Yeslab 华为实验手册

Release 1.0 苏函

在做本实验手册之前，请先完成 CCNA,CCNP 的实验，因为在那二部实验手册里，对原理性的描述更多一些，而在所有的实验实现的原理上，都是一样的。本实验手册的作用在于，当你遇到一台华为的设备时，不至于什么都不会做。

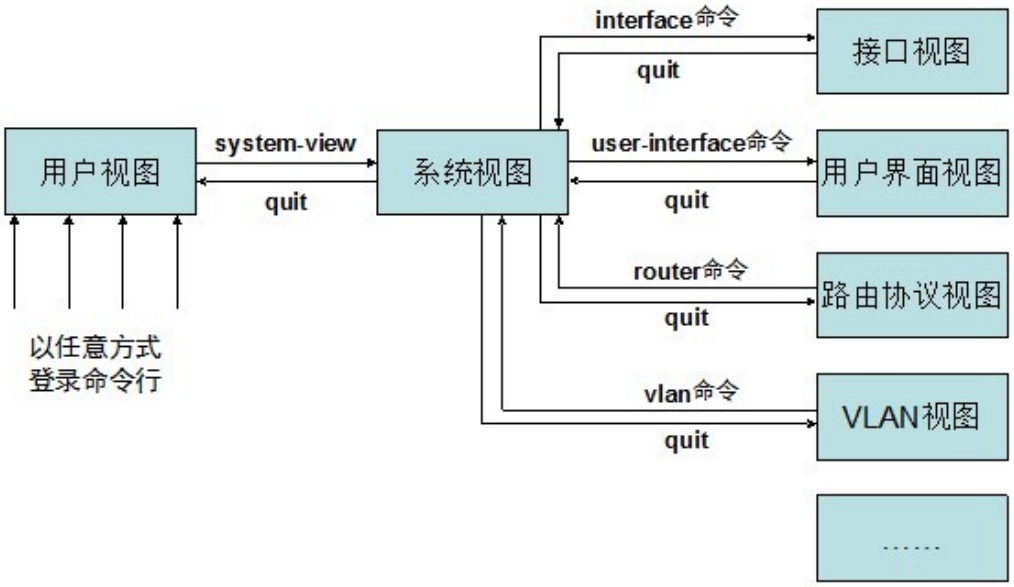
Yeslab-Learning on demand
www.yeslab.net
RS & ISP QQ Group:74041531
Tel: 010:82864660
Fax:010-82684760

目录

实验第一部分 基本配置

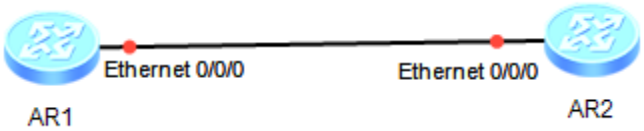
实验目的

掌握进入路由器各种模式，配置登陆 console 口密码，系统视图密码，保存配置，删除配置等等基本配置命令，这里不一一列举，请做下面的实验



- 用<Ctrl+Z>可以从任意视图直接回到用户视图

实验拓扑



实验步骤

1：以下为刚进入路由器时的默认模式--用户模式（用户视图），用大于号表示。

```
<Huawei>
```

2：用户模式进入特权模式（系统视图）：

```
<Huawei>system-view
Enter system view, return user view with Ctrl+Z.
[Huawei]
```

//以上是特权模式。用[]表示

3：从特权模式进入接口模式：

```
[Huawei] interface Ethernet 0/0/0
[Huawei-Ethernet0/0/0]
//以上为接口模式。
[Huawei-Ethernet0/0/0]quit  //退出命令
[Huawei]
//退出的时候打 quit
```

4：系统视图下设置一个名字：

```
[Huawei]sysname yeslab
[yeslab]
//可以发现路由器的名字已经变成了 Yeslab
```

5 ：配置 console 口密码

```
[yeslab]user-interface console 0     //进入 console 口配置模式
[yeslab-ui-console0]authentication-mode password
//以上设置 console 认证的模式是本地密码的方式，还可以选择不认证与 AAA 认证
[yeslab-ui-console0]set authentication password cipher yeslab
//以上是设备 console 口登陆密码为 yeslab，以后再登陆 console 将需要密码
[yeslab-ui-console0]quit
[yeslab]
```

6：路由协议模式：

```
[yeslab]rip
[yeslab-rip-1]quit
[yeslab]ospf
[yeslab-ospf-1]quit
[yeslab]isis
[yeslab]quit
//以上是进入和退出三种路由协议模式
```

7：设置系统时间和时区

```
<yeslab>clock timezone Beijing add 8
<yeslab>clock datetime 19:57:00 2013-5-13
```

8：设置 telnet 密码

```
[R2]user-interface vty 0 4   //进入 telnet 的虚拟接口 vty 下
[R2-ui-vty0-4]authentication-mode password
```

```
[R2-ui-vty0-4]set authentication password simple yeslab
[R2-ui-vty0-4]user privilege level 3   //设备登陆进来的权限为控制级
```

用户级别和命令级别对应关系表：

用户级别	命令级别	级别名称	说明
0	0	参观级	网络诊断工具命令（ping、tracert）、从本设备出发访问外部设备的命令（Telnet 客户端）等。
1	0、1	监控级	用于系统维护，包括 display 等命令。
2	0、1、2	配置级	业务配置命令，包括路由、各个网络层次的命令，向用户提供直接网络服务。
3～15	0、1、2、3	管理级	用于系统基本运行的命令，对业务提供支撑作用，包括文件系统、FTP、TFTP 下载、配置文件切换命令、备板控制命令、用户管理命令、命令级别设置命令、系统内部参数设置命令；用于业务故障诊断的 debugging 命令等。

9：保存配置文件

```
<yeslab>save
The current configuration will be written to the device.
Are you sure to continue?[Y/N]y
```

10：清空配置文件

```
<R1>reset saved-configuration
Warning: The action will delete the saved configuration in the device.
The configuration will be erased to reconfigure. Continue? [Y/N]:y
```

11：显示系统运行配置信息

```
[R1]display current-configuration
#
sysname R1
#
------省略以下部分------=
```

12：配置 IP 地址，并打开接口

```
[R1] interface Ethernet0/0/0
```

```
[R1-Ethernet0/0/0]ip address 12.1.1.1 255.255.255.0
[R1-Ethernet0/0/0]undo shutdown
```

```
[R2] interface Ethernet0/0/0
[R2-Ethernet0/0/0]ip address 12.1.1.2 255.255.255.0
[R2-Ethernet0/0/0]undo shutdown
```

13：显示接口状态：与 CISCO 的相比，所有的 show 命令都使用 display 代替

```
[R1]display interface Ethernet 0/0/0
Ethernet0/0/0 current state : UP
Line protocol current state : UP
Last line protocol up time : 2013-05-14 11:47:52 UTC-08:00
Description:
Route Port,The Maximum Transmit Unit is 1500
Internet Address is 12.1.1.1/24
IP Sending Frames' Format is PKTFMT_ETHNT_2, Hardware address is 5489-9897-
8142
Last physical up time   : 2013-05-14 11:42:05 UTC-08:00
Last physical down time : 2013-05-14 11:42:02 UTC-08:00
Current system time: 2013-05-14 11:53:16-08:00
Hardware address is 5489-9897-8142
    Last 300 seconds input rate 3 bytes/sec, 0 packets/sec
    Last 300 seconds output rate 3 bytes/sec, 0 packets/sec
    Input: 1040 bytes, 11 packets
    Output: 1040 bytes, 11 packets
    Input:
      Unicast: 10 packets, Multicast: 0 packets
      Broadcast: 1 packets
    Output:
      Unicast: 11 packets, Multicast: 0 packets
      Broadcast: 0 packets
    Input bandwidth utilization :    0%
    Output bandwidth utilization :    0%
```

14：测试网络连通性，此命令与 CISCO 一致。

```
[R1]ping 12.1.1.2
  PING 12.1.1.2: 56  data bytes, press CTRL_C to break
```

```
  Reply from 12.1.1.2: bytes=56 Sequence=1 ttl=255 time=60 ms
  Reply from 12.1.1.2: bytes=56 Sequence=2 ttl=255 time=40 ms
  Reply from 12.1.1.2: bytes=56 Sequence=3 ttl=255 time=10 ms
  Reply from 12.1.1.2: bytes=56 Sequence=4 ttl=255 time=50 ms
  Reply from 12.1.1.2: bytes=56 Sequence=5 ttl=255 time=30 ms

 --- 12.1.1.2 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 10/38/60 ms
```

15：跟踪沿跳路径：

【描述】

tracert 命令用来测试数据包从发送主机到目的地所经过的网关，主要用于检查网络连接是否可达，以及辅助分析网络在何处发生了故障。

tracert 命令的执行过程：首先发送一个 TTL 为 1 的数据包，因此第一跳发送回一个 ICMP 错误消息以指明此数据包不能被发送（因为 TTL 超时），之后此数据包被重新发送，TTL 为 2，同样第二跳返回 TTL 超时，这个过程不断进行，直到到达目的地。执行这些过程的目的是记录每一个 ICMP TTL 超时消息的源地址，以提供一个 IP 数据包到达目的地所经历的路径。当用 **ping** 命令测试发现网络出现故障后，可以用 **tracert** 测试网络何处有故障。**tracert** 命令的输出信息包括到达目的地所有网关的 IP 地址，如果某网关超时，

则输出“***”。

例：

```
[R1]tracert 12.1.1.2
 traceroute to  12.1.1.2(12.1.1.2), max hops: 30 ,packet length: 40,press CTRL_C to break
 1 12.1.1.2 30 ms  50 ms  30 ms
```

此命令后面有多个参数可选，如下：

-f：表示该测试项用于测试-f 开关是否正确，*first-TTL* 指定一个初始 TTL，它的范围是大于 0 小于最大 TTL，*first-TTL* 缺省为 1；

-m：表示该测试项用于测试-m 开关是否正确，*max-TTL* 指定一个最大 TTL，它的范围是大于初始 TTL，*max-TTL* 缺省为 30；

-p：表示该测试项用于测试-p 开关是否正确，*port* 是一个整数，该整数是目的主机的端口号，用户一般无须更改此选项；*port* 缺省为 33434；

-q：表示该测试项用于测试-q 开关是否正确，*nqueries* 是一个整数，该整数是每次发送的探测数据包的个数，它的范围是大于 0；*nqueries* 缺省为 3； **-w**：表示该测试项用于测试-wf 开关是否正确，*timeout* 是一个整数，该整数指明 IP 包的超时时间，单位为秒，它的范围是大于 0；*timeout* 缺省为 5s； *host*：目的主机的 IP 地址或远端系统的主机名。

13：查看路由表

```
[R1-Ethernet0/0/0]display ip routing
Route Flags: R - relay, D - download to fib
------------------------------------------------------------------------------
Routing Tables: Public
         Destinations : 4        Routes : 4

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

        12.1.1.0/24  Direct  0    0           D   12.1.1.1        Ethernet0/0/0
        12.1.1.1/32  Direct  0    0           D   127.0.0.1       Ethernet0/0/0
      127.0.0.0/8    Direct  0    0           D   127.0.0.1       InLoopBack0
      127.0.0.1/32   Direct  0    0           D   127.0.0.1       InLoopBack0
```

此时仅有直连的以及环回口的路由

14：关闭弹 LOG 信息：

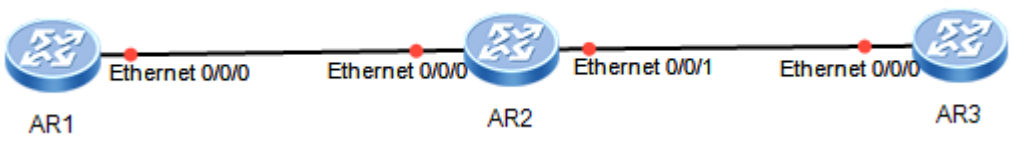
```
[R1]undo info-center enable
```

实验第二部分 静态路由

实验目的

使用静态路由使 IP 数据可达

实验拓扑



实验步骤

1：更名及配置 IP 地址

```
<Huawei>system-view
[Huawei]sysname R1
[R1]interface Ethernet 0/0/0
[R1-Ethernet0/0/0]ip address 12.1.1.1 255.255.255.0
[R1-Ethernet0/0/0]undo shutdown
```

```
<Huawei>system-view
[Huawei]sysname R2
[R2]interface Ethernet 0/0/0
[R2-Ethernet0/0/0]ip address 12.1.1.2 255.255.255.0
[R2-Ethernet0/0/0]undo shutdown
[R2]interface Ethernet 0/0/1
[R2-Ethernet0/0/1]ip add
[R2-Ethernet0/0/1]ip address 23.1.1.2 255.255.255.0
[R2-Ethernet0/0/1]undo shutdown
```

```
[Huawei]sysname R3
[R3-Ethernet0/0/0]ip address 23.1.1.3 255.255.255.0
[R3-Ethernet0/0/0]undo shutdown
```

2：在中间路由器 R2 上测试直连是否通

```
[R2-Ethernet0/0/1]ping 12.1.1.1
  PING 12.1.1.1: 56  data bytes, press CTRL_C to break
    Reply from 12.1.1.1: bytes=56 Sequence=1 ttl=255 time=160 ms
    Reply from 12.1.1.1: bytes=56 Sequence=2 ttl=255 time=20 ms
    Reply from 12.1.1.1: bytes=56 Sequence=3 ttl=255 time=10 ms
    Reply from 12.1.1.1: bytes=56 Sequence=4 ttl=255 time=30 ms
    Reply from 12.1.1.1: bytes=56 Sequence=5 ttl=255 time=40 ms

  --- 12.1.1.1 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 10/52/160 ms
```

```
[R2-Ethernet0/0/1]ping 23.1.1.3
  PING 23.1.1.3: 56  data bytes, press CTRL_C to break
```

```
  Reply from 23.1.1.3: bytes=56 Sequence=1 ttl=255 time=50 ms
  Reply from 23.1.1.3: bytes=56 Sequence=2 ttl=255 time=50 ms
  Reply from 23.1.1.3: bytes=56 Sequence=3 ttl=255 time=50 ms
  Reply from 23.1.1.3: bytes=56 Sequence=4 ttl=255 time=30 ms
  Reply from 23.1.1.3: bytes=56 Sequence=5 ttl=255 time=30 ms

 --- 23.1.1.3 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 30/42/50 ms
```

3：在 R1 上直接 ping R3 将不通，需要配置一条去方向以及回来方向的配认路由

```
[R1-Ethernet0/0/0]ping 23.1.1.3
 PING 23.1.1.3: 56  data bytes, press CTRL_C to break
  Request time out
```

在 R1 与 R3 上分别配置默认路由：

```
[R1]ip route-static 23.1.1.0 255.255.255.0 12.1.1.2
[R3]ip route-static 12.1.1.0 255.255.255.0 23.1.1.2
```

4：再测试连通性：

```
[R1]ping 23.1.1.3
 PING 23.1.1.3: 56  data bytes, press CTRL_C to break
  Reply from 23.1.1.3: bytes=56 Sequence=1 ttl=254 time=80 ms
  Reply from 23.1.1.3: bytes=56 Sequence=2 ttl=254 time=70 ms
  Reply from 23.1.1.3: bytes=56 Sequence=3 ttl=254 time=60 ms
  Reply from 23.1.1.3: bytes=56 Sequence=4 ttl=254 time=40 ms
  Reply from 23.1.1.3: bytes=56 Sequence=5 ttl=254 time=50 ms

 --- 23.1.1.3 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 40/60/80 ms
```

//以上表示成功通信了。

4：还可以为一条静态路由设备优先集，也就是 CISCO 中的管理距离（AD 值），以实现浮动路由等效果。如下：

```
[R1]ip route-static 23.1.1.0 255.255.255.0 12.1.1.2 preference 10
```

请再根据 NA 的实验手册，结合这里的华为的命令，在华为的路由器上面实现浮动静态，选择静态，汇总静态，默认静态，负载均衡的静态等效果。

查看路由表：

```
[R1]display ip routing
Route Flags: R - relay, D - download to fib
------------------------------------------------------------------------------
Routing Tables: Public
         Destinations : 5        Routes : 5

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

        12.1.1.0/24  Direct  0    0           D   12.1.1.1        Ethernet0/0/0
        12.1.1.1/32  Direct  0    0           D   127.0.0.1       Ethernet0/0/0
        23.1.1.0/24  Static  10   0          RD   12.1.1.2        Ethernet0/0/0
       127.0.0.0/8   Direct  0    0           D   127.0.0.1       InLoopBack0
       127.0.0.1/32  Direct  0    0           D   127.0.0.1       InLoopBack0
```

5：当想取消这条默认路由的时候，在前面打 UNDO

```
[R1]undo ip route-static 23.1.1.0 255.255.255.0 12.1.1.2 preference 10
```

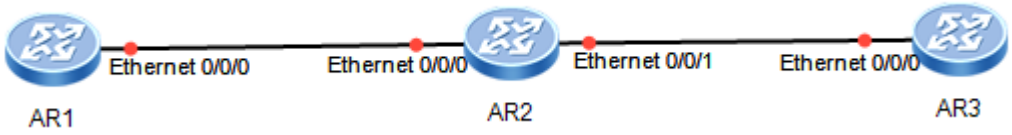
实验第三部分 RIP 协议

实验目的

熟悉在华为设备上配置 RIP 协议，汇总 RIP 路由，做明文的及 MD5 的认证，设置被动接口等

☐ 掌握 RIPv2 的配置方式
☐ 掌握 RIPv2 的汇总方式
☐ 掌握 RIPv2 的认证配置方式
☐ 掌握 RIPv2 中被动接口的配置及效果
☐ 掌握 RIPv2 不 RIPv1 的兼容配置

实验拓扑



实验 3.1 配置 RIP

实验步骤

1：首先请配置好 IP 地址，格式参考 NA 实验手册，或上面静态路由实验中的配置方法。再为每个路由器配置一个环回口：

```
[R1]interface LoopBack 1
[R1-LoopBack2]ip address 11.1.1.1 255.255.255.0
```

```
[R2-LoopBack1]ip address 22.1.1
[R2-LoopBack1]ip address 22.1.1.1 255.255.255.0
```

```
[R3]interface LoopBack 1
[R3-LoopBack1]ip address 33.1.1.1 255.255.255.0
```

2：配置 RIP，设置版本为 V2，关闭自动汇总

```
[R1]rip 1    //进程号可以用来区分本地的 RIP 进程。以限定路由传递的区域
[R1-rip-1]version 2    // 确定 RIP 的版本
[R1-rip-1]undo summary    //关闭自动汇总
[R1-rip-1]network 11.0.0.0    //主类宣告网段
[R1-rip-1]network 12.0.0.0
```

```
[R2]rip 1
[R2-rip-1]version 2
[R2-rip-1]undo summary
[R2-rip-1]network 12.0.0.0
[R2-rip-1]network 22.0.0.0
[R2-rip-1]network 23.0.0.0
```

```
[R3]rip 1
[R3-rip-1]version 2
[R3-rip-1]undo summary
[R3-rip-1]network 23.0.0.0
```

```
[R3-rip-1]network 33.0.0.0
```

3：查看路由的学习情况

```
[R1]display ip routing
Route Flags: R - relay, D - download to fib
------------------------------------------------------------------------------
Routing Tables: Public
         Destinations : 9        Routes : 9

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

        11.1.1.0/24  Direct  0    0           D   11.1.1.1        LoopBack2
        11.1.1.1/32  Direct  0    0           D   127.0.0.1       LoopBack2
        12.1.1.0/24  Direct  0    0           D   12.1.1.1        Ethernet0/0/0
        12.1.1.1/32  Direct  0    0           D   127.0.0.1       Ethernet0/0/0
        22.1.1.0/24  RIP     100  1           D   12.1.1.2        Ethernet0/0/0
        23.1.1.0/24  RIP     100  1           D   12.1.1.2        Ethernet0/0/0
        33.1.1.0/24  RIP     100  2           D   12.1.1.2        Ethernet0/0/0
       127.0.0.0/8   Direct  0    0           D   127.0.0.1       InLoopBack0
       127.0.0.1/32  Direct  0    0           D   127.0.0.1       InLoopBack0
```

4：测试连通性

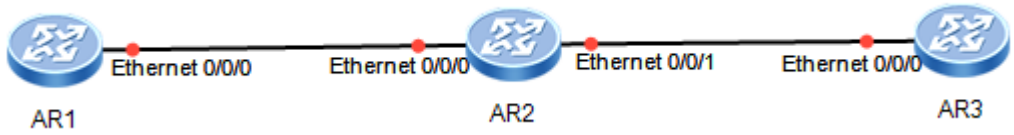
```
[R1]ping 33.1.1.1 ip-forwarding
  PING 33.1.1.1: 56  data bytes, press CTRL_C to break
    Reply from 33.1.1.1: bytes=56 Sequence=1 ttl=254 time=50 ms
    Reply from 33.1.1.1: bytes=56 Sequence=2 ttl=254 time=60 ms
    Reply from 33.1.1.1: bytes=56 Sequence=3 ttl=254 time=30 ms
    Reply from 33.1.1.1: bytes=56 Sequence=4 ttl=254 time=70 ms
    Reply from 33.1.1.1: bytes=56 Sequence=5 ttl=254 time=60 ms

  --- 33.1.1.1 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
round-trip min/avg/max = 30/54/70 ms
```

用环回口 PING。

```
[R1]ping -a 11.1.1.1 33.1.1.1
  PING 33.1.1.1: 56  data bytes, press CTRL_C to break
    Reply from 33.1.1.1: bytes=56 Sequence=1 ttl=254 time=40 ms
    Reply from 33.1.1.1: bytes=56 Sequence=2 ttl=254 time=50 ms
    Reply from 33.1.1.1: bytes=56 Sequence=3 ttl=254 time=60 ms
    Reply from 33.1.1.1: bytes=56 Sequence=4 ttl=254 time=20 ms
    Reply from 33.1.1.1: bytes=56 Sequence=5 ttl=254 time=40 ms

  --- 33.1.1.1 ping statistics ---
    5 packet(s) transmitted
    5 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 20/42/60 ms
```

实验 3.2 配置 RIP 手动汇总

实验拓扑



实验步骤

接着上面的实验

在 R3 上的回环口 loopback1 上配置多个子地址。

```
[R3]interface LoopBack1
[R3-LoopBack1]ip address 33.1.2.1 255.255.255.0 sub
[R3-LoopBack1]ip address 33.1.3.1 255.255.255.0 sub
```

由于是主类宣告，所以直接到 R1 上查看这些 RIP 的路由条目：

```
[R1]display ip routing protocol rip

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface
```

```
22.1.1.0/24  RIP    100  1        D   12.1.1.2        Ethernet0/0/0
23.1.1.0/24  RIP    100  1        D   12.1.1.2        Ethernet0/0/0
33.1.1.0/24  RIP    100  2        D   12.1.1.2        Ethernet0/0/0
33.1.2.0/24  RIP    100  2        D   12.1.1.2        Ethernet0/0/0
33.1.3.0/24  RIP    100  2        D   12.1.1.2        Ethernet0/0/0
```

可以收到这些明细的路由，与 CISCO 一样，在做汇总时都是到路由的出接口，这里在 R2 的出接口做：

```
[R2]interface e0/0/0
[R2-Ethernet0/0/0]rip summary-address 33.1.0.0 255.255.252.0
```

