(v2)(config-wireless)#**network 2**

WNC6000-1000-AC(v2)(config-network)#**ssid mp456**

WNC6000-1000-AC(v2)(config-network)#**vlan 40**

WNC6000-1000-AC(v2)(config-network)#exit

WNC6000-1000-AC(v2)(config-wireless)#**l2tunnel vlan-list add 1,40**

**//将vlan40的用户集中下发。**

WNC6000-1000-AC(v2)(config-wireless)#ap profile 2

WNC6000-1000-AC(v2)(config-ap-profile)#radio 1

WNC6000-1000-AC(v2)(config-ap-profile-radio)#**vap 1**

**//使能vap1**

WNC6000-1000-AC(v2)(config-ap-profile-vap)#enable

WNC6000-1000-AC(v2)#wireless ap profile apply 1

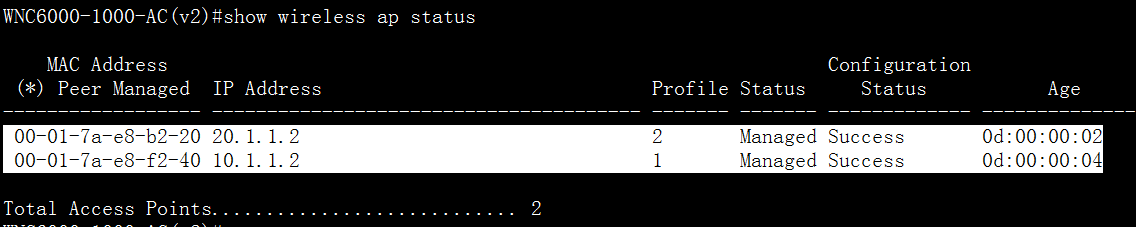
**//下发配置profile1**

WNC6000-1000-AC(v2)#wireless ap profile apply 2

**//下发配置profile2**

**验证：**

**查看是否下发配置成功：show wireless ap status**

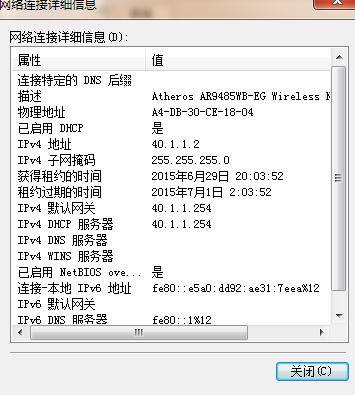


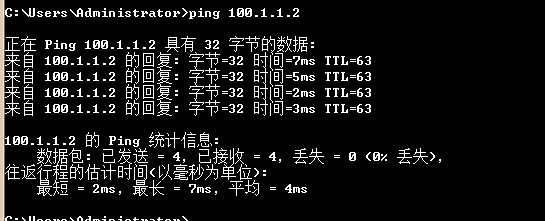
我们看到配置已成功下发到AP1和AP2。

打开PC无线网络，可以看到如下图所示两个网络。



点击任意一个均可以通过AP连接上网络。

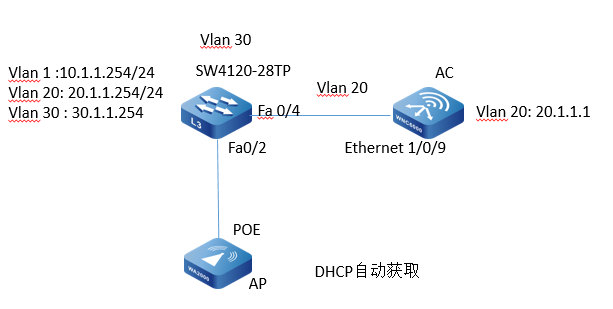




PC连接到了AP2释放出的网络信号，自动获取到40.1.1.2 IP地址。且能够与AC相互通信。

## AP三层自动注册到AC

实验拓扑如图：



实验要求：AP恢复出厂设置通过DHCP option43方式注册到AC。地址规划如图所示。

SW4120基本配置如下：

SW2(config)#vlan 20,30

SW2(config)#interface vlan1

SW2(config-if-vlan1)# ip address 10.1.1.254 255.255.255.0

SW2(config-if-vlan1)# exit

SW2(config)#interface vlan20

SW2(config-if-vlan20)# ip address 20.1.1.254 255.255.255.0

SW2(config-if-vlan20)# exit

SW2(config)#interface vlan30

SW2(config-if-vlan30)# ip address 30.1.1.254 255.255.255.0

SW2(config-if-vlan30)# exit

SW2(config)#ip dhcp pool vlan10

SW2(dhcp-config)# range 10.1.1.1 10.1.1.253 255.255.255.0

SW2(dhcp-config)# default-router 10.1.1.254

SW2(dhcp-config)# option 43 hex 80 07 00 00 01 14 01 01 01

**//option43字段：80为固定字段，07表示后面有7个字节，00 00为固定字段，01表示后面跟一个AC地址，后面四个字节是AC ip地址的十六进制表示**

SW2(dhcp-config)# exit

SW2(config)#ip dhcp pool vlan30

SW2(dhcp-config)# range 30.1.1.1 30.1.1.253 255.255.255.0

SW2(dhcp-config)# default-router 30.1.1.254

SW2(dhcp-config)# exit

SW2(config)#interface fa0/1-0/24

SW2(config-if-range)#switchport mode trunk

SW2(config-if-range)#switchport trunk allowed vlan all

AC基本配置：

WNC6000-1000-AC(v2)(config)#vlan 20

WNC6000-1000-AC(v2)(config-vlan20)#exit

WNC6000-1000-AC(v2)(config)#interface Vlan20

WNC6000-1000-AC(v2)(config-if-vlan20)# ip address 20.1.1.1 255.255.255.0

WNC6000-1000-AC(v2)(config-if-vlan20)#exit

WNC6000-1000-AC(v2)(config)#ip route 0.0.0.0/0 20.1.1.254

WNC6000-1000-AC(v2)(config)#wireless

WNC6000-1000-AC(v2)(config-wireless)# no auto-ip-assign

WNC6000-1000-AC(v2)(config-wireless)# enable

WNC6000-1000-AC(v2)(config-wireless)# static-ip 20.1.1.1

WNC6000-1000-AC(v2)(config-wireless)#network 1

WNC6000-1000-AC(v2)(config-network)#ssid text

WNC6000-1000-AC(v2)(config-network)#vlan 30

WNC6000-1000-AC(v2)(config-network)#exit

WNC6000-1000-AC(v2)(config-wireless)#ap database 00-01-7a-e8-f2-40

WNC6000-1000-AC(v2)(config-wireless)#ap profile 1

WNC6000-1000-AC(v2)(config-ap-profile)#hwtype 7

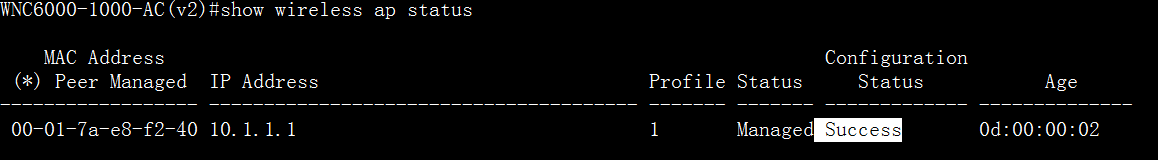
WNC6000-1000-AC(v2)(config)#Interface Ethernet1/0/9

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)# switchport mode trunk

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)#switchport trunk allowed vlan all

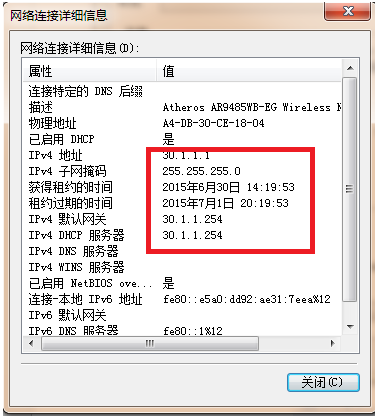
WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)#

查看是否下发配置成功：WNC6000-1000-AC(v2)#show wireless ap status



AP已成功注册到AC且配置已成功下发。



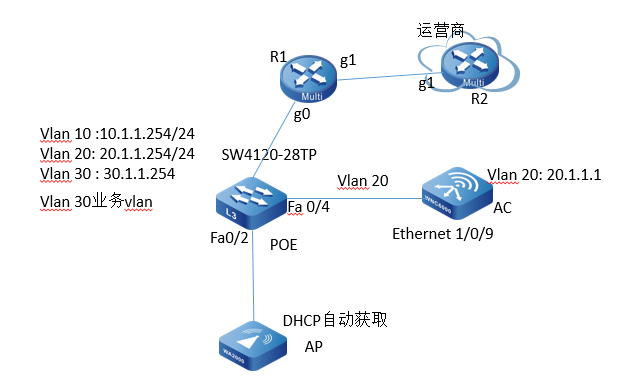


PC机已成功获取到IP地址。

## WPA2方式加密认证

实验要求：PC连接上WiFi使用WPA2加密方式，连上后能通过R1 NAT上公网。

实验拓扑如下：



分析：拓扑地址规划基本和AP三层注册一样，增加的部分为无线终端可以上公网，且终端连接网络时需要进行简单密码验证。

**SW4120配置：**

SW2(config)#vlan 20,30

SW2(config)#interface vlan1

SW2(config-if-vlan1)# ip address 10.1.1.254 255.255.255.0

SW2(config-if-vlan1)# exit

SW2(config)#interface vlan20

SW2(config-if-vlan20)# ip address 20.1.1.254 255.255.255.0

SW2(config-if-vlan20)# exit

SW2(config)#interface vlan30

SW2(config-if-vlan30)# ip address 30.1.1.254 255.255.255.0

SW2(config-if-vlan30)# exit

SW2(config)#ip dhcp pool vlan10

SW2(dhcp-config)# range 10.1.1.1 10.1.1.253 255.255.255.0

SW2(dhcp-config)# default-router 10.1.1.254

SW2(dhcp-config)# option 43 hex 80 07 00 00 01 14 01 01 01

//设置option43方式AP自动注册到AC

SW2(dhcp-config)# exit

SW2(config)#ip dhcp excluded-address 30.1.1.253

//排除30.1.1.253地址段。

SW2(config)#ip dhcp pool vlan30

SW2(dhcp-config)# range 30.1.1.1 30.1.1.252 255.255.255.0

SW2(dhcp-config)# default-router 30.1.1.254

SW2(dhcp-config)# exit

SW2(config)#ip route 0.0.0.0 0.0.0.0 30.1.1.253

SW2(config)#interface fa0/1-0/23

SW2(config-if-range)#switchport mode trunk

SW2(config-if-range)#switchport trunk allowed vlan all

SW2(config)#interface fa0/24

SW2(config-if-range)#switchport mode trunk

SW2(config-if-range)# switchport trunk pvid vlan 30

SW2(config-if-range)#switchport trunk allowed vlan all

**AC基本配置：**

WNC6000-1000-AC(v2)(config)#vlan 20

WNC6000-1000-AC(v2)(config-vlan20)#exit

WNC6000-1000-AC(v2)(config)#interface Vlan20

WNC6000-1000-AC(v2)(config-if-vlan20)# ip address 20.1.1.1 255.255.255.0

WNC6000-1000-AC(v2)(config-if-vlan20)#exit

WNC6000-1000-AC(v2)(config)#ip route 0.0.0.0 0.0.0.0 20.1.1.254

WNC6000-1000-AC(v2)(config)#wireless

WNC6000-1000-AC(v2)(config-wireless)# no auto-ip-assign

WNC6000-1000-AC(v2)(config-wireless)# enable

WNC6000-1000-AC(v2)(config-wireless)# static-ip 20.1.1.1

WNC6000-1000-AC(v2)(config-wireless)#network 1

WNC6000-1000-AC(v2)(config-network)#ssid text

WNC6000-1000-AC(v2)(config-network)#vlan 30

WNC6000-1000-AC(v2)(config-network)#security mode wpa-personal

//设置安全模式为WPA2个人模式

WNC6000-1000-AC(v2)(config-network)#wpa versions wpa2

//设置wpa版本

WNC6000-1000-AC(v2)(config-network)#wpa key maipu123

//设置本地认证密钥为maipu123

WNC6000-1000-AC(v2)(config-network)#exit

WNC6000-1000-AC(v2)(config-wireless)#ap database 00-01-7a-e8-f2-40

WNC6000-1000-AC(v2)(config-wireless)#ap profile 1

WNC6000-1000-AC(v2)(config-ap-profile)#hwtype 7

WNC6000-1000-AC(v2)(config)#Interface Ethernet1/0/9

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)# switchport mode trunk

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)#switchport trunk allowed vlan all

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)#

**R1配置：**

R1(config)#interface g0

R1(config-if-gigabitethernet0)#ip address 30.1.1.253 24

R1(config-if-gigabitethernet0)#exit

R1(config)#interface g1

R1(config-if-gigabitethernet1)#ip address 12.1.1.1 24

R1(config-if-gigabitethernet1)#exit

R1(config)#ip access-list standard 1

R1(config-std-nacl)#permit any

R1(config-std-nacl)#exit

R1(config)#ip nat inside source list 1 interface gigabitethernet 1 overload

//NAT转换到外网

R1(config)#interface g0

R1(config-if-gigabitethernet0)#ip nat inside

R1(config-if-gigabitethernet0)#exit

R1(config)#interface g1

R1(config-if-gigabitethernet1)#ip nat outside

R1(config-if-gigabitethernet1)#exit

R1(config)#ip route 0.0.0.0 0.0.0.0 12.1.1.2

**R2配置：**

R2(config)#interface g1

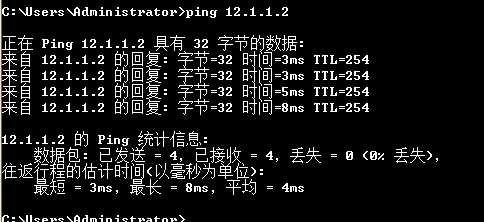
R2(config-if-gigabitethernet1)#ip address 12.1.1.2 24

