



# **Cisco Systems**

## **CCNA 2007 New Edition Lab Hand Book**

**First Edition**

Author SuYong(nash\_su)

Times Education Press

Copyright 2007 SuYong All rights reserved

ROOM327,Shaanxi Library 18 Changan North Road Xi'an



## LAB1-1 路由器基本操作

**实验目的：**1 熟悉路由器的各种模式  
2 熟悉各个模式的常用命令

### 一：路由器的基本配置

实验目的：熟练掌握路由器的基本配置命令

Router>enable

Router#

//进入特权模式

Router#disable

//退出特权模式

Router>

Router#configure terminal

//进入全局配置模式

路由器 CLI 命令行的命令自动补全功能：

Router#sh <按 Tab 键>

Router#show

配置路由器时间：

Router#clock set 13:01:01 10 july 2007

Router#show clock

//查看路由器当前时间

13:01:28.985 UTC Tue Jul 10 200

#### 1 路由器安全相关配置

Router(config)#enable password cisco

// 特权模式的明文密码

Router(config)#enable secret cisco

// 特权模式的密文密码

Router(config)#service password-encryption //将路由器中所有明文密码变为加密的形式

#### 2 修改路由器主机名

Router(config)#hostname STSD

STSD(config)#

//路由器命名为 STSD

#### 3 关闭域名解析

Router(config)#no ip domain-lookup

//关闭域名解析

配置此命令后，当我们输入错误的命令时，路由器不会对它进行域名解析，从而节省我们的时间。

#### 4 配置 con 口日志同步显示：

Router(config)#line con 0

Router(config-line)#logging synchronous

配置此命令后，在输入命令时就不会被弹出日志所打断

#### 5 设置标语信息

Router(config)#banner motd #{text}#

设置标识语信息时，以#做为分隔符，并按下回车键，描述语句的本地的一个标识，它只在本地可见，并且 CISCO 执行命令时会跳过它。

#### 6 Console 接口的安全配置

Router(config)#line console 0

```

Router(config-line)#password [password] //设置 con 口登录密码
Router(config-line)#login
Router(config-line)#exit
7 VTY 口的配置
Router(config)#line vty 0 4 //不同设备或不同 IOS 可能数量不同
Router(config-line)#password [password]
Router(config-line)#login
Router(config-line)#exit
8 配置以太网口
Router#conf t
Router(config)#int E0
Router(config-if)#ip address 192.168.1.1 255.255.255.0 //配置 IP 地址
Router(config-if)#no shutdown //激活该接口
9 配置串行接口（需要配置时钟频率）
Router#conf t
Router(config)#int S0
Router(config-if)#clock rate 64000 // DCE 设备配置时钟，DTE 设备不用配置
Router(config-if)#ip address 192.168.1.1 255.255.255.0 //配置 IP 地址
Router(config-if)#no shutdown
10 配置接口描述信息：
Router(config-if)#description link to std
11 配置主机名与 IP 地址映射：
Router(config)#ip host std 192.168.113.1
Ping std = ping 192.168.113.1
12 清除路由器启动配置：
Router#erase startup-config
13 保存路由器当前配置：
Router#copy running-config startup-config

```

## 二：常用 show 命令

查看路由器版本	Router#show version
查看路由器 flash	Router#show flash:
查看历史命令记录	Router#show history
查看接口物理相关信息	Router#show interfaces s0/0
查看接口协议相关信息	Router#show ip int s0/0
查看接口简要信息	Router#show ip int brief
查看路由器启动配置文件	Router#show running-config
查看路由器运行配置文件	Router#show running-config

---

查看当前登录的所有用户	Router#show users
查看路由器 ARP 表	Router#show arp

### 三：路由器模式详解

用户模式	Router>
特权模式	Router#
全局配置模式	Router(config)#
接口模式	Router(config-if)#
子接口模式	Router(config-subif)#
线路配置模式	Router(config-line)#
路由配置模式	Router(config-router)#

## LAB1-2 常用的 TCP/IP 测试命令-----PING 命令

实验目的：学习 ping 及扩展 ping 的用法

学习用 ping 命令检测网络故障

### 1 普通的 ping 命令的用法及结果

```
Router#ping 3.3.3.3
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/115/212 ms
```

### 2 扩张 ping

```
Router#ping
```

```
Protocol [ip]:
```

所使用的协议

```
Target IP address: 3.3.3.3
```

目标 IP 地址

```
Repeat count [5]:
```

PING 次数

```
Datagram size [100]:
```

数据包大小

```
Timeout in seconds [2]:
```

超时时间

```
Extended commands [n]: y
```

```
Source address or interface: 23.0.0.1
```

源地址

```
Type of service [0]:
```

服务类型

```
Set DF bit in IP header? [no]:
```

是否分片

```
Validate reply data? [no]:
```

```
Data pattern [0xABCD]:
```

```
Loose, Strict, Record, Timestamp, Verbose[none]:
```

疏松、严格的路由选择，时间戳

```
Sweep range of sizes [n]:
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
```

```
Packet sent with a source address of 23.0.0.1
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/85/216 ms
```

用 ping 命令可以检测的各个网络功能：

Ping 127.0.0.1-----ping 回环接口是测试基本的 TCP/IP 网络配置

Ping 主机 IP 地址----测试本地主机的 TCP/IP 地址配置

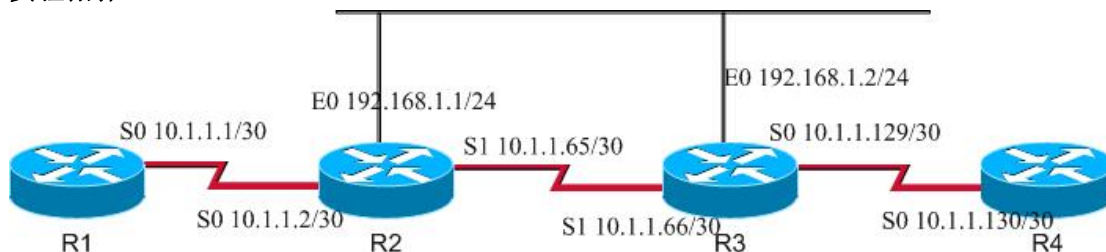
Ping 默认网关-----检测本地网络和其他网络的路由器是否可达

Ping 一个域名-----检测 DNS 是否正常工作

Ping 远端目的 IP 地址-----检测与远端主机的连通性

## LAB1-2 静态路由

实验拓扑:



实验目的: 掌握静态及默认路由的配置方法

实验一要求: R1 使用下一跳配置

R2 用出接口配置

R3 用出接口+下一跳

R4 上配置默认路由来访问其他网络

在 R2 上配置到网络 10.1.1.128/30 的负载均衡

实验步骤:

步骤 1 按如上拓扑做好底层基本 IP 配置, 并检测相邻设备之间的连通性

步骤 2 按实验要求在各台路由器上做配置

R1(config)#ip route 10.1.1.64 255.255.255.252 s0

R1(config)#ip route 10.1.1.128 255.255.255.252 s0

R2(config)#ip route 10.1.1.128 255.255.255.252 10.1.1.66

R2(config)#ip route 10.1.1.128 255.255.255.252 192.168.1.2

注意, 在配置负载均衡时, 如果要查看基于数据包的负载均衡, 需要关闭 Cisco CEF 功能:

R2(config)#no ip cef

R2(config)#int s1

R2(config-if)#no ip route-cache

R2(config)#int e0

R2(config-if)#no ip route-cache

R3(config)#ip route 10.1.1.0 255.255.255.252 s1 10.1.1.65

R4(config)#ip route 0.0.0.0 0.0.0.0 10.1.1.129

查看路由器路由表:

Router#show ip route

实验二要求: 在 R2 上配置到网络 10.1.1.128/30 的浮动静态路由, 要求正常情况下只使用以太网通信, 当以太网出现故障时, 自动切换到串行线路上

**实验步骤：**通过修改静态路由的管理距离，实现浮动静态路由功能

基于上面实验配置，修改 R2 部分配置，之前配置为：

```
R2(config)#ip route 10.1.1.128 255.255.255.252 10.1.1.66
```

修改为：

```
R2(config)#ip route 10.1.1.128 255.255.255.252 10.1.1.66 2
```

在命令后面加入修改管理距离的字段，将此路由的管理距离修改为 2。

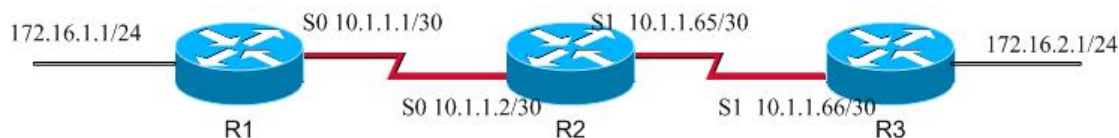
使用 show ip route 命令查看效果，当 e0 口 shutdown 后，自动加入管理距离为 2 的备份路由：

```
S      10.1.1.128 [2/0] via 10.1.1.66
```



## LAB2-2 RIPv1 的基本配置

实验拓扑:



**实验目的:** 掌握 RIPv1 的配置方法  
观察 RIPv1 的主类通告和自动汇总

**实验步骤:** 步骤 1 按如上拓扑做好底层配置, 并检查相邻设备的连通性  
步骤 2 在三台路由器上配置 RIPv1

```

R1(config)#router rip           //启动 RIP 进程
R1(config-router)#version 1     //指定使用版本为版本 1
R1(config-router)#network 172.16.0.0 //RIP 的主类通告
R1(config-router)#network 10.0.0.0
  
```

```

R2(config)#router rip
R2(config-router)#version 1
R2(config-router)#network 10.0.0.0
  
```

```

R3(config)#router rip
R3(config-router)#version 1
R3(config-router)#network 172.16.0.0
R3(config-router)#network 10.0.0.0
  
```

步骤 3 使用 `debug ip rip` 命令查看 RIPv1 的收发包的类型, 同时看 RIPv1 是以什么形式发更新包的, 使用 `debug ip udp` 命令验证 RIP 使用 UDP 协议, 端口号是 520。

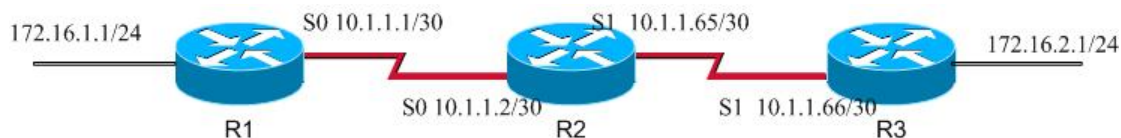
步骤 4 使用 `show ip route` 命令查看 RIP 的自动汇总情况  
使用 `show ip protocols` 命令查看路由协议相关信息

输出信息中的关键知识点:

计时器	<code>Sending updates every 30 seconds, next due in 27 seconds</code> <code>Invalid after 180 seconds, hold down 180, flushed after 240</code>
版本	<code>Default version control: send version 1, receive version 1</code>
自动汇总	<code>Automatic network summarization is in effect</code>
负载均衡路径	<code>Maximum path: 4</code>
管理距离	<code>Distance: (default is 120)</code>