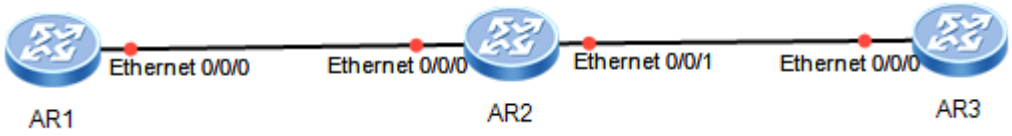
再到 R1 上去查看路由表：

```
[R1]display ip routing protocol rip

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

        22.1.1.0/24  RIP    100  1        D   12.1.1.2        Ethernet0/0/0
        23.1.1.0/24  RIP    100  1        D   12.1.1.2        Ethernet0/0/0
        33.1.0.0/22  RIP    100  2        D   12.1.1.2        Ethernet0/0/0
```

以上表示看到的就是一条汇总的路由。

实验 3.3 配置 RIP 明文与 MD5 认证

实验拓扑



实验步骤

接着上面的实验：

在 R1 与 R2 之间做明文的认证：

```
[R1]interface Ethernet 0/0/0
[R1-Ethernet0/0/0]rip authentication-mode simple yeslab
```

过一会去查看 R1 上的路由表，因为 RIP 的收敛比较快，也可以 shutdownR1 与 R2 之间的接口加快路由表的收敛

```
[R1]interface Ethernet 0/0/0
[R1-Ethernet0/0/0]shutdown

[R1]display ip routing-table protocol rip
```

在 R1 上将看不到任何 RIP 的路由

再去 R2 上查看

```
[R2]display ip routing-table protocol rip

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

        33.1.1.0/24  RIP     100  1           D   23.1.1.3        Ethernet0/0/1
        33.1.2.0/24  RIP     100  1           D   23.1.1.3        Ethernet0/0/1
        33.1.3.0/24  RIP     100  1           D   23.1.1.3        Ethernet0/0/1
```

R2 上也看不到 R1 的路由，

用 debug 命令去观察报错的信息：

```
<R1>terminal debugging
<R1>debugging rip 1 packet
```

```
May 14 2013 19:44:19.970.3-08:00 R1 RIP/7/DBG: 6: 12371: Authentication-mode - Simple: yeslab
May 14 2013 19:44:19.250.8-08:00 R1 RIP/7/DBG: 6: 11386: RIP 1: Authentication failure
```

以上信息表明了是一个简单的认证，并且也带上了认证的密码。提示认证失败。

```
关闭 debug
<R1>undo debugging all
```

接着在 R2 上配置 RIP 的明文认证

```
[R2]interface Ethernet 0/0/0
[R2-Ethernet0/0/0]rip authentication-mode simple yeslab
```

做完认证后再去查看路由表：

```
<R1>display ip routing protocol rip

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

      22.1.1.0/24   RIP     100  1         D   12.1.1.2        Ethernet0/0/0
      23.1.1.0/24   RIP     100  1         D   12.1.1.2        Ethernet0/0/0
      33.1.0.0/22   RIP     100  2         D   12.1.1.2        Ethernet0/0/0
```

路由又重新学到了。

再去 R2 与 R3 之间使用 RIP 的 MD5 认证，

```
[R2]interface Ethernet 0/0/1
[R2-Ethernet0/0/1]rip authentication-mode md5 usual yeslab
```

查看路由表

```
[R2]display ip routing-table protocol rip

      11.1.1.0/24   RIP     100  1         D   12.1.1.1        Ethernet0/0/0
```

看不到 R3 的路由信息了

再用 debug 查看信息：

```
<R2>terminal debugging
<R2>debugging rip 1 packet
```

```
May 14 2013 19:53:13.0.3-08:00 R2 RIP/7/DBG: 6: 12407: Authentication-mode - MD5 Digest:
143ff2b9.11bedfa1.6b6e47a0.9883cd6a

May 14 2013 19:53:12.610.6-08:00 R2 RIP/7/DBG: 6: 11386: RIP 1: Authentication failure
May 14 2013 19:53:12.610.7-08:00 R2 RIP/7/DBG: 6: 1676: RIP 1: Process message failed
```

//以上信息表明是一个 MD5 的认证，并且是由于认证不通过的原因造成的不能学习路由

到 R3 上做 MD5 的认证，

```
[R3]interface Ethernet 0/0/0
[R3-Ethernet0/0/0]rip authentication-mode md5 usual yeslab
```

做完后再去查看路由表：

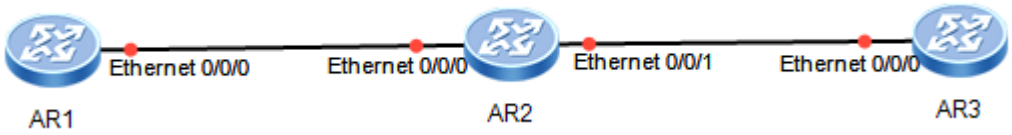
```
[R2]display ip routing-table protocol rip
```

```
Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

        11.1.1.0/24  RIP    100  1           D   12.1.1.1        Ethernet0/0/0
        33.1.1.0/24  RIP    100  1           D   23.1.1.3        Ethernet0/0/1
        33.1.2.0/24  RIP    100  1           D   23.1.1.3        Ethernet0/0/1
        33.1.3.0/24  RIP    100  1           D   23.1.1.3        Ethernet0/0/1
```

//以上表明路由被重新学习到了

实验 3.4 配置 RIP 的抑制接口

实验拓扑



实验步骤

此命令与 CISCO 的被动（passive）接口一样效果

在 R2 上抑制掉连接 R1 的接口，然后去观查现像：

```
[R2]rip 1
[R2-rip-1]silent-interface Ethernet 0/0/0
```

等待一会或 shutdown 再 undo shutdown 接口后观察结果：

```
<R1>display ip routing protocol rip
```

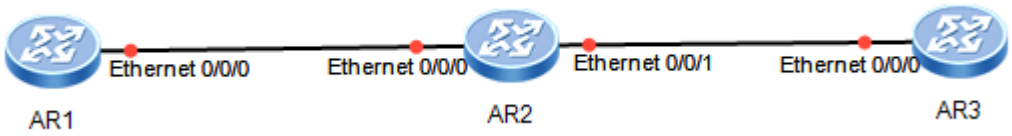
//以上表明 R1 再也收不到 R2 的路由了

```
[R2]display ip routing protocol rip

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

        11.1.1.0/24  RIP    100  1           D   12.1.1.1        Ethernet0/0/0
        33.1.1.0/24  RIP    100  1           D   23.1.1.3        Ethernet0/0/1
        33.1.2.0/24  RIP    100  1           D   23.1.1.3        Ethernet0/0/1
        33.1.3.0/24  RIP    100  1           D   23.1.1.3        Ethernet0/0/1
```

//以上表明 R2 能正常的接收 R1 的路由，也就是只收不发，与 CISCO 的被动接口是一样的效果。

实验 3.5 配置 RIPV1 与 V2 兼容

实验拓扑



实验步骤

虽然 RIP 的 V1 已经不被使用，这里也演示一下 V1 与 V2 的兼容。

修改 R3 的 RIP 的版本 1

```
[R3]rip 1
[R3-rip-1]version 1
```

修改后去 R2 查看路由表：

```
[R2]display ip routing protocol rip

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

      11.1.1.0/24  RIP     100  1           D   12.1.1.1        Ethernet0/0/0
```

//以上表明 R2 再也收不到 R3 的路由了。

```
[R3]display ip routing protocol rip
```

同样 R3 也收不到 R2 的，因为 V1 与 V2 默认是不兼容的。

再去查看 debug 消息：

```
<R3>debugging rip 1 packet
```

```
May 14 2013 20:04:07.150.1-08:00 R3 RIP/7/DBG: 6: 12227: RIP 1: Sending response on interface Ethernet0/0/0
from 23.1.1.3 to 255.255.255.255
```

```
May 14 2013 20:40:03.440.2-08:00 R3 RIP/7/DBG: 6: 12247: Packet: Version 1, Cmd response, Length 24
May 14 2013 20:04:21.290.2-08:00 R3 RIP/7/DBG: 6: 12236: RIP 1: Receive response from 23.1.1.2 on Ethernet0/0/0
May 14 2013 20:04:21.290.3-08:00 R3 RIP/7/DBG: 6: 12247: Packet: Version 2, Cmd response, Length 108
May 14 2013 20:40:53.370.9-08:00 R3 RIP/7/DBG: 6: 2500: RIP 1: Ignoring packet. This version is not configured.
```

//以上表明 RIP 收到的是 Version2,而发送的是 Version1,不兼容。数据包被 ignore

去 R2 上配置兼容命令：

```
[R2]interface Ethernet 0/0/1
[R2-Ethernet0/0/1]rip version 1
```

再去查看路由表：

```
[R2]display ip routing protocol rip

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

     11.1.1.0/24  RIP     100  1           D   12.1.1.1        Ethernet0/0/0
     33.0.0.0/8   RIP     100  1           D   23.1.1.3        Ethernet0/0/1
```

R3 上

```
<R3>display ip routing protocol rip

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

     11.0.0.0/8   RIP     100  2           D   23.1.1.2        Ethernet0/0/0
     12.0.0.0/8   RIP     100  1           D   23.1.1.2        Ethernet0/0/0
     22.0.0.0/8   RIP     100  1           D   23.1.1.2        Ethernet0/0/0
```

//以上表明双方都收到了对方的路由，但是都是汇总的，因为 V1 传递的是主类的路由

实验第四部分 OSPF 协议

实验目的

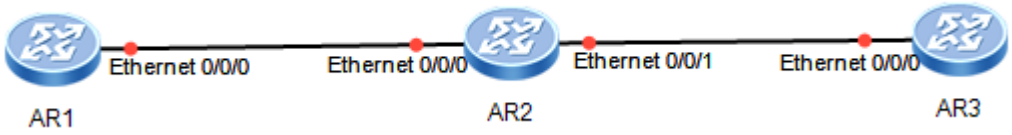
掌握单区域 OSPF 的配置方法

掌握 OSPF 区域认证的配置方法

修 改 OSPF 的 Loopback 的 32 位主机路由

掌握对 OSPF 接口代价值进行修改的方法
掌握 OSPF 中 Silent-interface 的配置方法
掌握使用 Display 查看 OSPF 各种状态的方法
掌握使用 Debug 命令查看 OSPF 邻接关系和进行故障排除的方法

实验拓扑



实验步骤

实验 4.1 单区域的 OSPF

1：首先配置 IP 地址，这里不再演示，请参考上面的实验。

2：配置单区域（area0 ）的 OSPF

```
[R1]ospf 1 router-id 11.1.1.1    //配置 OSPF 的进程号以及 route-ID
[R1-ospf-1]area 0                //进入到区域 0 里面。
[R1-ospf-1-area-0.0.0.0]network 12.1.1.1 0.0.0.0    //精确宣告一个 IP 地址。
[R1-ospf-1-area-0.0.0.0]network 11.1.1.1 0.0.0.0
[R1-ospf-1-area-0.0.0.0]authentication-mode md5 1 cipher yeslab  // 基于区域的 MD5 认证
```

在 R2 上配置

```
[R2]ospf 1 router-id 22.1.1.1
[R2-ospf-1]area 0
[R2-ospf-1-area-0.0.0.0]network 12.1.1.2 0.0.0.0
[R2-ospf-1-area-0.0.0.0]network 23.1.1.2 0.0.0.0
[R2-ospf-1-area-0.0.0.0]network 22.1.1.1 0.0.0.0
[R2-ospf-1-area-0.0.0.0]authentication-mode md5 1 cipher yeslab
```

在 R3 上去配置

```
[R3]ospf 1 router-id 33.1.1.1
[R3-ospf-1-area-0.0.0.0]network 33.1.1.1 0.0.0.0
[R3-ospf-1-area-0.0.0.0]network 23.1.1.3 0.0.0.0
```

```
[R3-ospf-1-area-0.0.0.0]authentication-mode md5 1 cipher yeslab
```

都配置好以后去查看路由表：

```
 [R1]display ip routing protocol ospf

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

      22.1.1.1/32  OSPF    10   1           D   12.1.1.2        Ethernet0/0/0
      23.1.1.0/24  OSPF    10   2           D   12.1.1.2        Ethernet0/0/0
      33.1.1.1/32  OSPF    10   2           D   12.1.1.2        Ethernet0/0/0
```

在 R2 上查看：

```
[R2]display ip routing protocol ospf

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

      11.1.1.1/32  OSPF    10   1           D   12.1.1.1        Ethernet0/0/0
      33.1.1.1/32  OSPF    10   1           D   23.1.1.3        Ethernet0/0/1
```

在 R3 上查看：

```
[R3]display ip routing protocol ospf

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

      11.1.1.1/32  OSPF    10   2           D   23.1.1.2        Ethernet0/0/0
      12.1.1.0/24  OSPF    10   2           D   23.1.1.2        Ethernet0/0/0
      22.1.1.1/32  OSPF    10   1           D   23.1.1.2        Ethernet0/0/0
```

使用 display ospf brief 命令查看 OSPF 运行的基本信息。

```
[R1]display ospf brief
————————省略一部分显示————————
 Area: 0.0.0.0          (MPLS TE not enabled)
 Authtype: MD5   Area flag: Normal     //区域 0 的认证类型
 SPF scheduled Count: 8
 ExChange/Loading Neighbors: 0
 Router ID conflict state: Normal
 Interface: 12.1.1.1 (Ethernet0/0/0)   //接口的的开销，状态，接口网络类型
 Cost: 1        State: DR        Type: Broadcast    MTU: 1500
```

```
Priority: 1
Designated Router: 12.1.1.1
Backup Designated Router: 12.1.1.2
Timers: Hello 10 , Dead 40 , Poll  120 , Retransmit 5 , Transmit Delay 1

Interface: 11.1.1.1 (LoopBack1)
Cost: 0       State: P-2-P     Type: P2P       MTU: 1500
Timers: Hello 10 , Dead 40 , Poll  120 , Retransmit 5 , Transmit Delay 1
```

再去查看邻居关系：

```
[R1]display ospf peer brief
        OSPF Process 1 with Router ID 11.1.1.1
                Peer Statistic Information
 ----------------------------------------------------------------------------
 Area Id          Interface                  Neighbor id      State
 0.0.0.0          Ethernet0/0/0                22.1.1.1       Full
 ----------------------------------------------------------------------------
```

//以上表示 R1 上能看到 R2 这个邻居，有邻居的 route-ID, 状态，本地接口和区域等

去查看 LSDB 表项：

```
[R1]display ospf lsdb
        OSPF Process 1 with Router ID 11.1.1.1
                Link State Database
                        Area: 0.0.0.0
 Type      LinkState ID    AdvRouter        Age  Len   Sequence   Metric
 Router    33.1.1.1        33.1.1.1         668  48    80000004   0
 Router    22.1.1.1        22.1.1.1         668  60    80000009   1
 Router    11.1.1.1        11.1.1.1         726  48    80000006   1
 Network   23.1.1.2        22.1.1.1         668  32    80000001   0
 Network   12.1.1.1        11.1.1.1         726  32    80000001   0
```

//以上输出表示能看到一类的和 2 类的 LSA，与 CISCO 没有区别。

再去查看一条详细的内容：

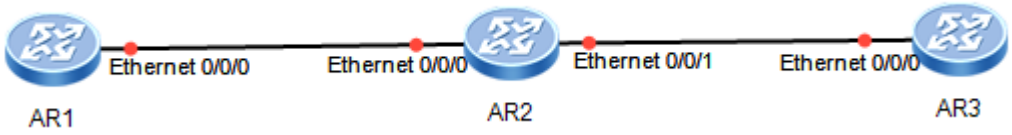
```
[R1]display ospf lsdb router 33.1.1.1
        OSPF Process 1 with Router ID 11.1.1.1
                Area: 0.0.0.0
            Link State Database
 Type      : Router
```

```
Ls id     : 33.1.1.1
Adv rtr   : 33.1.1.1
Ls age    : 694
Len       : 48
Options   :  E
seq#      : 80000004
chksum    : 0xc9d2
Link count: 2
 * Link ID: 33.1.1.1
   Data   : 255.255.255.255
   Link Type: StubNet
   Metric : 0
   Priority : Medium
 * Link ID: 23.1.1.2
   Data   : 23.1.1.3
   Link Type: TransNet
   Metric : 1
```

//以上输出表示 R3 通告了二项信息，一个是回环回，一个是连接 R2 的接口。具体理解请看 OSPF 的讲解视频。

实验 4.2 控制 DR/BDR 的举，观察邻居状态变迁

实验拓扑



实验步骤

接着上面的实验：

先去查看 R1 与 R2 之间谁是 DR

```
[R1]display ospf peer
        OSPF Process 1 with Router ID 11.1.1.1
              Neighbors
 Area 0.0.0.0 interface 12.1.1.1(Ethernet0/0/0)'s neighbors
 Router ID: 22.1.1.1         Address: 12.1.1.2
   State: Full  Mode:Nbr is  Master  Priority: 1
   DR: 12.1.1.1  BDR: 12.1.1.2  MTU: 0
   Dead timer due in 37  sec
```

```
  Retrans timer interval: 5
  Neighbor is up for 00:28:53
  Authentication Sequence: [ 2472]
```

以上表示 R1 为 DR。但在这里每个人得到的实验结果会不一样，因为 DR/BDR 的选举是非抢占的。

接下来修改 R1 的接口优先集。

```
[R1-Ethernet0/0/0]ospf dr-priority 10
```

然后 shutdown 二边的接口，打开 debug，观察 OSPF 的邻居建立过程。

```
<R1>debugging ospf 1 event
[R1]info-center enable
```

Undo shutdown 接口，观察以下信息

```
May 16 2013 17:43:39-08:00 R1 %%01PHY/1/PHY(l)[0]:    Ethernet0/0/0: change status to up
May 16 2013 17:43:39-08:00 R1 %%01IFNET/4/LINK_STATE(l)[1]:The line protocol IP on the interface Ethernet0/0/0 has entered the
UP state.
May 16 2013 17:43:39-08:00 R1 %%01OSPF/4/NBR_CHANGE_E(l)[2]:Neighbor changes event: neighbor status changed. (ProcessId=1,
NeighborAddress=12.1.1.2, NeighborEvent=HelloReceived, NeighborPreviousState=Down, NeighborCurrentState=Init)
May 16 2013 17:43:48-08:00 R1 DS/4/DATASYNC_CFGCHANGE:OID 1.3.6.1.4.1.2011.5.25.191.3.1 configurations have been changed.
The current change number is 23, the change loop count is 0, and the maximum number of records is 4095.
May 16 2013 17:43:58-08:00 R1 %%01OSPF/4/NBR_CHANGE_E(l)[3]:Neighbor changes event: neighbor status changed. (ProcessId=1,
NeighborAddress=12.1.1.2, NeighborEvent=2WayReceived, NeighborPreviousState=Init, NeighborCurrentState=2Way)
May 16 2013 17:44:20-08:00 R1 %%01OSPF/4/NBR_CHANGE_E(l)[4]:Neighbor changes event: neighbor status changed. (ProcessId=1,
NeighborAddress=12.1.1.2, NeighborEvent=AdjOk?, NeighborPreviousState=2Way, NeighborCurrentState=ExStart)
May 16 2013 17:44:25-08:00 R1 %%01OSPF/4/NBR_CHANGE_E(l)[5]:Neighbor changes event: neighbor status changed. (ProcessId=1,
NeighborAddress=12.1.1.2, NeighborEvent=NegotiationDone, NeighborPreviousState=ExStart, NeighborCurrentState=Exchange)
May 16 2013 17:44:25-08:00 R1 %%01OSPF/4/NBR_CHANGE_E(l)[6]:Neighbor changes event: neighbor status changed. (ProcessId=1,
NeighborAddress=12.1.1.2, NeighborEvent=ExchangeDone, NeighborPreviousState=Exchange, NeighborCurrentState=Loading)
May 16 2013 17:44:25-08:00 R1 %%01OSPF/4/NBR_CHANGE_E(l)[7]:Neighbor changes event: neighbor status changed. (ProcessId=1,
NeighborAddress=12.1.1.2, NeighborEvent=LoadingDone, NeighborPreviousState=Loading, NeighborCurrentState=Full)
```

再去 R2 上查看邻居信息：

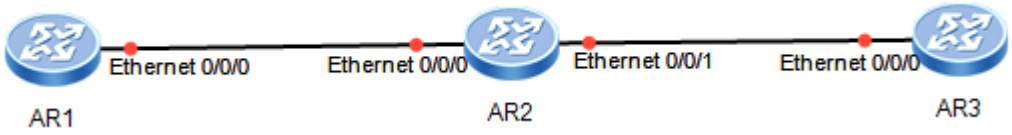
```
[R2]display ospf peer
        OSPF Process 1 with Router ID 22.1.1.1
                Neighbors
 Area 0.0.0.0 interface 12.1.1.2(Ethernet0/0/0)'s neighbors
 Router ID: 11.1.1.1          Address: 12.1.1.1
   State: Full  Mode:Nbr is  Slave  Priority: 10
   DR: 12.1.1.1  BDR: 12.1.1.2  MTU: 0
   Dead timer due in 37  sec
   Retrans timer interval: 5
   Neighbor is up for 00:02:36
   Authentication Sequence: [ 8265]
```

//以上表示 R1 的优先集已经被修改为 10 了。

实验 4.3 配置接口的网络类型

接着上面的实验：

实验拓扑



实验步骤

先去查看 R1 的路由表，将会发现 R1 上收到的 R2,R3 的环回口路由都是 32 位的主机路由：

```
[R1]display ip routing protocol ospf
Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface
      22.1.1.1/32   OSPF    10   1           D   12.1.1.2        Ethernet0/0/0
      23.1.1.0/24   OSPF    10   2           D   12.1.1.2        Ethernet0/0/0
      33.1.1.1/32   OSPF    10   2           D   12.1.1.2        Ethernet0/0/0
```

再去查看 R2 上环回口的网络类型。

```
[R2]display ospf interface LoopBack 1
        OSPF Process 1 with Router ID 22.1.1.1
                Interfaces
 Interface: 22.1.1.1 (LoopBack1)
 Cost: 0       State: P-2-P     Type: P2P       MTU: 1500
 Timers: Hello 10 , Dead 40 , Poll  120 , Retransmit 5 , Transmit Delay 1
```

//以上输出表示类型是 P2P,这些与 CISCO 是不一样的。CISCO 的回环口是 loopback，但是结局都是一样的，都是似送 32 位主机路由。

修改接口的网络类型如下：

```
[R2]interface LoopBack 1
[R2-LoopBack1]ospf network-type broadcast
```

修改后再去 R1 查看结果：

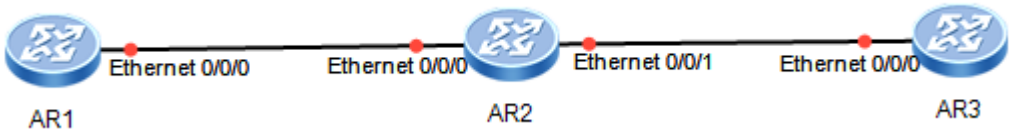
```
[R1]display ip routing protocol ospf

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

      22.1.1.0/24  OSPF    10   1           D   12.1.1.2        Ethernet0/0/0
      23.1.1.0/24  OSPF    10   2           D   12.1.1.2        Ethernet0/0/0
      33.1.1.1/32  OSPF    10   2           D   12.1.1.2        Ethernet0/0/0
```