R2(config-if-gigabitethernet1)#exit

配置完成后可以在手机或PC上搜索到text网络，如手机上连接该网络会看到要求输入密码的提示信息。如下图：（PC上一样）



当PC机连接上网络后，ping外网主机R2：

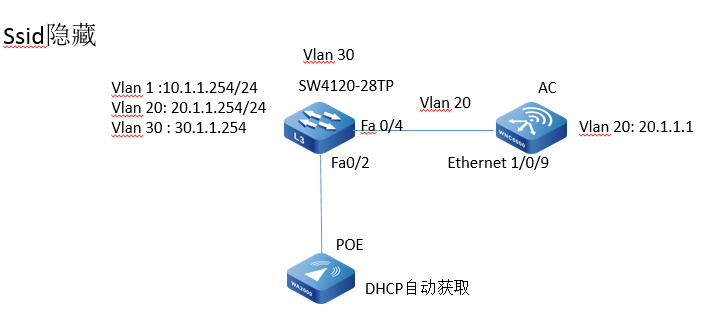


可以看到，此时PC机能访问外网了。

## 无线安全实验

### SSID隐藏

试验拓扑如图：



实验要求：在AC上通过配置命令实现下发给AP的SSID在无线网络环境下对一般用户不可见。

实验基本配置：

SW4120基本配置如下：

SW2(config)#vlan 20,30

SW2(config)#interface vlan1

SW2(config-if-vlan1)# ip address 10.1.1.254 255.255.255.0

SW2(config-if-vlan1)# exit

SW2(config)#interface vlan20

SW2(config-if-vlan20)# ip address 20.1.1.254 255.255.255.0

SW2(config-if-vlan20)# exit

SW2(config)#interface vlan30

SW2(config-if-vlan30)# ip address 30.1.1.254 255.255.255.0

SW2(config-if-vlan30)# exit

SW2(config)#ip dhcp pool vlan10

SW2(dhcp-config)# range 10.1.1.1 10.1.1.253 255.255.255.0

SW2(dhcp-config)# default-router 10.1.1.254

SW2(dhcp-config)# option 43 hex 80 07 00 00 01 14 01 01 01

SW2(dhcp-config)# exit

SW2(config)#ip dhcp pool vlan30

SW2(dhcp-config)# range 30.1.1.1 30.1.1.253 255.255.255.0

SW2(dhcp-config)# default-router 30.1.1.254

SW2(dhcp-config)# exit

SW2(config)#interface fa0/1-0/24

SW2(config-if-range)#switchport mode trunk

SW2(config-if-range)#switchport trunk allowed vlan all

**AC基本配置：**

WNC6000-1000-AC(v2)(config)#vlan 20

WNC6000-1000-AC(v2)(config-vlan20)#exit

WNC6000-1000-AC(v2)(config)#interface Vlan20

WNC6000-1000-AC(v2)(config-if-vlan20)# ip address 20.1.1.1 255.255.255.0

WNC6000-1000-AC(v2)(config-if-vlan20)#exit

WNC6000-1000-AC(v2)(config)#ip route 0.0.0.0/0 20.1.1.254

WNC6000-1000-AC(v2)(config)#wireless

WNC6000-1000-AC(v2)(config-wireless)# no auto-ip-assign

WNC6000-1000-AC(v2)(config-wireless)# enable

WNC6000-1000-AC(v2)(config-wireless)# static-ip 20.1.1.1

WNC6000-1000-AC(v2)(config-wireless)#network 2

WNC6000-1000-AC(v2)(config-network)#***hide-ssid //隐藏SSID命令***

WNC6000-1000-AC(v2)(config-network)#ssid yctext

WNC6000-1000-AC(v2)(config-network)#vlan 30

WNC6000-1000-AC(v2)(config-network)#exit

WNC6000-1000-AC(v2)(config-wireless)#ap database 00-01-7a-e8-f2-40

WNC6000-1000-AC(v2)(config-wireless)#ap profile 1

WNC6000-1000-AC(v2)(config-ap-profile)#hwtype 7

WNC6000-1000-AC(v2)(config-ap-profile)#radio 1

WNC6000-1000-AC(v2)(config-ap-profile-vap)#**vap 1**

WNC6000-1000-AC(v2)(config-ap-profile-vap)#**enable**

WNC6000-1000-AC(v2)(config-ap-profile-vap)#exit

WNC6000-1000-AC(v2)(config-ap-profile-radio)#exit

WNC6000-1000-AC(v2)(config-ap-profile)#exit

WNC6000-1000-AC(v2)(config-wireless)#exit

WNC6000-1000-AC(v2)(config)#Interface Ethernet1/0/9

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)# switchport mode trunk

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)#switchport trunk allowed vlan all

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)#

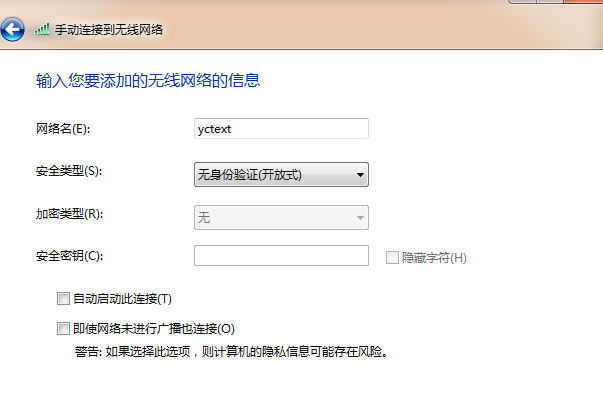
配置完后我们在PC上查看：



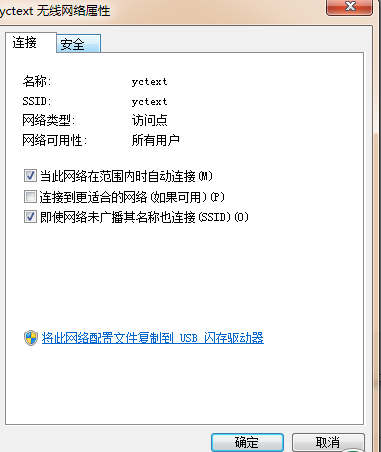
此时可以看到我们设置的ssid：yctext并不在网络列表中。在无线客户端中我们可以通过手动添加该SSID。

打开网络共享中心——管理无线网络——添加







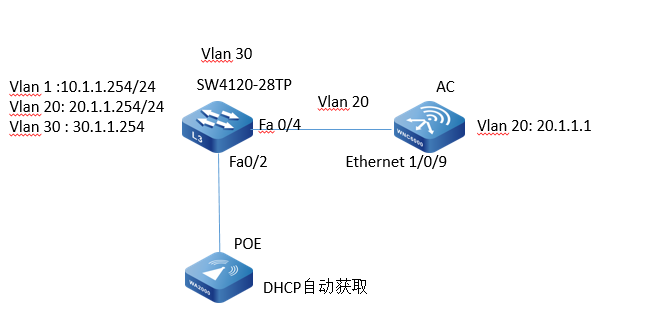




此时我们可以连接上该无线网络了。

### mac地址白名单/黑名单

实验拓扑和上述实验一相同，相关地址配置也一致。



SW4120基本配置如下：

SW2(config)#vlan 20,30

SW2(config)#interface vlan1

SW2(config-if-vlan1)# ip address 10.1.1.254 255.255.255.0

SW2(config-if-vlan1)# exit

SW2(config)#interface vlan20

SW2(config-if-vlan20)# ip address 20.1.1.254 255.255.255.0

SW2(config-if-vlan20)# exit

SW2(config)#interface vlan30

SW2(config-if-vlan30)# ip address 30.1.1.254 255.255.255.0

SW2(config-if-vlan30)# exit

SW2(config)#ip dhcp pool vlan10

SW2(dhcp-config)# range 10.1.1.1 10.1.1.253 255.255.255.0

SW2(dhcp-config)# default-router 10.1.1.254

SW2(dhcp-config)# option 43 hex 80 07 00 00 01 14 01 01 01

SW2(dhcp-config)# exit

SW2(config)#ip dhcp pool vlan30

SW2(dhcp-config)# range 30.1.1.1 30.1.1.253 255.255.255.0

SW2(dhcp-config)# default-router 30.1.1.254

SW2(dhcp-config)# exit

SW2(config)#interface fa0/1-0/24

SW2(config-if-range)#switchport mode trunk

SW2(config-if-range)#switchport trunk allowed vlan all

**AC基本配置：**

WNC6000-1000-AC(v2)(config)#vlan 20

WNC6000-1000-AC(v2)(config-vlan20)#exit

WNC6000-1000-AC(v2)(config)#interface Vlan20

WNC6000-1000-AC(v2)(config-if-vlan20)# ip address 20.1.1.1 255.255.255.0

WNC6000-1000-AC(v2)(config-if-vlan20)#exit

WNC6000-1000-AC(v2)(config)#ip route 0.0.0.0/0 20.1.1.254

WNC6000-1000-AC(v2)(config)#wireless

WNC6000-1000-AC(v2)(config-wireless)# no auto-ip-assign

WNC6000-1000-AC(v2)(config-wireless)# enable

WNC6000-1000-AC(v2)(config-wireless)# static-ip 20.1.1.1

WNC6000-1000-AC(v2)(config-wireless)#network 2

WNC6000-1000-AC(v2)(config-network)#ssid yctext

WNC6000-1000-AC(v2)(config-network)#vlan 30

WNC6000-1000-AC(v2)(config-network)#exit

WNC6000-1000-AC(v2)(config-wireless)#ap database 00-01-7a-e8-f2-40

WNC6000-1000-AC(v2)(config-wireless)#ap profile 1

WNC6000-1000-AC(v2)(config-ap-profile)#hwtype 7

WNC6000-1000-AC(v2)(config-ap-profile)#radio 1

WNC6000-1000-AC(v2)(config-ap-profile-vap)#**vap 1**

WNC6000-1000-AC(v2)(config-ap-profile-vap)#**enable**

WNC6000-1000-AC(v2)(config-ap-profile-vap)#exit

WNC6000-1000-AC(v2)(config-ap-profile-radio)#exit

WNC6000-1000-AC(v2)(config-ap-profile)#exit

WNC6000-1000-AC(v2)(config-wireless)#exit

WNC6000-1000-AC(v2)(config)#Interface Ethernet1/0/9

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)# switchport mode trunk

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)#switchport trunk allowed vlan all

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)#

白名单：单非黑即白，mac-authentication-mode white-list只是配置列表为白名单，没在列表里的都是黑名单，反之亦然。mac authentication local是开启黑白名单过滤功能。例如，配置默认下列表为白名单时，所有未加入列表的客户端都为黑名单，一旦开启mac authentication local，即所有未加入列表客户端都无法接入AP。

WNC6000-1000-AC(v2)(config-wireless)#mac-authentication-mode white-list

//配置列表为白名单

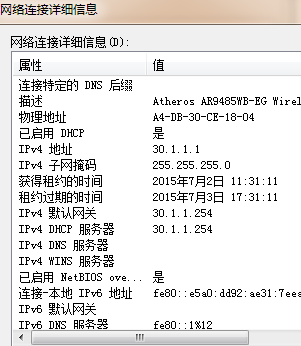
WNC6000-1000-AC(v2)(config-wireless)#known-client A4-DB-30-CE-18-04 action global-action //匹配黑白名单列表

WNC6000-1000-AC(v2)(config-wireless)#network 2 //关联network

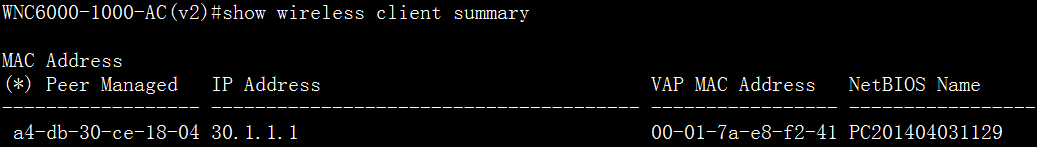
WNC6000-1000-AC(v2)(config-network)#mac authentication local

//开启黑白名单功能 根据mac地址黑白名单过滤功能

配置完白名单后本台PC机（mac A4-DB-30-CE-18-04）可以连接上yctext。但移动终端没有添加到白名单，所以无法连接到该网络。



加入到白名单的主机，可以认证成功



未加入白名单的主机，不能认证通过。

黑名单：单非黑即白，mac-authentication-mode black-list只是配置列表为黑名单，没在列表里的都是白名单，反之亦然。mac authentication local是开启黑白名单过滤功能。例如，配置默认下列表为白名单时，所有未加入列表的客户端都为黑名单，一旦开启mac authentication local，即所有未加入列表客户端都无法接入AP

WNC6000-1000-AC(v2)(config-wireless)#no mac-authentication-mode

//no掉上面的白名单。

WNC6000-1000-AC(v2)(config-wireless)#mac-authentication-mode black-list

//配置列表为黑名单

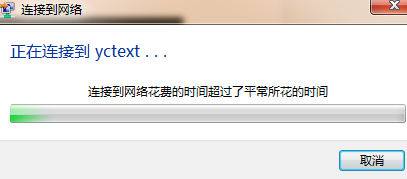
WNC6000-1000-AC(v2)(config-wireless)#known-client A4-DB-30-CE-18-04 action global-action //匹配黑白名单列表

WNC6000-1000-AC(v2)(config-wireless)#network 2 //关联network

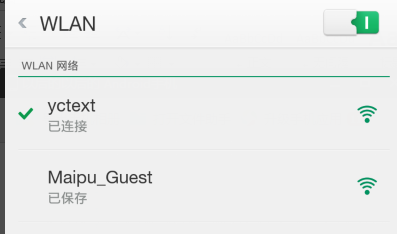
WNC6000-1000-AC(v2)(config-network)#mac authentication local