## LAB2-3 IPv2 基本配置

实验拓扑:



实验目的: 掌握 IPv2 的配置方法

观察 IPv2 默认的自动汇总

掌握如何修改 RIP 的计时器

实验步骤: 步骤 1 按如上拓扑图做好底层配置, 并检查相邻设备的连通性

步骤 2 在三台路由器上做 IPv2 的配置

```
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#network 172.16.0.0
R1(config-router)# network 10.0.0.0
```

```
R2(config)#router rip
R2(config-router)#version 2
R2(config-router)# network 10.0.0.0
```

```
R3(config)#router rip
R3(config-router)#version 2
R3(config-router)# network 172.16.0.0
R3(config-router)# network 10.0.0.0
```

步骤 3 用 show ip route 查看路由表, 观察是否存在自动汇总

步骤 4 关闭自动汇总, 再看路由表有什么变化

```
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#no auto-summary
```

```
R2(config)#router rip
R2(config-router)#version 2
R2(config-router)#no auto-summary
```

```
R3(config)#router rip
R3(config-router)#version 2
R3(config-router)#no auto-summary
```

步骤 5 在 R1 上修改 RIP 的四个计时器:

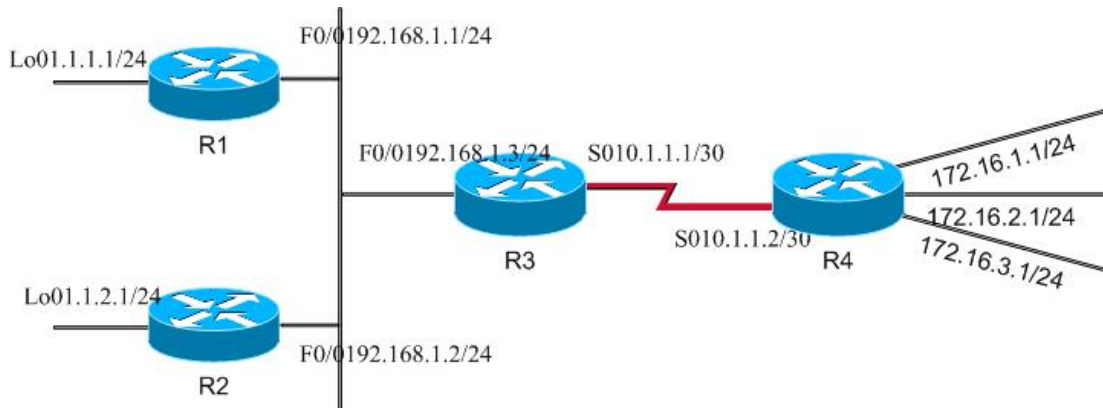
```
R1(config)#router rip
R1(config-router)#version 2
```

```
R1(config-router)#timers basic 20 120 120 160
```

四个值分别代表 update, Invalid, hold down, flushed

## LAB2-3 RIP 扩展实验

实验拓扑:



**实验目的:** 掌握 RIPv2 的手工汇总  
掌握 RIPv2 的被动接口和单播更新的配置  
掌握 RIPv2 认证的配置

**实验要求:** 只有 R1 可以学习到 R3 的路由信息, R2 学习不到  
R2 上配置缺省路由来到达其他网段  
R3 R4 之间启用 RIP MD5 认证  
在 R4 上对 172.16.1.0-172.16.3.0 进行手动汇总  
关闭 R4 的水平分割, 并用 debug 命令查看  
在上述条件下使全网互通

**实验步骤:** 步骤 1 按如上拓扑做好底层配置, 并检测相邻设备的连通性

步骤 2 在 R1、R3、R4 上启 RIP 协议

```
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#no auto-summary
R1(config-router)#network 1.0.0.0
R1(config-router)# network 192.168.1.0
```

```
R3(config)#router rip
R3(config-router)#version 2
R3(config-router)#no auto-summary
R3(config-router)# network 192.168.1.0
R3(config-router)# network 10.0.0.0
```

```
R4(config)#router rip
R4(config-router)#version 2
R4(config-router)#no auto-summary
R4(config-router)# network 172.16.0.0
```

```
R4(config-router)# network 10.0.0.0
```

步骤 3 在 R2 上配置默认路由

```
R2(config)#ip route 0.0.0.0 0.0.0.0 192.168.1.3
```

步骤 4 在 R3 上配置被动接口和单播更新,使得 R2 无法从 R3 学习路由信息

```
R3(config)#router rip
```

```
R3(config-router)#passive-interface E0
```

```
R3(config-router)#neighbor 192.168.1.1
```

可以使用 `debug ip rip` 命令观察 R3 的单播更新

步骤 5 在 R3、R4 之间做 MD5 认证

创建认证数据库:

```
R3(config)#key chain stsd
```

```
R3(config-keychain)#key 1
```

```
R3(config-keychain-key)#key-string cisco
```

```
R3(config-keychain-key)#exit
```

```
R3(config-keychain)#exit
```

```
R4(config)#key chain stsd
```

```
R4(config-keychain)#key 1
```

```
R4(config-keychain-key)#key-string cisco
```

```
R4(config-keychain-key)#exit
```

```
R4(config-keychain)#exit
```

在接口上启动 `rip md5` 认证:

```
R3(config)#int s0
```

```
R3(config-if)#ip rip authentication mode md5
```

```
R3(config-if)#ip rip authentication key-chain stsd
```

```
R4(config)#int s0
```

```
R4(config-if)#ip rip authentication mode md5
```

```
R4(config-if)#ip rip authentication key-chain stsd
```

步骤 6 在 R4 上做汇总

```
R4(config)#int s0
```

```
R4(config-if)#ip summary-address rip 172.16.0.0 255.255.252.0
```

#注意 RIP 只支持子网汇总, 不支持超网汇总

步骤 7 关闭 R4 s0 口上的水平分割并用 `debug ip rip` 查看结果

```
R4(config)#int s0
```

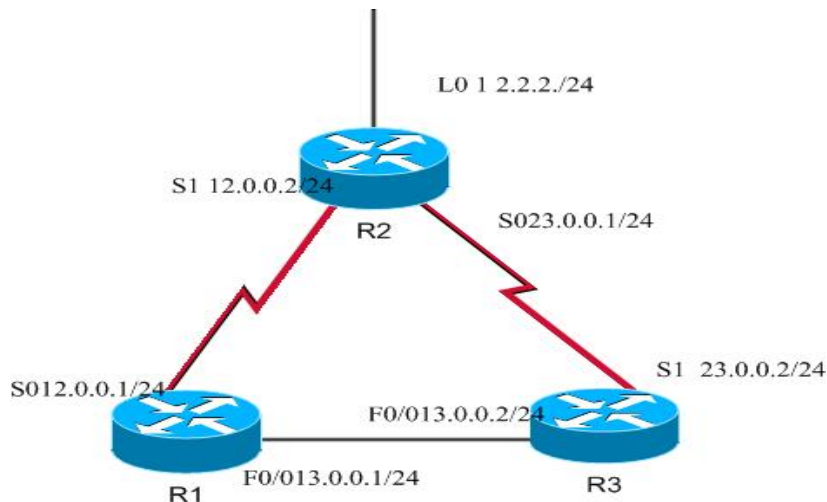
```
R4(config-if)#no ip split-horion
```

```
R4(config-if)#end
```

```
R4#debug ip rip
```

## LAB2-5 RIP 的等价负载均衡

实验拓扑:



**实验目的:** 通过实验现象熟悉负载均衡的概念, 及如何实现 RIP 的负载均衡。

掌握如何修改 RIP 的负载均衡条目数量

**实验要求:** 在全网互通的前提下实验 RIP 的负载均衡

**实验步骤:** 步骤 1 按如上拓扑图做好底层配置

步骤 2 在各个路由器上起 RIP, 是全网互通

```
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#no auto-summary
R1(config-router)#network 13.0.0.0
R1(config-router)#network 12.0.0.0

R2(config)#router rip
R2(config-router)#version 2
R2(config-router)#no auto-summary
R2(config-router)# network 23.0.0.0
R2(config-router)# network 12.0.0.0
R2(config-router)# network 2.2.2.0

R3(config)#router rip
R3(config-router)#version 2
R3(config-router)#no auto-summary
R3(config-router)# network 13.0.0.0
R3(config-router)# network 23.0.0.0
```

步骤 3 查看 R2 的路由表, 看是否有两条到达 13.0.0.0 网络的路由信息

步骤 4 如果要看到基于数据包的负载均衡, 需要关闭路由器的 CEF 功能:

```
R2(config)#int s1
R2(config-if)#no ip cef
R2(config-if)#no ip route-cache
```

```
R2(config)#int s0
R2(config-if)#no ip cef
R2(config-if)#no ip route-cache
```

步骤 5 做基于源地址是 2.2.2.2，目的地址是 13.0.0.1 的扩展 ping，并且用 debug ip packet 命令查看 RIP 等价负载均衡的特性。

步骤 6 修改 RIP 的最大负载均衡条目数量：

使用命令 show ip protocols 可以查看路由选择协议默认的最大负载均衡条目数量：

```
Maximum path: 4
```

可以看到默认是最多在 4 条线路上进行负载均衡。

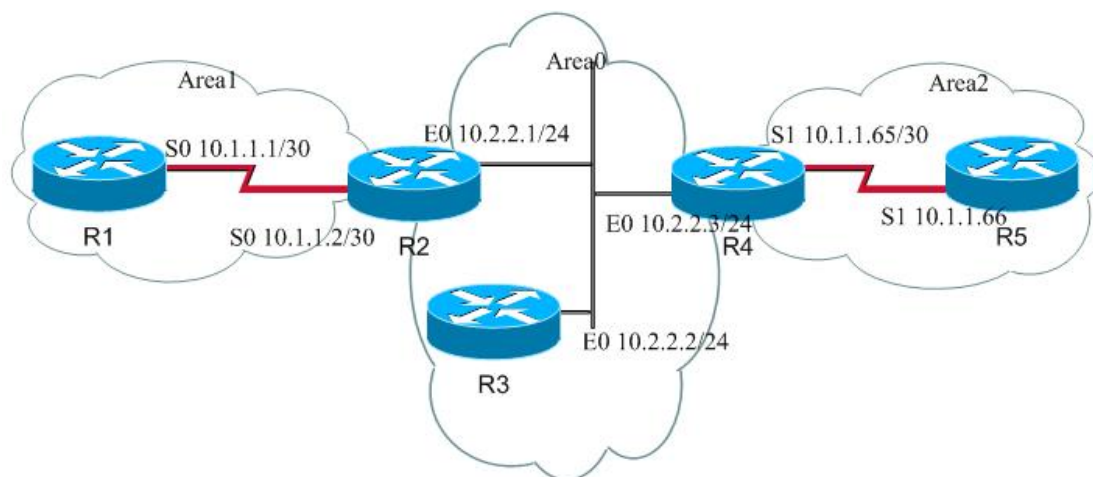
修改成 6 条：

```
R2(config)#router rip
R2(config-router)#maximum-paths 6
```