//以上表示 R1 收到 R2 的路由是 24 位的了。

实验 4.4 修改 OSPF 接口的 COST 值

接着上面的实验：

实验拓扑



实验步骤

通以上面的实验输出可以发现,LOOPBACK 口的 COST 值为 0,接下来把 R2 的环回口 COST 值修改为 99,再去 R1 看结果。

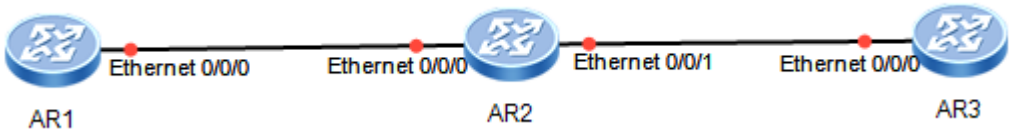
```
[R2]interface loop 1
[R2-LoopBack1]ospf cost 99
```

```
[R1]display ip routing protocol ospf
Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface
      22.1.1.0/24  OSPF    10   100         D   12.1.1.2        Ethernet0/0/0
      23.1.1.0/24  OSPF    10   2           D   12.1.1.2        Ethernet0/0/0
      33.1.1.1/32  OSPF    10   2           D   12.1.1.2        Ethernet0/0/0
```

实验 4.5 配置 OSPF 的被动接口

接着上面的实验：

实验拓扑



实验步骤

OSPF 的被动接口与 RIP 是不一样的，不再是只收不发，而是不收也不发任何 OSPF 的报文，会导至邻居关系都建立不起来，当然也学不到任何路由：

```
[R1]ospf 1
[R1-ospf-1]silent-interface Ethernet 0/0/0
[R1]display ospf peer
```

取消 E0/0/0 口的被动接口，去回环回：

```
[R1-ospf-1]undo silent-interface Ethernet 0/0/0
[R1-ospf-1]silent-interface LoopBack 1
```

再去 R2 上去查看结果：

```
[R2]display ip routing protocol ospf
Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

     11.1.1.1/32  OSPF    10   1           D   12.1.1.1        Ethernet0/0/0
     33.1.1.1/32  OSPF    10   1           D   23.1.1.3        Ethernet0/0/1
```

//以上表示 R2 正常收到的一个被 **silent** 掉的回环口的路由，也就是说被动接口只是不建立邻居关系，但是这个接口所在的路由依然会通告出去。

实验 4.6 配置多区域 的 OSPF

实验目的

掌握多区域 OSPF 的配置方法

☐ 掌握 OSPF 区域之间路由汇总的配置方法

☐ 掌握 OSPF 参考带宽的配置方法

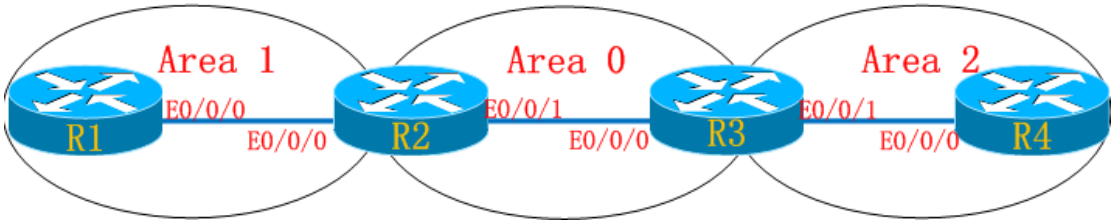
☐ 掌握 OSPF 引入外部路由的配置方法

☐ 掌握 OSPF 引入的外部路由时进行路由汇总的方法

☐ 掌握向 OSPF 导入缺省路由的方法

☐ 掌握对 OSPF 中各类路由的管理距离的修改方法

实验拓扑



实验步骤

1：先配 置 IP 地址 。

2：如图所示配置多区域的 OSPF

```
[R1]ospf 1 router-id 11.1.1.1
[R1-ospf-1]area 1
[R1-ospf-1-area-0.0.0.1]network 11.1.1.1 0.0.0.0
[R1-ospf-1-area-0.0.0.1]network 12.1.1.1 0.0.0.0
```

在 R2 上：

```
[R2]ospf 1  router-id 22.1.1.1
[R2-ospf-1]area 0
[R2-ospf-1-area-0.0.0.0]network 23.1.1.2  0.0.0.0
[R2-ospf-1-area-0.0.0.0]quit
[R2-ospf-1]area 1
[R2-ospf-1-area-0.0.0.1]network 12.1.1.2   0.0.0.0
[R2-ospf-1-area-0.0.0.1]network 22.1.1.1  0.0.0.0
```

//在不同的区域里宣告相关的网段即可。

```
[R3]ospf  1 router-id 33.1.1.1
[R3-ospf-1]area 0
```

```
[R3-ospf-1-area-0.0.0.0]network 23.1.1.3 0.0.0.0
[R3-ospf-1-area-0.0.0.0]quit
[R3-ospf-1]area 2
[R3-ospf-1-area-0.0.0.2]network 34.1.1.3 0.0.0.0
[R3-ospf-1-area-0.0.0.2]network 33.1.1.1  0.0.0.0
```

```
[R4]ospf 1  router-id 44.1.1.1
[R4-ospf-1]area 2
[R4-ospf-1-area-0.0.0.2]network 34.1.1.4 0.0.0.0
[R4-ospf-1-area-0.0.0.2]network 44.1.1.1 0.0.0.0
```

配置好以后去查看 R1 的路由表：

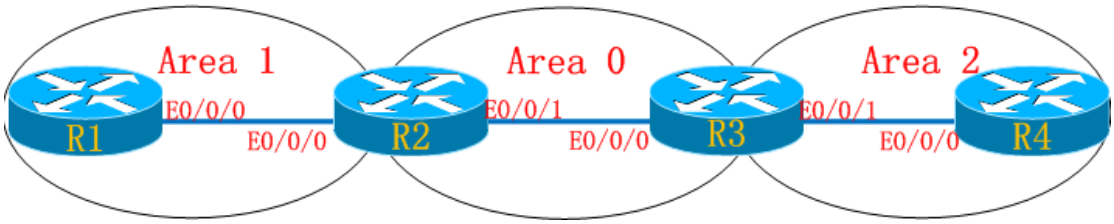
```
[R1]display ip routing protocol ospf
Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

        22.1.1.1/32  OSPF    10   1           D   12.1.1.2        Ethernet0/0/0
        23.1.1.0/24  OSPF    10   2           D   12.1.1.2        Ethernet0/0/0
        33.1.1.1/32  OSPF    10   2           D   12.1.1.2        Ethernet0/0/0
        34.1.1.0/24  OSPF    10   3           D   12.1.1.2        Ethernet0/0/0
        44.1.1.1/32  OSPF    10   3           D   12.1.1.2        Ethernet0/0/0
```

//以上表示 R1 学到到了 所有区域的路由，这里与 CISCO 不同的是，区域间的路由与区域内的路的的优先集是一样的。

实验 4.7 修改 OSPF 的参考带宽值。

接着上面的实验：

实验拓扑



实验步骤

在实际网络我们可能使用了千兆甚至万兆以太网。但是由于 OSPF 的默认参考带宽值为 100Mbps，并且接口代价值仅为整数，所以 OSPF 无法在带宽上区分百兆以太网和千兆及以上的以太网。

与 CISCO 一样，有一条命令可以去修改 OSPF 的参考带宽：

```
[R1-ospf-1]bandwidth-reference 1000
Info: Reference bandwidth is changed. Please ensure that the reference bandwidth that is configured for all the routers are the same.
```

//同时提醒修改应该是在所有路由器上进行的。

在 R1 上查看路由表，Cost 值已经发生了发化。

[R1-ospf-1]display ip routing protocol ospf

```
Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface
      22.1.1.1/32   OSPF    10   10          D   12.1.1.2        Ethernet0/0/0
      23.1.1.0/24   OSPF    10   11          D   12.1.1.2        Ethernet0/0/0
      33.1.1.1/32   OSPF    10   11          D   12.1.1.2        Ethernet0/0/0
      34.1.1.0/24   OSPF    10   12          D   12.1.1.2        Ethernet0/0/0
      44.1.1.1/32   OSPF    10   12          D   12.1.1.2        Ethernet0/0/0
```

//以上输出表示路由的 COST 值加大了。不再是用 10 的 8 次方去除以带宽，而是用 10 的 9 次方去除，所以值加大了 10 倍。当要适应更高带宽的网络时，就把参考带宽加大，但是一定要在所有路由器上都修改。否则会出现次优。

再去查看一下多区域的 OSPF 用的到 LSA。查看 OSPF 的 LSDB 表。

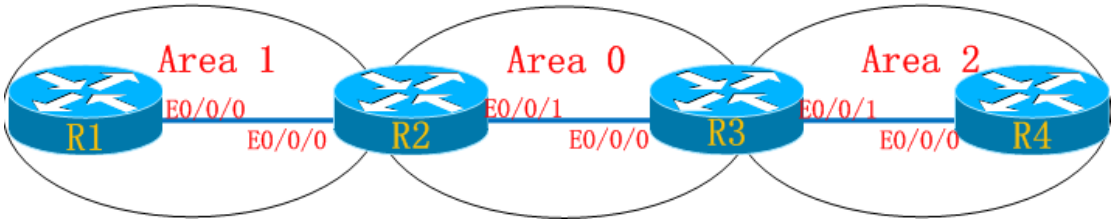
```
[R1]display ospf lsdb
                    Area: 0.0.0.1
 Type      LinkState ID    AdvRouter       Age  Len   Sequence   Metric
 Router    22.1.1.1        22.1.1.1        639  48    80000004   1
 Router    11.1.1.1        11.1.1.1        640  48    80000006   1
 Network   12.1.1.1        11.1.1.1        640  32    80000002   0
 Sum-Net   44.1.1.1        22.1.1.1        548  28    80000001   2
 Sum-Net   23.1.1.0        22.1.1.1        651  28    80000001   1
 Sum-Net   34.1.1.0        22.1.1.1        599  28    80000001   2
 Sum-Net   33.1.1.1        22.1.1.1        586  28    80000001   1
```

//由于是多区域的 OSPF，我们将看到三类的 LSA，sum-net 的 LSA。用来描述区域间的路由。

实验 4.8 引入外部路由（重分布）

紧接着上面的实验：

实验拓扑



实验步骤

在 R4 上引入直连的路由条目

首先在 R4 上创建几条新的直连路由：

```
[R4]interface LoopBack 0
[R4-LoopBack0]ip address 44.1.0.1 255.255.255.0 sub
[R4-LoopBack0]ip address 44.1.2.1 255.255.255.0 sub
[R4-LoopBack0]ip address 44.1.3.1 255.255.255.0 sub
```

然后把这些直连的路由重分布进 OSPF：

```
[R4-ospf-1]import-route direct
```

同时我们注意到，用此命令可以做与任何协议之间的重分布：

```
[R4-ospf-1]import-route ?
  bgp     Border Gateway Protocol (BGP) routes
  direct  Connected routes
  isis    Intermediate System to Intermediate System (IS-IS) routes
  limit   Limit the number of routes imported into OSPF
  ospf    Open Shortest Path First (OSPF) routes
  rip     Routing Information Protocol (RIP) routes
  static  Static routes
  unr     User Network Routes
```

再到 R1 上去查看路由条目：

[R1-ospf-1]display ip routing protocol ospf

```
Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface
      22.1.1.1/32  OSPF    10   10          D   12.1.1.2        Ethernet0/0/0
```

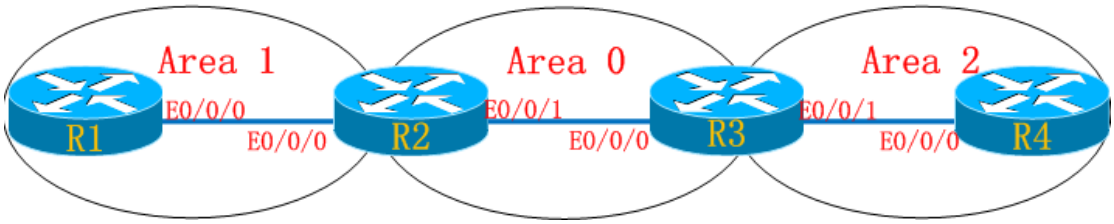
```
23.1.1.0/24  OSPF    10   11          D   12.1.1.2        Ethernet0/0/0
33.1.1.1/32  OSPF    10   11          D   12.1.1.2        Ethernet0/0/0
34.1.1.0/24  OSPF    10   12          D   12.1.1.2        Ethernet0/0/0
44.1.0.0/24  O_ASE   150  1           D   12.1.1.2        Ethernet0/0/0
44.1.1.0/24  O_ASE   150  1           D   12.1.1.2        Ethernet0/0/0
44.1.1.1/32  OSPF    10   12          D   12.1.1.2        Ethernet0/0/0
44.1.2.0/24  O_ASE   150  1           D   12.1.1.2        Ethernet0/0/0
44.1.3.0/24  O_ASE   150  1           D   12.1.1.2        Ethernet0/0/0
```

//上面红色部分路由表示的就是重分布进去的路由，优先集为 150.这些 CISCO 是不一样的。

实验 4.9 汇总 OSPF 路由

紧接着上面的实验：

实验拓扑



实验步骤

先做区域间的汇总，所以先在 R3 上多产生几条直连的路由，然后宣告进 OSPF。

```
[R3]interface LoopBack 0
[R3-LoopBack0]ip address 33.1.2.1 255.255.255.0 sub
[R3-LoopBack0]ip address 33.1.0.1 255.255.255.0 sub
[R3-LoopBack0]ip address 33.1.3.1 255.255.255.0 sub
[R3-LoopBack0]ospf enable 1 area 2   //华为也支持接口下的宣告。
```

现在去汇总这 4 条路由，由于这四条路由都是被宣告进了 OSPF 的区域 2.所以汇总也要进到 area2 里面进行：

```
[R3]ospf 1
[R3-ospf-1]area 2
[R3-ospf-1-area-0.0.0.2]abr-summary 33.1.0.0 255.255.252.0
```

然后到 R1 上去看现象：

```
[R1]display ip routing protocol ospf
Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface
        22.1.1.1/32  OSPF    10   10          D   12.1.1.2        Ethernet0/0/0
        23.1.1.0/24  OSPF    10   11          D   12.1.1.2        Ethernet0/0/0
        33.1.0.0/22  OSPF    10   11          D   12.1.1.2        Ethernet0/0/0
        34.1.1.0/24  OSPF    10   12          D   12.1.1.2        Ethernet0/0/0
        44.1.0.0/24  O_ASE   150  1           D   12.1.1.2        Ethernet0/0/0
        44.1.1.0/24  O_ASE   150  1           D   12.1.1.2        Ethernet0/0/0
        44.1.1.1/32  OSPF    10   12          D   12.1.1.2        Ethernet0/0/0
        44.1.2.0/24  O_ASE   150  1           D   12.1.1.2        Ethernet0/0/0
        44.1.3.0/24  O_ASE   150  1           D   12.1.1.2        Ethernet0/0/0
```

//以上表示成功汇总成了一条 22 位的路由。

再去汇总 R4 引于的几条外部路由，这要在 ASBR 上做：

```
 [R4-ospf-1]asbr-summary 44.1.0.0 255.255.252.0
```

再到 R1 查看结果：

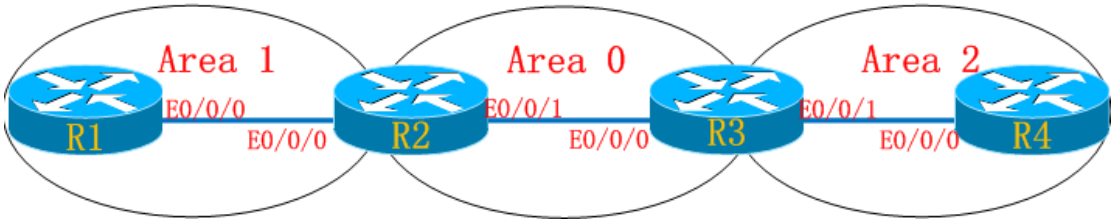
```
[R1]display ip routing protocol ospf
Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface
        22.1.1.1/32  OSPF    10   10          D   12.1.1.2        Ethernet0/0/0
        23.1.1.0/24  OSPF    10   11          D   12.1.1.2        Ethernet0/0/0
        33.1.0.0/22  OSPF    10   11          D   12.1.1.2        Ethernet0/0/0
        34.1.1.0/24  OSPF    10   12          D   12.1.1.2        Ethernet0/0/0
        44.1.0.0/22  O_ASE   150  2           D   12.1.1.2        Ethernet0/0/0
        44.1.1.1/32  OSPF    10   12          D   12.1.1.2        Ethernet0/0/0
```

//成功汇总了外部的四条路由。

实验 4.10 在 OSPF 中产生一条默认路由。

紧接着上面的实验：

实验拓扑



实验步骤

OSPF 可以很方便的引入一条默认路由：

在 R4 上：

```
[R4]ospf 1
[R4-ospf-1]default-route-advertise always
```

然后到 R1 上查看结果：

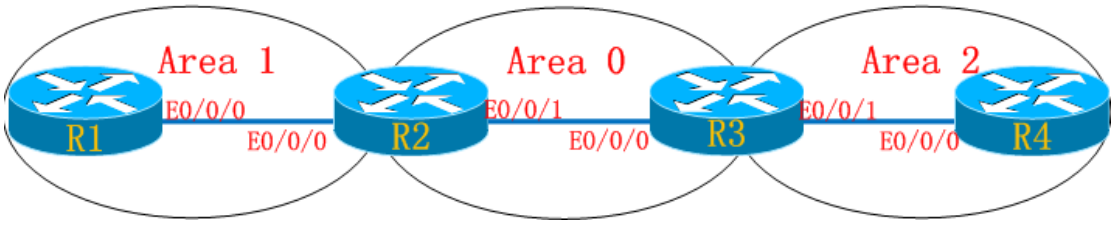
```
[R1]display ip routing protocol ospf
Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface
        0.0.0.0/0   O_ASE   150  1           D   12.1.1.2        Ethernet0/0/0
       22.1.1.1/32  OSPF    10   10          D   12.1.1.2        Ethernet0/0/0
       23.1.1.0/24  OSPF    10   11          D   12.1.1.2        Ethernet0/0/0
       33.1.0.0/22  OSPF    10   11          D   12.1.1.2        Ethernet0/0/0
       34.1.1.0/24  OSPF    10   12          D   12.1.1.2        Ethernet0/0/0
       44.1.0.0/22  O_ASE   150  2           D   12.1.1.2        Ethernet0/0/0
       44.1.1.1/32  OSPF    10   12          D   12.1.1.2        Ethernet0/0/0
```

//以上表示收到了一条默认路由，也是由 5 类的 LSA 产生的。

实验 4.11 修改 OSPF 路由的优先集

紧接着上面的实验：

实验拓扑



实验步骤

在华为设备上，默认的内部的 OSPF 路由优先集为 10，外部的为 150，这个值是可以修改的：

```
[R1]ospf 1
[R1-ospf-1]preference 20       //修改内部的为 20
[R1-ospf-1]preference ase 50   //修改外部的为 50
```

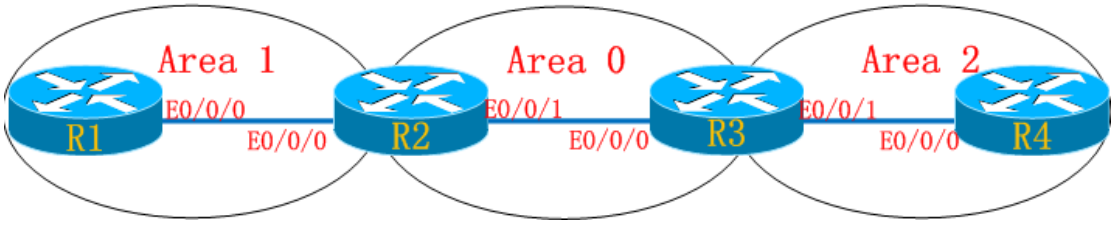
修改后再去查看路由表：

```
[R1]display ip routing-table protocol ospf
Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface
        0.0.0.0/0   O_ASE    50   1           D   12.1.1.2        Ethernet0/0/0
       22.1.1.1/32  OSPF    20   10          D   12.1.1.2        Ethernet0/0/0
       23.1.1.0/24  OSPF    20   11          D   12.1.1.2        Ethernet0/0/0
       33.1.0.0/22  OSPF    20   11          D   12.1.1.2        Ethernet0/0/0
       34.1.1.0/24  OSPF    20   12          D   12.1.1.2        Ethernet0/0/0
       44.1.0.0/22  O_ASE   50   2           D   12.1.1.2        Ethernet0/0/0
       44.1.1.1/32  OSPF    20   12          D   12.1.1.2        Ethernet0/0/0
```

实验 4.12 配置 OSPF 的 stub 区域

紧接着上面的实验：

实验拓扑



实验步骤

在 R1 与 R2 上配置之间的 area1 配置 stup 区域。以减少外部路由的接收，

```
[R1]ospf 1
[R1-ospf-1]area 1
[R1-ospf-1-area-0.0.0.1]stub
```

```
[R2]ospf 1
[R2-ospf-1]area 1
[R2-ospf-1-area-0.0.0.1]stub
```

配置完成后去 R1 上查看结果：

```
[R1]display ip routing protocol ospf
Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface
        0.0.0.0/0   OSPF    20   11          D   12.1.1.2        Ethernet0/0/0
       22.1.1.1/32  OSPF    20   10          D   12.1.1.2        Ethernet0/0/0
       23.1.1.0/24  OSPF    20   11          D   12.1.1.2        Ethernet0/0/0
       33.1.0.0/22  OSPF    20   11          D   12.1.1.2        Ethernet0/0/0
       34.1.1.0/24  OSPF    20   12          D   12.1.1.2        Ethernet0/0/0
       44.1.1.1/32  OSPF    20   12          D   12.1.1.2        Ethernet0/0/0
```

//如上所示，再也没有了外部的路由条目。但依然有很多区域间的路由。并且默认路由也由 R2 产生了一条区域内的默认。

在 ABR 上配置完全末节区域。以最大限度的减少 R1 的路由表：

```
[R2-ospf-1]area 1
[R2-ospf-1-area-0.0.0.1]stub no-summary
```

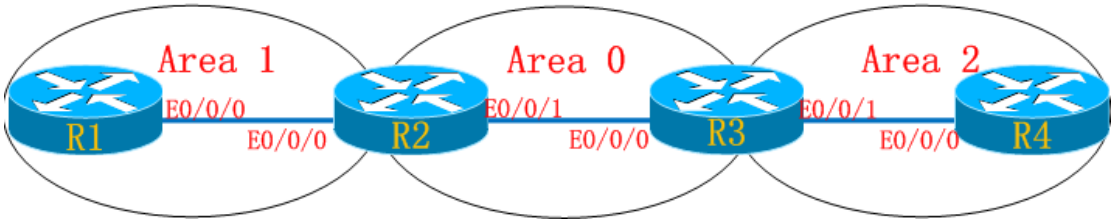
再到 R1 上查看路由表：

```
[R1]display ip routing protocol ospf
Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface
        0.0.0.0/0       OSPF    20   11          D   12.1.1.2        Ethernet0/0/0
       22.1.1.1/32      OSPF    20   10          D   12.1.1.2        Ethernet0/0/0
```

实验 4.13 配置 AREA2 为 NSSA 区域

紧接着上面的实验：

实验拓扑



实验步骤

Nssa 区域保留了 STUP 区域的特点，不接收外部的路由，但是在本区域却可以引入外部的路由，用 7 类的 LSA 来表示：

在 R3 上配置

```
[R3]ospf 1
[R3-ospf-1]area 2
[R3-ospf-1-area-0.0.0.2]nssa
```

```
[R4]ospf 1
[R4-ospf-1]area 2
[R4-ospf-1-area-0.0.0.2]nssa
```