//开启黑白名单功能 根据mac地址黑白名单过滤功能

此时可以看到，本台PC机无法连接上该无线网络了。



其他终端能够连接上yctext网络。

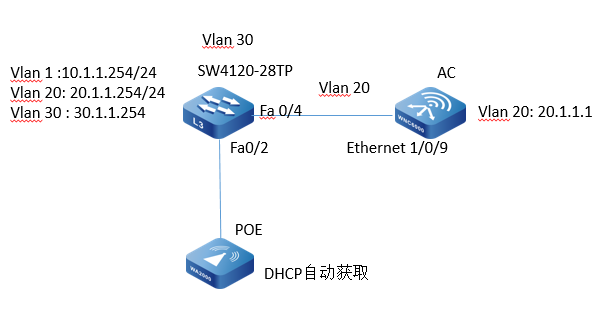


### 信道自动调整

当无线设备实际部署后，会收到周边新加设备或者变更信道后的AP的干扰，当同一信道干扰严重时会出现通信质量下降的情况，所以需要进行AP信道的自动调整。

技术实现：当AP检测到周围有相同设备与我司设备处于同一信道，并通过综合条件评价，例如信道利用率等等，当评价结果低于门限值时则进行信道的自动调整。

实验拓扑和实验一相同。相关基础配置不变



SW4120基本配置如下：

SW2(config)#vlan 20,30

SW2(config)#interface vlan1

SW2(config-if-vlan1)# ip address 10.1.1.254 255.255.255.0

SW2(config-if-vlan1)# exit

SW2(config)#interface vlan20

SW2(config-if-vlan20)# ip address 20.1.1.254 255.255.255.0

SW2(config-if-vlan20)# exit

SW2(config)#interface vlan30

SW2(config-if-vlan30)# ip address 30.1.1.254 255.255.255.0

SW2(config-if-vlan30)# exit

SW2(config)#ip dhcp pool vlan10

SW2(dhcp-config)# range 10.1.1.1 10.1.1.253 255.255.255.0

SW2(dhcp-config)# default-router 10.1.1.254

SW2(dhcp-config)# option 43 hex 80 07 00 00 01 14 01 01 01

SW2(dhcp-config)# exit

SW2(config)#ip dhcp pool vlan30

SW2(dhcp-config)# range 30.1.1.1 30.1.1.253 255.255.255.0

SW2(dhcp-config)# default-router 30.1.1.254

SW2(dhcp-config)# exit

SW2(config)#interface fa0/1-0/24

SW2(config-if-range)#switchport mode trunk

SW2(config-if-range)#switchport trunk allowed vlan all

**AC基本配置：**

WNC6000-1000-AC(v2)(config)#vlan 20

WNC6000-1000-AC(v2)(config-vlan20)#exit

WNC6000-1000-AC(v2)(config)#interface Vlan20

WNC6000-1000-AC(v2)(config-if-vlan20)# ip address 20.1.1.1 255.255.255.0

WNC6000-1000-AC(v2)(config-if-vlan20)#exit

WNC6000-1000-AC(v2)(config)#ip route 0.0.0.0/0 20.1.1.254

WNC6000-1000-AC(v2)(config)#wireless

WNC6000-1000-AC(v2)(config-wireless)# no auto-ip-assign

WNC6000-1000-AC(v2)(config-wireless)# enable

WNC6000-1000-AC(v2)(config-wireless)# static-ip 20.1.1.1

WNC6000-1000-AC(v2)(config-wireless)#network 2

WNC6000-1000-AC(v2)(config-network)#ssid yctext

WNC6000-1000-AC(v2)(config-network)#vlan 30

WNC6000-1000-AC(v2)(config-network)#exit

WNC6000-1000-AC(v2)(config-wireless)#ap database 00-01-7a-e8-f2-40

WNC6000-1000-AC(v2)(config-wireless)#ap profile 1

WNC6000-1000-AC(v2)(config-ap-profile)#hwtype 7

WNC6000-1000-AC(v2)(config-ap-profile)#radio 1

WNC6000-1000-AC(v2)(config-ap-profile-vap)#**vap 1**

WNC6000-1000-AC(v2)(config-ap-profile-vap)#**enable**

WNC6000-1000-AC(v2)(config-ap-profile-vap)#exit

WNC6000-1000-AC(v2)(config-ap-profile-radio)#exit

WNC6000-1000-AC(v2)(config-ap-profile)#exit

WNC6000-1000-AC(v2)(config-wireless)#exit

WNC6000-1000-AC(v2)(config)#Interface Ethernet1/0/9

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)# switchport mode trunk

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)#switchport trunk allowed vlan all

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)#

信道自动调整配置如下：

WNC6000-1000-AC(v2)(config-wireless)#channel-plan an history-depth 1

WNC6000-1000-AC(v2)(config-wireless)#channel-plan an interval 3

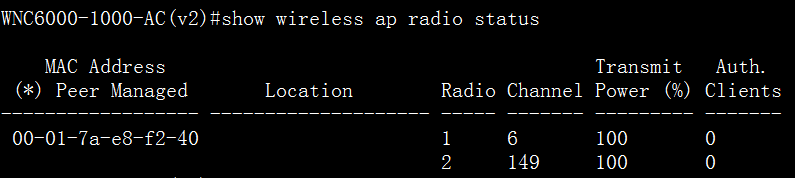
WNC6000-1000-AC(v2)(config-wireless)#channel-plan an mode interval

WNC6000-1000-AC(v2)(config-wireless)#channel-plan bgn history-depth 1

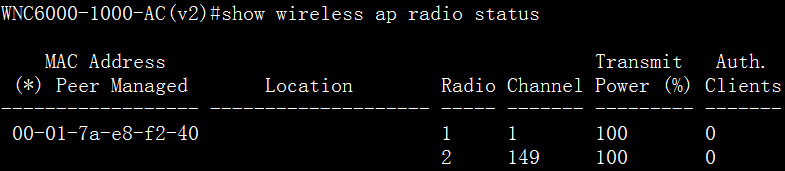
WNC6000-1000-AC(v2)(config-wireless)#channel-plan bgn interval 3

WNC6000-1000-AC(v2)(config-wireless)#channel-plan bgn mode interval

检验配置效果：调整前AP处于6信道：



AC经过信道调整方法测得当前1信道干扰更小，于是通知AP调整为1信道：



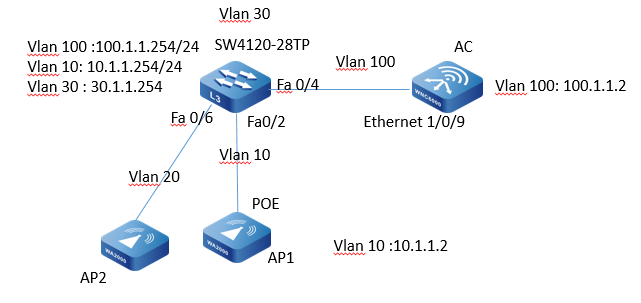
### 不同AP二层漫游

在客户部署了无线后，会有一些情况拿着电脑去另外的地方处理事情，此时两个地方获得的地址为同一个网段，所以就产生了二层漫游，产生漫游客户的网络不会中断。

当用户位置移动时，到达两个AP同时覆盖的范围时，PC会对两个信号强度做一个比较，同时PC内部会存在一个伐值，当信号强度低于阀值时，PC会自动关联至信号更强的AP上，AC就会快速更新表象来完成二层漫游。

漫游的动作完全是PC发起的与AP和AC无关。

实验拓扑：



**配置方法：**

两个AP由AC统一管理，并下发相同的配置。两个AP放置位置相隔一定的距离。然后通过在PC上不停的ping包并在两个AP间来回移动。看其是否会漫游丢包。

SW4210配置：

SW4120基本配置如下：

SW2(config)#vlan 20,30

SW2(config)#interface vlan1

SW2(config-if-vlan1)# ip address 10.1.1.254 255.255.255.0

SW2(config-if-vlan1)# exit

SW2(config)#interface vlan20

SW2(config-if-vlan20)# ip address 20.1.1.254 255.255.255.0

SW2(config-if-vlan20)# exit

SW2(config)#interface vlan30

SW2(config-if-vlan30)# ip address 30.1.1.254 255.255.255.0

SW2(config-if-vlan30)# exit

SW2(config)#ip dhcp pool vlan10

SW2(dhcp-config)# range 10.1.1.1 10.1.1.253 255.255.255.0

SW2(dhcp-config)# default-router 10.1.1.254

SW2(dhcp-config)# option 43 hex 80 07 00 00 01 14 01 01 01

SW2(dhcp-config)# exit

SW2(config)#ip dhcp pool vlan30

SW2(dhcp-config)# range 30.1.1.1 30.1.1.253 255.255.255.0

SW2(dhcp-config)# default-router 30.1.1.254

SW2(dhcp-config)# exit

SW2(config)#interface fa0/1-0/24

SW2(config-if-range)#switchport mode trunk

SW2(config-if-range)#switchport trunk allowed vlan all

AC基本配置：

WNC6000-1000-AC(v2)(config)#vlan 20

WNC6000-1000-AC(v2)(config-vlan20)#exit

WNC6000-1000-AC(v2)(config)#interface Vlan20

WNC6000-1000-AC(v2)(config-if-vlan20)# ip address 20.1.1.1 255.255.255.0

WNC6000-1000-AC(v2)(config-if-vlan20)#exit

WNC6000-1000-AC(v2)(config)#ip route 0.0.0.0/0 20.1.1.254

WNC6000-1000-AC(v2)(config)#wireless

WNC6000-1000-AC(v2)(config-wireless)# no auto-ip-assign

WNC6000-1000-AC(v2)(config-wireless)# enable

WNC6000-1000-AC(v2)(config-wireless)# static-ip 20.1.1.1

WNC6000-1000-AC(v2)(config-wireless)#network 2

WNC6000-1000-AC(v2)(config-network)#ssid yctext

WNC6000-1000-AC(v2)(config-network)#vlan 30

WNC6000-1000-AC(v2)(config-network)#exit

WNC6000-1000-AC(v2)(config-wireless)#ap database 00-01-7a-e8-f2-40

WNC6000-1000-AC(v2)(config-wireless)#ap database 00-01-7A-E8-B2-20

WNC6000-1000-AC(v2)(config-wireless)#ap profile 1

WNC6000-1000-AC(v2)(config-ap-profile)#hwtype 7

WNC6000-1000-AC(v2)(config-ap-profile)#hwtype 7

WNC6000-1000-AC(v2)(config-ap-profile)#radio 1

WNC6000-1000-AC(v2)(config-ap-profile-vap)#**vap 1**

WNC6000-1000-AC(v2)(config-ap-profile-vap)#**enable**

WNC6000-1000-AC(v2)(config-ap-profile-vap)#exit

WNC6000-1000-AC(v2)(config-ap-profile-radio)#exit

WNC6000-1000-AC(v2)(config-ap-profile)#exit

WNC6000-1000-AC(v2)(config-wireless)#exit

WNC6000-1000-AC(v2)(config)#Interface Ethernet1/0/9

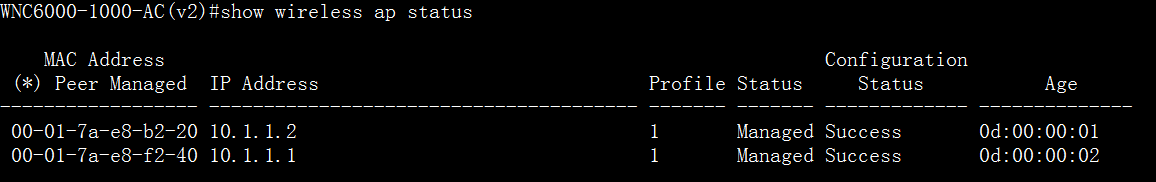
WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)# switchport mode trunk

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)#switchport trunk allowed vlan all

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)#end

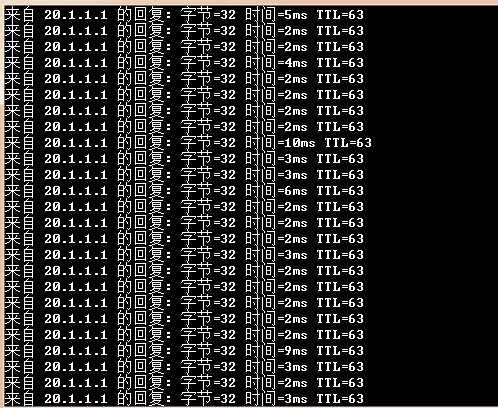
WNC6000-1000-AC(v2)#wireless ap profile apply 1

在AC上将配置信息下发到两个AP上，查看其是否均注册成功。



两个AP均已注册成功。

在PC上ping 包：



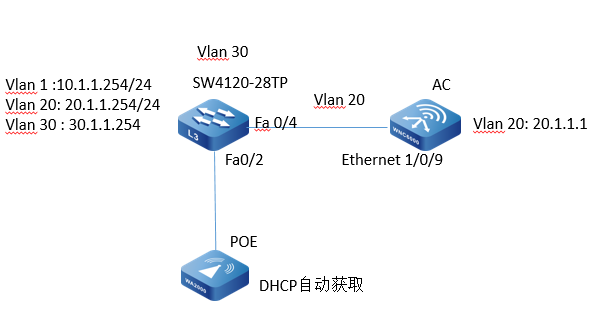
可以看到几乎漫游无丢包。

### 无线用户带宽控制：

带宽控制可以对无线客户端进行限速。

迈普设备目前支持以下几种带宽控制方式：基于单个用户、基于SSID、基于单个用户使用ACL以及基于AC出端口。

实验拓扑：



基本配置同实验一

SW4120基本配置如下：

SW2(config)#vlan 20,30

SW2(config)#interface vlan1

SW2(config-if-vlan1)# ip address 10.1.1.254 255.255.255.0

SW2(config-if-vlan1)# exit

SW2(config)#interface vlan20

SW2(config-if-vlan20)# ip address 20.1.1.254 255.255.255.0

SW2(config-if-vlan20)# exit

SW2(config)#interface vlan30

SW2(config-if-vlan30)# ip address 30.1.1.254 255.255.255.0

SW2(config-if-vlan30)# exit

SW2(config)#ip dhcp pool vlan10

SW2(dhcp-config)# range 10.1.1.1 10.1.1.253 255.255.255.0

SW2(dhcp-config)# default-router 10.1.1.254

SW2(dhcp-config)# option 43 hex 80 07 00 00 01 14 01 01 01

SW2(dhcp-config)# exit

SW2(config)#ip dhcp pool vlan30

SW2(dhcp-config)# range 30.1.1.1 30.1.1.253 255.255.255.0

SW2(dhcp-config)# default-router 30.1.1.254

SW2(dhcp-config)# exit

SW2(config)#interface fa0/1-0/24

SW2(config-if-range)#switchport mode trunk

SW2(config-if-range)#switchport trunk allowed vlan all

**AC基本配置：**

WNC6000-1000-AC(v2)(config)#vlan 20

WNC6000-1000-AC(v2)(config-vlan20)#exit

WNC6000-1000-AC(v2)(config)#interface Vlan20

WNC6000-1000-AC(v2)(config-if-vlan20)# ip address 20.1.1.1 255.255.255.0

WNC6000-1000-AC(v2)(config-if-vlan20)#exit

WNC6000-1000-AC(v2)(config)#ip route 0.0.0.0/0 20.1.1.254

WNC6000-1000-AC(v2)(config)#wireless

WNC6000-1000-AC(v2)(config-wireless)# no auto-ip-assign

WNC6000-1000-AC(v2)(config-wireless)# enable

WNC6000-1000-AC(v2)(config-wireless)# static-ip 20.1.1.1

WNC6000-1000-AC(v2)(config-wireless)#network 2

WNC6000-1000-AC(v2)(config-network)#ssid yctext

WNC6000-1000-AC(v2)(config-network)#vlan 30

WNC6000-1000-AC(v2)(config-network)#exit

WNC6000-1000-AC(v2)(config-wireless)#ap database 00-01-7a-e8-f2-40

WNC6000-1000-AC(v2)(config-wireless)#ap profile 1

WNC6000-1000-AC(v2)(config-ap-profile)#hwtype 7

WNC6000-1000-AC(v2)(config-ap-profile)#radio 1

WNC6000-1000-AC(v2)(config-ap-profile-vap)#**vap 1**

WNC6000-1000-AC(v2)(config-ap-profile-vap)#**enable**

WNC6000-1000-AC(v2)(config-ap-profile-vap)#exit

WNC6000-1000-AC(v2)(config-ap-profile-radio)#exit

WNC6000-1000-AC(v2)(config-ap-profile)#exit

WNC6000-1000-AC(v2)(config-wireless)#exit

WNC6000-1000-AC(v2)(config)#Interface Ethernet1/0/9

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)# switchport mode trunk

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)#switchport trunk allowed vlan all