不同 IOS 版本可能支持的最大数量不同，12.3 之前最多 6 条，之后最多 16 条。默认都是 4 条。

**LAB2-6 OSPF 的基本配置及 DR /BDR 选举的实验**

实验拓扑:



**实验目的:** 掌握 OSPF 的基本配置  
 掌握手工指定 RID  
 掌握如何修改 OSPF 的接口优先级  
 观察 DR BDR 选举的过程

**实验要求:** R3 当选为 DR  
 R2 为 BDR  
 R4 不参与选举  
 全网互通

**实验步骤:** 步骤 1 按如上拓扑做好底层配置, 并检测相邻设备的连通性

步骤 2 在三台路由器上起 OSPF 协议

```
R1(config)#interface lo0 //通过环回接口限定 ospf 的 rid
R1(config-if)#ip address 1.1.1.1 255.255.255.0
R1(config)#router ospf 100
R1(config-router)#router-id 1.1.1.1 //手工指定 ospf 的 rid
R1(config-router)#network 10.1.1.0 0.0.0.3 area 1
```

```
R2(config)#interface lo0
R2(config-if)#ip address 2.2.2.2 255.255.255.0
R2(config)#router ospf 100 //注意 OSPF 进程号的本地特性
R2(config-router)#router-id 2.2.2.2
R2(config-router)#network 10.1.1.0 0.0.0.3 area 1
R2(config-router)#network 10.2.2.0 0.0.0.255 area 0
```

```
R3(config)#interface lo0
R3(config-if)#ip address 3.3.3.3 255.255.255.0
```



```
R3(config)#router ospf 100
R3(config-router)#router-id 3.3.3.3

R3(config-router)#network 10.2.2.0 0.0.0.255 area 0

R4(config)#interface lo0
R4(config-if)#ip address 4.4.4.4 255.255.255.0
R4(config)#router ospf 100
R4(config-router)#router-id 4.4.4.4
R4(config-router)#network 10.1.1.64 0.0.0.3 area 2
R4(config-router)#network 10.2.2.0 0.0.0.255 area 0
```

步骤 3 用 `show ip ospf neighbors` 查看 DR BDR 的情况

步骤 4 R4 的 E0 接口优先级改为 0，使其不参与 DR/BDR 选举：

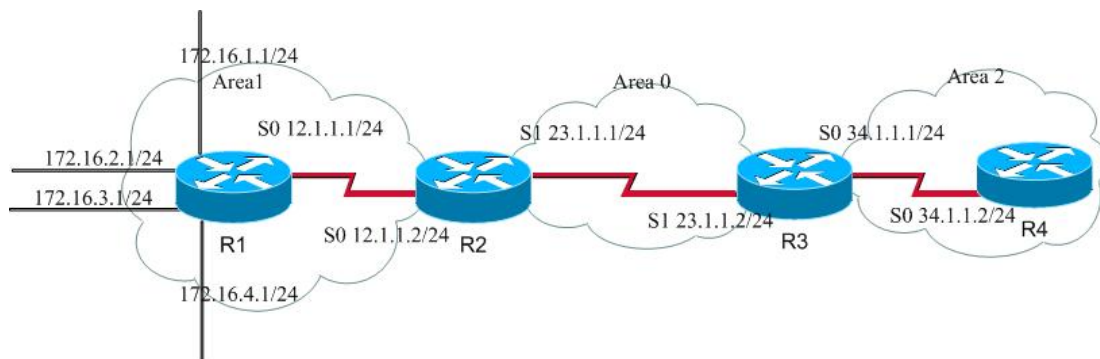
```
R4(config)#int E0
R4(config-if)#ip ospf priority 0
```

注意，因为 R2 与 R3 默认 E0 口优先级均为 1，所以通过 RID 进行 DR/BDR 选举，在本实验中，因为 R3 的 RID 为 3.3.3.3，大于 R2 的 2.2.2.2，所以不需要修改 R2、R3 的优先级，如果需要保证 R3 的 DR 地位，可以将 R3 的优先级设置成较高值。

步骤 5 使用 `clear ip ospf process` 重启 ospf 进程之后观察 DR BDR 的情况及接口的优先级。本步骤可以省略，但如果配置速度过慢，因为 OSPF DR 的非抢占特性，先配置的路由器可能会成为 DR，此时就需要重启 OSPF 进程。

## LAB2-7 OSPF 扩展实验

实验拓扑:



**实验目的:** 掌握 OSPF 的区域汇总

掌握 OSPF 的简单口令以及 MD5 认证

掌握如何修改 hello、dead 间隔及 cost 的值

**实验要求:** 做 area1 在向 Area0 通告时进行汇总

在 R1 与 R2 之间做区域的简单口令认证

在 R2 与 R3 之间做区域的 MD5 认证

修改 R3 R4 之间的 hello、dead 间隔为 5、20

修改 R4 的 s0 口的 cost 为 80

在所有路由器上开启 OSPF 邻居日志

**实验步骤:** 步骤 1 按如上拓扑做好底层配置，并检测相邻设备之间的连通性

步骤 2 在网络中各个路由器上启 OSPF 协议

```
R1(config)#router ospf 100
R1(config-router)#network 172.16.1.0 0.0.0.255 area 1
R1(config-router)#network 172.16.2.0 0.0.0.255 area 1
R1(config-router)#network 172.16.3.0 0.0.0.255 area 1
R1(config-router)#network 172.16.4.0 0.0.0.255 area 1
R1(config-router)#network 12.1.1.0 0.0.0.255 area 1
R1(config-router)#log-adjacency-changes
//打开 OSPF 邻居状态日志
```

```
R2(config)#router ospf 100
R2(config-router)#network 12.1.1.0 0.0.0.255 area 1
R2(config-router)#network 23.1.1.0 0.0.0.255 area 0
R2(config-router)#log-adjacency-changes
```

```
R3(config)#router ospf 100
R3(config-router)#network 23.1.1.0 0.0.0.255 area 0
```

```
R3(config-router)#network 34.1.1.0 0.0.0.255 area 2
R3(config-router)#log-adjacency-changes
```

```
R4(config)#router ospf 100
R4(config-router)#network 34.1.1.0 0.0.0.255 area 2
R4(config-router)#log-adjacency-changes
```

步骤 2 在 R2 上对区域 1 进行路由汇总

```
R2(config)#router ospf 100
R2(config-router)#area 1 range 172.16.0.0 255.255.248.0
```

步骤 3 在 R2、R3 之间做区域的 MD5 认证

```
R2(config)#int s1
R2(config-if)#ip ospf message-digest-key 1 md5 cisco
//定义认证密钥
```

```
R2(config)#router ospf 100
R2(config-router)#area 0 authentication message-digest
```

```
R3(config)#int s1
R3(config-if)#ip ospf message-digest-key 1 md5 cisco
R3(config)#router ospf 100
R3(config-router)#area 0 authentication message-digest
```

步骤 4 在 R1 与 R2 之间配置简单口令认证

```
R1(config)#int s0
R1(config-if)#ip ospf authentication-key stsd
//配置密码为 stsd
R1(config-router)#area 1 authentication
//在区域 1 中开启认证
```

```
R2(config)#int s0
R2(config-if)#ip ospf authentication-key stsd
R2(config-router)#area 1 authentication
```

步骤 5 修改 hello、dead 间隔的值

```
R3(config)#int s0
R3(config-if)#ip ospf hello-interval 5
R3(config-if)#ip ospf dead-interval 20
```

```
R4(config)#int s0
```

```
R4(config-if)#ip ospf hello-interval 5
R4(config-if)#ip ospf dead-interval 20
```

步骤6 修改 R4 的 s0 接口的 cost

```
R4(config)#int s0
R4(config-if)#ip ospf cost 80
```

使用命令 `show ip ospf interface s0` 查看修改结果，包括 RID,优先级，所属区域,计时器以及开销值等信息。

OSPF 常用 show 命令：

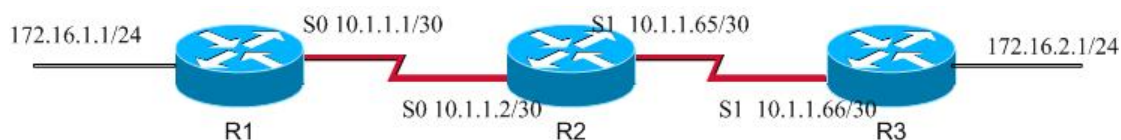
Router#show ip protocol	显示路由器上运行的所有路由选择协议
Router#show ip route	显示完整的路由表
Router#show ip ospf	显示OSPF路由进程的基本信息
Router#show ip ospf interface	显示接口的OSPF相关信息
Router#show ip ospf interface fastethernet 0/0	显示特定接口Fa0/0的OSPF信息
Router#show ip ospf border-routers	显示所有边界路由器信息
Router#show ip ospf neighbor	显示OSPF的邻居表
Router#show ip ospf neighbor detail	显示OSPF邻居的详细信息
Router#show ip ospf database	显示OSPF的数据库

OSPF 常用 debug 命令

Router#debug ip ospf events	显示所有OSPF事件
Router#debug ip ospf adjacency	显示OSPF邻居建立过程，可用来观察DR/BDR选举
Router#debug ip ospf packets	显示所有OSPF数据包

## LAB2-9 EIGRP 基本配置

实验拓扑:



实验目的: 掌握 EIGRP 的基本配置  
观察 EIGRP 的自动汇总情况

实验步骤: 步骤 1 按如上拓扑做好底层配置, 并检测相邻设备之间的连通性  
步骤 2 在三台路由器上启 EIGRP 协议

```
R1(config)#router eigrp 100 //注意 EIGRP AS 号的全局意义
R1(config-router)#network 172.16.1.0 0.0.0.255
R1(config-router)#network 10.1.1.0 0.0.0.3
```

```
R2(config)#router eigrp 100
R2(config-router)#network 10.1.1.64 0.0.0.3
R2(config-router)#network 10.1.1.0 0.0.0.3
```

```
R3(config)#router eigrp 100
R3(config-router)#network 172.16.1.0 0.0.0.255
R3(config-router)#network 10.1.1.64 0.0.0.3
```

步骤 3 用 `show ip route` 命令查看路由表, 观察 EIGRP 的默认自动汇总特性

步骤 4 关闭路由器上的自动汇总, 然后再看其路由表有什么变化

```
R1(config)#router eigrp 100
R1(config-router)#no auto-summary
```

```
R2(config)#router eigrp 100
R2(config-router)#no auto-summary
```

```
R3(config)#router eigrp 100
R3(config-router)#no auto-summary
```

## LAB2-10 EIGRP 的扩展实验

实验拓扑：



**实验目的：** 掌握 EIGRP 的手工汇总  
 掌握 EIGRP 的 MD5 认证  
 掌握如何修改 K 值  
 掌握如何配置 EIGRP 所占用的最大带宽

**实验要求：** 在 R1 上做汇总  
 在 R2 、 R3 之间做 EIGRP 的 MD5 认证  
 修改 R1 R2 之间的默认 K 值，修改为 1 0 0 0 0  
 修改 R2 S1 接口上 EIGRP 所占用的最大带宽为 30%