配置完成后到 R4 上查看路由表，此时的 R4 上将看不到任何区域外的路由，

```
[R4]display ip routing-table protocol ospf
Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface
        0.0.0.0/0   O_NSSA  150  1           D   34.1.1.3        Ethernet0/0/0
       11.1.1.1/32  OSPF    10   3           D   34.1.1.3        Ethernet0/0/0
       12.1.1.0/24  OSPF    10   3           D   34.1.1.3        Ethernet0/0/0
       22.1.1.1/32  OSPF    10   2           D   34.1.1.3        Ethernet0/0/0
       23.1.1.0/24  OSPF    10   2           D   34.1.1.3        Ethernet0/0/0
       33.1.0.1/32  OSPF    10   1           D   34.1.1.3        Ethernet0/0/0
       33.1.1.1/32  OSPF    10   1           D   34.1.1.3        Ethernet0/0/0
       33.1.2.1/32  OSPF    10   1           D   34.1.1.3        Ethernet0/0/0
       33.1.3.1/32  OSPF    10   1           D   34.1.1.3        Ethernet0/0/0
```

//但是依然有很多区域间路由。

在 ABR 上配置完全次末节区域：

```
[R3-ospf-1-area-0.0.0.2]nssa no-summary
```

然后再到 R4 看结果，发现没有区域间的路由了，只有区域内的路由了。

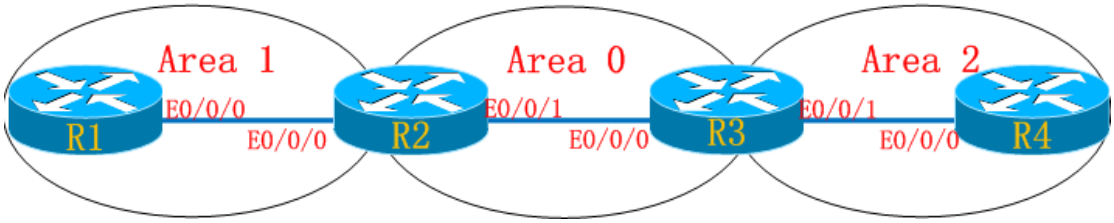
```
[R4]display ip routing-table protocol ospf
```

Destination/Mask	Proto	Pre	Cost	Flags	NextHop	Interface
0.0.0.0/0	OSPF	10	2	D	34.1.1.3	Ethernet0/0/0
33.1.0.1/32	OSPF	10	1	D	34.1.1.3	Ethernet0/0/0
33.1.1.1/32	OSPF	10	1	D	34.1.1.3	Ethernet0/0/0
33.1.2.1/32	OSPF	10	1	D	34.1.1.3	Ethernet0/0/0
33.1.3.1/32	OSPF	10	1	D	34.1.1.3	Ethernet0/0/0

实验 4.14 配置区域间的路由过滤

紧接着上面的实验：

实验拓扑



实验步骤

在 R2 上过滤掉 R1 的环回口路由，让 R3 不能再学到。

先写 ACL 去拒绝掉不想要的路由，放过其它的所有。

```
[R2]acl number 2000                    //配置基本标准的 ACL
[R2-acl-basic-2000]rule deny source 11.1.1.0 0.0.0.255   //拒绝掉不想要的路由
[R2-acl-basic-2000]rule permit
```

```
[R2]ospf 1
[R2-ospf-1]area 1
[R2-ospf-1-area-0.0.0.1]filter 2000 export
```

//应用到 area 1 的 out 方向。

再去查看 R3 的路由表：

```
[R3]display ip routing-table protocol ospf
```

```
     12.1.1.0/24  OSPF    10   2          D   23.1.1.2        Ethernet0/0/0
     22.1.1.1/32  OSPF    10   1          D   23.1.1.2        Ethernet0/0/0
   44.1.0.0/22  O_NSSA  150  2          D   34.1.1.4        Ethernet0/0/1
     44.1.1.1/32  OSPF    10   1          D   34.1.1.4        Ethernet0/0/1
```

//以上表示 R3 再也没有收到 11.1.1.0/24 那条路由了。

在 R1 上产生另一条路由，看是否会被拒绝：

```
[R1]interface LoopBack 1
[R1-LoopBack1]ip address 11.1.2.1 255.255.255.0
[R1-LoopBack1]ospf enable 1 area 1
```

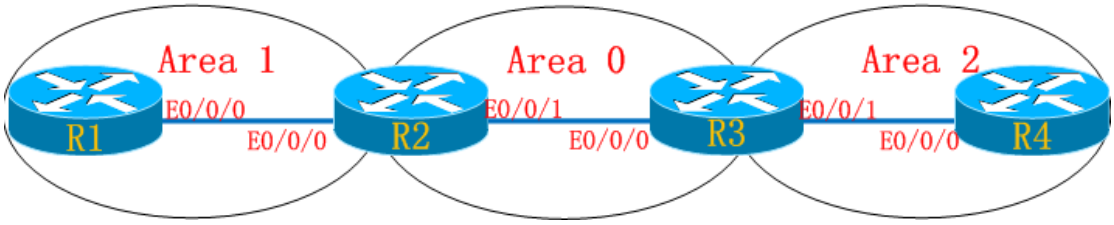
```
[R3]display ip routing-table protocol ospf
     11.1.2.1/32  OSPF    10   2          D   23.1.1.2        Ethernet0/0/0
     12.1.1.0/24  OSPF    10   2          D   23.1.1.2        Ethernet0/0/0
     22.1.1.1/32  OSPF    10   1          D   23.1.1.2        Ethernet0/0/0
   44.1.0.0/22  O_NSSA  150  2          D   34.1.1.4        Ethernet0/0/1
     44.1.1.1/32  OSPF    10   1          D   34.1.1.4        Ethernet0/0/1
```

//以上表示能够正常学到，刚做的过滤只过滤掉了指定的路由。

实验 4.15 OSPF 故障排除

紧接着上面的实验：

实验拓扑



实验目的

☐ 掌握 OSPF 中区域号码不匹配故障的排除方法

☐ 掌握 OSPF 中掩码不匹配故障的排除方法

☐ 掌握 OSPF 中 Hello 时间不匹配的故障排除方法

☐ 掌握 OSPF 中 Router-id 冲突故障的排除方法

☐ 掌握 OSPF 认证相关故障的排除方法

实验步骤

错误 1：route-ID 一致的错误。

在 R1 与 R2 之间模拟一些故障，以观察排除过程

在 R2 上修改 route-ID,和 R1 的一致：

```
[R2]ospf 1 router-id 11.1.1.1
Info: The configuration succeeded. You need to restart the OSPF process to validate the new router ID.
```

必须要重启 OSPF 的进程新的 route-ID 才能生效：

```
<R2>reset ospf process
Warning: The OSPF process will be reset. Continue? [Y/N]:y
```

到 R1 上查看邻居：

```
[R1]display ospf peer brief
```

//将看不到任何邻居，因为 route-ID 是一致的，

查看错误信息：

```
<R1>reset ospf counters    //此命令是清楚以前显示的次数
```

```
<R1>display ospf error     //此命令以查看 OSPF 的错误信息
General packet errors:
 0    : IP: received my own packet        0    : Bad packet
 0    : Bad version                       0    : Bad checksum
 0    : Bad area id                       0    : Drop on unnumbered interface
 0    : Bad virtual link                  0    : Bad authentication type
 0    : Bad authentication key            0    : Packet too small
 0    : Packet size > ip length           0    : Transmit error
 0    : Interface down                    0    : Unknown neighbor
 0    : Bad net segment                   0    : Extern option mismatch
 1    : Router id confusion
```

//以上红色部分指定的出错的原因是 router-ID 配置错误。错误的数字还会不停的涨大。

还可以使用 debug 命令来排除这个错误

```
[R1]info-center enable
<R1>terminal debugging
<R1>debugging ospf event
May 17 2013 19:17:03-08:00
R1 %%01OSPF/4/CONFLICT_ROUTERID_INTF(l)[0]:OSPF Router id conflict is detected on interface.
(ProcessId=1, RouterId=11.1.1.1, AreaId=0.0.0.1, InterfaceName=Ethernet0/0/0,  IpAddr=12.1.1.1,
PacketSrcIp=12.1.1.2)
```

//以上信息的红色部分可以很清楚的表明出错的接口与邻居的 IP 地址。这样就能很快的排除这个错误。

去 R2 修正这个错误：

```
[R2]ospf 1 router-id 22.1.1.1
<R2>reset ospf process
Warning: The OSPF process will be reset. Continue? [Y/N]:y
```

再到 R1 上查看邻居，发现邻居已经成功建立

```
[R1]display ospf peer brief
        OSPF Process 1 with Router ID 11.1.1.1
                Peer Statistic Information
 ----------------------------------------------------------------------------
 Area Id          Interface                     Neighbor id      State
 0.0.0.1          Ethernet0/0/0                   22.1.1.1       Full
 ----------------------------------------------------------------------------
```

错误 2：区域 ID 配置错误

在 R2 上去制造这个错误：

```
[R2]ospf 1
[R2-ospf-1]area 1
[R2-ospf-1-area-0.0.0.1]undo network 12.1.1.2 0.0.0.0
```

//先去掉 R2 正确的宣告。

然后再把连接 R1 的接口宣告到一个错误的区域 0 里面

```
[R2-ospf-1-area-0.0.0.1]quit
[R2-ospf-1]area 0
[R2-ospf-1-area-0.0.0.0]network 12.1.1.2 0.0.0.0
```

然后到 R1 查看邻居关系：

```
[R1]display ospf peer brief
```

//将看不到任何的邻居关系。

再去查看错误的信息：

```
[R1]display ospf error
```

```
General packet errors:
 0    : IP: received my own packet      0    : Bad packet
 0    : Bad version                     0    : Bad checksum
 9    : Bad area id                     0    : Drop on unnumbered interface
 0    : Bad virtual link                0    : Bad authentication type
 0    : Bad authentication key          0    : Packet too small
 0    : Packet size > ip length         0    : Transmit error
 0    : Interface down                  0    : Unknown neighbor
 0    : Bad net segment                 0    : Extern option mismatch
 0    : Router id confusion
```

// 如红色部分所示，指出了错误为区域 ID。

去 R2 上恢复配置

```
[R2-ospf-1-area-0.0.0.0]undo network 12.1.1.2 0.0.0.0
[R2-ospf-1-area-0.0.0.0]quit
[R2-ospf-1]area 1
[R2-ospf-1-area-0.0.0.1]network 12.1.1.2 0.0.0.0
[R2-ospf-1-area-0.0.0.1]
```

错误 3：掩码配置错误。

依然是去 R2 上模拟这个错误,配置一个错误的掩码

```
[R2]interface Eth0/0/0
[R2-Ethernet0/0/0]ip address 12.1.1.2 255.255.255.128
```

到 R1 上去发现这个错误提醒信息

```
[R1]display ospf peer brief
[R1]display ospf error
General packet errors:
 0    : IP: received my own packet      16   : Bad packet
 0    : Bad version                     0    : Bad checksum
 0    : Bad area id                     0    : Drop on unnumbered interface
 0    : Bad virtual link                0    : Bad authentication type
 0    : Bad authentication key          0    : Packet too small
 0    : Packet size > ip length         0    : Transmit error
```

```
0    : Interface down                0    : Unknown neighbor
0    : Bad net segment               0    : Extern option mismatch
0    : Router id confusion
HELLO packet errors:
16   : Netmask mismatch              0    : Hello timer mismatch
0    : Dead timer mismatch           0    : Virtual neighbor unknown
0    : NBMA neighbor unknown         0    : Invalid Source Address
```

//红色部分指明了出错的位置，

通过 debug 信息可以看到更详细的信息：

```
[R1]info-center enable
<R1>debugging ospf packet hello
May 17 2013 19:39:20.750.2-08:00 R1 RM/6/RMDEBUG:  Source Address: 12.1.1.2
May 17 2013 19:39:20.750.3-08:00 R1 RM/6/RMDEBUG:  Destination Address: 224.0.0.5
May 17 2013 19:39:20.750.4-08:00 R1 RM/6/RMDEBUG:  Ver# 2, Type: 1 (Hello)
May 17 2013 19:39:20.750.5-08:00 R1 RM/6/RMDEBUG:  Length: 44, Router: 22.1.1.1
May 17 2013 19:39:20.750.6-08:00 R1 RM/6/RMDEBUG:  Area: 0.0.0.1, Chksum: da18
May 17 2013 19:39:20.750.7-08:00 R1 RM/6/RMDEBUG:  AuType: 00
May 17 2013 19:39:20.750.8-08:00 R1 RM/6/RMDEBUG:  Key(ascii): 0 0 0 0 0 0 0 0
May 17 2013 19:39:20.750.9-08:00 R1 RM/6/RMDEBUG:  Net Mask: 255.255.255.128
May 17 2013 19:39:20.750.10-08:00 R1 RM/6/RMDEBUG:  Hello Int: 10, Option:
May 17 2013 19:39:20.750.11-08:00 R1 RM/6/RMDEBUG:  Rtr Priority: 1, Dead Int: 40
May 17 2013 19:39:20.750.12-08:00 R1 RM/6/RMDEBUG:  DR: 12.1.1.2
May 17 2013 19:39:20.750.13-08:00 R1 RM/6/RMDEBUG:  BDR: 0.0.0.0
May 17 2013 19:39:20.750.14-08:00 R1 RM/6/RMDEBUG:  # Attached Neighbors: 0
May 17 2013 19:39:20.750.15-08:00 R1 RM/6/RMDEBUG:
```

再去 R2 恢复这个错误。

```
[R2]interface Ethernet 0/0/0
[R2-Ethernet0/0/0]ip address 12.1.1.2 255.255.255.0
```

错误 4：Hello 时间不一致

再去 R2 上修改 hello 时间：

```
[R2]interface Ethernet 0/0/0
[R2-Ethernet0/0/0]ospf
[R2-Ethernet0/0/0]ospf timer hello 5
```

去 R1 查看错误信息：

```
<R1>display ospf error
General packet errors:
 0    : IP: received my own packet        54   : Bad packet
 0    : Bad version                       0    : Bad checksum
 0    : Bad area id                       0    : Drop on unnumbered interface
 0    : Bad virtual link                  0    : Bad authentication type
 0    : Bad authentication key            0    : Packet too small
 0    : Packet size > ip length           0    : Transmit error
 0    : Interface down                    0    : Unknown neighbor
 0    : Bad net segment                   0    : Extern option mismatch
 0    : Router id confusion

HELLO packet errors:
 49   : Netmask mismatch                  5    : Hello timer mismatch
 0    : Dead timer mismatch               0    : Virtual neighbor unknown
 0    : NBMA neighbor unknown             0    : Invalid Source Address
```

//以上红色部分指示出错信息是 hello 时间不一致。

去 R2 上恢复正确配置：

```
[R2-Ethernet0/0/0]undo ospf timer hello
```

错误 5：认证错误。

在 R1 与 R2 之间做基于接口的认证，一边使用明文的，一边使用 MD5 的认证方法。

```
[R1]interface Ethernet 0/0/0
[R1-Ethernet0/0/0]ospf authentication-mode simple plain yeslab
```

```
[R2]interface Ethernet 0/0/0
[R2-Ethernet0/0/0]ospf authentication-mode md5 1 plain yeslab
```

到 R1 上查看错误信息：

```
[R1]display ospf error
General packet errors:
```

```
0    : IP: received my own packet      301   : Bad packet
0    : Bad version                     0     : Bad checksum
0    : Bad area id                     0     : Drop on unnumbered interface
0    : Bad virtual link                14    : Bad authentication type
0    : Bad authentication key          0     : Packet too small
0    : Packet size > ip length         0     : Transmit error
0    : Interface down                  0     : Unknown neighbor
0    : Bad net segment                 0     : Extern option mismatch
0    : Router id confusion
```

//如上红色部分所示，报一个认证类型的错误。

去 R1 上修改正确的认证类型：

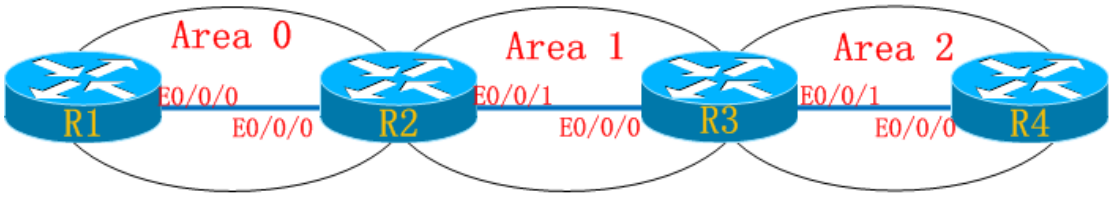
```
[R1-Ethernet0/0/0]ospf authentication-mode md5 1 plain yeslab
```

实验 4.16 OSPF 的虚链路

实验目的

利用虚链路把远离的普通区域连接到骨干区域

实验拓扑



实验步骤

如图所示配置好 IP 地址及 OSPF 的区域。

如图所示。Area 2 没有与 area 0 相连接。也就是说 area2 是没有 ABR。R3 虽然连接了 AREA1，也连接了 AREA2。但是由于不是 ABR，并不会把区域间学到的路由引入到另一个区域。做实现验证：

其中 R2 的环回口在 AREA0，R3 的环回口在 AREA1

```
[R1]ospf 1 router-id 11.1.1.1
```

```
[R1-ospf-1]area 0
[R1-ospf-1-area-0.0.0.0]network 12.1.1.1 0.0.0.0
[R1-ospf-1-area-0.0.0.0]network 11.1.1.1 0.0.0.0
```

```
[R2]ospf 1
[R2-ospf-1]area 0
[R2-ospf-1-area-0.0.0.0]network 12.1.1.2 0.0.0.0
[R2-ospf-1-area-0.0.0.0]network 22.1.1.1 0.0.0.0
[R2-ospf-1]area 1
[R2-ospf-1-area-0.0.0.1]network 23.1.1.2 0.0.0.0
```

```
[R3]ospf 1 router-id 33.1.1.1
[R3-ospf-1]area 1
[R3-ospf-1-area-0.0.0.1]network 23.1.1.3 0.0.0.0
[R3-ospf-1-area-0.0.0.1]network 33.1.1.1 0.0.0.0
[R3-ospf-1-area-0.0.0.1]quit
[R3-ospf-1]area 2
[R3-ospf-1-area-0.0.0.2]network 34.1.1.3 0.0.0.0
```

```
[R4]ospf 1 router-id 44.1.1.1
[R4-ospf-1]area 2
[R4-ospf-1-area-0.0.0.2]network 34.1.1.4 0.0.0.0
[R4-ospf-1-area-0.0.0.2]network 44.1.1.1 0.0.0.0
```

所有的接口都宣告到相关的区域后，去查看邻居关系：

```
[R2]display ospf peer brief
        OSPF Process 1 with Router ID 22.1.1.1
                Peer Statistic Information
 ----------------------------------------------------------------------------
 Area Id          Interface                     Neighbor id      State
 0.0.0.0          Ethernet0/0/0                   11.1.1.1       Full
 0.0.0.1          Ethernet0/0/1                   33.1.1.1       Full
 ----------------------------------------------------------------------------
```

再到 R3 上查看：

```
[R3]display ospf peer brief
        OSPF Process 1 with Router ID 33.1.1.1
```

```
            Peer Statistic Information
--------------------------------------------------------------------------
Area Id        Interface                Neighbor id     State
0.0.0.1        Ethernet0/0/0            22.1.1.1        Full
0.0.0.2        Ethernet0/0/1            44.1.1.1        Full
--------------------------------------------------------------------------
```

//以上输出表示所有的路由器邻居关系都已经正常起来了。

再分别到 R1 与 R4 上查看路由表：

```
[R1]display ip routing-table protocol ospf
Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

      22.1.1.1/32   OSPF    10   1           D   12.1.1.2        Ethernet0/0/0
      23.1.1.0/24   OSPF    10   2           D   12.1.1.2        Ethernet0/0/0
      33.1.1.1/32   OSPF    10   2           D   12.1.1.2        Ethernet0/0/0
```

//以上输出表示 R1 并没有学到 R4 的路由，

```
[R4]display ip routing-table protocol ospf
```

R4 上查看时将会看到 R4 学不到任何路由器的路由。

如上所述，由于 R3 不是 ABR，他不会把从 area2 学到的 R4 的路由告诉给 area1，也不会把从 area1 学到的路由告诉 area2.

这路需要在 area1 的 R2 与 R3 之间建立一条虚链路。

在 R2 上：

```
[R2]ospf 1
[R2-ospf-1]area 1
[R2-ospf-1-area-0.0.0.1]vlink-peer 33.1.1.1
```

//注意，命令后指定的不是一个 IP 地址，而是对方路由器的 route-ID,所以，一个路由器的 route-ID,不能随意的变动。

再到 R3 上：

```
[R3]ospf 1
[R3-ospf-1]area 1
[R3-ospf-1-area-0.0.0.1]vlink-peer 22.1.1.1
```

查看验证这条虚链路是否 UP：

```
[R2]display ospf vlink
        OSPF Process 1 with Router ID 22.1.1.1
              Virtual Links
 Virtual-link Neighbor-id  -> 33.1.1.1, Neighbor-State: Full
 Interface: 23.1.1.2 (Ethernet0/0/1)
 Cost: 1  State: P-2-P  Type: Virtual
 Transit Area: 0.0.0.1
 Timers: Hello 10 , Dead 40 , Retransmit 5 , Transmit Delay 1
```

//以上表示这条虚链路已经建立成功了。

再去查看 R1 和 R4 的路由表：

```
[R1]display ip routing-table protocol ospf
Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface
      22.1.1.1/32  OSPF    10   1           D   12.1.1.2        Ethernet0/0/0
      23.1.1.0/24  OSPF    10   2           D   12.1.1.2        Ethernet0/0/0
      33.1.1.1/32  OSPF    10   2           D   12.1.1.2        Ethernet0/0/0
      34.1.1.0/24  OSPF    10   3           D   12.1.1.2        Ethernet0/0/0
      44.1.1.1/32  OSPF    10   3           D   12.1.1.2        Ethernet0/0/0
```

再到 R4 上：

```
[R4]display ip routing-table protocol ospf
Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface
      11.1.1.1/32  OSPF    10   3           D   34.1.1.3        Ethernet0/0/0
      12.1.1.0/24  OSPF    10   3           D   34.1.1.3        Ethernet0/0/0
      22.1.1.1/32  OSPF    10   2           D   34.1.1.3        Ethernet0/0/0
      23.1.1.0/24  OSPF    10   2           D   34.1.1.3        Ethernet0/0/0
      33.1.1.1/32  OSPF    10   1           D   34.1.1.3        Ethernet0/0/0
```

再到 R4 查看 LSDB：

```
[R4]display ospf lsdb
Type      LinkState ID    AdvRouter       Age  Len   Sequence   Metric
 Router     44.1.1.1       44.1.1.1       1642  48   80000005      1
 Router     33.1.1.1       33.1.1.1        118  36   80000006      1
 Network    34.1.1.4       44.1.1.1       1642  32   80000003      0
 Sum-Net    12.1.1.0       33.1.1.1        118  28   80000001      2
 Sum-Net    23.1.1.0       33.1.1.1        118  28   80000001      1
```

```
Sum-Net   11.1.1.1      33.1.1.1        118  28   80000001     2
Sum-Net   22.1.1.1      33.1.1.1        118  28   80000001     1
Sum-Net   33.1.1.1      33.1.1.1        118  28   80000001     0
```

//以上表示 R3 做为区域 2 的 ABR，把所有区域间的路由通告给了 R4.