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)#

**下面只举例基于单个用户的QOS。**

WNC6000-1000-AC(v2)(config)#wireless

WNC6000-1000-AC(v2)(config-wireless)#ap client-qos  **//全局开启QoS**

WNC6000-1000-AC(v2)(config-wireless)#network 2

WNC6000-1000-AC(v2)(config-network)#client-qos enable **//network下开启QoS**

WNC6000-1000-AC(v2)(config-network)#**client-qos bandwidth-limit up 512**

**//上传速率限制为512KB**

WNC6000-1000-AC(v2)(config-network)#**client-qos bandwidth-limit down 2048**

**//下载速率限制为2MB**

WNC6000-1000-AC(v2)(config-network)#end

WNC6000-1000-AC(v2)#wireless ap profile apply 1  **//配置下发到AP**

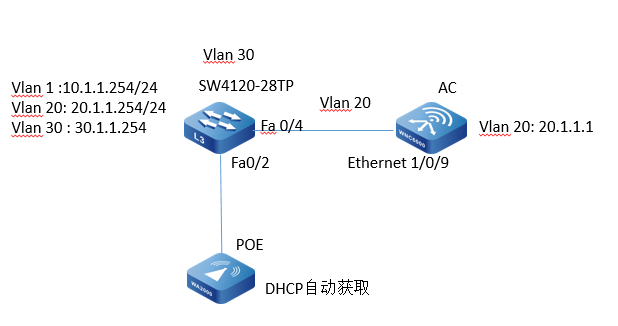
**验证：**将两台PC连接上同一个AP的网络，其中一台设置为服务器端并提供一个下载文件，利用测速软件在另一台PC上可以查看到效果。

### 基于时间段控制AP在线时间

基于时间段控制AP的在线时间（SSID和RADIO） ：AC控制AP在有效时间内广播SSID，用户在有效的时间内才能搜索到SSID，其他时间内搜索不到。

这里只举例基于SSID控制AP在线时间。

实验拓扑及其基本配置同实验一



实验要求：基于时间段控制AP的在线时间（SSID和RADIO） ：AC控制AP在有效时间内广播SSID，用户在有效的时间内才能搜索到SSID，其他时间内搜索不到。

SW4120基本配置如下：

SW2(config)#vlan 20,30

SW2(config)#interface vlan1

SW2(config-if-vlan1)# ip address 10.1.1.254 255.255.255.0

SW2(config-if-vlan1)# exit

SW2(config)#interface vlan20

SW2(config-if-vlan20)# ip address 20.1.1.254 255.255.255.0

SW2(config-if-vlan20)# exit

SW2(config)#interface vlan30

SW2(config-if-vlan30)# ip address 30.1.1.254 255.255.255.0

SW2(config-if-vlan30)# exit

SW2(config)#ip dhcp pool vlan10

SW2(dhcp-config)# range 10.1.1.1 10.1.1.253 255.255.255.0

SW2(dhcp-config)# default-router 10.1.1.254

SW2(dhcp-config)# option 43 hex 80 07 00 00 01 14 01 01 01

SW2(dhcp-config)# exit

SW2(config)#ip dhcp pool vlan30

SW2(dhcp-config)# range 30.1.1.1 30.1.1.253 255.255.255.0

SW2(dhcp-config)# default-router 30.1.1.254

SW2(dhcp-config)# exit

SW2(config)#interface fa0/1-0/24

SW2(config-if-range)#switchport mode trunk

SW2(config-if-range)#switchport trunk allowed vlan all

**AC基本配置：**

WNC6000-1000-AC(v2)(config)#vlan 20

WNC6000-1000-AC(v2)(config-vlan20)#exit

WNC6000-1000-AC(v2)(config)#interface Vlan20

WNC6000-1000-AC(v2)(config-if-vlan20)# ip address 20.1.1.1 255.255.255.0

WNC6000-1000-AC(v2)(config-if-vlan20)#exit

WNC6000-1000-AC(v2)(config)#ip route 0.0.0.0/0 20.1.1.254

WNC6000-1000-AC(v2)(config)#wireless

WNC6000-1000-AC(v2)(config-wireless)# no auto-ip-assign

WNC6000-1000-AC(v2)(config-wireless)# enable

WNC6000-1000-AC(v2)(config-wireless)# static-ip 20.1.1.1

WNC6000-1000-AC(v2)(config-wireless)#network 2

WNC6000-1000-AC(v2)(config-network)#ssid yctext

WNC6000-1000-AC(v2)(config-network)#vlan 30

WNC6000-1000-AC(v2)(config-network)#exit

WNC6000-1000-AC(v2)(config-wireless)#ap database 00-01-7a-e8-f2-40

WNC6000-1000-AC(v2)(config-wireless)#ap profile 1

WNC6000-1000-AC(v2)(config-ap-profile)#hwtype 7

WNC6000-1000-AC(v2)(config-ap-profile)#radio 1

WNC6000-1000-AC(v2)(config-ap-profile-vap)#**vap 1**

WNC6000-1000-AC(v2)(config-ap-profile-vap)#**enable**

WNC6000-1000-AC(v2)(config-ap-profile-vap)#exit

WNC6000-1000-AC(v2)(config-ap-profile-radio)#exit

WNC6000-1000-AC(v2)(config-ap-profile)#exit

WNC6000-1000-AC(v2)(config-wireless)#exit

WNC6000-1000-AC(v2)(config)#Interface Ethernet1/0/9

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)# switchport mode trunk

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)#switchport trunk allowed vlan all

WNC6000-1000-AC(v2)(config-if-ethernet1/0/9)#

**关键配置如下：**

WNC6000-1000-AC(v2)(config-wireless)#network 2

WNC6000-1000-AC(v2)(config-wireless)#no hide-ssid //打开实验一中隐藏的SSID

WNC6000-1000-AC(v2)(config-network)#time-limit from 15:50 to 15:51 weekday all

//(每天的15：50分到15：51之间AP不会广播network 2下的SSID)

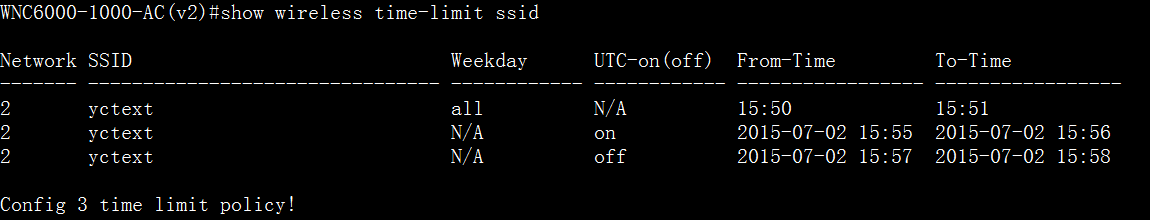
WNC6000-1000-AC(v2)(config-network)#time-limit-UTC from 2015-07-2 15:55 to 2015-07-2 15:56 on

//从2015年7月2日15:55开始到2015年7月2日15：56是广播network 2下的SSID

WNC6000-1000-AC(v2)(config-network)#time-limit-UTC from 2015-07-2 15:57 to 2015-07-2 15:58 off

//从2015年7月2日15:57开始到2015年7月2日15：58是不广播network 2下的SSID

WNC6000-1000-AC(v2)(config-network)#



制定时间段内不能搜索到yctext这个网络。

# 第六章 语音实验部分

## 语音网关的基本配置操作

**实验目的：**

1、熟悉IP语音网关设备硬件结构；

2、学习了解通过telnet方式配置语音网关的方法；

3、熟悉语音网关shell系统的基本操作方法；

**实验拓扑：**



**实验环境说明：**

本实验中，使用一台PC机和一台IP语音网关，使用直通双绞线连接语音网关的LAN0口和PC机的网卡接口。

**实验要求：**

1、将路由器设备从包装盒中取出，按规范进行硬件安装；

2、使用直通双绞线连接语音网关的LAN0口和PC机的网卡接口；

3、启动语音网关，从电脑上通过telnet程序进入语音网关shell系统，设置语音网关的设备名、认证用户登陆密码，并检验配置结果；

**实验步骤：**

**1、IP语音网关设备的硬件安装；**

* 设备安装
* 使用带有RJ11接头的线缆，将电话终端接入FXS端口，以及将电话线接入FXO端口。
* 使用10/100M Base-Tx的以太网网线，将本地网络设备接入LAN0～LAN3。
* 使用直通网线，通过WAN口接入外网或交换机。
* 将交流电源线的插入VG A600的电源输入端
* 将交流电源线插入电源插座，VG A600开始上电运行。
* 安全注意事项
* 妥善放置VG A600网关，以免因跌落造成严重损失。
* 妥善布线，请勿让任何重物压在电源线上，并避免踩踏接线。
* 推荐用户使用UPS 不间断电源，一方面可以避免由于市电故障对网络系统的影响，另一方面也可以避免电源干扰。
* 推荐用户使用时保证可靠接地（零地电压小于5V），避免因接地不好带来设备烧毁。
* VG A600应该保持干燥通风，避免水淋，保持清洁，远离热源。
* 不用VG A600时，请拔掉电源线。
* VG A600上不要放置重物，避免阳光直射。

**提示**：如果长时间使用VG A600，设备外壳会有一定程度的发热，请不必担心属于正常现象，可以继续使用。

**2、通过telnet方式登陆语音网关shell系统；**

VG A600启动后（SYS灯开始闪烁），可以通过telnet登陆VG A600的LAN口。具体步骤如下；

* 将PC机网卡接口通过局域网与IP语音网关的以太网口连接
* 在LAN上的一台PC机上运行Telnet客户端应用程序
* 设置Telnet终端首选项

其设置内容是：终端->首选选项->模拟选项，设为VT100/ANSI。

🕭注意：

在配置Telnet客户端程序时，一定要将“本地响应(回显)”选项取消，否则将会导致用户输入的内容重复显示，影响Shell子系统命令编辑功能的正常使用。

* 键入IP语音网关的IP地址，与IP语音网关建立Telnet连接；VG A600的默认IP地址是192.168.0.1，子网掩码255.255.255.0，用户名是admin，密码是admin。
* 主机名置为IP语音网关的IP地址(如192.168.0.1；
* 端口置为Telnet(23)；
* 终端类型置为ANSI类型；



其余的操作与通过Console接口进行配置相同。

**3、IP语音网关设备基本系统参数设置：**

第一步，登陆进入语音网关操作系统，通过telnet连接语音网关地址，操作如下：

Connect to 192.168.0.1 ...done

User Access Verification

login:**admin** /要求输入用户名，默认为admin

password: /要求输入用户密码，默认为admin

gateway>**enable** /进入特权用户模式

password: /要求输入enable密码，默认为admin

gateway#

第二步，进行系统参数修改，操作如下：

gateway# **configure terminal**  /进入全局配置模式

gateway(config)#**hostname VG1** /修改设备名为VG1

VG1(config)#**enable password 0 test** /修改enable密码为test

VG1(config)#

第三步，检查系统运行参数，操作如下：

VG1#**show running-config** /检查当前系统运行的参数

信息将显示如下：

Building Configuratio

! Current configuration : 1712 bytes

!

! Last configuration change at UTC THU JAN 01 00:07:55 1970 by admin

! Flash config last updated at UTC THU JAN 01 00:07:43 1970 by admin

!