**实验步骤：** 步骤 1 按如上拓扑做好底层配置，并检测相邻设备之间的连通性

步骤 2 在三台路由器上启 EIGRP 协议

```
R1(config)#router eigrp 100
R1(config-router)#no auto-summary
R1(config-router)#network 192.168.1.0 0.0.0.255
R1(config-router)#network 192.168.2.0 0.0.0.255
R1(config-router)#network 192.168.3.0 0.0.0.255
R1(config-router)#network 12.1.1.0 0.0.0.255
```

```
R2(config)#router eigrp 100
R2(config-router)#no auto-summary
R2(config-router)#network 12.1.1.0 0.0.0.255
R2(config-router)#network 23.1.1.0 0.0.0.255
```

```
R3(config)#router eigrp 100
R3(config-router)#no auto-summary
R3(config-router)#network 23.1.1.0 0.0.0.255
```

步骤 3 在 R1 上做汇总

```
R1(config)#int s0
R1(config-if)#ip summary-address eigrp 100 192.168.0.0
255.255.252.0
```

步骤 4 在 R2 R3 之间做 md5 认证

```
R2(config)#key chain stsd
R2(config-keychain)#key 1
```

```
R2(config-keychain-key)#key-string cisco
R2(config)#int s1
R2(config-if)#ip authentication mode eigrp 100 md5
R2(config-if)#ip authentication key-chain eigrp 100 stsd
```

```
R3(config)#key chain stsd
R3(config-keychain)#key 1
R3(config-keychain-key)#key-string cisco
R3(config)#int s1
R3(config-if)#ip authentication mode eigrp 100 md5
R3(config-if)#ip authentication key-chain eigrp 100 stsd
```

#### 步骤 5 修改 R1 R2 之间的 k 值

```
R1(config)#router eigrp 100
R1(config-router)#metric weights 0 1 0 0 0 0
#第一个 0 为 tos 值，没有使用过，始终为 0。后面是顺序的 K1-K5
```

```
R2(config)#router eigrp 100
R2(config-router)#metric weights 0 1 0 0 0 0
```

#### 步骤 6 修改 R2 s1 接口 EIGRP 所占用的最大带宽为 30%

```
R2(config)#int s1
R2(config-if)#ip bandwidth-percent eigrp 100 30
//单位为%
```

默认情况下 EIGRP 所占用链路的最大带宽为 50%

#### 常用 EIGRP show 命令

Router#show ip eigrp neighbors	显示 EIGRP 邻居
Router#show ip eigrp neighbors detail	显示 EIGRP 邻居详细信息
Router#show ip eigrp interfaces	查看接口的 EIGRP 信息
Router#show ip eigrp topology	查看 EIGRP 拓扑表
Router#show ip route eigrp	显示所有 EIGRP 路由

#### 常用 Debug 命令:

Router#debug eigrp packet	显示所有 EIGRP 数据包
---------------------------	----------------

---

Router#debug eigrp neighbor	显示EIGRP邻居状态过程
-----------------------------	---------------



## LAB2-11 标准与扩展 ACL 实验

### 一 标准访问控制列表实验：

实验拓扑：



实验目的：掌握标准与扩展 ACL 的配置

实验要求：拒绝 R1 到 R3 的所有流量

实验步骤：步骤 1 按如上拓扑做好底层配置，并检测相邻设备之间的连通性

步骤 2 起静态路由，使全网互通

```
R1(config)#ip route 10.1.1.64 255.255.255.252 10.1.1.2
```

```
R3(config)#ip route 10.1.1.0 255.255.255.252 10.1.1.65
```

步骤 3 在 R3 上做标准的 ACL 使 R1 不能访问 R3

```
R3(config)#access-list 1 deny 10.1.1.1 0.0.0.0
```

或者使用命令

```
R3(config)#access-list 1 deny host 10.1.1.1
```

因为访问控制列表默认有一个拒绝所有的隐含条目，所以需要在最后加入一条：

```
R3(config)#access-list 1 permit any
```

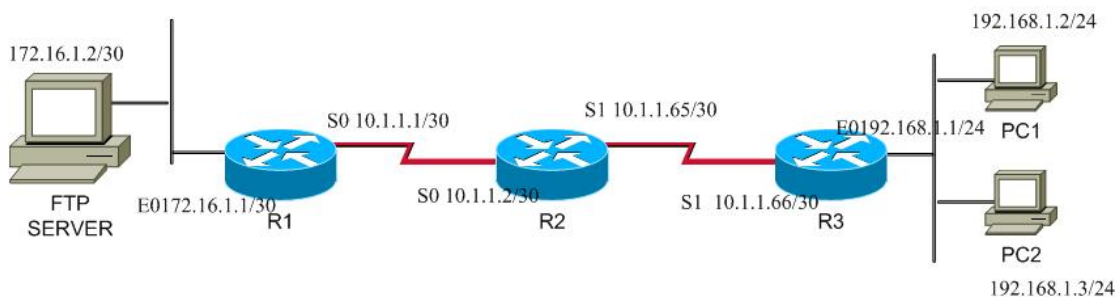
标准 ACL 要放置在离目标近的地方，所以配置在 R3 的 S1 的入向上面：

```
R3(config)#int s1
```

```
R3(config-if)#ip access-group 1 in
```

### 二 扩展访问控制列表实验：

实验拓扑：



实验目的：掌握扩展访问控制列表的配置

掌握如何使用扩展的访问控制列表实现网络的安全性

实验要求：拒绝任何来自 192.168.1.0 网络的 icmp 流量

只有 PC1 可以访问 FTP 服务器

实验步骤：步骤 1 按如上拓扑在做好底层配置，在三台路由器上启 RIPv2 协议，使之互通

```
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#no auto-summary
R1(config-router)#network 10.0.0.0
R1(config-router)#network 172.16.0.0
```

```
R2(config)#router rip
R2(config-router)#version 2
R2(config-router)#no auto-summary
R2(config-router)#network 10.0.0.0
```

```
R3(config)#router rip
R3(config-router)#version 2
R3(config-router)#no auto-summary
R3(config-router)#network 10.0.0.0
R3(config-router)#network 192.168.1.0
```

步骤 3 在 R3 上做扩展的 ACL，拒绝来自 192.168.1.0 网络的 icmp 流量

```
R3(config)#access-list 102 deny icmp 192.168.1.0 0.0.0.255 any
R3(config)#access-list 102 permit ip any any
R3(config)#int e0
R3(config-if)#ip access-group 102 in
```

步骤 4 在 R1 上做扩展的 ACL 使得只有 PC1 可以访问 FTP 服务器

注意 FTP 使用两个端口号，20 与 21：

```
R1(config)#access-list 110 permit tcp 192.168.1.1 0.0.0.0
172.16.1.2 0.0.0.0 eq 21
R1(config)#access-list 110 permit tcp 192.168.1.1 0.0.0.0
172.16.1.2 0.0.0.0 eq 20
R1(config)#int s0
R1(config-if)#ip access-group 110 in
```

### 三 命名的访问控制列表试验：

实验需求与拓扑图通上，将访问控制列表转换成命名的 ACL：

```
R3(config)#ip access-list extended deny_icmp
R3(config-ext-nacl)#deny icmp 192.168.1.0 0.0.0.255 any
R3(config-ext-nacl)#permit ip any any
R3(config)#int e0
R3(config-if)#ip access-group deny_icmp in

R1(config)#ip access-list extended deny_ftp
R1(config-ext-nacl)#permit tcp 192.168.1.1 0.0.0.0 172.16.1.2
0.0.0.0 eq 20
```

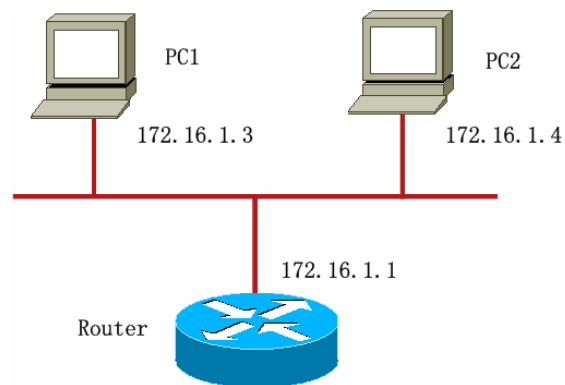
```
R1(config-ext-nacl)#permit tcp 192.168.1.1 0.0.0.0 172.16.1.2 0.0.0.0 eq 21
```

```
R1(config)#int s0  
R1(config-if)#ip access-group deny_ftp in
```

命名 ACL 的最大优点在于可以修改其中任意一条，而使用编号的 ACL 则不能。

#### 四：使用 ACL 对 vty 线路进行限制：

实验拓扑：



**实验需求：**在 Router 的 VTY 线路上通过 ACL 限制访问  
只有 PC1 能够远程登录 Router

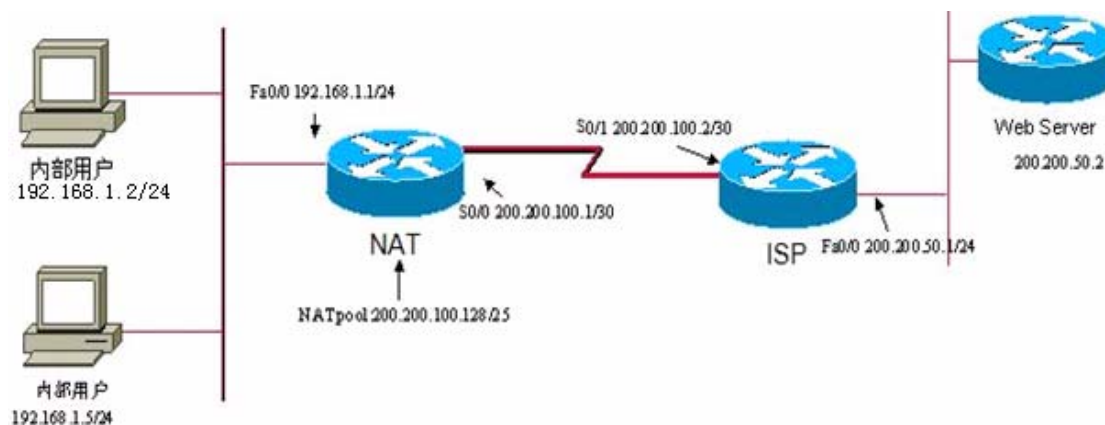
**实验步骤：**

```
Router(config)#access-list 1 permit host 172.16.1.3  
//因为 ACL 有隐含拒绝特性，所以不需要显式拒绝 PC2
```

```
Router(config)#line vty 0 15  
Router(config-line)#password stsd  
Router(config-line)#login  
Router(config-line)#access-class 1 in
```

## LAB2-12 静态 NAT、动态 NAT

实验拓扑:



**实验目的:** 熟悉网络地址转换协议  
 掌握静态 NAT 和 动态 NAT 的配置  
 分析静态 NAT 和 动态 NAT 的区别  
 使用 show 命令来检查 NAT 的运行情况

**实验要求:** 按拓扑图来配置静态 NAT 和 动态 NAT

**实验步骤:** 步骤 1 按如上拓扑做好底层配置, 并检验相邻设备之间的连通性

步骤 2 在 NAT 上配置一条到 internet 的缺省路由

```
NAT(config)#ip route 0.0.0.0 0.0.0.0 200.200.100.2
```

因为内部主机与 internet 通信时转换成 200.200.100.128/25 网络的公有 IP, 所以在 ISP 上配置一条到 200.200.100.128/25 网络的静态路由

```
ISP(config)#ip route 200.200.100.128 255.255.255.128 200.200.100.1
```

步骤 3 创建一条定义了所有内部用户的标准的访问控制列表

```
NAT(config)#access-list 1 permit 192.168.1.0 0.0.0.255
```

步骤 4 静态 NAT 是一对一的, 即一个内部主机对应一个地址池中的公有地址, 所以静态 NAT 的配置如下

```
NAT(config)#ip nat inside source static 192.168.1.2 200.200.100.129
NAT(config)#ip nat inside source static 192.168.1.5 200.200.100.130
```

步骤 5 因为静态 NAT 要位所有内部主机配置地址转换, 所以提出了一种简单的替代方法, 配置地址池, 即动态 NAT

```
NAT(config)#ip nat pool public 200.200.100.129 200.200.100.254 netmask 255.255.255.128
```

```
NAT(config)#ip nat inside source list 1 pool public
```

步骤 6 指定 NAT 的内部接口和外部接口

```
NAT(config)#int f0/0
```

```
NAT(config-if)#ip nat inside
```

```
NAT(config)#int s0/0
```

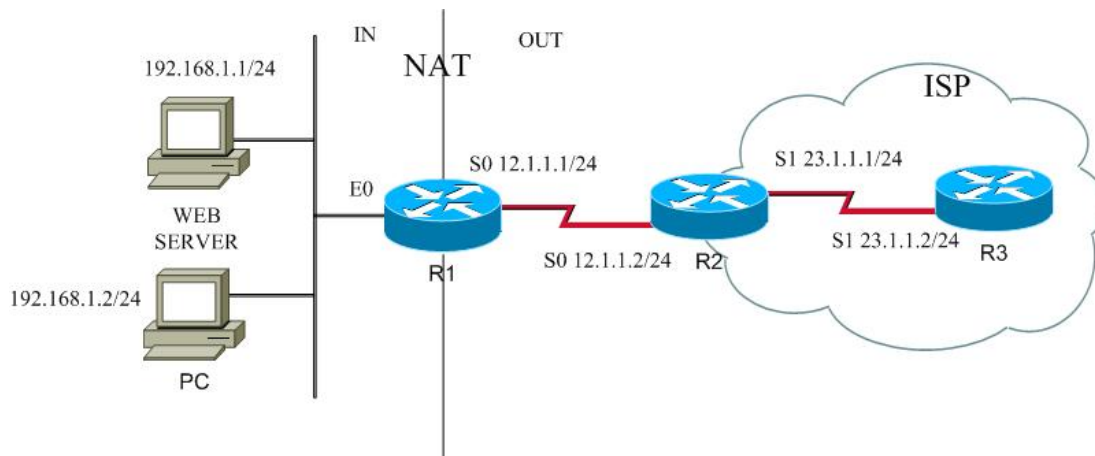
```
NAT(config-if)#ip nat outside
```

常用查看命令：

Router#show ip nat translations	显示NAT转换表
Router#show ip nat statistics	显示当前 NAT 状态
Router#clear ip nat translations*	在超时前清楚所有NAT转换

## LAB2-13 NAT+DHCP+ACL 综合实验

实验拓扑:



**实验目的:** 掌握 NAT 的原理及其实现方法  
掌握 DHCP 的原理及其配置

**实验要求:** 在 R1 上做静态 NAT 使 WEB 服务器可以被访问, E0 口 IP 地址作为内网网关 192.168.1.254

在 R1 上做 PAT 使的内网可以访问 Internet

R1 配置成 DHCP 服务器, 为 192.168.1.0 网络分配 IP 地址。

R2 上使用 ACL 使得内网的地址不会出现在外网中

DHCP 需求: 地址池网段-192.168.1.0/24

排除地址: 192.168.1.1~192.168.1.50

DNS: 218.30.19.40

网关: 192.168.1.254

Domain: am007.com

租约期: 无限

实验步骤:

步骤 1 将 R1 配置成 DHCP 服务器

```
R1(config)#ip dhcp pool std
R1(dhcp-config)#network 192.168.1.0 255.255.255.0
R1(dhcp-config)#default-router 192.168.1.1
R1(dhcp-config)#dns-server 218.30.19.40
R1(dhcp-config)#domain-name am007.com
R1(dhcp-config)#lease infinite
配置排除地址:
R1(config)#ip dhcp excluded-address 192.168.1.1
192.168.1.50
```

使路由器接口通过 DHCP 获取 IP 地址可以使用命令 `ip address dhcp`

步骤 2 在 R1 上做静态 NAT，使 WEB 服务器可以通过 12.1.1.3 被访问

```
R1(config)#ip nat inside source static 192.168.1.1 12.1.1.3
```

步骤 3 在 R1 上做 PAT，使内网可以访问 Internet

```
R1(config)#access-list 1 deny 192.168.1.1 0.0.0.0
```

```
R1(config)#access-list 1 permit 192.168.1.0 0.0.0.255
```

```
R1(config)#ip nat inside source list 1 interface s0 overload
```

步骤 4 指定 NAT 的内部接口和外部接口

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

```
R1(config-if)#ip nat inside
```

```
R1(config)#interface s0
```

```
R1(config-if)#ip nat outside
```

步骤 5 在 R2 上做 ACL，防止内网地址在外网中出现

```
R2(config)#access-list 1 deny 192.168.1.0 0.0.0.255
```

```
R2(config)#access-list 1 permit any
```

```
R2(config)#interface s0
```

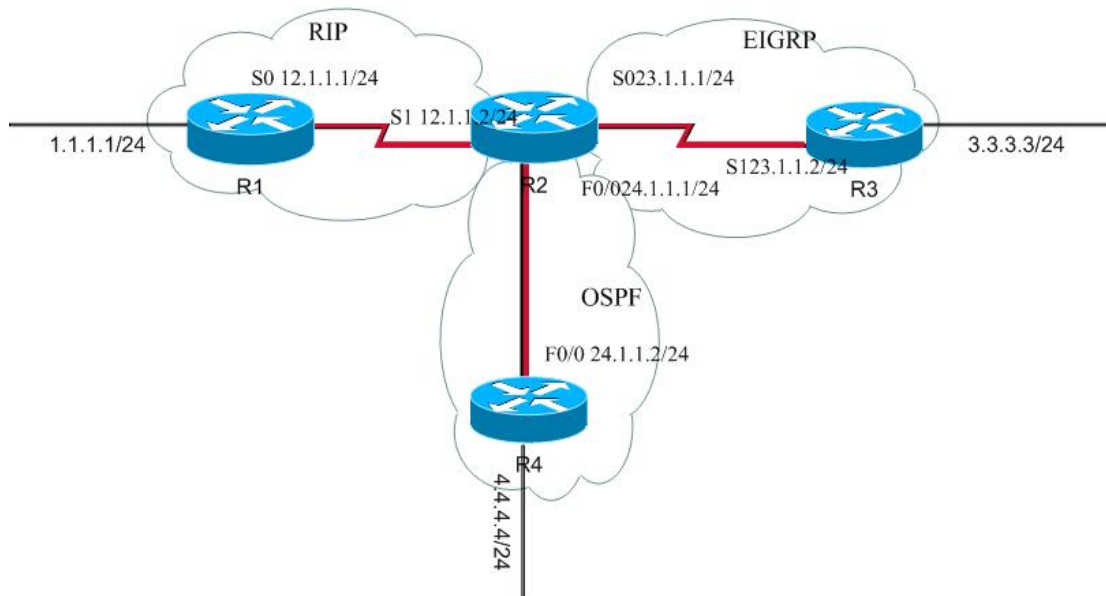
```
R2(config-if)#ip access-group 1 in
```

常用 DHCP 以及 NAT 查看命令：

Router#show ip nat translations	显示NAT转换表
Router#show ip nat statistics	显示NAT状态
Router#clear ip nat translations*	清除所有NAT转换
Router#show ip dhcp binding	显示已有的DHCP绑定信息
Router#show ip dhcp database	显示当前使用的DHCP数据库
Router#show ip dhcp server Statistics	列出DHCP服务运行状态信息

## LAB2-15 路由重发布实验

实验拓扑



实验目的：掌握不同路由协议之间重发布的配置方法

实验要求：全网互通，不能使用静态路由以及缺省路由

实验步骤：步骤 1 按 LAB 2 的配置方法做好底层配置，

步骤 2 在 R1 上起 RIP v2，R2 上起 RIP ,EIGRP ,OSPF 三种协议，R3 上 EIGRP，R4 上 OSPF。

```

R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#network 1.1.1.0
R1(config-router)#network 12.1.1.0

R2(config)#router rip
R2(config-router)#version 2
R2(config-router)#network 23.1.1.0
R2(config-router)#network 12.1.1.0
R2(config-router)#network 24.1.1.0
R2(config)#router eigrp 100
R2(config-router)#network 12.1.1.0 0.0.0.255
R2(config-router)#network 23.1.1.0 0.0.0.255
R2(config-router)#network 24.1.1.0 0.0.0.255
R2(config)#router ospf 100

```



```
R2(config-router)#network 12.1.1.0 0.0.0.255 area 0
R2(config-router)#network 23.1.1.0 0.0.0.255 area 0
R2(config-router)#network 24.1.1.0 0.0.0.255 area 0
```

```
R3(config)#router eigrp 100
R3(config-router)#network 23.1.1.0 0.0.0.255
R3(config-router)#network 3.3.3.0 0.0.0.255
```

```
R4(config)#router ospf 100
R4(config-router)#network 24.1.1.0 0.0.0.255 area 0
R4(config-router)#network 4.4.4.0 0.0.0.255 area 0
```

步骤 3 在 R2 上做重发布的配置

```
R2(config)#router rip
R2(config-router)#redistribute eigrp 100 metric 2
R2(config-router)#redistribute ospf 100 metric 2

R2(config)#router eigrp 100
R2(config-router)#redistribute rip metric 2000 1 255 1
1500 //EIGRP 的复合度量值
R2(config-router)#redistribute ospf 100 metric 2000 1 255
1500

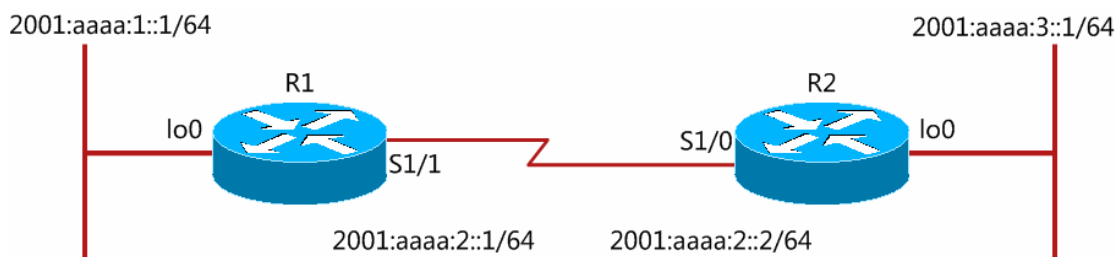
R2(config)#router ospf 100
R2(config-router)#redistribute rip metric 64 subnets
R2(config-router)#redistribute eigrp 100 metric 64
subnets
```