在不连续的区域 0 的时候，也会出现同样的问题，同样需要虚链路来解决，这里就不再演示了，同学们自己去设计验证。

虚链路的排障

还需要注意的一点是，虚链路也是属于区域 0 的，如果做了区域 0 的认证，需要在起虚链路的路由器上也基于区域 0 做认证。

在 R1 与 R2 上做基于区域 0 的认证：

```
[R1]ospf 1
[R1-ospf-1]area 0
[R1-ospf-1-area-0.0.0.0]authentication-mode md5 1 plain yeslab
```

```
[R2]ospf 1
[R2-ospf-1]area 0
[R2-ospf-1-area-0.0.0.0]authentication-mode md5 1 plain yeslab
```

做完后，在 R3 上不做区域 0 的认证，在 R3 上查看这条虚链路：

```
[R3]display ospf vlink
        OSPF Process 1 with Router ID 33.1.1.1
                Virtual Links
 Virtual-link Neighbor-id  -> 22.1.1.1, Neighbor-State: Down
 Interface: 23.1.1.3 (Ethernet0/0/0)
 Cost: 1  State: P-2-P  Type: Virtual
 Transit Area: 0.0.0.1
 Timers: Hello 10 , Dead 40 , Retransmit 5 , Transmit Delay 1
```

//以上显示这条虚链路是 down 掉的。

用上面学过的命令去查看错误点：

```
[R3]display ospf error
```

```
        OSPF Process 1 with Router ID 33.1.1.1
                OSPF error statistics
General packet errors:
 0    : IP: received my own packet      15   : Bad packet
 0    : Bad version                      0    : Bad checksum
 0    : Bad area id                      0    : Drop on unnumbered interface
 0    : Bad virtual link                 15   : Bad authentication type
 0    : Bad authentication key           0    : Packet too small
 0    : Packet size > ip length          0    : Transmit error
 0    : Interface down                   0    : Unknown neighbor
 0    : Bad net segment                  0    : Extern option mismatch
```

//以上输出表示认证类型出错.

到 R3 上去做基于区域的认证：

```
[R3]ospf 1
[R3-ospf-1]area 0
[R3-ospf-1-area-0.0.0.0]authentication-mode md5 1 plain yeslab
```

再去查看虚链路是否起来：

```
[R3]display ospf v
[R3]display ospf vlink
        OSPF Process 1 with Router ID 33.1.1.1
                Virtual Links
Virtual-link Neighbor-id  -> 22.1.1.1, Neighbor-State: Full
Interface: 23.1.1.3 (Ethernet0/0/0)
 Cost: 1  State: P-2-P  Type: Virtual
 Transit Area: 0.0.0.1
 Timers: Hello 10 , Dead 40 , Retransmit 5 , Transmit Delay 1
```

//认证通过后，虚链路再次起来了。

实验 4.17 在 FR 环境中配置 OSPF

实验目的

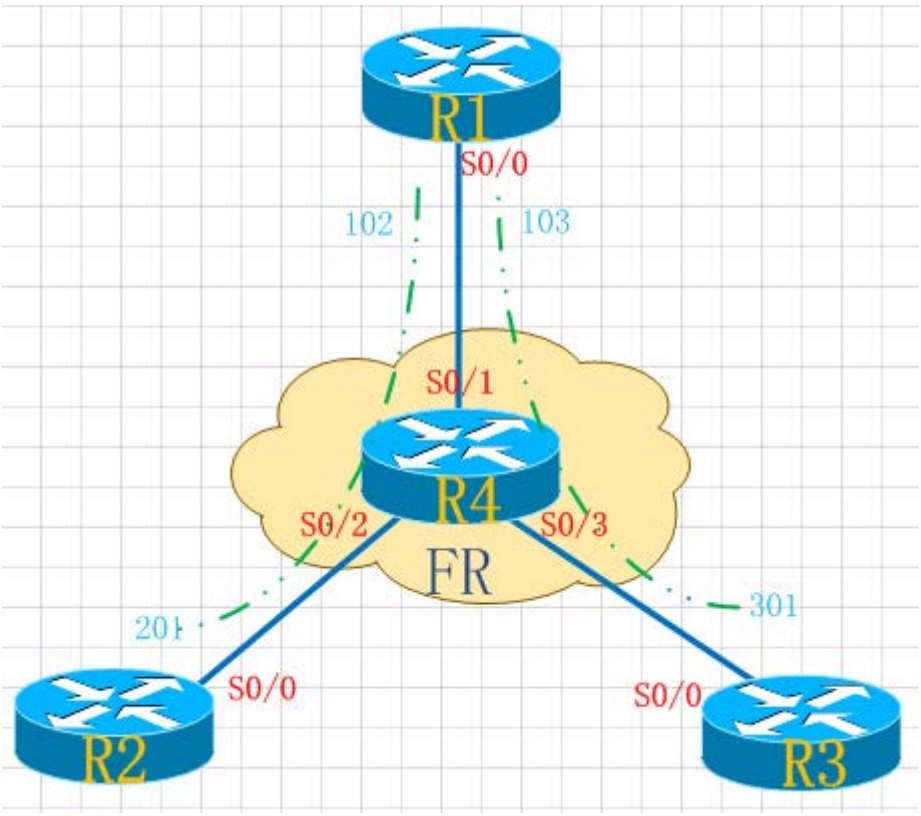
掌握在 NBMA 网络中手工配置 OSPF 邻居的方法

☐ 掌握在 NBMA 网络中配置 OSPF 的特点

☐ 掌握在 FR 中设置 P2MP 型网络类型

☐ 掌握在 FR 中建立子接口设置 P2P 型网络类型

实验拓扑



实验步骤

用模拟器里的帧中继交换机配置如图所示的 DLCI 号对应关系。

```
在 DTE 端，也就是 R1,R2,R3 上做如下配置：
[R1]interface Serial 0/0/0
[R1-Serial0/0/0]link-protocol fr ietf     //在接口封装帧中继
Warning: The encapsulation protocol of the link will be changed.
Continue? [Y/N]:y
[R1-Serial0/0/0]undo fr inarp             //关闭反向的 ARP
[R1-Serial0/0/0]ip add 123.1.1.1 255.255.255.0
[R1-Serial0/0/0]fr map ip 123.1.1.2 102 broadcast     //手动配置映射
[R1-Serial0/0/0]fr map ip 123.1.1.3 103 broadcast
```

```
[R1-Serial0/0/0]undo shutdown
```

```
[R2]interface Serial 0/0/0
[R2-Serial0/0/0]link-protocol fr ietf
[R2-Serial0/0/0]undo fr inarp
[R2-Serial0/0/0]ip add 123.1.1.2 255.255.255.0
[R2-Serial0/0/0]fr map ip 123.1.1.1 201 broadcast
```

```
[R3]interface Serial 0/0/0
[R3-Serial0/0/0]link-protocol fr ietf
[R3-Serial0/0/0]undo fr inarp
[R3-Serial0/0/0]ip add 123.1.1.3 255.255.255.0
[R3-Serial0/0/0]fr map ip 123.1.1.1 301 broadcast
```

配置后验证直连是否正常通信：

```
[R1-Serial0/0/0]ping 123.1.1.3
  PING 123.1.1.3: 56  data bytes, press CTRL_C to break
    Reply from 123.1.1.3: bytes=56 Sequence=1 ttl=255 time=20 ms
    Reply from 123.1.1.3: bytes=56 Sequence=2 ttl=255 time=10 ms
    Reply from 123.1.1.3: bytes=56 Sequence=3 ttl=255 time=50 ms
    Reply from 123.1.1.3: bytes=56 Sequence=4 ttl=255 time=20 ms
    Reply from 123.1.1.3: bytes=56 Sequence=5 ttl=255 time=30 ms
```

```
[R1-Serial0/0/0]ping 123.1.1.2
  PING 123.1.1.2: 56  data bytes, press CTRL_C to break
    Reply from 123.1.1.2: bytes=56 Sequence=1 ttl=255 time=60 ms
    Reply from 123.1.1.2: bytes=56 Sequence=2 ttl=255 time=30 ms
    Reply from 123.1.1.2: bytes=56 Sequence=3 ttl=255 time=60 ms
    Reply from 123.1.1.2: bytes=56 Sequence=4 ttl=255 time=20 ms
    Reply from 123.1.1.2: bytes=56 Sequence=5 ttl=255 time=40 ms
```

分别查看帧中继映射表项：

```
[R1]display fr map-info
Map Statistics for interface Serial0/0/0 (DTE)
  DLCI = 102, IP 123.1.1.2, Serial0/0/0
    create time = 2013/05/20 19:52:16, status = ACTIVE
    encapsulation = ietf, vlink = 1, broadcast
```

```
  DLCI = 103, IP 123.1.1.3, Serial0/0/0
    create time = 2013/05/20 19:52:21, status = ACTIVE
encapsulation = ietf, vlink = 2, broadcast
```

```
[R2]display fr map-info
Map Statistics for interface Serial0/0/0 (DTE)
  DLCI = 201, IP 123.1.1.1, Serial0/0/0
    create time = 2013/05/20 19:53:24, status = ACTIVE
encapsulation = ietf, vlink = 1, broadcast
```

```
[R3]display fr map-info
Map Statistics for interface Serial0/0/0 (DTE)
  DLCI = 301, IP 123.1.1.1, Serial0/0/0
    create time = 2013/05/20 19:54:07, status = ACTIVE
encapsulation = ietf, vlink = 1, broadcast
```

接着配置 OSPF：

```
[R1]ospf 1 router-id 11.1.1.1
[R1-ospf-1]area 0
[R1-ospf-1-area-0.0.0.0]network 11.1.1.1 0.0.0.0
[R1-ospf-1-area-0.0.0.0]network 123.1.1.1 0.0.0.0
```

```
[R2]ospf 1 router-id 22.1.1.1
[R2-ospf-1]area 0
[R2-ospf-1-area-0.0.0.0]network 22.1.1.1 0.0.0.0
[R2-ospf-1-area-0.0.0.0]network 123.1.1.2 0.0.0.0
```

```
[R3]ospf 1 router-id 33.1.1.1
[R3-ospf-1]area 0
[R3-ospf-1-area-0.0.0.0]network 33.1.1.1 0.0.0.0
[R3-ospf-1-area-0.0.0.0]network 123.1.1.3 0.0.0.0
```

去查看接口的 OSPF 网络类型：

```
[R2]display ospf interface Serial 0/0/0
        OSPF Process 1 with Router ID 22.1.1.1
                Interfaces
 Interface: 123.1.1.2 (Serial0/0/0)
 Cost: 1562    State: DROther    Type: NBMA      MTU: 1500
 Priority: 0
 Designated Router: 123.1.1.1
```

```
Backup Designated Router: 0.0.0.0
Timers: Hello 30 , Dead 120 , Poll  120 , Retransmit 5 , Transmit Delay 1
```

由于默认的网络类型是 NBMA，所以需要手动指邻居，并且必须二边都指才可以：

```
[R1]ospf 1
[R1-ospf-1]peer 123.1.1.2
[R1-ospf-1]peer 123.1.1.3
```

```
[R2-ospf-1]peer 123.1.1.1
```

```
[R3-ospf-1]peer 123.1.1.1
```

查看邻居表的时候将看到 NBMA 所特有的一种状态：

```
[R1]display ospf peer brief

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

 Area Id          Interface                Neighbor id      State
 0.0.0.0          Serial0/0/0              0.0.0.0          Attempt
 0.0.0.0          Serial0/0/0              0.0.0.0          Attempt
```

在 CISCO 的实验手册及课程中都着重讲过，在 NBMA 的网络中，必须控制 DR 在 hub 点上，所以我们修改 R2,R3 的优先集为 0

```
[R2]interface Serial 0/0/0
[R2-Serial0/0/0]ospf dr-priority 0
```

```
[R3]interface Serial 0/0/0
[R3-Serial0/0/0]ospf dr-priority 0
```

再去查看 R1 的邻居表：

```
[R1]display ospf peer brief
        OSPF Process 1 with Router ID 11.1.1.1
                Peer Statistic Information
 ----------------------------------------------------------------------------

 Area Id          Interface                Neighbor id      State
 0.0.0.0          Serial0/0/0              22.1.1.1         Full
 0.0.0.0          Serial0/0/0              33.1.1.1         Full
```

//以上表示邻居都正常起来了

再分别去查看路由表：

```
[R1]display ip routing-table protocol ospf
```

```
Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface
      22.1.1.1/32  OSPF    10   1562      D   123.1.1.2       Serial0/0/0
      33.1.1.1/32  OSPF    10   1562      D   123.1.1.3       Serial0/0/0
```

```
[R2]display ip routing protocol ospf
Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface
      11.1.1.1/32  OSPF    10   1562      D   123.1.1.1       Serial0/0/0
      33.1.1.1/32  OSPF    10   1562      D   123.1.1.3       Serial0/0/0
```

```
[R3]display ip routing protocol ospf
Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface
      11.1.1.1/32  OSPF    10   1562      D   123.1.1.1       Serial0/0/0
      22.1.1.1/32  OSPF    10   1562      D   123.1.1.2       Serial0/0/0
```

路由都正常学到了。再去 R2 上去 pingR3：

```
[R2]ping -a 22.1.1.1 33.1.1.1
  PING 33.1.1.1: 56  data bytes, press CTRL_C to break
    Request time out
    Request time out
    Request time out
    Request time out
   Request time out
```

//结果是有路由，但是不通，原因很简单，没有二层的映射关系，分别到 R2 与 R3 上去做二层的映射关系：

```
[R2-Serial0/0/0]fr map ip 123.1.1.3 201 broadcast
[R3-Serial0/0/0]fr map ip 123.1.1.2 301 broadcast
```

再测试：

```
[R2]ping -a 22.1.1.1 33.1.1.1
  PING 33.1.1.1: 56  data bytes, press CTRL_C to break
    Reply from 33.1.1.1: bytes=56 Sequence=1 ttl=254 time=100 ms
    Reply from 33.1.1.1: bytes=56 Sequence=2 ttl=254 time=30 ms
    Reply from 33.1.1.1: bytes=56 Sequence=3 ttl=254 time=70 ms
    Reply from 33.1.1.1: bytes=56 Sequence=4 ttl=254 time=60 ms
    Reply from 33.1.1.1: bytes=56 Sequence=5 ttl=254 time=40 ms
```

实验 4.18 修改 OSPF 的网络类型为点到多点:

分别到 R1,R2,R3 上修改接口的 OSPF 网络类型：

```
[R1]interface Serial 0/0/0
[R1-Serial0/0/0]ospf network-type p2mp
```

```
[R2]interface Serial 0/0/0
[R2-Serial0/0/0]ospf network-type p2mp
```

```
[R3]interface Serial 0/0/0
[R3-Serial0/0/0]ospf network-type p2mp
```

再去查看邻居表：

```
[R1]display ospf peer
        OSPF Process 1 with Router ID 11.1.1.1
              Neighbors
 Area 0.0.0.0 interface 123.1.1.1(Serial0/0/0)'s neighbors
 Router ID: 22.1.1.1          Address: 123.1.1.2
   State: Full  Mode:Nbr is  Master  Priority: 0
   DR: None   BDR: None   MTU: 0
   Dead timer due in 112 sec
   Retrans timer interval: 5
   Neighbor is up for 00:00:19
   Authentication Sequence: [ 0 ]
```

//以上表示没有了 DR/BDR 的选举

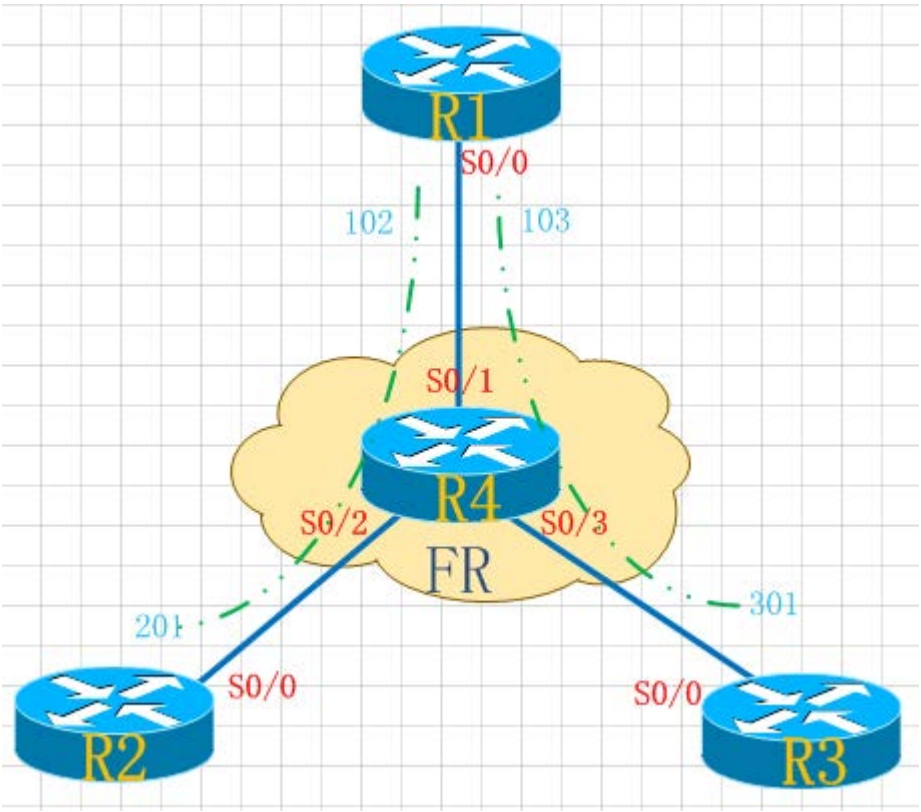
再去查看接口的 OSPF 网络类型：

```
[R2]display ospf interface s
[R2]display ospf interface Serial 0/0/0
        OSPF Process 1 with Router ID 22.1.1.1
              Interfaces
Interface: 123.1.1.2 (Serial0/0/0)
 Cost: 1562    State: P-2-P     Type: P2MP      MTU: 1500
 Timers: Hello 30 , Dead 120 , Poll  120 , Retransmit 5 , Transmit Delay 1
```

//以上表示网络类型变成了点到多点。

用以上的命令可以修改 OSPF 的各种网络类型，这里不再一一演示。明白网络类型的意义，自己多修改 OSPF 的网络类型。

实验 4.19 点到点的子接口运行 OSPF



先在 R1 上配置点到点的子接口及 IP 地址。

[R1]interface LoopBack 0 [R2-LoopBack0]ip add 11.1.1.1 255.255.255.0
[R1]interface Serial 0/0/0 [R1-Serial0/0/0]link-protocol fr //首先在主接口封装帧中继 [R1-Serial0/0/0]undo shutdown
[R1]interface Serial 0/0/0.12 p2p // 起一个点到点的子接口。 [R1-Serial0/0/0.12]ip address 12.1.1.1 255.255.255.0

```
[R1-Serial0/0/0.12]undo shutdown
[R1-Serial0/0/0.12]fr dlci 102        // 给这个接口一个 DLCI 号
```

```
[R1]interface Serial 0/0/0.13 p2p
[R1-Serial0/0/0.13]ip address 13.1.1.1 255.255.255.0
[R1-Serial0/0/0.13]undo shutdown
[R1-Serial0/0/0.13]fr dlci 103
```

在 R2 上：

```
[R2]interface Serial 0/0/0
[R2-Serial0/0/0]link-protocol fr
[R2]interface Serial 0/0/0.12 p2p
[R2-Serial0/0/0.12]ip address 12.1.1.2 255.255.255.0
[R2-Serial0/0/0.12]undo shutdown
[R2-Serial0/0/0.12]fr dlci 201
```

```
[R2]interface LoopBack 0
[R2-LoopBack0]ip add 22.1.1.1 255.255.255.0
```

在 R3 上：

```
[R3]interface Serial 0/0/0
[R3-Serial0/0/0]link-protocol fr
[R3]interface Serial 0/0/0.13 p2p
[R3-Serial0/0/0.13]ip address 13.1.1.3 255.255.255.0
[R3-Serial0/0/0.13]undo shutdown
[R3-Serial0/0/0.13]fr dlci 301
```

```
[R3]interface LoopBack 0
[R3-LoopBack0]ip address 33.1.1.1 255.255.255.0
```

帧中继的配置 OK 后，先测试一下直连是否能通，直连通了以后，再配置以下的 OSPF：

```
[R1]ospf 1 router-id 11.1.1.1
[R1-ospf-1]area 0
[R1-ospf-1-area-0.0.0.0]network 11.1.1.1 0.0.0.0
[R1-ospf-1-area-0.0.0.0]network 12.1.1.1 0.0.0.0
[R1-ospf-1-area-0.0.0.0]network 13.1.1.1 0.0.0.0
```

```
[R2]ospf 1 router-id 22.1.1.1
[R2-ospf-1]area 0
```

```
[R2-ospf-1-area-0.0.0.0]network 12.1.1.2 0.0.0.0
[R2-ospf-1-area-0.0.0.0]network 22.1.1.1 0.0.0.0
```

```
[R3]ospf 1 router-id 33.1.1.1
[R3-ospf-1]area 0
[R3-ospf-1-area-0.0.0.0]network 13.1.1.3 0.0.0.0
[R3-ospf-1-area-0.0.0.0]network 33.1.1.1 0.0.0.0
```

配置好以后去查看 OSPF 的邻居关系，发现并没有起邻居。查看接口的网络类型发现，依然是 NBMA,所以需要修改 OSPF 的网络类型为点到点。

```
[R1]display ospf interface Serial 0/0/0.12
        OSPF Process 1 with Router ID 11.1.1.1
              Interfaces
 Interface: 12.1.1.1 (Serial0/0/0.12)
 Cost: 1562    State: Waiting    Type: NBMA      MTU: 1500
 Priority: 1
 Designated Router: 0.0.0.0
 Backup Designated Router: 0.0.0.0
 Timers: Hello 30 , Dead 120 , Poll  120 , Retransmit 5 , Transmit Delay 1
```

分别去 R1,R2,R3 上修改 OSPF 的网络类型：

```
[R1]interface Serial 0/0/0.12
[R1-Serial0/0/0.12]ospf network-type p2p
[R1]interface Serial 0/0/0.13
[R1-Serial0/0/0.13]ospf network-type p2p
```

```
[R2]interface Serial 0/0/0.12
[R2-Serial0/0/0.12]ospf network-type p2p
```

```
[R3]interface Serial 0/0/0.13
[R3-Serial0/0/0.13]ospf network-type p2p
```

修改好后再去 R1 上查看邻居和路由表，发现一切都 OK 了。

```
[R1]display ip routing-table protocol ospf
Route Flags: R - relay, D - download to fib
Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface
```

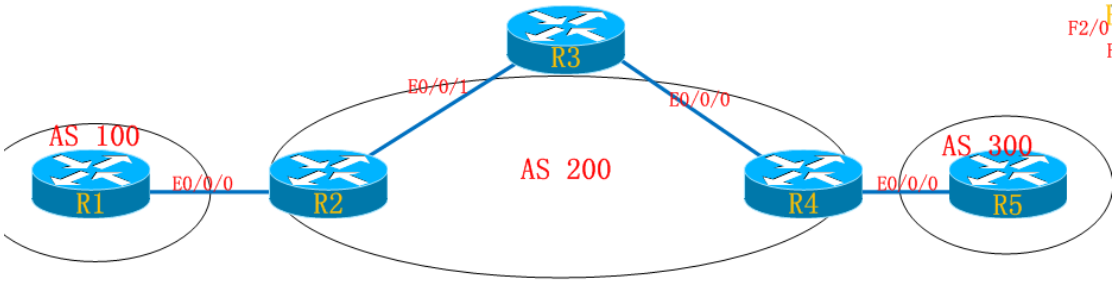
```
22.1.1.1/32  OSPF   10   1562      D   12.1.1.2      Serial0/0/0.12
33.1.1.1/32  OSPF   10   1562      D   13.1.1.3      Serial0/0/0.13
```