!software version 8.0.71 /设备软件系统信息

!software image file VPM600A-VG-8.0.71.bin

!compiled on Feb 11 2009, 09:02:47

hostname VG1 /设备名称

no service password-encrypt

no service new-encrypt

service login-secure

enable password QWRVPWQW encrypt /enable密码（密文）

user admin privilege 15 password 0 admin /系统预设的用户名和密码

user guest password 0 guest

aaa new-model /系统要求对登陆用户进行认证

aaa authentication login default local

vlan 1 /VGA600的交换以太口默认属于VLAN1

description default

port 0-3 untagged

exit

mode router /目前工作在路由模式

interface fastethernet0 /设备的WAN口，用于外联网络

exit

interface switchethernet0 /VLAN1所属的三层接口，用于LAN口所接用户远程接入本设备

ip address 192.168.0.1 255.255.255.0

vlan 1

exit

ip http server vg\_a\_1.1.75.rom /设备提供HTTP服务，支持WEB配置方式

callrouting-conf /默认的拨号路由配置

dial-peer 1 pots

destination-pattern 401

port 1/0

exit

dial-peer 2 pots

destination-pattern 402

port 1/1

exit

dial-peer 3 pots

destination-pattern 403

port 1/2

exit

dial-peer 4 pots

destination-pattern 404

port 1/3

exit

dial-peer 5 pots

destination-pattern xx.

port 2/0

exit

dial-peer 100 voip

destination-pattern xx.

backup pstn

session-target ras

exit

exit

fxs-card 1 /语音FXS接口信息

channel 0 3 callid enable

channel 0 3 enable

exit

fxo-card 2 /语音FXO接口信息

channel 0 0 callid enable

channel 0 0 enable

exit

dialplan terminator # /支持以＃作为拨号结束键

voicesrv-conf /语音服务功能配置

exit

h323 start slow

h323 send-dtmf h245-string

h323 call-diversion default

h323 h245Tunnel off

h323 bearer-cap 3100hz

h323 fill-send-complete enable

h323 grq\_interval 40

h323 call-thrust-ttl 10

call-transfer disable

call-transfer consultation

user-config enable

exit

interface loopback0 /内部环回接口

exit

stun-conf

exit

!end

第四步，保存当前配置：

VG1#**write** /保存当前配置参数到文件系统中

Are you sure to override /flash/startup (Yes|No)?**y** /yes或y表示确认保存

Building Configuration...done

重启IP语音网关，可以检验配置信息是否正确保存。

**相关命令信息详解：**

**1、****配置系统名称：**

MyPower VG网关在出厂时，其缺省的系统名称是gateway。在使用过程中，用户可以根据自己的需要，随时改变系统的名称，并且这种改动是立即生效的，即新的系统名称将会在下一次系统提示符的显示中出现。下面的命令即是将系统名称由“gateway”改为“VG6000”：

操作的步骤是：

|  |  |
| --- | --- |
| **命令** | **描述** |
| gateway#configure terminal | 进入全局配置模式 |
| gateway(config)#hostname *VG6000* | 修改系统名称 |
| VG6000(config)# | 在下一次系统提示符的显示中，使用新的系统名称 |

**2、配置系统用户：**

为了增强系统的安全性，MyPower VG网关允许配置一些系统用户，和AAA配置一起，可以限制只有配置了的用户才能通过终端、Telnet等访问MyPower VG网关，而其它用户则无法访问。

加用户

gateway#configure terminal

|  |  |
| --- | --- |
| **命令** | **描述** |
| gateway(config)#user *maipu* password 0 *maipu* | 为系统添加用户“maipu”，用户对应的口令为“maipu”，其中“0”表示口令为明文方式 |

看系统用户信息：

|  |  |
| --- | --- |
| **命令** | **描述** |
| gateway#show user | 在特权用户模式下执行上述命令后，可查看已注册用户 |

删除系统用户：

gateway#configure terminal

|  |  |
| --- | --- |
| **命令** | **描述** |
| gateway(config)#no user *maipu* | 删除系统用户“maipu” |

该命令执行后，将从本地用户数据库中删除用户“maipu”。

🕭 注意：

在MyPower VG网关中出现的口令和有关密码，可以通过全局配置命令no service password-encrypt和service password-encrypt决定是否加密。如果有service password-encrypt配置，则在配置用户名和口令时为：user maipu password 7 XPXXXOYTYO

**3、配置enable密码和超时时间：**

在全局配置模式下，通过命令enable password和enable timeout来设定。

|  |  |
| --- | --- |
| **命令** | **描述** |
| gateway(config)#enable password {*0* |level| *<string>*} | 配置超级用户口令 |
| gateway(config)#enable timeout *<0-35791392>* | 配置超时时间，enable timeout 0 表示永不超时 |

🕮 注：

超时时间的缺省值为300秒，即5分钟。如设置为0，则永远都不会超时。

**4、show命令：**

系统命令show可以查看的信息分为以下几类：

* 系统软、硬件资源信息
* 系统统计信息
* 系统配置信息
* 系统基本信息

表4-3 系统部分show命令举例

|  |  |
| --- | --- |
| **命令** | **功能** |
| Stack | 显示系统中各个任务堆栈的使用情况 |
| memory | 显示系统内存信息 |
| Mbuf | 显示系统缓冲区信息 |
| process | 显示系统任务/进程信息 |
| device | 显示系统物理、逻辑设备信息 |
| interface | 显示系统网络接口信息 |
| Hosts | 显示系统内部主机表信息 |
| Arp | 显示系统ARP表信息 |
| Ip | 显示IP层（包括TCP、UDP）的统计信息 |
| startup-config | 显示系统启动配置文件内容 |
| running-config | 显示系统运行配置内容 |
| About | 显示系统版权信息 |
| version | 显示系统硬件、软件版本信息 |

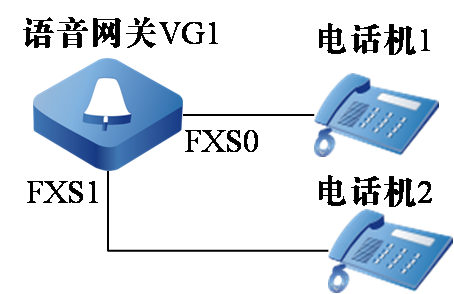
## 语音接口的基本参数操作

**实验目的：**

1、熟悉IP语音网关的各类语音接口；

2、掌握FXS/FXO的基本配置方法；

**实验拓扑：**



**实验环境说明：**

本实验中，使用IP语音网关，使用RJ11接头的电话线连接语音网关的FXS0、FXS1口和电话机1和电话机2。

本实验中的语音网关使用PC机通过telnet方式进行配置。

1、按照实验环境连接设备，在电话机上进行查询初始电话号码，进行电话互通测试；

2、进入语音网关shell界面，查看当前电话相关配置；

3、修改FXS0和FXS1的参数设置，测试修改效果；

**实验步骤：**

**1、本网关预设号码测试：**

MyPower VG 网关提供以下接口：

* Console，用于配置和调试。
* 10M/100M以太口。
* E1接口，只有VG2000、VG M6500、VG M6000支持。
* FXS接口，用于连接普通电话机。
* FXO接口，用于连接PSTN电话线。

在VGA600上有固化的4个FXS接口和1个FXO接口。本次实验我们就使用其中的FXS接口。

第一步，按照环境要求连接设备和线缆，并将VGA600语音网关上电，等到前面板的“SYS”指示灯开始闪烁时，语音网关进入工作状态。

第二步，在电话机上拨“＃33＃”进行本机查号，可以听到语音报号，电话机1的号码是“401”，电话机2的号码是“402”，这就是FXS0和FXS1的初始

第三步，从电话机1拨打“402”，应该可以拨通电话机2，并可以进行正常通话，并且可以在电话机2上看到来电显示为401。反之亦然。

**2、登陆语音网关，查看语音接口默认配置；**

第一步，登陆进入语音网关操作系统，通过telnet连接语音网关地址，操作如下：

Connect to 192.168.0.1 ...done

User Access Verification

login:**admin** /要求输入用户名，默认为admin

password: /要求输入用户密码，默认为admin

gateway>**enable** /进入特权用户模式

password: /要求输入enable密码，默认为admin

gateway#

第二步，检查语音接口运行参数，操作如下：

gateway #**show voice card** /检查语音接口情况

fxs-card 1 /fxs卡，编号1

channel 0 3 callid enable /0－3通道（即fxs0-fxs3）启用来电显示功能

channel 0 3 enable /0－3通道（即fxs0-fxs3）启用

exit

fxo-card 2 /fx0卡，编号2

channel 0 0 callid enable /0通道（即fxo0）启用来电显示功能

channel 0 0 enable /0通道（即fxo0）启用

exit

dialplan terminator # /拨号可以＃结束

从配置参数可以看到，系统已经默认启动了对应的语音接口。如果没有启动，我们可以用channel X Y enable命令启动。

第三步，检查默认的电话号码设置，操作如下：

gateway #**show voice route** /查看拨号路由信息

callrouting-conf /拨号路由配置

dial-peer 1 pots /POTS拨号端1号

destination-pattern 401 /电话号码401

port 1/0 /对应语音口port1/0（即FXS0）

exit

dial-peer 2 pots /POTS拨号端2号

destination-pattern 402

port 1/1

exit

dial-peer 3 pots /POTS拨号端3号

destination-pattern 403

port 1/2

exit

dial-peer 4 pots /POTS拨号端4号

destination-pattern 404

port 1/3

exit

dial-peer 5 pots /POTS拨号端5号

destination-pattern xx.

port 2/0

exit

dial-peer 100 voip /POTS拨号端5号

destination-pattern xx.