#注意在向 ospf 重发布是需要加上 subnets，否则只重发布主类网

重发布完成后注意查看 EIGRP 的路由表，观察 EIGRP 的管理距离的变化。

## LAB2-16 IPv6 实验

实验拓扑:



**实验目的:** 掌握 IPv6 基本编址原理  
掌握如何在 cisco 路由器上面配置 IPv6 地址  
掌握如何配置 IPv6 静态路由

**实验需求:** 在 R1、R2 上面配置 IPv6 地址  
配置 IPv6 静态路由, 使全网互通

实验步骤:

步骤一: 在 R1、R2 上配置 IPv6 地址:

```
R1(config)#int lo0
R1(config-if)#ipv6 address 2001:aaaa:1::1/64
R1(config-if)#exit
R1(config)#int s1/1
R1(config-if)#ipv6 address 2001:aaaa:2::1/64
R1(config-if)#clock rate 64000
R1(config-if)#no shutdown
R1(config-if)#end
```

```
R2(config)#int s1/0
R2(config-if)#ipv6 address 2001:aaaa:2::1/64
R2(config-if)#clock rate 64000
R2(config-if)#no shutdown
R2(config-if)#exit
R2(config)#int lo0
R2(config-if)#ipv6 address 2001:aaaa:3::1/64
R2(config-if)#end
```

步骤二: 开启 R1、R2 的 IPv6 路由功能

```
R1#conf t
R1(config)#ipv6 unicast-routing
```

```
R2#conf t
```

```
R2(config)#ipv6 unicast-routing
```

步骤三：在 R1、R2 上面配置 IPv6 静态路由

```
R1(config)#ipv6 route 2001:aaaa:3::/64 s1/1
```

```
R2(config)#ipv6 route 2001:aaaa:1::/64 s1/0
```

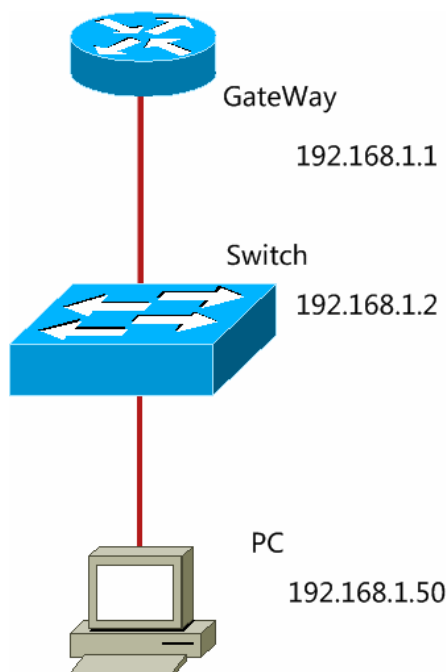
步骤四：通过 ping 命令验证网络连通性

IPv6 常用查看命令：

show ipv6 route	查看 IPv6 路由表
show ipv6 protocols	查看当前运行的 IPv6 协议
show ipv6 interface	查看接口的 IPv6 信息

## LAB3-1 交换机基本操作以及端口安全

实验拓扑：



**实验目的：** 掌握交换机基本操作  
掌握交换机端口安全配置

**实验需求：** 配置交换机主机名，密码，网关等。  
在交换机 fa0/3 口上面配置接口安全，规定最大学习 MAC 地址数量为 10，其余丢弃。

**实验步骤：**

### 步骤一：基本配置

主机名，密码等基本配置与路由器相同，在此不做过多讲解。

```
Switch(config)#ip default-gateway 192.168.1.1
//设置网关地址为 192.168.1.1（不设置不能通过 IP 访问交换机）
Switch(config)#interface vlan 1
Switch(config-if)#ip address 192.168.1.2 255.255.255.0
Switch(config-if)#no shutdown
//设置交换机管理 IP 地址为 192.168.1.2
Switch(config)#int fa0/1
Switch(config-if)#duplex full
Switch(config-if)#speed 100
//手工设置交换机双工模式以及速度
```

### 步骤二：设置交换机端口安全

```
Switch(config-if)#int fa0/3
Switch(config-if)#switchport mode access
//必须先将交换机接口设置为接入接口
Switch(config-if)#switchport port-security maximum 10
//将接口学习的最大 MAC 地址数量设置为 10
Switch(config-if)#switchport port-security violation
protect
//将超过最大地址数量的数据帧的动作为丢弃
```

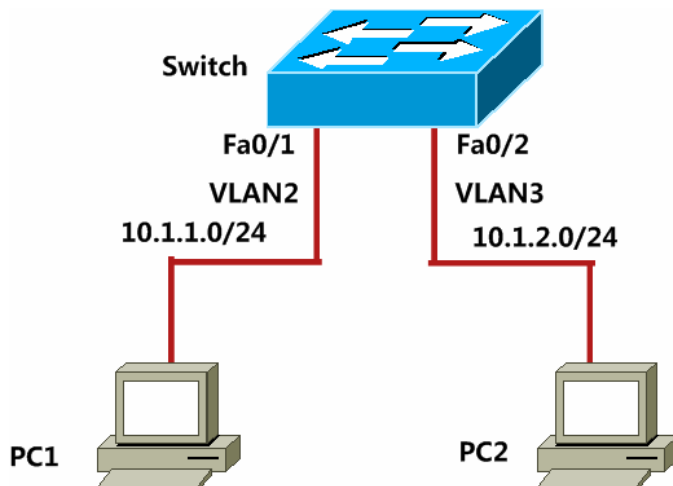
交换机常见查看命令：

show flash:	查看交换机 flash 内容
show mac-address-table	查看交换机 MAC 地址表
show post	查看交换机开机自检信息

## LAB3-2 VLAN 及 Trunk 实验

### 一 基本 VLAN 试验

实验拓扑：



实验目的：掌握 VLAN 基本配置  
掌握静态接口 VLAN 划分

实验需求：在交换机 switch 上面创建两个 VLAN，vlan2 和 vlan3，vlan 名称分别为 HR、ENG  
将主机 PC1、PC2 分别划入两个 VLAN 中

实验步骤：

步骤一：创建 VLAN2 与 VLAN3

```
Switch#conf t
Switch(config)#vlan 2
Switch(config-vlan)#name HR
Switch(config-vlan)#exit
Switch(config)#vlan 3
Switch(config-vlan)#name ENG
```

步骤二：将与两台 PC 连接的两个接口分别划入两个 VLAN 中

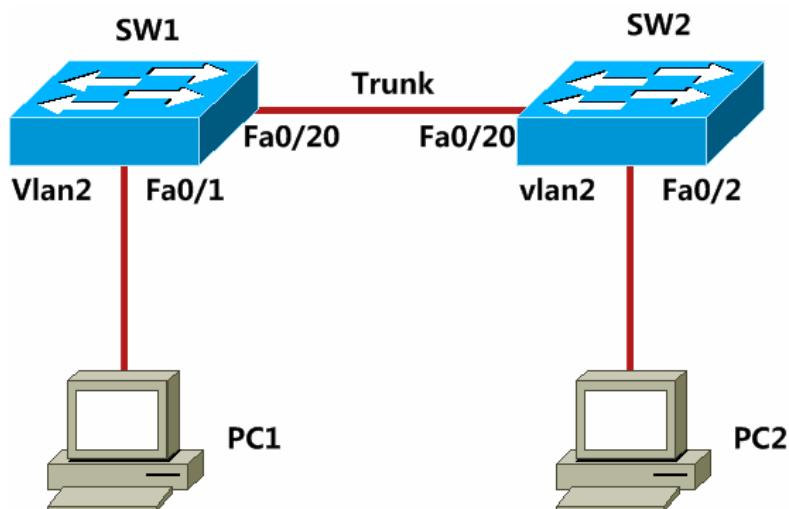
```
Switch(config)#int fa0/1
Switch(config-if)#switchport mode access
//将接口模式设置为接入接口
Switch(config-if)#switchport access vlan 2
//将接口划入指定 VLAN 中
Switch(config-if)#end
Switch(config)#int fa0/2
Switch(config-if)#switchport mode access
```

```
Switch(config-if)#switchport access vlan 3
Switch(config-if)#end
```

步骤三：使用 `show vlan` 命令查看实验结果，在 PC1 以及 PC2 上使用 `ping` 命令查看连通性。

## 二 Vlan Trunk 实验

实验拓扑：



实验目的：掌握 Trunk 的基本配置

实验需求：在交换机 SW1 与 SW2 上创建 VLAN2，并将两个交换机之间的链路设置成 trunk 链路，使 vlan 数据能够跨越交换机

实验步骤：

步骤一：在交换机 SW1、SW2 上创建 VLAN2，并将其与主机连接的接口划入对应 VLAN 中

```
SW1#conf t
SW1(config)#vlan 2
SW1(config-vlan)#name HR
SW1(config-vlan)#exit
SW1(config)#int fa0/1
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 2
```

```
SW2#conf t
SW2(config)#vlan 2
SW2(config-vlan)#name HR
SW2(config-vlan)#exit
SW2(config)#int fa0/2
```

```
SW2(config-if)#switchport mode access
SW2(config-if)#switchport access vlan 2
```

步骤二：将两个交换机之间的链路配置为 Trunk 链路

```
SW1(config)#int fa0/20
SW1(config-if)#switchport trunk encapsulation dot1q
```

//如果交换机支持多种 trunk 封装，则必须先指定所使用的封装。如 3550 交换机。如果只支持一种封装，则没有此命令，如 2950。

```
SW1(config-if)#switchport mode trunk
```

```
SW2(config)#int fa0/20
SW2(config-if)#switchport trunk encapsulation dot1q
SW2(config-if)#switchport mode trunk
```

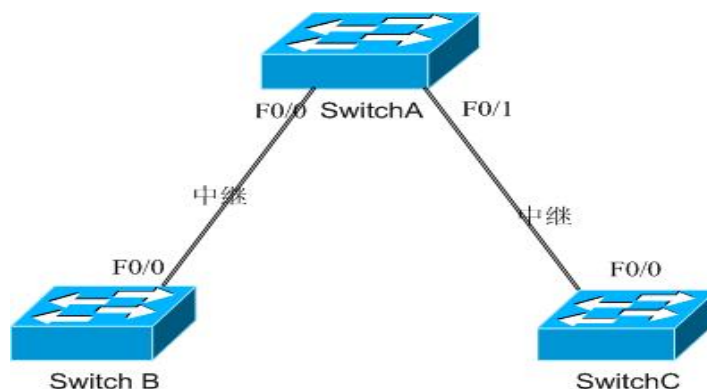
常用查看命令：

show vlan	查看当前交换机中的所有 vlan 信息
show interfaces trunk	查看当前交换机中所有 trunk 接口
show interfaces switchport	查看所有接口的交换相关信息