实验第五部分 BGP 协议

实验目的

- ☐ 掌握区域内部 BGP 的配置方法
- ☐ 掌握多区域 BGP 的配置方法
- ☐ 观察 BGP 的邻居表和数据库
- ☐ 掌握 BGP 更新源的配置方法
- ☐ 掌握 EBGP 多跳的配置方法
- ☐ 观察 IBGP 和 EBGP 中路由下一跳的发化
- ☐ 掌握 IBGP 中下一跳的配置
- ☐ 掌握 BGP 的 Network 命令的配置方法

实验拓扑



实验步骤

1:配置 IP 地址:

```
[R1]interface Ethernet 0/0/0
[R1-Ethernet0/0/0]ip address 12.1.1.1 255.255.255.0
[R1-Ethernet0/0/0]undo shutdown
[R1]interface LoopBack 0
[R1-LoopBack0]ip address 11.1.1.1 255.255.255.0
```

```
[R2]interface Ethernet 0/0/0
```

```
[R2-Ethernet0/0/0]ip address 12.1.1.2 255.255.255.0
[R2-Ethernet0/0/0]undo shutdown
[R2]interface Ethernet 0/0/1
[R2-Ethernet0/0/1]ip address 23.1.1.2 255.255.255.0
[R2-Ethernet0/0/1]undo shutdown
[R2]interface LoopBack 0
[R2-LoopBack0]ip address 22.1.1.1 255.255.255.0
```

```
[R3]interface Ethernet 0/0/1
[R3-Ethernet0/0/1]ip address 23.1.1.3 255.255.255.0
[R3-Ethernet0/0/1]undo shutdown
[R3-Ethernet0/0/1]quit
[R3]interface LoopBack 0
[R3-LoopBack0]ip address 33.1.1.1 255.255.255.0
[R3]interface Ethernet 0/0/0
[R3-Ethernet0/0/0]ip address 34.1.1.3 255.255.255.0
[R3-Ethernet0/0/0]undo shutdown
```

```
[R4]interface Ethernet 0/0/0
[R4-Ethernet0/0/0]ip address 34.1.1.4 255.255.255.0
[R4-Ethernet0/0/0]undo shutdown
[R4-Ethernet0/0/0]quit
[R4]interface Ethernet 0/0/1
[R4-Ethernet0/0/0]ip address 45.1.1.4 255.255.255.0
[R4-Ethernet0/0/0]undo shutdown
[R4]interface LoopBack 0
[R4-LoopBack0]ip address 44.1.1.1 255.255.255.0
```

```
[R5]interface Ethernet 0/0/1
[R5-Ethernet0/0/0]ip address 45.1.1.5 255.255.255.0
[R5-Ethernet0/0/0]undo shutdown
[R5]interface LoopBack 0
```

2：配置 AS200 内的 OSPF

```
[R2]ospf 1 router-id 22.1.1.1
[R2-ospf-1]area 0
[R2-ospf-1-area-0.0.0.0]network 23.1.1.2 0.0.0.0
```

```
[R2-ospf-1-area-0.0.0.0]network 22.1.1.1 0.0.0.0
```

```
[R3]ospf 1 router-id 33.1.1.1
[R3-ospf-1]area 0
[R3-ospf-1-area-0.0.0.0]network 23.1.1.3 0.0.0.0
[R3-ospf-1-area-0.0.0.0]network 33.1.1.1 0.0.0.0
[R3-ospf-1-area-0.0.0.0]network 34.1.1.3 0.0.0.0
```

```
[R4]ospf 1 router-id 44.1.1.1
[R4-ospf-1]area 0
[R4-ospf-1-area-0.0.0.0]network 34.1.1.4 0.0.0.0
[R4-ospf-1-area-0.0.0.0]network 44.1.1.1 0.0.0.0
```

3：配置 BGP，

在 R1 与 R2 之间配置 EBGP 的邻居关系：

```
[R1]bgp 100
[R1-bgp]peer 12.1.1.2 as-number 200
```

```
[R2]bgp 200
[R2-bgp]peer 12.1.1.1 as-number 100
```

查看 TCP 连接：

[R1]display tcp status

```
TCPCB    Tid/Soid Local Add:port      Foreign Add:port     VPNID  State
1cffb7c8 59 /1    0.0.0.0:23          0.0.0.0:0            -1     Listening
15c6e484 106/1    0.0.0.0:179         12.1.1.2:0           0      Listening
15c6f264 106/4    12.1.1.1:179        12.1.1.2:54272       0      Established
```

查看邻居表：

```
[R1]display bgp peer
 BGP local router ID : 12.1.1.1
 Local AS number : 100
 Total number of peers : 1                 Peers in established state : 1
  Peer            V          AS  MsgRcvd  MsgSent  OutQ  Up/Down       State PrefRcv
  12.1.1.2        4         200        4        5     0 00:02:39 Established       0
```

查看详细信息：

```
[R1]display bgp peer verbose
  BGP Peer is 12.1.1.2,  remote AS 200
  Type: EBGP link
  BGP version 4, Remote router ID 12.1.1.2
  Update-group ID: 1
  BGP current state: Established, Up for 00h03m58s
  BGP current event: RecvKeepalive
  BGP last state: OpenConfirm
  BGP Peer Up count: 1
  Received total routes: 0
  Received active routes total: 0
  Advertised total routes: 0
  Port:  Local - 179      Remote - 54272
  Configured: Connect-retry Time: 32 sec
  Configured: Active Hold Time: 180 sec   Keepalive Time:60 sec
  Received  : Active Hold Time: 180 sec
  Negotiated: Active Hold Time: 180 sec   Keepalive Time:60 sec
  Peer optional capabilities:
  Peer supports bgp multi-protocol extension
  Peer supports bgp route refresh capability
  Peer supports bgp 4-byte-as capability
  Address family IPv4 Unicast: advertised and received
Received: Total 5 messages
          Update messages          0
          Open messages            1
          KeepAlive messages       4
          Notification messages    0
          Refresh messages         0
Sent: Total 6 messages
          Update messages          0
          Open messages            2
          KeepAlive messages       4
          Notification messages    0
          Refresh messages         0
Authentication type configured: None
Last keepalive received: 2013-05-27 00:34:03-08:00
Minimum route advertisement interval is 30 seconds
```

```
Optional capabilities:
Route refresh capability has been enabled
4-byte-as capability has been enabled
Peer Preferred Value: 0
Routing policy configured:
No routing policy is configured
```

在 R2,R3,R4 之间配置 IBGP 用环回口建立邻居关系

```
[R2-bgp]peer 33.1.1.1 as-number 200
[R2-bgp]peer 33.1.1.1 connect-interface LoopBack 0
[R2-bgp]peer 33.1.1.1 next-hop-local ?
[R2-bgp]peer 44.1.1.1 as-number 200
[R2-bgp]peer 44.1.1.1 connect-interface LoopBack 0
[R2-bgp]peer 44.1.1.1 next-hop-local ?
```

```
[R3]bgp 200
[R3-bgp]peer 22.1.1.1 as-number 200
[R3-bgp]peer 22.1.1.1 next-hop-local
[R3-bgp]peer 22.1.1.1 connect-interface LoopBack 0
[R3-bgp]peer 44.1.1.1 as-number 200
[R3-bgp]peer 44.1.1.1 connect-interface LoopBack 0
```

```
[R4-bgp]peer 33.1.1.1 as-number 200
[R4-bgp]peer 33.1.1.1 connect-interface LoopBack 0
[R4-bgp]peer 33.1.1.1 next-hop-local
[R4-bgp]peer 22.1.1.1 as-number 200
[R4-bgp]peer 22.1.1.1 connect-interface LoopBack 0
[R4-bgp]peer 22.1.1.1 next-hop-local
[R4-bgp]peer 45.1.1.5 as-number 300
```

```
[R5]bgp 300
[R5-bgp]peer 45.1.1.4 as-number 200
```

宣告路由：

```
[R1]bgp 100
[R1-bgp]network 11.1.1.0 255.255.255.0
```

```
[R5-bgp]network 55.1.1.0 255.255.255.0
```

查看 BGP 表：

```
[R5]display bgp routing-table

 BGP Local router ID is 55.1.1.1
 Status codes: * - valid, > - best, d - damped,
               h - history,  i - internal, s - suppressed, S - Stale
               Origin : i - IGP, e - EGP, ? - incomplete


 Total Number of Routes: 2
      Network          NextHop        MED        LocPrf    PrefVal Path/Ogn

 *>   11.1.1.0/24      45.1.1.4                  0         200 100i
 *>   55.1.1.0/24      0.0.0.0        0          0         i
```

再查看路由表：

```
[R5]display ip routing-table protocol bgp
Route Flags: R - relay, D - download to fib
------------------------------------------------------------------------------
Public routing table : BGP
         Destinations : 1        Routes : 1

BGP routing table status : <Active>
         Destinations : 1        Routes : 1

Destination/Mask    Proto   Pre  Cost      Flags NextHop         Interface

       11.1.1.0/24  EBGP    255  0           D   45.1.1.4        Ethernet0/0/1

BGP routing table status : <Inactive>
         Destinations : 0        Routes : 0
```

测试连通性：

```
[R1]ping -a 11.1.1.1 55.1.1.1
  PING 55.1.1.1: 56  data bytes, press CTRL_C to break
```

```
Reply from 55.1.1.1: bytes=56 Sequence=1 ttl=252 time=90 ms
Reply from 55.1.1.1: bytes=56 Sequence=2 ttl=252 time=90 ms
Reply from 55.1.1.1: bytes=56 Sequence=3 ttl=252 time=80 ms
Reply from 55.1.1.1: bytes=56 Sequence=4 ttl=252 time=100 ms
Reply from 55.1.1.1: bytes=56 Sequence=5 ttl=252 time=80 ms

--- 55.1.1.1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 80/88/100 ms
```

实验第 6 部分 交换

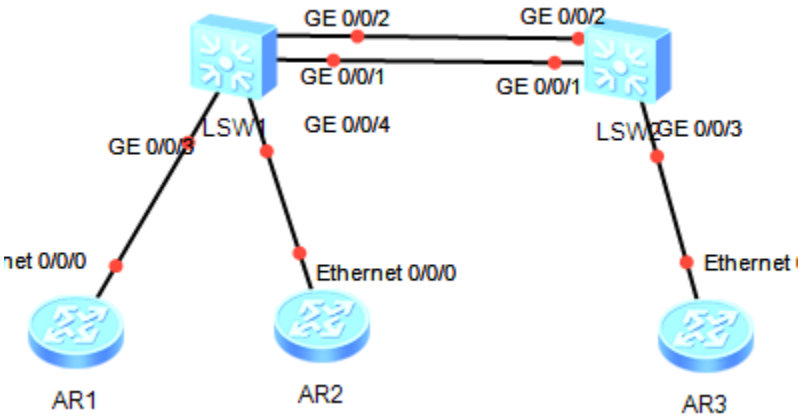
实验目的□

掌握 VLAN 的配置方法

□ 掌握 Eth-trunk 的配置方法

□ 理解 Hybrid 接口类型的应用场景

实验拓扑



如图所示由二台交换机组成的交换网络,SW1 与 SW2，其中 R1,R2,为这个网络中的二台 PC，而 R3 去模拟一台服务器。

为了网络的速度和可靠，安全等考虑，R1,R2 应该处于不同的 VLAN，R3 与 R2 处于同一 VLAN，并且还要求 R1 也能够正常访问 R3。

实验步骤

给所有设备配置 IP 地址：

```
[R1]interface Ethernet 0/0/0
[R1-Ethernet0/0/0]ip address 10.1.1.1 255.255.255.0
[R1-Ethernet0/0/0]undo shutdown
```

```
[R2]interface Ethernet 0/0/0
[R2-Ethernet0/0/0]ip address 10.1.1.2 255.255.255.0
[R2-Ethernet0/0/0]undo shutdown
```

```
[R3]interface Ethernet 0/0/0
[R3-Ethernet0/0/0]ip address 10.1.1.3 255.255.255.0
[R3-Ethernet0/0/0]undo shutdown
```

实验 6.1：配置 SW1 与 SW2 之间的链路聚合

链路聚合可以有效的提高效路带宽，和提高链路的高可用性。

1：创建 eth-trunk 接口：

```
[SW1]interface Eth-Trunk 1
[SW2]interface Eth-Trunk 1
```

2：使能发送 BPDU 与配置模式为 lacp-static.

```
[SW1-Eth-Trunk1]bpdu enable
[SW1-Eth-Trunk1]mode lacp-static
```

```
[SW2-Eth-Trunk1]bpdu enable
[SW2-Eth-Trunk1]mode lacp-static
```

3：把相关接口加入到刚创建好的聚合端口：

```
[SW1]interface GigabitEthernet 0/0/1
[SW1-GigabitEthernet0/0/1]eth-trunk 1
[SW1-GigabitEthernet0/0/1]quit
[SW1]interface GigabitEthernet 0/0/2
[SW1-GigabitEthernet0/0/2]eth-trunk 1
```

在 SW2 上：

```
[SW2]interface GigabitEthernet 0/0/1
[SW2-GigabitEthernet0/0/1]eth-trunk 1
[SW2-GigabitEthernet0/0/1]quit
[SW2]interface GigabitEthernet 0/0/2
[SW2-GigabitEthernet0/0/2]eth-trunk 1
```

验证结果：

```
[SW1]display eth-trunk
Eth-Trunk1's state information is:
Local:
LAG ID: 1                   WorkingMode: STATIC
Preempt Delay: Disabled     Hash arithmetic: According to SIP-XOR-DIP
System Priority: 32768      System ID: 4c1f-cc3c-c593
Least Active-linknumber: 1  Max Active-linknumber: 8
Operate status: up          Number Of Up Port In Trunk: 2
--------------------------------------------------------------------------------
ActorPortName        Status   PortType PortPri PortNo PortKey PortState Weight
GigabitEthernet0/0/1 Selected 1000TG   32768   2     401    10111100  1
GigabitEthernet0/0/2 Selected 1000TG   32768   3     401    10111100  1

Partner:
--------------------------------------------------------------------------------
ActorPortName        SysPri   SystemID       PortPri PortNo PortKey PortState
GigabitEthernet0/0/1 32768    4c1f-cc8e-9107 32768   2     401    10111100
GigabitEthernet0/0/2 32768    4c1f-cc8e-9107 32768   3     401    10111100
```

//以上红色部分显示了运行的模式，负载均衡的方式，以及被捆进来的接口，和各个接口的状态。此时二个接口 down 掉任何一个，eth-trunk 接口都不会 down 掉。

实验 6.2 配置 VLAN

在 SW1 与 SW2 上各自创建二个 VLAN，用于分离 R1,与 R2. R3 与 R2 保持在同一个 VLAN.

1：创建 VLAN：

```
[SW1]vlan batch 10 20
```

```
[SW2]vlan batch 10 20
```

2：将 SW1 上连接 R1,R2 的二个接口确定为 access 模式，并加入相应的 VLAN。

```
[SW1]interface GigabitEthernet 0/0/3
[SW1-GigabitEthernet0/0/3]port link-type access
[SW1-GigabitEthernet0/0/3]port default vlan 10
[SW1-GigabitEthernet0/0/3]quit
```

```
[SW1]interface GigabitEthernet 0/0/4
[SW1-GigabitEthernet0/0/4]port link-type access
[SW1-GigabitEthernet0/0/4]port default vlan 20
```

把 SW2 上的 R3 划到 VLAN20 里面：

```
[SW2]interface GigabitEthernet 0/0/3
[SW2-GigabitEthernet0/0/3]port link-type access
[SW2-GigabitEthernet0/0/3]port default vlan 20
```

再把 SW1 与 SW2 之间的链路配置成 trunk。，Trunk 接口默认只允许 VLAN 1 通过，所以配置允许 VLAN10、VLAN20 通过

```
[SW1]interface Eth-Trunk 1
[SW1-Eth-Trunk1]port link-type trunk
[SW1-Eth-Trunk1]port trunk allow-pass vlan 10 20
```

```
[SW2]interface Eth-Trunk 1
[SW2-Eth-Trunk1]port link-type trunk
[SW2-Eth-Trunk1]port trunk allow-pass vlan 10 20
```

验证 VLAN 的配置：

```
[SW1]display vlan summary
static vlan:
Total 3 static vlan.
  1 10 20
```

//以上表示手动创建的 2 个 VLAN。

```
[SW1]display vlan
--------------------------------------------------------------------------------
1    common  UT:GE0/0/5(D)     GE0/0/6(D)     GE0/0/7(D)     GE0/0/8(D)
             GE0/0/9(D)     GE0/0/10(D)    GE0/0/11(D)    GE0/0/12(D)
             GE0/0/13(D)    GE0/0/14(D)    GE0/0/15(D)    GE0/0/16(D)
             GE0/0/17(D)    GE0/0/18(D)    GE0/0/19(D)    GE0/0/20(D)
             GE0/0/21(D)    GE0/0/22(D)    GE0/0/23(D)    GE0/0/24(D)
             Eth-Trunk1(U)
```

```
10   common  UT:GE0/0/3(U)
          TG:Eth-Trunk1(U)
20   common  UT:GE0/0/4(U)
          TG:Eth-Trunk1(U)

VID  Status  Property      MAC-LRN Statistics Description
--------------------------------------------------------------------------------
1    enable  default       enable  disable    VLAN 0001
10   enable  default       enable  disable    VLAN 0010
20   enable  default       enable  disable    VLAN 0020
```

再用以下命令查看接口属于的 VLAN。

```
[SW1]display port vlan
Port                        Link Type    PVID  Trunk VLAN List
--------------------------------------------------------------------------------
Eth-Trunk1                  trunk        1          1 10 20
GigabitEthernet0/0/1        hybrid       0     -
GigabitEthernet0/0/2        hybrid       0     -
GigabitEthernet0/0/3        access       10    -
GigabitEthernet0/0/4        access       20    -
GigabitEthernet0/0/5        hybrid       1     -
```

在 R2 上去测试连通性：

```
[R2]ping 10.1.1.3
  PING 10.1.1.3: 56  data bytes, press CTRL_C to break
    Reply from 10.1.1.3: bytes=56 Sequence=1 ttl=255 time=530 ms
    Reply from 10.1.1.3: bytes=56 Sequence=2 ttl=255 time=80 ms
    Reply from 10.1.1.3: bytes=56 Sequence=3 ttl=255 time=80 ms
    Reply from 10.1.1.3: bytes=56 Sequence=4 ttl=255 time=100 ms
    Reply from 10.1.1.3: bytes=56 Sequence=5 ttl=255 time=100 ms
```

//以上表示 R2 能够通过 trunk 链路与处于同一个 VLAN 的 R3 正常通信。

但是 R1 依然不能 ping 通 R3，因为他们在不同的 VLAN。

实验 6.3 hybrid 接口

紧接上面的实验，用 hybrid 接口实现 R1 与 R3 不同 VLAN 之间的通信。

Hybrid 接口通常用来隔离处于同一个网段下的接口通信。

Hybrid 接口可以实现 R1 和 R3 之间二层模式下的通信。

在 SW1 上：

```
[SW1-GigabitEthernet0/0/3]undo port default vlan
[SW1-GigabitEthernet0/0/3]port link-type hybrid    // 配置接口的模式为 hybrid，默认就是此模式，可不打此命令。
[SW1-GigabitEthernet0/0/3]port hybrid pvid vlan 10
 //把此接口加入到 VLAN10.默认属于 VLAN1
[SW1-GigabitEthernet0/0/3]port hybrid untagged vlan 10 20
//让接口允许发送 20 的流量，并且不打标记。
```

```
[SW2-GigabitEthernet0/0/3]undo port default vlan
[SW2-GigabitEthernet0/0/3]port link-type hybrid
[SW2-GigabitEthernet0/0/3]port hybrid pvid vlan 20
[SW2-GigabitEthernet0/0/3]port hybrid untagged vlan 10 20
```

再去测试连通性：

```
[R1]ping 10.1.1.3
  PING 10.1.1.3: 56  data bytes, press CTRL_C to break
    Reply from 10.1.1.3: bytes=56 Sequence=1 ttl=255 time=110 ms
    Reply from 10.1.1.3: bytes=56 Sequence=2 ttl=255 time=60 ms
    Reply from 10.1.1.3: bytes=56 Sequence=3 ttl=255 time=120 ms
    Reply from 10.1.1.3: bytes=56 Sequence=4 ttl=255 time=80 ms
    Reply from 10.1.1.3: bytes=56 Sequence=5 ttl=255 time=80 ms
```

//R1 与 R3 不在一个 VLAN，也实现了通信。但是他们都是属于同一个三层子网。

实验 6.4 hybrid 端口另一个实验

功能简介

1 交换机端口链路类型介绍

交换机以太网端口共有三种链路类型：Access、Trunk 和 Hybrid。

□ Access 类型的端口只能属于 1 个 VLAN，一般用于连接计算机的端口；

□ Trunk 类型的端口可以属于多个 VLAN，可以接收和发送多个 VLAN 的报文，一般用于交换机之间连接的端口；

□ Hybrid 类型的端口可以属于多个 VLAN，可以接收和发送多个 VLAN 的报文，可以用于交换机之间连接，也可以用于连接用户的计算机。

其中，Hybrid 端口和 Trunk 端口的相同之处在于两种链路类型的端口都可以允许多个 VLAN 的报文发送时打标签；不同之处在于 Hybrid 端口可以允许多个 VLAN 的报文发送时不打标签，而 Trunk 端口只允许缺省 VLAN 的报文发送时不打标签。

三种类型的端口可以共存在一台以太网交换机上，但 Trunk 端口和 Hybrid 端口之间不能直接切换，只能先设为 Access 端口，再设置为其他类型端口。例如：Trunk 端口不能直接被设置为 Hybrid 端口，只能先设为 Access 端口，再设置为 Hybrid 端口。

2 各类型端口使用注意事项

配置 Trunk 端口或 Hybrid 端口，并利用 Trunk 端口或 Hybrid 端口发送多个 VLAN 报文时一定要注意：本端端口和对端端口的缺省 VLAN ID(端口的 PVID)要保持一致。

当在交换机上使用 isolate-user-vlan 来进行二层端口隔离时，参与此配置的端口的链路类型会自动变成 Hybrid 类型。

Hybrid 端口的应用比较灵活，主要为满足一些特殊应用需求。此类需求多为在无法下发访问控制规则的交换机上，利用 Hybrid 端口收发报文时的处理机制，来完成对同一网段的 PC 机之间的二层访问控制。

3 各类型端口在接收和发送报文时的处理

1) 端口接收报文时的处理：

端口接收到的报文类型	报文帧结构中携带 VLAN 标记	报文帧结构中不携带 VLAN 标记
Access 端口	丢弃该报文	为该报文打上 VLAN 标记为本端口的 PVID
Trunk 端口	判断本端口是否允许携带该 VLAN 标记的报文通过。如果允许则报文携带原有 VLAN 标记进行转发，否则丢弃该报文	同上
Hybrid 端口	同上	同上

2) 端口发送报文时的处理：

Access 端口	剥掉报文所携带的 VLAN 标记，进行转发
Trunk 端口	首先判断报文所携带的 VLAN 标记是否和端口的 PVID 相等。如果相等，则剥掉报文所携带的 VLAN 标记，进行转发；否则报文将携带原有的 VLAN 标记进行转发
Hybrid 端口	首先判断报文所携带的 VLAN 标记在本端口需要做怎样的处

	理。如果是 untagged 方式转发，则处理方式同 Access 端口； 如果是 tagged 方式转发，则处理方式同 Trunk 端口

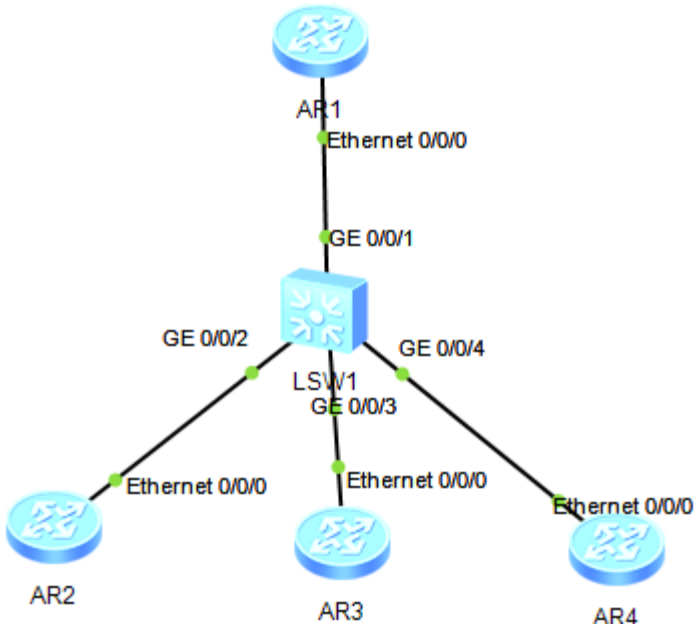
Tag 通常是对进入端口的报文打上标签，untagged 则是对出端口的报文去除标签。

简记: Access 端口只连接到计算机，此种类型的端口只能和相同 vlan-id 主机通信；连接到计算机，从计算机接收报文时打上标记，发送到计算机剥离标记；

Trunk 端口常用于交换机之间的连接，可以接收和发送多个 vlan-id 标记的报文，即多种标签报文混合在 trunk 管道上混跑，只有允许的才可以发送和接收。发送和接收都携带报文标签。可设置一个 PVID 标签,与此 PVID 相同的 VLAN-ID 数据,接收加上 PVID,发送剥离此 PVID.