backup pstn

session-target ras

exit

exit

从配置参数可以看到，系统已经为FXS0－FXS3设置了号码：401到404。故两台电话可以互通。

**3、修改语音接口参数并验证效果：**

第一步，测试热线拨号功能

配置命令如下：

gateway #**configure terminal** /进入全局配置模式

gateway (config)#**fxs-card 1** /进入fxs 1号卡的配置模式

gateway (config-fxs-card)# **channel 0 0 hot-line number 402** /通道0热线拨号402

完成配置后，拿起电话1的听筒，静等5秒（默认），可以自动拨号402，与电话机2通话。

使用**channel 0 0 hot-line** **wait-time X**可以设置热线拨号等待时间。

第二步，测试来电显示开关命令

配置命令如下：

gateway (config)#**fxs-card 1** /进入fxs 1号卡的配置模式

gateway (config-fxs-card)#**channel 0 0 callid disable** /将通道0的来电显示功能关闭

完成配置后，从电话机2拨打电话机1的号码401，仍然可以拨通，但从电话机1上已经看不到来电显示了。

再使用**channel 0 0 callid enable**可以恢复。

第三步，测试语音接口开关命令

配置命令如下：

gateway (config)#**fxs-card 1** /进入fxs 1号卡的配置模式

gateway (config-fxs-card)#**channel 0 0 disable** /将通道0关闭

完成配置后，拿起电话1听筒，将不能听到任何声音，从电话机2也不能拨通401。

再使用channel 0 0 enable可以恢复。

**相关命令信息详解：**

**语音网关FXS接口配置方法**

MyPower VG系列网关均支持FXS的配置，但是进入FXS卡配置的模式有少许的差别。

MyPower VG M6500、VG M6000进入FXS卡配置命令为：

gateway(config)#card cardNum

|  |  |
| --- | --- |
| **命令** | **描述** |
| cardNum | cardNum为卡槽位，进入相应的卡配置模式 |

MyPower VG2000、VG800、VG A600设备进入FXS卡配置命令为：

gateway(config)#fxs-card c*ardNum*

|  |  |
| --- | --- |
| **命令** | **描述** |
| cardNum | cardNum为卡槽位，进入相应的卡配置模式 |

进入FXS卡配置模式后：

gateway(config-fxs-card)#

|  |  |
| --- | --- |
| **命令** | **描述** |
| channel *<0-31> <0-31>* fsk starttime *<1-50>* | 配置FSK的开始时间  第一个<0-31>为起始通道号  第二个<0-31>为结束通道号  时间值为1—50，单位为TICK（1/60S） |
| channel *<0-31> <0-31>* fsk sendtime *<60-150>* | 配置FSK的发送时间  第一个<0-31>为起始通道号  第二个<0-31>为结束通道号  时间值为60—150，单位为TICK（1/60S） |
| channel *<0-31> <0-31>* {enable | disable} | 配置通道的状态  第一个<0-31>为起始通道号  第二个<0-31>为结束通道号  enable：使能通道，disable：禁止通道 |
| channel *<0-31> <0-31>* dspvolume in *<-14-6>* | 配置dsp的输入音量，单位为db  MyPower VG M6500、VG M6000的范围是<-14-6>，默认为-5db；  MyPower VG2000、VG800、VG A600的范围是<-10-10>，默认为0db |
| channel *<0-31> <0-31>* dspvolume out *<-14-6>* | 配置dsp的输出音量，单位为db  MyPower VG M6500、VG M6000的范围是<-14-6>，默认为-5db；  MyPower VG2000、VG800、VG A600的范围是<-10-10>，默认为0db |
| channel *<0-31> <0-31>* reverse-polarity | 启用极性反转功能 |
| channel *<0-31> <0-31>* no reverse-polarity | 禁用极性反转功能 |
| channel <0-15> <0-15> payload <1-5> | 配置payload，默认为1  🕮 注：该命令只适用于MyPower VG2000、VG800、VG A600设备 |
| channel *<0-15> <0-15>* volume in *<-3-3>* | 配置codec的输入音量，单位为db，默认为0db  适用于MyPower VG2000、VG800、VG A600设备 |
| channel *<0-15> <0-15>* volume out *<-3-3>* | 配置codec的输出音量，单位为db，默认为0db  适用于MyPower VG2000、VG800、VG A600设备 |
| channel *<0-31> <0-31>* callid {enable | disable} | 配置主叫号码显示功能  第一个<0-31>为起始通道号  第二个<0-31>为结束通道号  enable：使能主叫号码显示，disable：禁止主叫号码显示 |
| channel *<0-31> <0-31>* echo-length *<0-128>* | 配置回波抵消长度，单位ms  第一个<0-31>为起始通道号  第二个<0-31>为结束通道号  时间值：0－128ms  默认缺省值：32ms  🕮 注：该命令只存在于MyPowr VG M6500、VG M6000版本中 |
| channel *<0-31> <0-31>* hot-line number *<string>* [<CR> | increase] | 配置热线拨号号码  第一个<0-31>为起始通道号  第二个<0-31>为结束通道号  <string> E.164号码  <CR> 配置相同的热线号码  increase 配置递增的热线号码 |
| channel *<0-31> <0-31>*  hot-line wait-time *<0-5>* | 配置热线拨号等待时间 单位：s  第一个<0-31>为起始通道号  第二个<0-31>为结束通道号  <0-5> 等待时间0－5s  默认缺省值：5s |
| channel <0-31> <0-31> jitterbuf <0-200> | 配置消除抖动时长 单位：ms  第一个<0-31>为起始通道号  第二个<0-31>为结束通道号  <0-200> 消抖时间0－200ms  默认缺省值： 0ms  🕮 注：该命令只存在于VG M6500、VG M6000版本中 |
| channel <0-15> <0-15> jitter-buffer <10-20> <2-8> | 配置消除抖动 单位：个数  第一个<0-15>为起始通道号  第二个<0-15>为结束通道号  <10-20> 最多的缓存语音包的个数(缺省为15)  <2-8> 可以放音的缓存语音包的个数(缺省为5)  🕮 注：该命令只适用于MyPower VG2000、VG800、VG A600设备 |
| channel *<0-31> <0-31>* dtmf-silent *<4-100>* | 配置拨号时间间隔  第一个<0-31>为起始通道号  第二个<0-31>为结束通道号  🕮 注：对于MyPower VG M6500、VG M6000，单位：TICK，长度：4-100(缺省为5)  🕮 注：对于MyPower VG2000、VG800、VG A600设备,单位：10ms 长度：4-100(缺省为8) |
| channel *<0-31> <0-31>* pulse-dial {enable | disable} | 配置脉冲拨号功能  第一个<0-31>为起始通道号  第二个<0-31>为结束通道号  enable:使能脉冲拨号；disable:禁止脉冲拨号  默认值：disable  🕮 注：该命令只适用于MyPower VG M6500、VG M6000设备 |
| channel *<0-31> <0-31>* scf on {nosid | pt13 | sid} | 配置静音压缩功能  第一个<0-31>为起始通道号  第二个<0-31>为结束通道号  on 使能静音压缩功能  nosid 不传送SID  pt13 传送PT13（只适用于g711编码）  sid 传送SID |
| channel *<0-31> <0-31>* scf off | 配置静音压缩功能  off 禁止静音压缩功能 |
| channel *<0-31> <0-31>* dtmf-send {enable | disable} | 配置是否启用DOD功能  第一个<0-31>为起始通道号  第二个<0-31>为结束通道号  enable:使能DOD功能；disable:禁止DOD功能  默认值：disable  🕮 注：该命令在MyPower VG M6500、VG M6000、VG2000、VG800、VG A600设备下，起始与结束通道号范围有所不同  DOD：Direct Outward Dialing，外线直拨 |
| channel *<0-31> <0-31>* wait-dial-time *<0-10>* | 配置DOD功能的参数：从FXS检测到摘机到拨号的时间间隔。  第一个<0-31>为起始通道号  第二个<0-31>为结束通道号  <0-10>：单位为秒，如果没有配置，默认为0秒，即没有时间间隔  🕮 注：该命令在MyPower VG M6500、VG M6000、VG2000、VG800、VG A600设备下，起始与结束通道号范围有所不同 |
| channel *<0-15> <0-15>* logic-terminal-type { E1| FXO | FXS } | 配置FXS卡上的通道的逻辑终端类型  🕮 注：该命令在MyPower VG M6500、VG M6000、VG2000、VG800、VG A600设备下，起始与结束通道号范围有所不同 |
| channel <0-15> <0-15> trunk-code <string> | 配置FXS上的通道的中继编码  🕮 注：该命令在MyPower VG M6500、VG M6000、VG2000、VG800、VG A600设备下，起始与结束通道号范围有所不同 |
| channel <0-15> <0-15> unit-code <string> | 配置FXS上的通道的单位编码  🕮 注：该命令在MyPower VG M6500、VG M6000、VG2000、VG800、VG A600设备下，起始与结束通道号范围有所不同 |
| ajustimp {600R | 900R | 900RC} | 阻抗调整  支持三种模式，600R，900R，900RC、1，2，3对应D/A增益分别为0dB、-3.5dB、-7dB  默认为600R、-3.5dB |
| pulse-width <1-40> <1-40> | 脉冲宽度，需要配置一个上限和下限，单位是10ms  默认脉冲宽度：下限为3，上限为12 |
| debounce *<0-10>* | 摘机消除抖动的时间，单位10ms，默认值为1 |
| dial-space *<10-100>* | 位间隔时间，默认值30 |
| ringfrequency {*25Hz* | *50Hz*} | 调节振铃频率，默认为25Hz  🕮 注：该命令只适用于MyPower VG2000、VG800、VG A600设备 |
| channel <0-31> <0-31> fax-passthru enable|disable | 配置通道支持传真透传功能。  第一个<0-31>为起始通道号  第二个<0-31>为结束通道号  🕮 注：该命令仅存在于MyPower VG M6500、VG M6000、 |

🕮 注：

1、通道必须在禁止呼叫时（也就是disable状态），才能配置。

2、所有增益的配置参数，以0db为中界线，负的越多音量越小，正的越多，音量越大

3、DSP的增益调整请慎重进行， DSP输出输入增益可能引起回声问题。

4、仅当通话的对端普遍反映声音过小或过大时，酌情调整输入增益，而当本地电话通话时普遍反映声音过大或过小时调整输出增益。

## 语音网关电话互通配置

**实验目的：**

1、熟悉IP语音网关的电话互通的配置步骤；

2、掌握IP语音网关拨号端的配置方法；

**实验拓扑：**



**实验环境说明：**

本实验中，使用两台IP语音网关，分别使用FXS0口通过RJ11接头的电话线连电话机1和电话机2，并通过WAN口分别连接路由器R1的F0和F1接口。

本实验中的语音网关使用PC机通过telnet方式进行配置。

**实验要求：**

1、按照实验环境连接设备，保证两台语音网关的WAN口可以互通；

2、配置两台语音网关的语音参数，设置两边FXS接口的电话号码别为1001和2001，并保证1001和2001电话互通；

3、将两台语音网关的所有FXS口都设置上电话号码，语音网关1的号码为1001－1004，语音网关2的号码为2001－2004，保证两边任意电话互通；

**实验步骤：**

**1、实验准备：**

首先按照环境要求将所有设备、线缆连接好，然后进行各设备的基本系统设置。

对路由器设置设备名：R1，enable密码：test；

对语音网关1设置设备名：VG1，enable密码：test；

对语音网关2设置设备名：VG2，enable密码：test；

**2、路由器接口地址设置：**

路由器需要设置的参数范例如下：

interface fastethernet0

ip address 192.168.1.1 255.255.255.0 /用于连接VG1

exit

interface fastethernet1

ip address 192.168.2.1 255.255.255.0 /用于连接VG2

exit

**3、语音网关接口地址和路由设置：**

语音网关VG1设置：

VG1(config)#**interface fastethernet0** /进入F0口（WAN口）设置模式

VG1(config-if-fastethernet0)#**ip address 192.168.1.2 255.255.255.0** /配置IP地址

VG1(config-if-fastethernet0)#exit /退出接口配置模式

VG1(config)#**ip route 0.0.0.0 0.0.0.0 192.168.1.1** /设置默认网关，指向路由器F0口地址

语音网关VG2设置：

VG2(config)#**interface fastethernet0** /进入F0口（WAN口）设置模式

VG2(config-if-fastethernet0)#**ip address 192.168.2.2 255.255.255.0** /配置IP地址

VG2(config-if-fastethernet0)#exit /退出接口配置模式

VG2(config)#**ip route 0.0.0.0 0.0.0.0 192.168.2.1** /设置默认网关，指向路由器F1口地址

设置完成后，测试两台语音网关的互通性，可以在语音网关1上对语音网关2的WAN口地址进行ping操作，范例如下：

VG1#**ping 192.168.2.2** /ping VG2的F0地址

Press key (ctrl + shift + 6) interrupt it.

Sending 5, 76-byte ICMP Echos to 192.168.2.1 , timeout is 2 seconds:

!!!!!

Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms. /全部ping通，表示路由通

**4、语音接口和拨号路由配置；**

第一步，检查语音接口运行参数，确保FXS接口已经启用，操作如下：

gateway #**show voice card** /检查语音接口情况

fxs-card 1 /fxs卡，编号1

channel 0 3 callid enable /0－3通道（即fxs0-fxs3）启用来电显示功能

channel 0 3 enable /0－3通道（即fxs0-fxs3）启用

exit

fxo-card 2 /fx0卡，编号2

channel 0 0 callid enable /0通道（即fxo0）启用来电显示功能

channel 0 0 enable /0通道（即fxo0）启用

exit

dialplan terminator # /拨号可以＃结束

从配置参数可以看到，系统已经默认启动了对应的语音接口。如果没有启动，我们可以用channel X Y enable命令启动。

第二步，拨号路由设置，操作如下：

语音网关1操作：

VG1(config)#**callrouting-conf** /进入拨号路由配置模式

VG1(config-callroute)#**no dial-peer 1** /删除1号拨号端

VG1(config-callroute)#**dial-peer 1 pots** /重新配置1号拨号端，类型为pots

VG1(config-dial-peer)#**destination-pattern 1001** /配置对应电话号码为1001

VG1(config-dial-peer)#**port 1/0** /指定对应的语音口为1/0（即FXS0）

VG1(config-callroute)#**dial-peer 10 voip** /配置10号拨号端，类型为VOIP

VG1(config-dial-peer)#**destination-pattern 2001** /配置对应电话号码为2001

VG1(config-dial-peer)#**codec g729**  /配置语音压缩编码格式为G.729

VG1(config-dial-peer)#**session-target 192.168.2.2** /配置对应的网关地址为VG2 F0地址

VG1(config-dial-peer)#exit /退回全局配置模式

VG1(config)#**interface fastethernet0** /进入F0接口配置模式

VG1(config-if-fastethernet0)#**h323-gateway voip interface** /指定本接口作为RAS协议接口

语音网关2操作：

VG2(config)#**callrouting-conf** /进入拨号路由配置模式

VG2(config-callroute)#**no dial-peer 1** /删除1号拨号端

VG2(config-callroute)#**dial-peer 1 pots** /重新配置1号拨号端，类型为pots

VG2(config-dial-peer)#**destination-pattern 2001** /配置对应电话号码为2001

VG2(config-dial-peer)#**port 1/0** /指定对应的语音口为1/0（即FXS0）

VG2(config-callroute)#**dial-peer 10 voip** /配置10号拨号端，类型为VOIP

VG2(config-dial-peer)#**destination-pattern 1001** /配置对应电话号码为1001

VG2(config-dial-peer)#**codec g729**  /配置语音压缩编码格式为G.729

VG2(config-dial-peer)#**session-target 192.168.1.2** /配置对应的网关地址为VG2 F0地址

VG2(config-dial-peer)#exit /退回全局配置模式

VG2(config)#**interface fastethernet0** /进入F0接口配置模式

VG2(config-if-fastethernet0)#**h323-gateway voip interface** /指定本接口作为RAS协议接口

注意，**h323-gateway voip interface**用于在语音网关之间互通使用H.323机制互通，如果没有本条目，则配置的dial –peer voip将不能生效。

配置完成后，通过show callroute命令检查目前语音网关的拨号路由表，应该可以看到FXS0修改后的号码以及指向对端语音网关的号码，范例如下（以VG1为例）：

VG1#**show callroute** /查看目前语音网关有效的拨号路由条目

Active callroute:

--------------------------------------------------------------------------

Type=POTS, Binded:1/1 , Preference=10, CalledNum=402

Type=POTS, Binded:1/2 , Preference=10, CalledNum=403

Type=POTS, Binded:1/3 , Preference=10, CalledNum=404

Type=POTS, Binded:1/0 , Preference=10, CalledNum=1001

/类型POTS，对应1/0语音接口（FXS0），优先级10（默认），号码1001

Type=VOIP, Binded:192.168.2.2 , Preference=10, CalledNum=2001

/类型VOIP，对应对端网关192.168.2.2，优先级10（默认），号码2001

**5、验证网关之间两台电话互通效果：**

配置完成后，使用电话1与电话2互拨，应该可以拨通，并且能正确看到对方的来电号码。

在拨号时，可以在语音网关上使用如下命令查看目前的呼叫情况：

VG1#**show call active voice brief** /查看当前的语音呼叫情况

CallId:5;Status:ACTIVE;Connections:2

PhoneNum:1001;connType:GWCI;Status:ALERTING;ProtocolStack:FXO/FXS;LocalIp:0x

c0a80102,LocalPort:32768;RemoteIp:0x0,RemotePort:0;Codec:none;AeName:1/0;

PhoneNum:2001;connType:GWCO;Status:ALERTING;ProtocolStack:H323;LocalIp:0x0,L

ocalPort:0;RemoteIp:0xc0a80102,RemotePort:32768;Codec:G729;

/从1001到2001的语音呼叫

TotalActiveCalls: 1

**6、拨号路由的批量配置：**

以上实验步骤中，我们只是在一台网关上配置了一个电话号码，现在，我们采用批量配置的方式为所有FXS口配置上电话号码。

第一步，首先清除目前语音网关的相关的拨号路由，两台网关上做同样的操作，范例如下：

VG1(config)#callrouting-conf

VG1(config-callroute)#no dial-peer 1

VG1(config-callroute)#no dial-peer 2

VG1(config-callroute)#no dial-peer 3

VG1(config-callroute)#no dial-peer 4

VG1(config-callroute)#no dial-peer 10

第二步，接下来进行批量号码配置：

语音网关1操作：

VG1(config-callroute)#**dial-peer 1 pots slot 1**  /配置1号拨号端，类型为pots，对应1号插槽（fxs card 1）

VG1(config-dial-peer)#**channel 0 3 destination-pattern 1001 increase** /配置通道0－3（即FXS0－FXS3）的号码从1001开始递增，即1001、1002、1003、1004

VG1(config-dial-peer)#exit

语音网关2操作：

VG2(config-callroute)#**dial-peer 1 pots slot 1**  /配置1号拨号端，类型为pots，对应1号插槽（fxs card 1）

VG2(config-dial-peer)#**channel 0 3 destination-pattern 2001 increase** /配置通道0－3（即FXS0－FXS3）的号码从2001开始递增，即2001、2002、2003、2004