## LAB3-3 VTP 实验

实验拓扑:



实验目的: 掌握 VLAN 及 VTP 的配置

实验要求: VTP 域名为 cisco, 密码为 password

SwitchA 配置成 Server 模式, 创建 VLAN 10 和 VLAN 20 名为 aa bb, 并开启 VTP 修剪功能

SwitchB 配置成客户模式, 并将端口 1—10 加入 VLAN10, 11—20 划分到 VLAN20

SwitchC 配置成透明模式

实验步骤: 步骤 1 因为 VTP 信息之在 trunk 接口上发送, 所以先将交换机相连的接口配置成 trunk 接口

```
SwitchA(config)#int f0/0
```

如果交换机支持多种 VTP 封装的话, 则需要先指定封装

```
SwitchA(config-if)#switchport trunk encapsulation dot1q
//2950 上无此命令
```

```
SwitchA(config-if)#switchport mode trunk
```

```
SwitchA(config)#int f0/1
```

```
SwitchA(config-if)#switchport trunk encapsulation dot1q
```

```
SwitchA(config-if)#switchport mode trunk
```

```
SwitchB(config)#int f0/0
```

```
SwitchB(config-if)#switchport trunk encapsulation dot1q
```

```
SwitchB(config-if)#switchport mode trunk
```

```
SwitchC(config)#int f0/0
```

```
SwitchC(config-if)#switchport trunk encapsulation dot1q
```

```
SwitchC(config-if)#switchport mode trunk
```

步骤 2 在 SwitchA 上做 VTP 的配置并创建 vlan

```
SwitchA(config)#vtp mode server
```

```
SwitchA(config)#vtp domain stsd
SwitchA(config)#vtp password cisco
SwitchA(config)#vtp pruning
SwitchA#vlan 10 name aa
SwitchA#vlan 20 name bb
```

步骤 3 在 SwitchB 上做配置

```
SwitchB(config)#vtp mode client
SwitchB(config)#vtp domain stsd
SwitchB(config)#vtp password cisco
```

步骤 4 在 SwitchC 上做配置

```
SwitchC(config)#vtp mode transparent
```

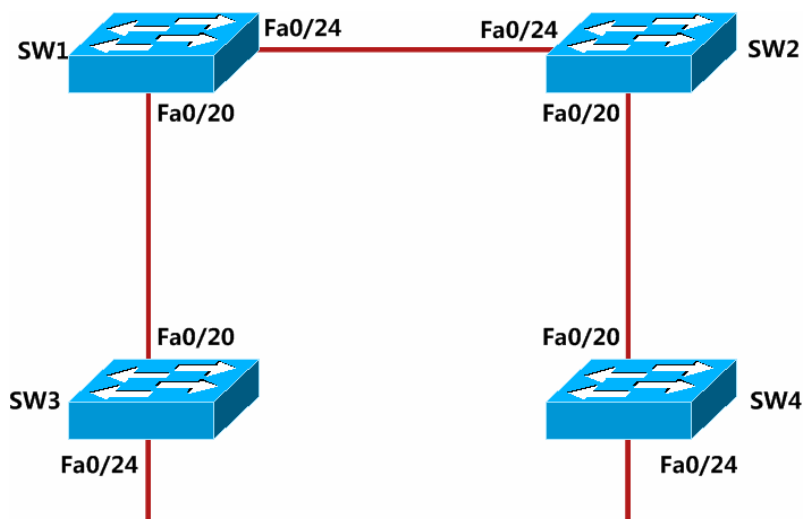
步骤 5 在 SwitchB 上查看 vlan 信息，看是否可以看到创建的 vlan10 和 vlan20

可以尝试在 SwitchB 上面创建 vlan，观察 client 模式的交换机是否能够创建 vlan。在 transparent 上面创建 vlan，并查看 vlan 信息的保存位置。

## LAB3-4 STP 实验

### 一 标准生成树

实验拓扑：



实验目的：观察生成树的运行原理

掌握生成树的常见参数修改，如生成树优先级、接口开销、接口优先级等  
学会控制生成树的主根备份根

实验需求：SW1 成为网络中的主根，SW2 为备份根

修改 SW3 的 FA0/24 口的优先级设置为 64

手工修改 SW3 与 SW4 的 Fa0/20 的接口开销为 5

实验步骤：

步骤一：cisco 交换机上面生成树是默认启用的，默认运行的生成树是 PVST+。可通过命令 `show spanning-tree` 查看生成树运行情况。

步骤二：手工在交换机上启动生成树（默认是自动启动的，本步骤非必须）

因为默认运行的是 PVST+，所以生成树的修改是基于 VLAN 的：

`Switch(config)#spanning-tree vlan 2` //在 VLAN2 上面开启生成树

`Switch(config)#no spanning-tree vlan 2` //在 VLAN2 上关闭生成树

步骤三：修改生成树优先级，使 SW1 成为主根，SW2 为备份根：

`SW1(config)#spanning-tree vlan 1 priority 24576`

`SW2(config)#spanning-tree vlan 1 priority 28672`

注意，由于生成树的 `system-id-extend` 特性，所以生成树优先级必须是 4096 倍数。

也可以使用 Cisco 交换机提供的交换机根设置的宏命令：

`SW1(config)#spanning-tree vlan 1 root primary`

将 SW1 设置为主根，交换机会自动将自己 VLAN1 的优先级设置的比网络中

其他交换机的低，保证自己被选举为主根。

```
SW2(config)#spanning-tree vlan 1 root secondary
```

将 SW2 设置为备份根，SW2 会自动将自己 VLAN1 的生成树优先级设置的比总根高，但是比其他交换机低，以作为主根的备份。

步骤四：修改 SW3 的 Fa0/24 的接口优先级为 100

```
SW3(config)#int fa0/24
```

```
SW3(config-if)#spanning-tree vlan 1 port-priority 64
```

注意,交换机生成树接口优先级必须以 16 递增。

步骤五：修改 SW3、SW4 的 Fa0/20 口的接口开销为 5

```
SW3(config)#int fa0/20
```

```
SW3(config-if)#spanning-tree vlan 1 cost 5
```

```
SW4(config)#int fa0/20
```

```
SW4(config-if)#spanning-tree vlan 1 cost 5
```

步骤六：使用命令 show spanning-tree 查看实验结果

扩展命令：

Switch(config)#spanning-tree vlan 5 hello-time 4	修改生成树的 hello 计时器
Switch(config)#spanning-tree vlan 5 forward-time 20	修改生成树的转发计时器
Switch(config)#spanning-tree vlan 5 max-age 25	修改生成树的最大年龄计时器

常用查看命令：

Switch#show spanning tree	查看生成树的运行信息
Switch#show spanning tree active	只显示激活接口的生成树信息
Switch#show spanning tree detail	显示生成树运行详细信息
Switch#show spanning tree interface fa0/24	查看特性接口的生成树信息
Switch#show spanning tree summary	显示生成树运行信息汇总

## 二 快速生成树

实验拓扑：与之前实验相同

**实验目的：**掌握快速生成树的运行原理

**实验需求：**在所有交换机上面启动快速生成树  
观察快速生成树的运行过程

**实验步骤：**

步骤一：在所有交换机上面启动快速生成树

**SW1(config)#spanning-tree mode rapid-pvst**

**SW2(config)#spanning-tree mode rapid-pvst**

**SW3(config)#spanning-tree mode rapid-pvst**

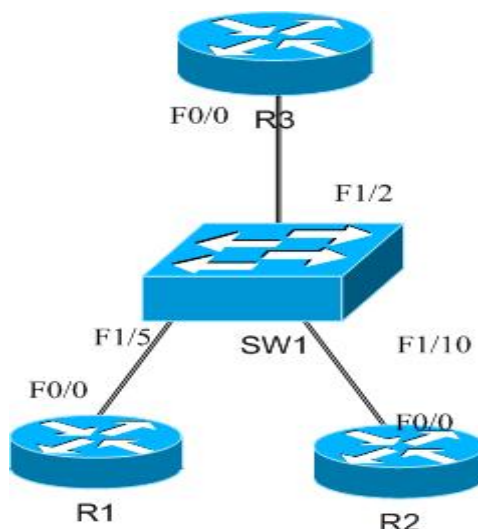
**SW4(config)#spanning-tree mode rapid-pvst**

步骤二：使用命令 **show spanning-tree** 查看快速生成树的运行情况

## LAB3-5 VLAN 间路由实验

一 单臂路由:

实验拓扑:



- 实验要求:
- 1- R1 R2 两台路由器模拟成主机
  - 2- R1 的 F0/0 口的 ip 为 192.168.1.2 默认网关为 192.168.1.1
  - 3- R2 的 F0/0 口的 ip 为 192.168.2.2 默认网关为 192.168.2.1
  - 4- 在 SW1 上划分 VLAN5 VLAN10, 并将 R1 划分到 VLAN5 中, 将 R2 划分到 VLAN10 中
  - 5- R3 上配置 VLAN 间路由
  - 6- R1 可以 ping 通 R2

实验步骤: 步骤 1 将 R1 R2 模拟成主机进行配置

```
R1(config)#no ip routing //关闭路由器路由功能
R1(config)#ip default-gateway 192.168.1.1 //设置网关
R1(config)#int fa0/0
R1(config-if)#ip add 192.168.1.2 255.255.255.0
R1(config-if)#no shutdown

R2(config)#no ip routing
R2(config)#ip default-gateway 192.168.2.1
R2(config)#int fa0/0
R2(config-if)#ip add 192.168.2.2 255.255.255.0
R2(config-if)#no shutdown
```

步骤 2 对 SW1 进行配置

```
SW1#conf t
SW1(config)#vlan 5
SW1(config-vlan)#exit
```

```
SW1(config)#vlan 10
SW1(config-vlan)#exit
```

```
SW1(config)#int fa1/5
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 5
SW1(config-if)#exit
SW1(config)#int fa1/10
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 10
SW1(config-if)#exit
```

交换机与路由器之间的线路必须是 **trunk**，封装必须一致

```
SW1(config)#int fa1/2
SW1(config-if)#switchport mod trunk
SW1(config-if)#switchport trunk encapsulation dot1q
SW1(config-if)#no shutdown
SW1(config-if)#exit
```

步骤 3 对 R3 进行配置

```
R3(config)#ip routing
```

```
R3(config)#int fa0/0
R3(config-if)#no shutdown
```