Hybrid 端口可连接计算机，也可以连接在交换机之间。Hybrid 端口连接到计算机时，都要在 hybrid 命令后加上 untagged，因为计算机的网卡是不能接收带标签报文的。虽然 hybrid 端口在声明时，只声明其属于某一个 vlan，但如果在 hybrid 命令[SwitchA-Ethernet0/1]port hybrid vlan 20 30 100 untagged 命令后加上多个 vlan-id，则表明该 vlan 可以和多个 vlan-id 的主机通信。这是和 access 端口不相同的地方。这个有点类似于把一个主机划分到多个 vlan 中，或者更准确地讲，一个 vlan 内主机可以和多个 vlan 通信。当 Hybrid 端口用于连接在交换机之间时，则与 trunk 端口一样使用。

实验拓扑



实验步骤

『配置环境参数』

1. R1 连接到端口 G0/0/1，属于 VLAN10, R2、R3 和 R4,分别连接到交换机 Switch1 的端口 G0/0/2 、G0/0/3 和 G0/0/4，端口分属于 VLAN20、30 和 40，

2. R1 的 IP 地址为 10.1.1.1/24，RC2 的 IP 地址为 10.1.1.2/24，R3 的 IP 地址为 10.1.1.3/24，R4 的 IP 是 10.1.1.4/24。

『组网需求』

1. R2 和 R3 之间可以互访；
2. R2 和 R4 之间可以互访；
3. R2、R3 和 R4 都可以访问 R1；
4. 其余的路由器间访问均禁止。

『交换机 Hybrid 端口配置流程』

利用 Hybrid 端口的特性¯**一个端口可以属于多个不同的 VLAN，**来完成分属不同 VLAN 内的同网段 PC 机的访问需求。

1：先创建 VLAN:

```
[SW1]vlan batch 10 20 30 40
```

```
[SW1]interface GigabitEthernet 0/0/1
[SW1-GigabitEthernet0/0/1]port hybrid pvid vlan 10
[SW1-GigabitEthernet0/0/1]port hybrid untagged vlan 10 20 30 40
```

```
[SW1]interface GigabitEthernet 0/0/2
[SW1-GigabitEthernet0/0/2]port hybrid pvid vlan 20
[SW1-GigabitEthernet0/0/2]port hybrid untagged vlan 10 20 30 40
```

```
[SW1]interface GigabitEthernet 0/0/3
[SW1-GigabitEthernet0/0/3]port hybrid pvid vlan 30
[SW1-GigabitEthernet0/0/3]port hybrid untagged vlan 10 20 30
```

```
[SW1]interface GigabitEthernet 0/0/4
[SW1-GigabitEthernet0/0/4]port hybrid pvid vlan 40
[SW1-GigabitEthernet0/0/4]port hybrid untagged vlan 10 20 40
```

到各个路由器上配置 IP 地址：

```
[R1]interface Ethernet 0/0/0
[R1-Ethernet0/0/0]ip address 10.1.1.1 255.255.255.0
[R1-Ethernet0/0/0]undo shutdown
```

```
[R2]interface Ethernet 0/0/0
[R2-Ethernet0/0/0]ip address 10.1.1.2 255.255.255.0
```

```
[R2-Ethernet0/0/0]undo shutdown
```

```
[R3]interface Ethernet 0/0/0
[R3-Ethernet0/0/0]ip address 10.1.1.3 255.255.255.0
[R3-Ethernet0/0/0]undo shutdown
```

```
[R4-Ethernet0/0/0]
[R4-Ethernet0/0/0]ip address 10.1.1.4 255.255.255.0
[R4-Ethernet0/0/0]undo shutdown
```

测试连通性：

R1 应该能 ping 通所有路由器：

```
[R1]ping -c 1 10.1.1.2
  Reply from 10.1.1.2: bytes=56 Sequence=1 ttl=255 time=60 ms
[R1]ping -c 1 10.1.1.3
  Reply from 10.1.1.3: bytes=56 Sequence=1 ttl=255 time=90 ms
[R1]ping -c 1 10.1.1.4
  Reply from 10.1.1.4: bytes=56 Sequence=1 ttl=255 time=50 ms
```

R2 也能 PING 通所有，

而 R3，和 R4 之间却不能通信：

```
[R3]ping 10.1.1.4
  PING 10.1.1.4: 56  data bytes, press CTRL_C to break
    Request time out
    Request time out
```

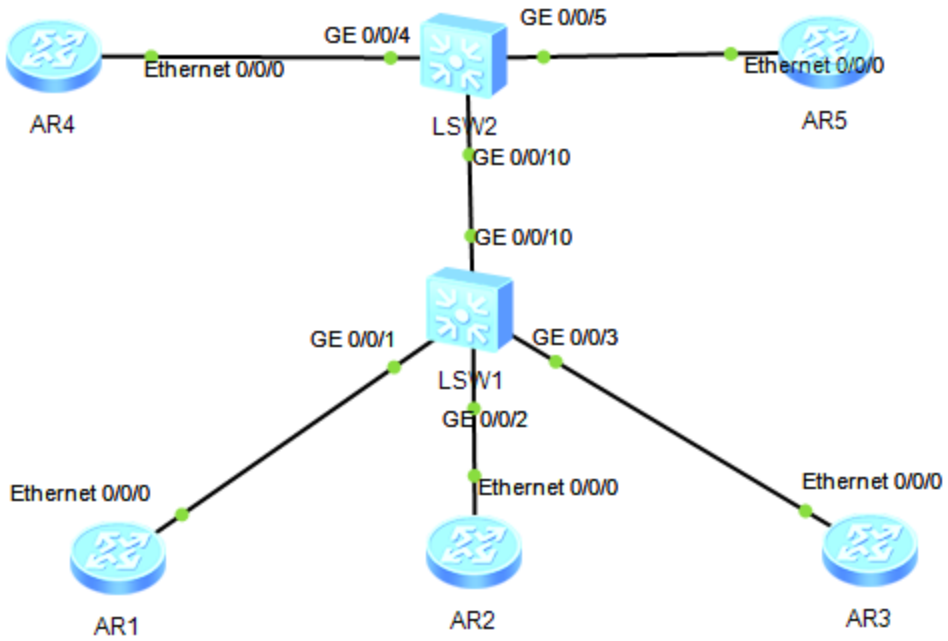
【补充说明】

对于 Hybrid 端口来说，可以同时属于多个 VLAN。这些 VLAN 分别是该 Hybrid 端口的 PVID，以及手工配置的”untagged”及”tagged”方式的 VLAN。一定要注意对应端口的 VLAN 配置，保证报文能够被端口进行正常的收发处理。

此应用在二层网络中，对相同网段的主机进行访问权限的控制。

实验 6.5 复杂的 hybrid 端口实验

实验拓扑



『配置环境参数』

1. R1、R2 和 R3 分别连接到交换机 Switch1 的端口 G0/0/1 、G0/0/2 和 G0/0/3，端口分属于 VLAN10、20 和 30；R4 和 R5 分别连接到交换机 Switch2 的端口 G0/0/4 和 G0/0/5，端口分属于 VLAN10 和 20；
2. Switch1 通过端口 G0/0/10，连接到 Switch2 的端口 G0/0/10；均不是 Trunk 端口；
3. R1 的 IP 地址为 10.1.1.1/24，R2 的 IP 地址为 10.1.1.2/24，R3 的 IP 地址为 10.1.1.3/24，R4 的 IP 地址为 10.1.1.4/24，R5 的 IP 地址为 10.1.1.5/24。

『组网需求』

1. R1 和 R3 之间可以互访；
2. R2 和 R3 之间可以互访；
3. R1 和 R4 之间可以互访； 本就在同一个 VLAN
4. R2 和 R5 之间可以互访； 本就在同一个 VLAN。
5. 其余 PC 之间均禁止互相访问。

在 SW1 上与 SW2 上创建 VLAN。

```
[SW1]vlan batch 10 20 30
```

```
[SW2]vlan batch 10 20
```

在 SW1 上配置分别连接 R2,R3，R4 的接口：

```
[SW1]interface GigabitEthernet 0/0/1
[SW1-GigabitEthernet0/0/1]port hybrid pvid vlan 10  //加入 VLAN10，收到的未标记的流量属于 VLAN10
[SW1-GigabitEthernet0/0/1]port hybrid untagged vlan 10 30   //发送数据时剥离 VLAN10，30
```

```
[SW1]interface GigabitEthernet 0/0/2
[SW1-GigabitEthernet0/0/2]port hybrid pvid vlan 20
[SW1-GigabitEthernet0/0/2]port hybrid untagged vlan 20 30
```

```
[SW1]interface GigabitEthernet 0/0/3
[SW1-GigabitEthernet0/0/3]port hybrid pvid vlan 30
[SW1-GigabitEthernet0/0/3]port hybrid untagged vlan 10 20 30
```

配置交换机与交换机之间的接口，

```
[SW1]interface GigabitEthernet 0/0/10
[SW1-GigabitEthernet0/0/10]port hybrid tagged vlan 10 20
```

//以上命令表示允许通过 VLAN 10 20。并且会打上标记。当然默认的 VLAN1 也是允许通过的。

在到 SW2 上配置：

```
[SW2]interface GigabitEthernet 0/0/4
[SW2-GigabitEthernet0/0/4]port hybrid pvid vlan 10
[SW2-GigabitEthernet0/0/4]port hybrid untagged vlan 10
```

```
[SW2-GigabitEthernet0/0/4]int g0/0/5
[SW2-GigabitEthernet0/0/5]port hybrid pvid vlan 20
[SW2-GigabitEthernet0/0/5]port hybrid untagged vlan 20
```

```
[SW2]interface GigabitEthernet 0/0/10
[SW2-GigabitEthernet0/0/10]port hybrid tagged vlan 10 20
```

到路由器上配置 IP 地址：

```
[R1]interface Ethernet 0/0/0
[R1-Ethernet0/0/0]ip address 10.1.1.1 255.255.255.0
[R1-Ethernet0/0/0]undo shutdown
```

```
[R2]interface Ethernet 0/0/0
[R2-Ethernet0/0/0]ip address 10.1.1.2 255.255.255.0
[R2-Ethernet0/0/0]undo shutdown
```

```
[R3]interface Ethernet 0/0/0
[R3-Ethernet0/0/0]ip address 10.1.1.3 255.255.255.0
[R3-Ethernet0/0/0]undo shutdown
```

```
[R4]interface Ethernet 0/0/0
[R4-Ethernet0/0/0]ip address 10.1.1.4 255.255.255.0
[R4-Ethernet0/0/0]undo shutdown
```

```
[R5]interface Ethernet 0/0/0
[R5-Ethernet0/0/0]ip address 10.1.1.5 255.255.255.0
[R5-Ethernet0/0/0]undo shutdown
```

测试：

```
<R3>ping -c 1 10.1.1.1
  Reply from 10.1.1.1: bytes=56 Sequence=1 ttl=255 time=40 ms
```

```
<R3>ping -c 1 10.1.1.2
  Reply from 10.1.1.2: bytes=56 Sequence=1 ttl=255 time=60 ms
```

```
<R1>ping -c 1 10.1.1.4
 Reply from 10.1.1.4: bytes=56 Sequence=1 ttl=255 time=140 ms
```

```
<R2>ping -c 1 10.1.1.5
  Reply from 10.1.1.5: bytes=56 Sequence=1 ttl=255 time=100 ms
```

其它都是不通的。

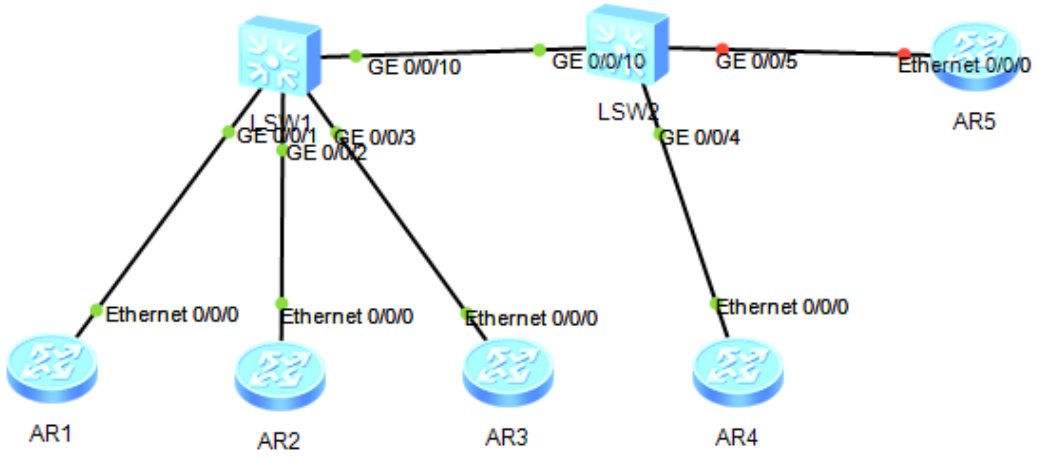
实验 6.6 GVRP

技术简介：

GVRP 可以在网络中传播 VLAN 信息，提高了网络管理员配置 VLAN 的效率。

通过将 SW1 的 G0/0/10 接口配置为 Fixed 模式，SW2 的 G0/0/10 接口配置为 Normal 模式，使得 SW2 能够学习 SW1 的 VLAN 配置。

实验拓扑



配置 SW1 与 SW2 之间的接口为 trunk，并且允许所有 VLAN 通过:

在 SW1 上：

```
[SW1]interface GigabitEthernet 0/0/10
[SW1-GigabitEthernet0/0/10]port link-type trunk
[SW1-GigabitEthernet0/0/10]port trunk allow-pass vlan all
```

```
[SW2]interface GigabitEthernet 0/0/10
[SW2-GigabitEthernet0/0/10]port link-type trunk
[SW2-GigabitEthernet0/0/10]port trunk allow-pass vlan all
```

启用交换机的 GVRP 功能：

```
[SW1]gvrp
[SW2]gvrp
```

查看是否启用了 GVRP

```
[SW1]display gvrp status
 Info: GVRP is enabled
```

```
[SW2]display gvrp status
 Info: GVRP is enabled
```

配置 SW1 的 G0/0/10 接口为 fixed 模式。

```
[SW1]interface GigabitEthernet 0/0/10
[SW1-GigabitEthernet0/0/10]gvrp
[SW1-GigabitEthernet0/0/10]gvrp registration fixed
[SW1-GigabitEthernet0/0/10]bpdu enable
```

配置 SW2 的 G0/0/10 接口为 normal 模式

```
[SW2]interface GigabitEthernet 0/0/10
[SW2-GigabitEthernet0/0/10]gvrp
[SW2-GigabitEthernet0/0/10]gvrp registration normal
[SW2-GigabitEthernet0/0/10]bpdu enable
```

验证刚才的配置：

```
[SW1]display gvrp statistics
  GVRP statistics on port GigabitEthernet0/0/10
    GVRP status                 : Enabled
    GVRP registrations failed   : 0
    GVRP last PDU origin        : 4c1f-cc79-f2f6
    GVRP registration type      : Fixed
```

```
[SW2]display gvrp statistics
  GVRP statistics on port GigabitEthernet0/0/10
    GVRP status                 : Enabled
    GVRP registrations failed   : 0
    GVRP last PDU origin        : 4c1f-cc05-446e
    GVRP registration type      : Normal
```

接下来在 SW1 上创造 VLAN 10，100，200

```
[SW1]vlan batch 10 100 200
Info: This operation may take a few seconds. Please wait for a moment...done.
```

查看 VLAN：

```
[SW1]display vlan summary
static vlan:
```

```
Total 4 static vlan.
  1 10 100 200
dynamic vlan:
Total 0 dynamic vlan.
reserved vlan:
Total 0 reserved vlan.
```

到 SW2 上查看 VLAN 是否学到：

```
[SW2]display vlan summary
static vlan:
Total 1 static vlan.
  1
dynamic vlan:
Total 3 dynamic vlan.
  10 100 200
reserved vlan:
Total 0 reserved vlan.
```

//以上表示 SW2 正常的学到了 SW1 的 VLAN。

再到 SW2 创建一个 VLAN：

```
[SW2]vlan 300
```

查看 VLAN：

```
[SW2]display vlan summary
static vlan:
Total 2 static vlan.
  1 300
dynamic vlan:
Total 3 dynamic vlan.
  10 100 200
reserved vlan:
Total 0 reserved vlan.
```

到 SW1 上查看是否学到这个 VLAN：

```
[SW1]display vlan summary
static vlan:
```

```
Total 4 static vlan.
  1 10 100 200
dynamic vlan:
Total 0 dynamic vlan.
reserved vlan:
Total 0 reserved vlan.
```

//以上表示 SW1 并没有学到这个 VLAN

修改 SW1 的接口为 normal 类型：

```
[SW1]interface GigabitEthernet 0/0/10
[SW1-GigabitEthernet0/0/10]gvrp registration normal
```

再去查看 VLAN，发现已经正常学习到了：

```
[SW1]display vlan summary
static vlan:
Total 4 static vlan.
  1 10 100 200
dynamic vlan:
Total 1 dynamic vlan.
  300
reserved vlan:
Total 0 reserved vlan.
```

实验 6.7 MUX VLAN

技术简介：

MUX VLAN 可以实现处于相同网段的设备划入不同 VLAN 后，虽然二层通信是隔离的，但是还可以和同一个指定 VLAN 通信。并且还能实现禁止相同 VLAN 内不同设备之间的通信。 等同于 CISCO 的私有 VLAN。

实验拓扑

如上图

实验步骤

将 VLAN 10 配置为 MUX VLAN 的主 VLAN，VLAN 100 和 200 配置为从 VLAN。

通过配置各 PC 与交换机相连接口的类型实现：

1：所有 PC 均可和与 R5 通信----R5 属于主 VLAN

2：R3 和 R4 不能和其他 VLAN 通信，同时也不能互相通信，---R3,R4 属于孤立 VLAN

3：R1 与 R2 相互之间可以通信，但不能访问 R3,R4。----R1,R2 属于团体 VLAN.