VG2(config-dial-peer)#exit

第三步，修改VOIP拨号端

语音网关1操作：

VG1(config-callroute)#**dial-peer 10 voip** /配置10号拨号端，类型为VOIP

VG1(config-dial-peer)#**destination-pattern 200X** /配置对应电话号码为200开头的4位号

VG1(config-dial-peer)#**codec g729**  /配置语音压缩编码格式为G.729

VG1(config-dial-peer)#**session-target 192.168.2.2** /配置对应的网关地址为VG2 F0地址

VG1(config-dial-peer)#exit /退回全局配置模式

语音网关2操作：

VG2(config-callroute)#**dial-peer 10 voip** /配置10号拨号端，类型为VOIP

VG2(config-dial-peer)#**destination-pattern 1001** /配置对应电话号码为100开头的4位号

VG2(config-dial-peer)#**codec g729**  /配置语音压缩编码格式为G.729

VG2(config-dial-peer)#**session-target 192.168.1.2** /配置对应的网关地址为VG2 F0地址

VG2(config-dial-peer)#exit /退回全局配置模式

配置完成后，通过show callroute命令检查目前语音网关的拨号路由表，应该可以看到FXS0修改后的号码以及指向对端语音网关的号码，范例如下（以VG1为例）：

VG1#**show callroute** /查看目前语音网关有效的拨号路由条目

Active callroute:

--------------------------------------------------------------------------

Type=POTS, Binded:1/0 , Preference=10, CalledNum=1001

Type=POTS, Binded:1/1 , Preference=10, CalledNum=1002

Type=POTS, Binded:1/2 , Preference=10, CalledNum=1003

Type=POTS, Binded:1/3 , Preference=10, CalledNum=1004

/4个FXS口的号码分别是1001到1004

Type=VOIP, Binded:192.168.2.2 , Preference=10, CalledNum=200x

/对端网关192.168.2.2上的号码是200开头的4位号码

**7、验证网关之间任意电话互通效果：**

配置完成后，使用电话1与电话2分别接在两台语音网关上的任意FXS口上，应该都可以拨通，并且能正确看到对方的来电号码。

在拨号时，可以在语音网关上使用如下命令查看目前的呼叫情况：

VG1#**show call active voice brief** /查看当前的语音呼叫情况

CallId:9;Status:ACTIVE;Connections:2

PhoneNum:1002;connType:GWCI;Status:ALERTING;ProtocolStack:FXO/FXS;LocalIp:0x

c0a80102,LocalPort:32768;RemoteIp:0x0,RemotePort:0;Codec:none;AeName:1/1;

PhoneNum:2003;connType:GWCO;Status:ALERTING;ProtocolStack:H323;LocalIp:0x0,L

ocalPort:0;RemoteIp:0xc0a80102,RemotePort:32768;Codec:G729;

/从1002到2003的语音呼叫

TotalActiveCalls: 1

TotalActiveCalls: 1

**相关命令详解：**

**1、VOIP拨号端配置相关命令**

VoIP拨号端相关配置命令

对于MyPower VG M6500、VG M6000，MyPower VG2000版本，拨号端的范围是1-1024：

gateway(config-callroute)#dial-peer <1-1024> voip

对于MyPower VG800，MyPower VG A600版本，拨号端的范围是1-255

gateway(config-callroute)#dial-peer <1-255> voip

gateway(config-dial-peer)#？

|  |  |
| --- | --- |
| **命令** | **描述** |
| codec {g711a | g711u | g723 | g729 | g729a | g729wb} | 配置H323的语音编码类型为g711a、g711u、g723、g729、g729a 、g729wb。当从ip呼出时，此处配置的编码模式生效。默认缺省值为g729 |
| destination-pattern *<string>* | 配置E.164 电话号码 |
| fax-protocol t38 | 配置T38传真协议 |
| fax-protocol pass-through | 配置传真为透传方式（MyPower VG M6000、MyPower VG M6500暂不支持） |
| fax-protocol disable | 配置禁止此拨号端的传真功能  🕮 注：禁止后即使有全局的传真配置，对于拔号端也不生效 |
| no | 取消命令 |
| session-target {*A.B.C.D* | ras | sip-server} | 配置VoIP端指向被叫IP地址或网守或者SIP服务器 |
| num-trans-index *<1-100>* {called | calling} | 配置的号码转换规则  <1-100>: 号码变换规则索引  called: 将规则用于被叫号码  calling: 将规则用于主叫号码 |
| preference *<1－20>* | 配置优先级（1-20），优先级随数字增加而降低。缺省为10，20为路由不可用 |
| backup | 配置IPSWITCH切换功能，实现当前所用IP链路出现故障后完成IP-TO-PSTN或IP-TO-IP切换，保障通话从备份拨号端重新路由：  ip:  切换到其他配置的IP拨号端，使用原来的被叫号码重新路由.  ip prefix <string>:  切换到其他配置的IP拨号端，在原来的被叫号码前加上前缀号码作为新的被叫号码重新路由.  pstn :  切换到其他配置的PSTN拨号端(包含VOE1及FXO端口的POTS)，使用原来的被叫号码重新路由.  pstn prefix <string>:  切换到其他配置的PSTN拨号端(包含VOE1及FXO端口的POTS)，在原来的被叫号码前加上前缀号码作为新的被叫号码重新路由.  callee-busy {enable | disable}:  启用或者禁用被叫遇忙时路由切换功能(仅在H.323协议下有效). |
| restrict-calling <string> | 配置主叫限制，当主叫号码在已经配置了的主叫限制表里，则该条路由将不能被该次呼叫使用。 |
| sip user-phone | 配置发起呼叫时，携带sip用户电话标记信息 |
| no save digitmap | 将配置的路由号码从digitmap中删除 |
| exit | 退出当前配置模式 |

**2、POTS拨号端配置相关命令**

1．POTS拨号端相关配置命令

对于MyPower VG M6500、VG M6000，MyPower VG2000版本，拨号端的范围是1-1024：

gateway(config-callroute)#dial-peer *<1-1024>* pots

对于MyPower VG800，MyPower VG A600版本，拨号端的范围是1-255

gateway(config-callroute)#dial-peer *<1-255>* pots

gateway(config-dial-peer)#？

|  |  |
| --- | --- |
| **命令** | **描述** |
| port <string> | 配置POTS端对应的语音口 |
| destination-pattern *<string>* | 路由号码匹配规则，可配置完全号码匹配，前缀匹配 |
| channel <0-15> <0-15> register {disable | enable} | 可以指定某些号码不注册网守或SIP注册服务器.及注册网守或SIP注册服务器，默认要进行注册 |
| preference *<1-20>* | 配置优先级，（1-20），优先级随数字增加而降低。缺省为10，20为路由不可用 |
| codec {g711a | g711u | g723 | g729 | g729a | g729wb} | 配置H323的语音编码类型为g711a、g711u、g723、g729、g729a 、g729wb。当从ip呼入到本端口时生效 |
| hidden calling-number {disable | enable} | 配置此号码发起呼叫时，是否进行主叫号码隐藏。默认情况下，不进行主叫号码隐藏 |
| exit | 退出当前配置模式 |

2．POTS拨号端批量配置相关命令

gateway(config-callroute)#dial-peer *<1-1024>* pots slot *<0-8>*

gateway(config-dial-peer)#

|  |  |
| --- | --- |
| **命令** | **描述** |
| channel *<0-31> <0-31>* destination-pattern *<string>* increase | 批量配置POTS端口路由规则  <0-31> POTS起始端口号  <0-31> POTS结束端口号  <string> 路由号码匹配规则，可配置完全号码匹配，前缀匹配  increase配置号码从起始端口号到结束端口号递增 |
| channel *<0-31> <0-31>* register-gk {disable | enable} | 批量配置POTS端口的号码是否注册网守  <0-31> POTS起始端口号  <0-31> POTS结束端口号  disabl表示号码不注册网守，Enable表示注册网守.此功能主要用于拨打大小号时和网守配合.默认注册网守 |
| channel *<0-31> <0-31>* num-trans-index *<1-100>* {called | calling} | 批量配置POTS端口的号码转换规则  <0-31> POTS起始端口号  <0-31> POTS结束端口号  <1-100> 号码变换规则  called 对于被叫号码做号码变换;calling 对于主叫号码做号码变换 |
| channel *<0-31> <0-31>* preference *<1－20>* | 批量配置POTS端口的优先级  <0-31> POTS起始端口号  <0-31> POTS结束端口号  <1-20> 配置优先级，优先级随数字增加而降低。缺省为10，20为路由不可用 |
| channel *<0-31> <0-31>* codec {g711a | g711u | g723 | g729 | g729a | g729wb} | 批量配置POTS端口语音编码类型为g711a、g711u、g723、g729、g729a、g729wb。为呼叫到本端口时所配置的编码优先生效。默认配置为g729  <0-31>POTS起始端口号  <0-31>POTS结束端口号 |
| channel *<0-31> <0-31>* preference *<1－20>* | 批量配置POTS端口的优先级  <0-31> POTS起始端口号  <0-31> POTS结束端口号  <1-20> 配置优先级，优先级随数字增加而降低。缺省为10，20为路由不可用 |
| channel *<0-31> <0-31>* hidden calling-number {disable | enable} | 配置此端口对应的号码发起呼叫时，是否进行主叫号码隐藏。默认情况下，不进行主叫号码隐藏 |
| channel *<0-31> <0-31>* no num-trans-index *<1-100>* {calling | called} | 去掉号码转换 |
| restrict-calling *<string>* | 配置主叫限制，当主叫号码在已经配置了的主叫限制表里，则该条路由将不能被该次呼叫使用。 |
| fxo-channel codec {g711a|g711u|g723|g729|g729a|g729wb} | 批量配置混合卡的fxo端口的拨号端的编码方式（只针对VG6000系统的有效） |
| fxo-channel destination-pattern <*string*> | 批量配置混合卡的fxo端口的拨号端的目的地址（只针对VG6000系统的有效） |
| fxo-channel num-trans-index *<1-100>* called | 批量配置混合卡的fxo端口的拨号端的被叫号码转换规则（只针对VG6000系统的有效） |
| fxo-channel num-trans-index *<1-100>* calling | 批量配置混合卡的fxo端口的拨号端的主叫号码转换规则（只针对VG6000系统的有效） |
| fxo-channel preference *<1-20>* | 批量配置混合卡的fxo端口的拨号端的优先级（只针对VG6000系统的有效） |
| fxo-channel no destination-pattern | 批量取消混合卡的fxo端口的拨号端的目的地址（只针对VG6000系统的有效） |
| fxo-channel no num-trans-index *<1-100>* called | 批量取消混合卡的fxo端口的拨号端的被叫号码转换规则（只针对VG6000系统的有效） |
| fxo-channel no num-trans-index *<1-100>* calling | 批量取消混合卡的fxo端口的拨号端的主叫号码转换规则（只针对VG6000系统的有效） |
| exit | 退出当前配置模式 |

🕮 注：

在同一个拨号端下，channel和fxo-channel系列的命令不能同时使用。二者只会有一个有效。

**3、配置编码相关命令**

gateway(config-dial-peer)#

|  |  |
| --- | --- |
| **命令** | **描述** |
| codec {g711a | g711u | g723 | g729 | g729a | g729wb} | 配置路由到VoIP的优选编码，默认使用g729 |

🕮 注：

网关是PSTN与IP之间的进行信令和媒体格式进行转换的设备，编码在PSTN模式pots/voe1和IP模式voip下都可以配置。配在voip下，是指使用此匹配模式destination-pattern呼叫到ip的优选编码，在PSTN下，是指使用从IP呼到此匹配模式优先使用的编码。

语音网关中进行IP通话时需要进行语音编解码，常用的编码方式有G723、G729、G711。G723、G729使用较复杂的编解码算法，编码后的码率较低（几Kbps/S）。G711使用简单的编解码算法，编码后的码率较高（64Kbps/S）。

在网关中完成编解码功能的DSP部件处理能力是有限的。使用复杂的编码虽然码率低，但DSP资源消耗多，能并发处理的语音通道少。相反，使用简单的编码方式虽然码率高，但DSP资源消耗少，能并发处理的语音通道多。

MyPower VG M6500、VG M6000产品中，为了方便用户根据自己的网络情况进行编码方式选择，除了提供传统的混合编码模式外（同时支持G711、G723、G729），还提供了单编码模式的配置选项。用户可以根据自己的网络情况选择合适的编码来进行IP通话。网络带宽较窄时可选用IP通话并发数量较少，但带宽占用小的G723、G729编码方式。网络带宽较充裕时，可选用G711编码提高IP通话的并发数。

**4、****当前语音呼叫的端口状态信息的显示命令**

下面两个命令用来查看当前语音呼叫的端口状态信息，通话状态信息等。

|  |  |
| --- | --- |
| 命令 | 描述 |
| gateway#show voice call summary | 该功能主要用于查看网关上各个端口的当前的语音呼叫的状态信息。包括编码方式，呼叫状态，端口状态。 |
| gateway#show call active voice brief | 该功能主要用于查看网关上当前的所有的通话状态的简要的统计信息。 |

**5、显示呼叫路由信息命令**

gateway#show callroute？

|  |  |
| --- | --- |
| **命令** | **描述** |
| called <srting> | 显示特定号码的激活路由信息 |
| pots | 显示POTS的激活路由信息 |
| voe1 | 显示E1的激活路由信息 |
| voip | 显示IP的激活路由信息 |
| <CR> | 显示所有的激活路由信息 |

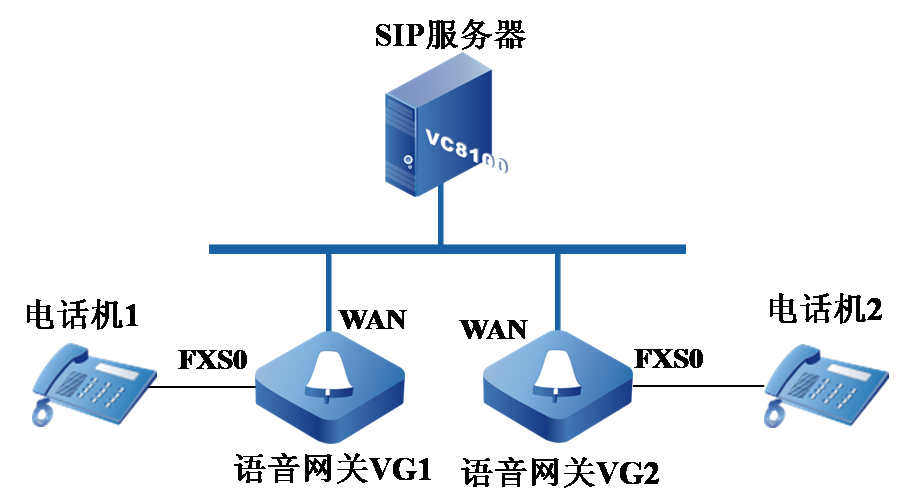
## 通过SIP服务器实现电话互通

**实验目的：**

1、熟悉IP语音网关的通过SIP服务器实现电话互通的配置步骤；

2、掌握IP语音网关注册SIP服务器的配置方法；

**实验拓扑：**



**实验环境说明：**

本实验中，使用两台IP语音网关和一台SIP服务器，三台设备都接入以太网交换机。

两台IP语音网关分别使用FXS0口通过RJ11接头的电话线连电话机1和电话机2。

本实验中的语音网关使用PC机通过telnet方式进行配置。

**实验要求：**

1、按照实验环境连接设备，保证两台语音网关的WAN口和SIP服务器的以太口可以互通；

2、SIP服务器由讲师设置，允许网关的号码注册；

3、配置两台语音网关的语音参数，设置两边FXS接口的电话号码别为1001－1004和2001－2004，并注册SIP服务器，通过SIP服务器让两台网关的任意FXS接口号码互通；

**实验步骤：**

**1、实验准备：**

首先按照环境要求将所有设备、线缆连接好，然后进行各设备的基本系统设置。

对SIP服务器设置域名：test：地址：192.168.2.1（教师设置）

对语音网关1设置设备名：VG1，enable密码：test；

对语音网关2设置设备名：VG2，enable密码：test；

**2、SIP服务器相关参数设置：**

SIP服务器设置完域名、地址后，需要设置两台语音的电话号码作为用户名，并设置密码为test，需要设置的用户为1001－1004，2001－2004，密码统一设置为test。（由讲师设置）

在如下界面中进行设置：



注意：需要配置的内容有用户标识（即号码）、PIN和确认PIN（可以和SIP密码一致）、SIP密码；

**3、语音网关接口地址设置：**

语音网关VG1设置：

VG1(config)#**interface fastethernet0** /进入F0口（WAN口）设置模式

VG1(config-if-fastethernet0)#**ip address 192.168.2.2 255.255.255.0** /配置IP地址

语音网关VG2设置：

VG2(config)#**interface fastethernet0** /进入F0口（WAN口）设置模式

VG2(config-if-fastethernet0)#**ip address 192.168.2.3 255.255.255.0** /配置IP地址

设置完成后，测试两台语音网关与SIP服务器的互通性，可以在语音网关1上对SIP服务器的以太口地址进行ping操作，范例如下：

VG1#**ping 192.168.2.1** /ping 服务器以太口地址

Press key (ctrl + shift + 6) interrupt it.

Sending 5, 76-byte ICMP Echos to 192.168.2.1 , timeout is 2 seconds:

!!!!!

Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms. /全部ping通，表示路由通

**4、语音接口和拨号路由配置；**

第一步，检查语音接口运行参数，确保FXS接口已经启用，操作如下：

gateway #**show voice card** /检查语音接口情况

fxs-card 1 /fxs卡，编号1

channel 0 3 callid enable /0－3通道（即fxs0-fxs3）启用来电显示功能

channel 0 3 enable /0－3通道（即fxs0-fxs3）启用

exit

fxo-card 2 /fx0卡，编号2

channel 0 0 callid enable /0通道（即fxo0）启用来电显示功能

channel 0 0 enable /0通道（即fxo0）启用

exit

dialplan terminator # /拨号可以＃结束

从配置参数可以看到，系统已经默认启动了对应的语音接口。如果没有启动，我们可以用channel X Y enable命令启动。

第二步，首先清除目前语音网关的相关的拨号路由，两台网关上做同样的操作，范例如下：

VG1(config)#callrouting-conf

VG1(config-callroute)#no dial-peer 1

VG1(config-callroute)#no dial-peer 2

VG1(config-callroute)#no dial-peer 3

VG1(config-callroute)#no dial-peer 4

VG1(config-callroute)#no dial-peer 5

VG1(config-callroute)#no dial-peer 100

第三步，配置向SIP服务器发起注册的接口

语音网关1操作：

VG1(config)#**interface fastethernet0** /进入F0接口配置模式

VG1(config-if-fastethernet0)# **sip-gateway voip interface** /指定本接口作为SIP协议接口

VG1(config-if-fastethernet0)# **sip-gateway voip proxy 192.168.2.1**

/指定SIP服务器代理地址为192.168.2.1

VG1(config-if-fastethernet0)# **sip-gateway voip registrar 192.168.2.1**

/指定SIP注册服务器地址为192.168.2.1

VG1(config-if-fastethernet0)# **sip-gateway voip password test** /指定注册时的全局密码为test

语音网关2操作：

VG1(config)#**interface fastethernet0** /进入F0接口配置模式

VG1(config-if-fastethernet0)# **sip-gateway voip interface** /指定本接口作为SIP协议接口

VG1(config-if-fastethernet0)# **sip-gateway voip proxy 192.168.2.1**

/指定SIP服务器代理地址为192.168.2.1

VG1(config-if-fastethernet0)# **sip-gateway voip registrar 192.168.2.1**

/指定SIP注册服务器地址为192.168.2.1

VG1(config-if-fastethernet0)# **sip-gateway voip password test** /指定注册时的全局密码为test

第四步，拨号路由设置，操作如下：

语音网关1操作：

VG1(config-callroute)#**dial-peer 1 pots slot 1**  /配置1号拨号端，类型为pots，对应1号插槽（fxs card 1）

VG1(config-dial-peer)#**channel 0 3 destination-pattern 1001 increase** /配置通道0－3（即FXS0－FXS3）的号码从1001开始递增，即1001、1002、1003、1004

VG1(config-dial-peer)#exit

语音网关2操作：

VG2(config-callroute)#**dial-peer 1 pots slot 1**  /配置1号拨号端，类型为pots，对应1号插槽（fxs card 1）

VG2(config-dial-peer)#**channel 0 3 destination-pattern 2001 increase** /配置通道0－3（即FXS0－FXS3）的号码从2001开始递增，即2001、2002、2003、2004

VG2(config-dial-peer)#exit

第五步，修改VOIP拨号端

语音网关1操作：

VG1(config-callroute)#**dial-peer 10 voip** /配置10号拨号端，类型为VOIP

VG1(config-dial-peer)#**destination-pattern 200X** /配置对应电话号码为200开头的4位号

VG1(config-dial-peer)#**codec g729**  /配置语音压缩编码格式为G.729

VG1(config-dial-peer)#**session-target sip-server** /通过SIP代理服务器发起呼叫

VG1(config-dial-peer)#exit /退回全局配置模式

语音网关2操作：

VG2(config-callroute)#**dial-peer 10 voip** /配置10号拨号端，类型为VOIP

VG2(config-dial-peer)#**destination-pattern 1001** /配置对应电话号码为100开头的4位号

VG2(config-dial-peer)#**codec g729**  /配置语音压缩编码格式为G.729

VG2(config-dial-peer)#**session-target sip-server** /通过SIP代理服务器发起呼叫

VG2(config-dial-peer)#exit /退回全局配置模式

第六步，启动SIP注册

语音网关1和2都做如下操作：

VG1(config)# **sip-gateway** /通过SIP协议向注册服务器注册本地号码

配置完成后，通过show callroute命令检查目前语音网关的拨号路由表，应该可以看到FXS0修改后的号码以及指向对端语音网关的号码，范例如下（以VG1为例）：

VG1#**show callroute** /查看目前语音网关有效的拨号路由条目

Active callroute:

--------------------------------------------------------------------------

Type=POTS, Binded:1/0 , Preference=10, CalledNum=1001

Type=POTS, Binded:1/1 , Preference=10, CalledNum=1002

Type=POTS, Binded:1/2 , Preference=10, CalledNum=1003

Type=POTS, Binded:1/3 , Preference=10, CalledNum=1004

/4个FXS口的号码分别是1001到1004

Type=VOIP, Binded:ras , Preference=10, CalledNum=200x

/通过SIP服务器访问的号码，200开头的4位号码

最后一条表示，已经成功注册上SIP服务器了。

而讲师也可以在SIP服务器上查询网关注册的号码进行确认。

**5、验证网关之间两台电话互通效果：**

配置完成后，使用电话1与电话2互拨，应该可以拨通，并且能正确看到对方的来电号码。

在拨号时，可以在语音网关上使用如下命令查看目前的呼叫情况：

VG1#**show call active voice brief** /查看当前的语音呼叫情况

CallId:5;Status:ACTIVE;Connections:2

PhoneNum:1001;connType:GWCI;Status:ALERTING;ProtocolStack:FXO/FXS;LocalIp:0x

c0a80102,LocalPort:32768;RemoteIp:0x0,RemotePort:0;Codec:none;AeName:1/0;

PhoneNum:2001;connType:GWCO;Status:ALERTING;ProtocolStack:H323;LocalIp:0x0,L

ocalPort:0;RemoteIp:0xc0a80102,RemotePort:32768;Codec:G729;

/从1001到2001的语音呼叫

TotalActiveCalls: 1

再将电话换接别的FXS口，进行拨打测试，应该都可以拨通。

**相关命令详解：**

**1、配置向SIP服务器发起注册的接口相关命令**

在接口下启动SIP协议，使用户可以使用SIP方式创建和删除一个对话。用户可以在点对点的通信中使用SIP，也可以通过代理服务器与对方建立连接。同时本地网关的电话号码可以定期的向SIP的注册服务器发送注册报文，以更新服务器上的用户信息。

gateway(config)#interface fastethernet *1*

|  |  |
| --- | --- |
| **命令** | **描述** |
| sip-gateway voip localPort <5000-10000> | 配置sip协议使用的本地端口号，默认为5060 |
| sip-gateway voip interface | 指定该接口做为网关的SIP协议接口 |
| sip-gateway voip proxy *A.B.C.D* | 指定网关使用的SIP代理服务器的IP地址，A.B.C.D为SIP代理服务器的IP地址 |
| sip-gateway voip proxy domain-name *<string>* | 指定网关使用的SIP代理服务器的域名，string为SIP代理服务器的域名 |
| sip-gateway voip proxyPort <1-65535> | 指定远端代理服务器的端口，默认为5060 |
| sip-gateway voip registrar *A.B.C.D* | 指定网关使用的SIP注册服务器的IP地址，A.B.C.D为SIP注册服务器的IP地址 |
| sip-gateway voip registrar domain-name <*string>* | 指定网关使用的SIP注册服务器的域名，string为SIP注册服务器的域名 |
| sip-gateway voip registrarPort <1-65535> | 指定注册服务器的端口，默认为5060 |
| sip-gateway voip registrar expires *<200-3600>* | 指定SIP终端注册的超时时间（秒），默认为3600秒 |
| sip-gateway voip username *<string>* | 若注册服务器需要身份验证，则在此指定注册时的全局用户名（也可在线卡的dial-peer下配置每一个号码的用户名，此种情况优先级最高） |
| sip-gateway voip password <*string>* | 若注册服务器需要身份验证，则在此指定注册时的全局密码（也可在线卡的dial-peer下配置每一个号码的密码，此种情况优先级最高） |
| sip-gateway voip retry-invite <1-5> | 指定发起呼叫对端不可达未收到响应时INVITE的重传次数，默认为5 |
| sip-gateway voip local-area *A.B.C.D A.B.C.D* | 指定SIP本地域，  前一个IP是指网段号，后一个IP是掩码。  当使用STUN穿越NAT的时候需要该配置实现内网通信 |
| sip-gateway voip signal-extern-ip A.B.C.D | 指定SIP信令外部地址,在需要穿越静态NAT时使用，当启用了STUN模式穿越NAT时，不要配置此项。 |
| sip-gateway voip media-extern-ip A.B.C.D {<CR> | useforlocal} | 指定媒体外部地址,在需要静态穿越NAT时使用，当启用了STUN模式穿越NAT时，不要配置此项。Useforlocal参数用于指定本地媒体发包源地址采用本地址，在多IP上联时使用 |
| ip name-server srv { enable | disable } | 指定使能或禁用SRV域名查询功能, 使能命令在配置的服务器地址为域名时使用。 |

🕮 注：

**在使用STUN实现SIP穿越NAT后，内网网关之间要实现通话，必须配置本地域，并且指定静态路由。**

本地域匹配规则：远端IP按位与本地域掩码，如果结果等于本地域，则认为该远端IP属于本地域，通过SIP拨打该IP所在网关的电话时，不使用STUN。

**端口配置限定：配置localport、proxyport、registrarport，必须在sip协议down时或接口shutdown时进行配置；其他配置在sip协议UP时才能配置。**

**在使用SIP功能时，强烈建议不要更改环回接口loopback0的地址，更改后SIP功能会受影响，如果确实需要使用环回接口，请使用loopback1等接口。**

**当网关设备在NAT后面，SIP服务器或其它需要通信的网关在NAT外面，需要配置SIP信令外部地址和媒体外部地址，同时还需在NAT上配置内部地址到外部地址的映射关系。**

**在使用SRV域名查询功能时，如果DNS服务器不支持SRV，网关上开启SRV功能，不会影响注册和通话。**

**在使用通过SIP的呼叫转接功能时，转接消息refer消息中refer-to头域如果是带的域名，则需要网关能够解析到该域名，即网关需要配置对应的域名服务器，否则可能导致域名解析阻塞，而使转接失败。**

**2、配置voip拨号端使用SIP服务方式相关命令**

gateway(config-callroute)#dial-peer *1* voip

|  |  |
| --- | --- |
| **命令** | **描述** |
| destination-pattern *<string>* | 配置目的端电话号码 |
| session-target *A.B.C.D* | 直接向指定的IP地址发起呼叫，A.B.C.D为被叫IP地址 |
| session-target sip-server | 通过Proxy Server发起呼叫 |

**3、启动SIP注册相关命令**

gateway(config)#

|  |  |
| --- | --- |
| **命令** | **描述** |
| sip-gateway | 网关通过SIP协议向注册服务器注册本地的电话号码 |
| no sip-gateway | 网关通过SIP协议向注册服务器注销已经注册的本地电话号码 |