在 SW1 上配置：

```
[SW1]vlan 10
[SW1-vlan10]mux-vlan
[SW1-vlan10]subordinate group 100
[SW1-vlan10]subordinate separate 200
```

在 SW 上配置：

```
[SW2]vlan batch 10 100 200
[SW2]vlan 10
[SW1-vlan10]mux-vlan
[SW2-vlan10]subordinate group 100
[SW2-vlan10]subordinate separate 200
```

将 R5 连接的接口划入 VLAN10，并且开启 mux vlan 功能

```
[SW2]interface GigabitEthernet 0/0/5
[SW2-GigabitEthernet0/0/5]port link-type access
[SW2-GigabitEthernet0/0/5]port default vlan 10
[SW2-GigabitEthernet0/0/5]port mux-vlan enable
```

将 R1,R2 所对应的接口划入团体 VLAN100,并开启 MUX vlan 功能：

```
[SW1]interface GigabitEthernet 0/0/1
[SW1-GigabitEthernet0/0/1]port link-type access
[SW1-GigabitEthernet0/0/1]port default vlan 100
[SW1-GigabitEthernet0/0/1]port mux-vlan enable
```

```
[SW1]interface GigabitEthernet 0/0/2
[SW1-GigabitEthernet0/0/2]port link-type access
[SW1-GigabitEthernet0/0/2]port default vlan 100
[SW1-GigabitEthernet0/0/2]port mux-vlan enable
```

再将 R3,R4 所对应的接口划入孤立 VLAN200，并开启 mux VLAN 功能：

```
[SW1]interface GigabitEthernet 0/0/3
```

```
[SW1-GigabitEthernet0/0/3]port link-type access
[SW1-GigabitEthernet0/0/3]port default vlan 200
[SW1-GigabitEthernet0/0/3]port mux-vlan enable
```

```
[SW2]interface GigabitEthernet 0/0/4
[SW2-GigabitEthernet0/0/4]port link-type access
[SW2-GigabitEthernet0/0/4]port default vlan 200
[SW2-GigabitEthernet0/0/4]port mux-vlan enable
```

查看配置结果：

```
[SW1]display mux-vlan
Principal Subordinate Type        Interface
------------------------------------------------------------------------------
10        -          principal
10        200          separate     GigabitEthernet0/0/3
10        100          group        GigabitEthernet0/0/1 GigabitEthernet0/0/2
------------------------------------------------------------------------------
```

```
[SW2]display mux-vlan
Principal Subordinate Type        Interface
------------------------------------------------------------------------------
10        -          principal    GigabitEthernet0/0/5
10        200          separate     GigabitEthernet0/0/4
10        100          group
------------------------------------------------------------------------------
```

配置 IP 地址，验证结果：

```
[R1]interface Ethernet 0/0/0
[R1-Ethernet0/0/0]ip address 10.1.1.1 255.255.255.0
[R1-Ethernet0/0/0]undo shutdown
```

```
[R2]interface Ethernet 0/0/0
[R2-Ethernet0/0/0]ip address 10.1.1.2 255.255.255.0
[R2-Ethernet0/0/0]undo shutdown
```

```
[R3]interface Ethernet 0/0/0
```

```
[R3-Ethernet0/0/0]ip address 10.1.1.3 255.255.255.0
[R3-Ethernet0/0/0]undo shutdown
```

```
[R4]interface Ethernet 0/0/0
[R4-Ethernet0/0/0]ip address 10.1.1.4 255.255.255.0
[R4-Ethernet0/0/0]undo shutdown
```

```
[R5]interface Ethernet 0/0/0
[R5-Ethernet0/0/0]ip address 10.1.1.5 255.255.255.0
[R5-Ethernet0/0/0]undo shutdown
```

在 R1 上测试：

```
[R1]ping -c 1 10.1.1.2
  Reply from 10.1.1.2: bytes=56 Sequence=1 ttl=255 time=60 ms
[R1]ping -c 1 10.1.1.3
  Request time out
[R1]ping -c 1 10.1.1.4
  Request time out
[R1]ping -c 1 10.1.1.5
  Reply from 10.1.1.5: bytes=56 Sequence=1 ttl=255 time=90 ms
```

实验 6.8 STP

实验目的

了解 STP、RSTP 与 MSTP。

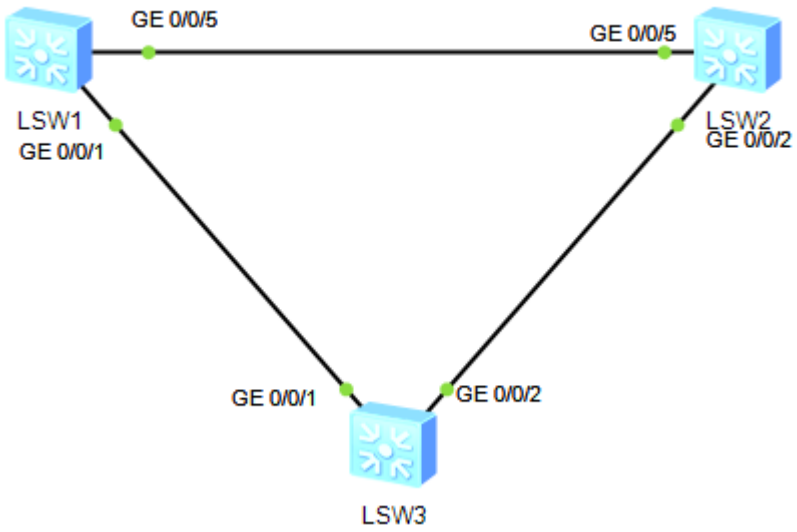
☐ 掌握修改网桥优先级影响根网桥选举的方法

☐ 掌握影响根端口和指定端口选举的方法

☐ 掌握配置 RSTP 的方法以及 STP 不 RSTP 的相兼容问题

☐ 掌握配置 MSTP 实现不同 VLAN 负载均衡的方法

实验拓扑



实验步骤

1：配置 STP。

```
[SW1]stp enable
[SW1]stp mode stp
```

```
[SW2]stp enable
[SW2]stp mode stp
```

```
[SW3]stp enable
[SW3]stp mode stp
```

2：

查看 STP 状态信息：

```
[SW1]display stp
-------[CIST Global Info][Mode STP]-------
CIST Bridge             :32768.4c1f-cc58-bd76     //自己的桥 ID
Config Times            :Hello 2s MaxAge 20s FwDly 15s MaxHop 20
Active Times            :Hello 2s MaxAge 20s FwDly 15s MaxHop 20 //时间值
CIST Root/ERPC          :32768.4c1f-cc58-bd76 / 0   //根桥的 ID，与去往根的开销
CIST RegRoot/IRPC       :32768.4c1f-cc58-bd76 / 0
CIST RootPortId         :0.0
BPDU-Protection         :Disabled
TC or TCN received      :8
TC count per hello      :0
STP Converge Mode       :Normal
Time since last TC      :0 days 0h:2m:23s
Number of TC            :8
```

```
Last TC occurred        :GigabitEthernet0/0/1
----[Port1(GigabitEthernet0/0/1)][FORWARDING]----
 Port Protocol       :Enabled
```

//以上信息表示 SW1 默认就是根桥了，因为 SW1 的 MAC 地址最小。

由于 SW1 为根桥，所以 SW1 上的所有接口都就应该为指定接口，查看接口状态：

```
[SW1]display stp brief
 MSTID  Port                        Role   TP State           Protection
   0    GigabitEthernet0/0/1       DESI  FORWARDING      NONE
   0    GigabitEthernet0/0/5       DESI  FORWARDING      NONE
```

接着在 SW2 上查看端口状态：

```
[SW2]display stp brief
 MSTID  Port              Role      STP  State              Protection
   0    GigabitEthernet0/0/2       ALTE    DISCARDING       NONE
   0    GigabitEthernet0/0/5       ROOT  FORWARDING      NONE
```

//以上输出表示 SW2 的 G0/0/5 端口为根端口，因为这个接口离根的 COST 值是最低的。

再查看 SW3:

```
[SW3]display stp brief
 MSTID  Port                        Role   STP State         Protection
   0    GigabitEthernet0/0/1       ROOT  FORWARDING      NONE
   0    GigabitEthernet0/0/2       DESI   FORWARDING      NONE
```

//以上表示 SW3 的根端口也是直连 SW1 的端口 G0/0/1,而在 SW3 与 SW2 之间的链路，SW3 的 G0/0/2 被选为了指定接口。

通过控制优先集的方式控制 SW1 为网络的根：

配置 SW1 为主根，SW 为备份根

```
[SW1]stp priority 4096
[SW2]stp priority 8192
```

到 SW3 上查看现象：

```
[SW3]display stp
-------[CIST Global Info][Mode STP]-------
CIST Bridge         :32768.4c1f-cc68-4ecb  //自己的桥 ID
Config Times        :Hello 2s MaxAge 20s FwDly 15s MaxHop 20
Active Times        :Hello 2s MaxAge 20s FwDly 15s MaxHop 20
```

```
CIST Root/ERPC        :4096 .4c1f-cc58-bd76 / 1  //根桥的 ID 优先集变为了 4096 了
CIST RegRoot/IRPC   :32768.4c1f-cc68-4ecb / 0
CIST RootPortId     :128.1
BPDU-Protection     :Disabled
TC or TCN received  :27
TC count per hello  :2
STP Converge Mode   :Normal
Time since last TC  :0 days 0h:0m:3s
Number of TC        :7
Last TC occurred    :GigabitEthernet0/0/1
----[Port1(GigabitEthernet0/0/1)][FORWARDING]----
```

根端口控制：

此时 SW3 上的根端口是连接 SW1 的接口，如果我们想修改 SW3 的根端口，就要去修改去往根的开销，把 G0/0/1 的开销修改到大于从 G0/0/2 口到 SW1 的开销：

```
[SW3]interface GigabitEthernet 0/0/1
[SW3-GigabitEthernet0/0/1]stp cost 3
```

再去查看 SW3 的端口信息：

```
[SW3]display stp brief
 MSTID  Port                        Role  STP State     Protection
   0    GigabitEthernet0/0/1       ALTE  DISCARDING    NONE
   0    GigabitEthernet0/0/2       ROOT  LEARNING      NONE
```

//以上表示 G0/0/2 变为了根端口。

配置 IP 地址测试 STP 收敛的时间：

```
[SW1]interface Vlanif 1
[SW1-Vlanif1]ip address 10.1.1.1 255.255.255.0
[SW1-Vlanif1]undo shutdown
```

```
[SW2interface Vlanif 1
[SW2-Vlanif1]ip address 10.1.1.2 25.255.255.0
[SW1-Vlanif1]undo shutdown
```

```
[SW3]interface Vlanif 1
[SW3-Vlanif1]ip address 10.1.1.3 255.255.255.0
```

```
[SW3-Vlanif1]undo shutdown
```

在 SW1 上去 ping SW2,此时走的链路是肯定是 SW1 与 SW2 这间的直连链路：

```
[SW1]ping -c 1000 10.1.1.2
 Reply from 10.1.1.2: bytes=56 Sequence=41 ttl=255 time=30 ms
  Request time out
  Request time out
  Request time out
  Request time out
  Request time out
  Request time out
  Request time out
  Request time out
  Request time out
  Request time out
  Request time out
  Request time out
  Request time out
  Request time out
  Request time out
  Request time out
  Reply from 10.1.1.2: bytes=56 Sequence=57 ttl=255 time=90 ms
```

然后 shutdownG0/0/5 口：

```
[SW2]interface GigabitEthernet 0/0/5
[SW2-GigabitEthernet0/0/5]shut
[SW2-GigabitEthernet0/0/5]shutdown
```

//通过这个实验我们发现，SW1 与 SW2 之间重新通过 SW3 可达的间隔时间长达 30 秒。

实验 6.9 快速生成树。

修改 SW1,SW2,SW3 的配置及验证：

```
[SW1]stp mode rstp
[SW2]stp mode rstp
```

```
[SW3]stp mode rstp
```

```
[SW1]display stp
-------[CIST Global Info][Mode RSTP]-------
```

//再查看的时候，已经运行 RSTP 了.

再通过 undo shutdown,再 shutdown 掉 G0/0/5，观察收敛的时间：

```
[SW2]interface GigabitEthernet 0/0/5
[SW2-GigabitEthernet0/0/5]shutdown
```

```
[SW1]ping -c 10000 10.1.1.2
Reply from 10.1.1.2: bytes=56 Sequence=33 ttl=255 time=10 ms
  Request time out
  Reply from 10.1.1.2: bytes=56 Sequence=35 ttl=255 time=60 ms
```

//只丢了一个包，收敛更快了。

实验 6.10 MSTP 的配置

1：先创建 20 个 VLAN 分别为 11-30。

```
[SW1]vlan batch 11 to 30
[SW1]vlan batch 11 to 30
[SW1]vlan batch 11 to 30
```

2：配置三条链路为 trunk，并且允许所有 VLAN 通过.

```
[SW1]interface GigabitEthernet 0/0/5
[SW1-GigabitEthernet0/0/5]port link-type trunk
[SW1-GigabitEthernet0/0/5]port trunk allow-pass vlan all
[SW1]interface GigabitEthernet 0/0/1
[SW1-GigabitEthernet0/0/5]port link-type trunk
[SW1-GigabitEthernet0/0/5]port trunk allow-pass vlan all
```

```
[SW2]interface GigabitEthernet 0/0/5
[SW2-GigabitEthernet0/0/5]port link-type trunk
[SW2-GigabitEthernet0/0/5]port trunk allow-pass vlan all
```

```
[SW2]interface GigabitEthernet 0/0/2
[SW2-GigabitEthernet0/0/2]port link-type trunk
[SW2-GigabitEthernet0/0/2]port trunk allow-pass vlan all
```

```
[SW3]interface GigabitEthernet 0/0/1
[SW3-GigabitEthernet0/0/5]port link-type trunk
[SW3-GigabitEthernet0/0/5]port trunk allow-pass vlan all
[SW3]interface GigabitEthernet 0/0/2
[SW3-GigabitEthernet0/0/5]port link-type trunk
[SW3-GigabitEthernet0/0/5]port trunk allow-pass vlan all
```

3：配置 MST

```
[SW1]stp region-configuration
[SW1-mst-region]region-name yeslab
[SW1-mst-region]instance 1 vlan 11 to 20
[SW1-mst-region]instance 2 vlan 21 to 30
[SW1-mst-region]active region-configuration
```

```
[SW2]stp region-configuration
[SW2-mst-region]region-name yeslab
[SW2-mst-region]instance 1 vlan 11 to 20
[SW2-mst-region]instance 2 vlan 21 to 30
[SW2-mst-region]active region-configuration
```

```
[SW3]stp region-configuration
[SW3-mst-region]region-name yeslab
[SW3-mst-region]instance 1 vlan 11 to 20
[SW3-mst-region]instance 2 vlan 21 to 30
[SW3-mst-region]active region-configuration
```

4：分别验证 MSTP：

```
[SW1]display stp region-configuration
 Oper configuration
   Format selector    :0
   Region name        :yeslab
   Revision level     :0
```

```
  Instance   VLANs Mapped
    0        1 to 10, 31 to 4094
    1        11 to 20
    2        21 to 30
```

5：再去确定实例 1 的根为 SW1，备份根为 SW2，实例 2 的根为 SW2,备份根为 SW1。

```
[SW1]stp instance 1 priority 4096
[SW1]stp instance 2 priority 8192
```

```
[SW2]stp instance 1 priority 8192
[SW2]stp instance 2 priority 4192
```

6：到 SW3 上查看信息：

```
[sw3]display stp brief
 MSTID  Port                      Role  STP State          Protection
   0    GigabitEthernet0/0/1      ROOT  FORWARDING         NONE
   0    GigabitEthernet0/0/2      DESI  FORWARDING         NONE
   1    GigabitEthernet0/0/1      ROOT  FORWARDING         NONE
   1    GigabitEthernet0/0/2      ALTE  DISCARDING         NONE
   2    GigabitEthernet0/0/1      ALTE  DISCARDING         NONE
   2    GigabitEthernet0/0/2      ROOT  FORWARDING         NONE
```

//以上表示在 SW3 上的实例 1 里，根端口为 g0/0/1,
而在实例 2 里，根端口为 g0/0/2
从而实现了负载均衡。

实验 6.11 配置根防护

紧接上面的实验：

1：在 SW1 上查看 instance 1 里面的根为 SW1。

```
[SW1]display stp instance 1
-------[MSTI 1 Global Info]-------
MSTI Bridge ID          :4096.4c1f-cc58-bd76
MSTI RegRoot/IRPC   :4096.4c1f-cc58-bd76 / 0
```

2：启用 SW1 上 G0/0/1 和 G0/0/5 的根防护

```
[SW1]interface GigabitEthernet 0/0/1
[SW1-GigabitEthernet0/0/1]stp root-protection
[SW1]interface GigabitEthernet 0/0/5
[SW1-GigabitEthernet0/0/5]stp root-protection
```

3：修改 SW3 中实例 1 的优先集为 0

```
[sw3]stp instance 1 priority 0
```

4：查看 SW1 上端口的变化

```
[SW1]display stp brief
 MSTID  Port                    Role  STP State     Protection
   1    GigabitEthernet0/0/1     DESI  DISCARDING     ROOT
   1    GigabitEthernet0/0/5     DESI  DISCARDING     ROOT
```

//以上显示 SW1 上的端口依然是指定端口，但是却是丢弃状态。

再查看 SW1 上和 STP 信息：

```
[SW1]display stp
-------[CIST Global Info][Mode MSTP]-------
CIST Bridge            :4096 .4c1f-cc58-bd76
Config Times           :Hello 2s MaxAge 20s FwDly 15s MaxHop 20
Active Times           :Hello 2s MaxAge 20s FwDly 15s MaxHop 20
CIST Root/ERPC         :4096 .4c1f-cc58-bd76 / 0
```

//以上表示虽然 SW3 的优先集改为了 0,但是 SW1 依然认为自己是根。因为启用了根防护：

5：去掉 G0/0/1 口的根防护：

```
[SW1-GigabitEthernet0/0/1]undo
[SW1-GigabitEthernet0/0/1]undo stp root-protection
```

6：查看结果：

```
MSTID  Port                    Role  STP State     Protection
   1    GigabitEthernet0/0/1     ROOT  FORWARDING     NONE
   1    GigabitEthernet0/0/5     DESI  DISCARDING     ROOT
```

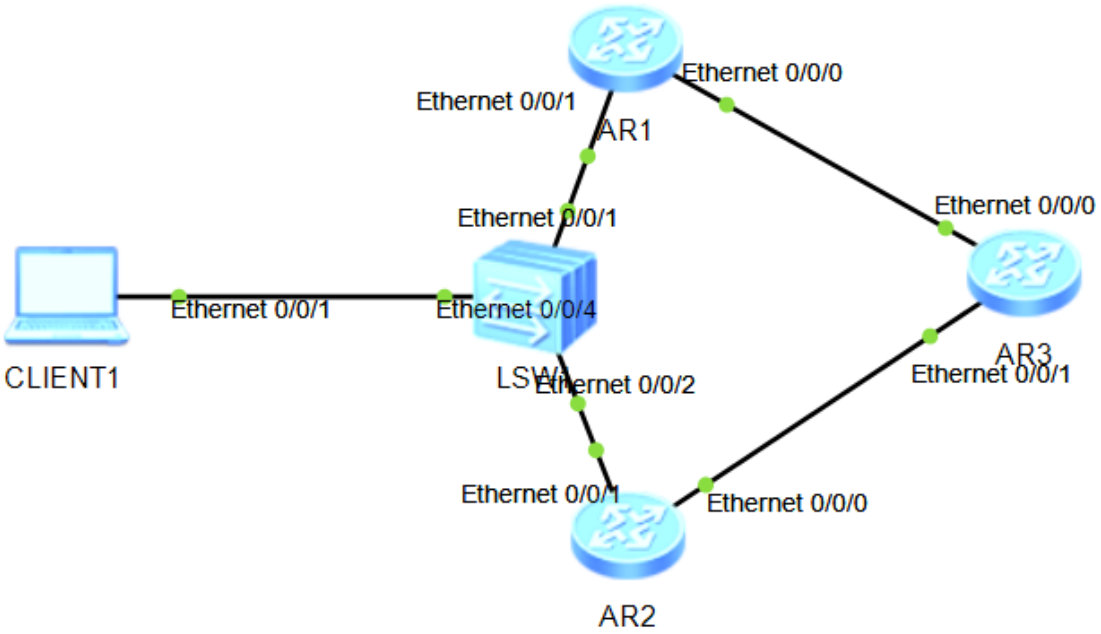
去掉根防护以后，SW1 的 G0/0/1 口就成为了根端口。

实验 6.12 VRRP

实验目的

掌握 VRRP 的配置

实验拓扑



实验步骤

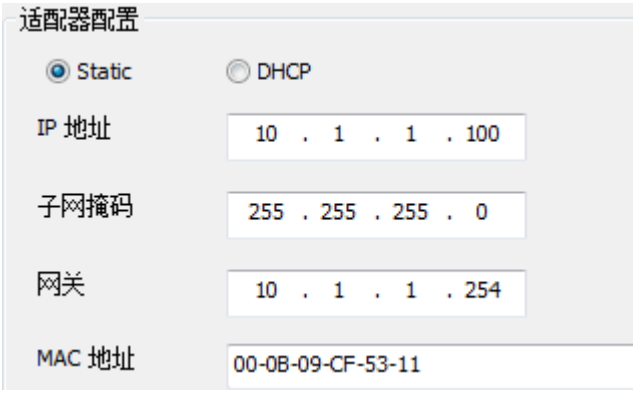
1：配置 IP 地址：

```
[R1]interface Ethernet 0/0/0
[R1-Ethernet0/0/0]ip add 13.1.1.1 255.255.255.0
[R1-Ethernet0/0/0]undo shutdown
[R1]interface Ethernet 0/0/1
[R1-Ethernet0/0/0]ip add 10.1.1.1 255.255.255.0
[R1-Ethernet0/0/0]undo shutdown
```

```
[R2]interface Ethernet 0/0/0
[R2-Ethernet0/0/0]ip add 23.1.1.1 255.255.255.0
[R2-Ethernet0/0/0]undo shutdown
[R2]interface Ethernet 0/0/1
[R2-Ethernet0/0/0]ip add 10.1.1.2 255.255.255.0
[R2-Ethernet0/0/0]undo shutdown
```

```
[R3]interface Ethernet 0/0/0
[R3-Ethernet0/0/0]ip add 13.1.1.3 255.255.255.0
[R3-Ethernet0/0/0]undo shutdown
[R3]interface Ethernet 0/0/1
[R3-Ethernet0/0/0]ip add 23.1.1.3 255.255.255.0
[R3-Ethernet0/0/0]undo shutdown
[R3] interface LoopBack 0
[R3-Ethernet0/0/0]ip add 33.1.1.1 255.255.255.0
```

配置主机的 IP 地址为 10.1.1.100



2：配置 OSPF 协议让路由可达：

```
[R1]ospf 1 router-id 11.1.1.1
[R1-ospf-1-area-0.0.0.0]network 10.1.1.1 0.0.0.0
[R1-ospf-1-area-0.0.0.0]network 13.1.1.1 0.0.0.0
```

```
[R2]ospf 1 router-id 22.1.1.1
[R2-ospf-1-area-0.0.0.0]network 10.1.1.2 0.0.0.0
[R2-ospf-1-area-0.0.0.0]network 23.1.1.2 0.0.0.0
```

```
[R3]ospf 1 router-id 33.1.1.1
[R3-ospf-1-area-0.0.0.0]network 13.1.1.3 0.0.0.0
[R3-ospf-1-area-0.0.0.0]network 23.1.1.3 0.0.0.0
[R3-ospf-1-area-0.0.0.0]network 33.1.1.1 0.0.0.0
```

3：配置 VRRP：

```
[R1]interface Ethernet 0/0/1
```

```
[R1-Ethernet0/0/1]vrrp vrid 1 virtual-ip 10.1.1.254
[R1-Ethernet0/0/1]vrrp vrid 1 priority 110
```

```
[R2]interface Ethernet 0/0/1
[R2-Ethernet0/0/1]vrrp vrid 1 virtual-ip 10.1.1.254
```

4：查看结果

```
[R1]display vrrp brief
VRID  State        Interface              Type     Virtual IP
------------------------------------------------------------------
1     Master       Eth0/0/1               Normal   10.1.1.254
------------------------------------------------------------------
Total:1     Master:1     Backup:0     Non-active:0
```

```
 [R2]display vrrp brief
VRID  State        Interface              Type     Virtual IP
------------------------------------------------------------------
1     Backup       Eth0/0/1               Normal   10.1.1.254
------------------------------------------------------------------
Total:1     Master:0     Backup:1     Non-active:0
```

```
5:查看详细信息：
[R1]display vrrp
  Ethernet0/0/1 | Virtual Router 1
    State : Master
    Virtual IP : 10.1.1.254
    Master IP : 10.1.1.1
    PriorityRun : 110
    PriorityConfig : 110
    MasterPriority : 110
    Preempt : YES   Delay Time : 0 s
    TimerRun : 1 s
    TimerConfig : 1 s
    Auth type : NONE
    Virtual MAC : 0000-5e00-0101
    Check TTL : YES
```

```
    Config type : normal-vrrp
    Create time : 2013-05-26 23:43:09 UTC-08:00
Last change time : 2013-05-26 23:43:12 UTC-08:00
```

在 PC 上 PINGR3。

```
PC>ping 33.1.1.1

Ping 33.1.1.1: 32 data bytes, Press Ctrl_C to break
From 33.1.1.1: bytes=32 seq=1 ttl=254 time=141 ms
From 33.1.1.1: bytes=32 seq=2 ttl=254 time=47 ms
From 33.1.1.1: bytes=32 seq=3 ttl=254 time=62 ms
From 33.1.1.1: bytes=32 seq=4 ttl=254 time=47 ms
From 33.1.1.1: bytes=32 seq=5 ttl=254 time=62 ms

--- 33.1.1.1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 47/71/141 ms

PC>
```

6：观察网关切换：

```
[R1]interface Ethernet 0/0/1
[R1-Ethernet0/0/1]shutdown
```

到 R2 上查看 VRRP 信息：

```
[R2]display vrrp
  Ethernet0/0/1 | Virtual Router 1
    State : Master
    Virtual IP : 10.1.1.254
    Master IP : 10.1.1.2
    PriorityRun : 100
    PriorityConfig : 100
    MasterPriority : 100
    Preempt : YES   Delay Time : 0 s
    TimerRun : 1 s
    TimerConfig : 1 s
    Auth type : NONE
    Virtual MAC : 0000-5e00-0101
    Check TTL : YES
```

```
    Config type : normal-vrrp
    Create time : 2013-05-26 23:44:20 UTC-08:00
    Last change time : 2013-05-26 23:48:06 UTC-08:00
```

//以上表示 R2 变为了 master.