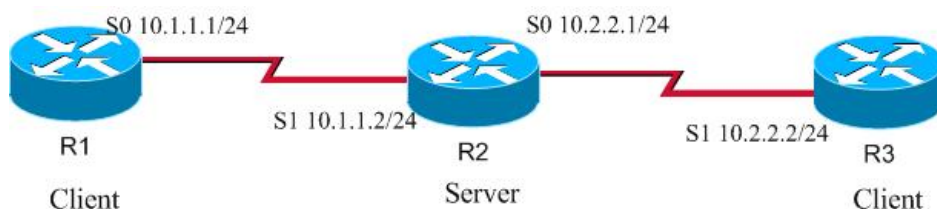
```
R3(config)#int fa0/0.5
R3(config-subif)#encapsulation dot1q 5
R3(config-subif)#ip add 192.168.1.1 255.255.255.0
R3(config-subif)#no shutdown
R3(config-subif)#exit
R3(config)#int fa0/0.10
R3(config-subif)#encapsulation dot1q 10
R3(config-subif)#ip add 192.168.2.1 255.255.255.0
R3(config-subif)#no shutdown
R3(config-subif)#exit
```

步骤 4 验证结果，R1 ping R2 看结果如何

注意 Cisco 限制用于单臂路由路由器的接口必须是 100M 口。但是 IOS 有 BUG，部分 2600 的 10M 口也可以完成本实验。

## LAB4-1 PPP 协议及其认证

实验拓扑：



实验目的：掌握 PPP 的两种认证方法的配置，即 PAP 与 CHAP

掌握单向认证和双向认证的配置

掌握 debug ppp authentication 命令

实验要求：PAP 认证中，R1 R2 之间配置 PAP 认证，R2 R3 之间配置 PAP 认证

CHAP 认证时，都配置 CHAP 认证

实验步骤：PPP PAP 认证

步骤 1 按如上拓扑做好底层配置，使全网互通

步骤 2 R2 做服务器，配置如下

```
R2(config)#username R1 password cisco
R2(config)#username R3 password cisco
```

```
R2(config)#int s1
R2(config-if)#encapsulation ppp
R2(config-if)#ppp authentication pap
R2(config)#int s0
R2(config-if)#encapsulation ppp
R2(config-if)#ppp authentication pap
```

步骤 3 R1、R3 做客户机配置如下

```
R1(config)#int s0
R1(config-if)#encapsulation ppp
R1(config-if)#ppp pap send-username R1 password cisco
```

```
R3(config)#int s1
R3(config-if)#encapsulation ppp
R3(config-if)#ppp pap send-username R3 password cisco
```



步骤 4 先将接口 shutdown, 再 no shutdown, 在 R2 上用 debug ppp authentication 命令查看 PPP 的认证过程

#### PPP CHAP 认证

步骤 1 做好底层配置, 使全网互通

步骤 2 进行双向认证的配置, R1、R2、R3 的配置如下

```
R1(config)#username R2 password cisco
R1(config)#int s0
R1(config-if)#encapsulation ppp
R1(config-if)# ppp authentication chap

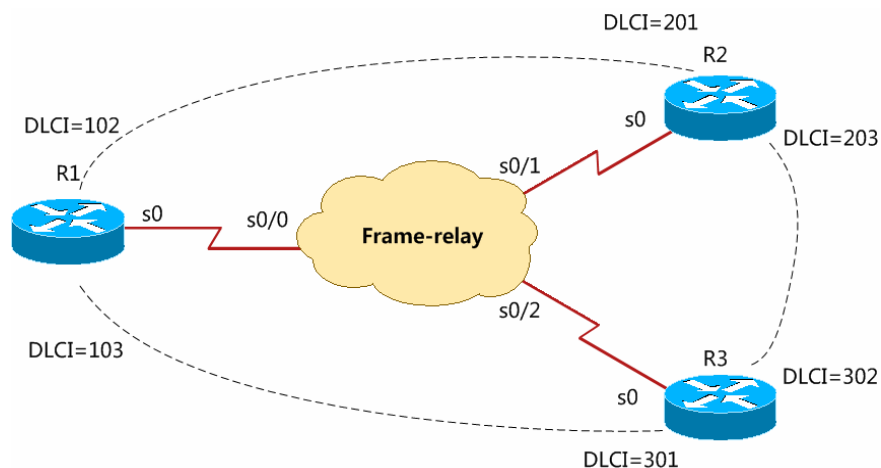
R2(config)#username R1 password cisco
R2(config)#username R3 password cisco
R2(config)#int s0
R2(config-if)#encapsulation ppp
R2(config-if)# ppp authentication chap
R2(config)#int s1
R2(config-if)#encapsulation ppp
R2(config-if)# ppp authentication chap

R3(config)#username R2 password cisco
R3(config)#int s1
R3(config-if)#encapsulation ppp
R3(config-if)# ppp authentication chap
```

步骤 3 查看 PPP 的认证情况

## LAB4-2 Frame-Relay 实验

实验拓扑：



### 一 基本 FR 实验

**实验目的：**掌握帧中继原理，FR 交换机基本配置，FR 客户端基本配置

掌握点对点、点对多点 FR 子接口的配置

观察 FR 环境中距离矢量型路由选择协议的更新问题以及解决方式

**实验需求：**所有路由器帧中继封装均为 ietf，LMI 类型为 ansi。

配置 full-mesh 的 FR 网络，要求相互能够 ping 通。

客户端所属网段为 10.1.1.0/24。

**实验步骤：**步骤一：帧中继交换机配置

中间的 frame-relay 网云我们使用一台路由器作为帧中继交换机，主机名为 FR  
`FR(config)#frame-relay switching` //全局打开路由器帧中继交换功能

```
FR(config)#int s0/0
FR(config-if)#encapsulation frame-relay ietf
//配置封装类型为 IETF
FR(config-if)#frame-relay intf-type dce
FR(config-if)#clock rate 64000
FR(config-if)#frame-relay lmi-type ansi
//配置 LMI 类型为 ANSI
FR(config-if)#frame-relay route 102 interface s0/1 201
FR(config-if)#frame-relay route 103 interface s0/2 301
FR(config-if)#no shutdown
```

```
FR(config)#int s0/1
FR(config-if)#encapsulation frame-relay ietf
FR(config-if)#frame-relay intf-type dce
FR(config-if)#clock rate 64000
```

```
FR(config-if)#frame-relay lmi-type ansi
FR(config-if)#frame-relay route 201 interface s0/0 102
FR(config-if)#frame-relay route 203 interface s0/2 302
FR(config-if)#no shutdown
```

```
FR(config)#int s0/2
FR(config-if)#encapsulation frame-relay ietf
FR(config-if)#frame-relay intf-type dce
FR(config-if)#clock rate 64000
FR(config-if)#frame-relay lmi-type ansi
FR(config-if)#frame-relay route 103 interface s0/2 301
FR(config-if)#frame-relay route 302 interface s0/1 203
FR(config-if)#no shutdown
```

步骤二：帧中继客户端配置：

```
R1(config)#int s0
R1(config-if)#encapsulation frame-relay ietf
R1(config-if)#frame-relay lmi-type ansi
R1(config-if)#ip add 10.1.1.1 255.255.255.0
R1(config-if)#no shutdown
```

```
R2(config)#int s0
R2(config-if)#encapsulation frame-relay ietf
R2(config-if)#frame-relay lmi-type ansi
R2(config-if)#ip add 10.1.1.2 255.255.255.0
R2(config-if)#no shutdown
```

```
R3(config)#int s0
R3(config-if)#encapsulation frame-relay ietf
R3(config-if)#frame-relay lmi-type ansi
R3(config-if)#ip add 10.1.1.3 255.255.255.0
R3(config-if)#no shutdown
```

步骤三：通过 ping 命令测试 FR 连通性。通过帧中继相关 show 命令查看信息：

```
FR#show frame-relay route
```

```
FR#show frame-relay pvc
```

```
FR#show frame-relay lmi
```

## 二 静态 IP/DLCI 映射配置：

实验目的：掌握 inverse arp 原理

掌握静态 IP/DLCI 映射配置

**实验需求：**关闭帧中继交换机的 inverse arp 功能，R1、R2、R3 通过静态映射命令进行配置。

**实验步骤：**

**步骤一：**基于实验一的配置，关闭 FR 上的 inverse-arp 功能

```
FR(config)#int s0/0
FR(config-if)#no frame-relay inverse-arp
FR(config)#int s0/1
FR(config-if)#no frame-relay inverse-arp
FR(config)#int s0/2
FR(config-if)#no frame-relay inverse-arp
```

**步骤二：**在 R1、R2、R3 上面配置静态 IP/DLCI 映射

```
R1(config)#int s0
R1(config-if)#frame-relay map ip 10.1.1.2 102
R1(config-if)#frame-relay map ip 10.1.1.3 103
```

```
R2(config)#int s0
R2(config-if)#frame-relay map ip 10.1.1.1 201
R2(config-if)#frame-relay map ip 10.1.1.3 203
```

```
R3(config)#int s0
R3(config-if)#frame-relay map ip 10.1.1.1 301
R3(config-if)#frame-relay map ip 10.1.1.2 302
```

**步骤三：**使用 show 命令查看结果，并使用 ping 检测连通性

```
R1#show frame-relay map
Serial0 (up): ip 10.1.1.2 dlci 102(0x66,0x1860), static,
IETF, status defined, active
Serial0 (up): ip 10.1.1.3 dlci 103(0x67,0x1870), static,
IETF, status defined, active
```

由以上信息可得知，当前映射为手工静态映射

使用 ping 命令检测连通性：

```
R1#ping 10.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/60/80 ms
R1#ping 10.1.1.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/61/96 ms
```

### 三 距离矢量型路由选择协议在帧中继网络中的问题

实验拓扑：与之前实验相同

实验目的：观察距离矢量型路由选择协议在帧中继网络中的运行问题

实验需求：在之前实验的基础上，启动 EIGRP 路由，并且路由器之间可以相互学习路由信息

实验步骤：

步骤一：按照之前实验需求完成基本配置

步骤二：在所有路由器上启动一个环回接口，方便之后测试

```
R1(config)#int lo0
R1(config-if)#ip add 172.16.1.1 255.255.255.0
```

```
R2(config)#int lo0
R2(config-if)#ip add 172.16.2.1 255.255.255.0
```

```
R3(config)#int lo0
R3(config-if)#ip add 172.16.3.1 255.255.255.0
```

步骤三：在 R1、R2、R3 上面启动 EIGRP 进程，通告所有网段：

```
R1(config)#router eigrp 100
R1(config-router)#no auto-summary
R1(config-router)#network 10.1.1.0 0.0.0.255
R1(config-router)#network 172.16.1.0 0.0.0.255
```

```
R2(config)#router eigrp 100
R2(config-router)#no auto-summary
R2(config-router)#network 10.1.1.0 0.0.0.255
R2(config-router)#network 172.16.2.0 0.0.0.255
```

```
R3(config)#router eigrp 100
R3(config-router)#no auto-summary
R3(config-router)#network 10.1.1.0 0.0.0.255
R3(config-router)#network 172.16.3.0 0.0.0.255
```

步骤四：查看 R2、R3 路由器的路由表，观察水平分割在 FR 网络中对 EIGRP 更新的影响

步骤五：关闭 R1 的 s0 接口的 EIGRP 的水平分割，保证 EIGRP 路由更新

```
R1(config-if)#no ip split-horizon eigrp 100
```

帧中继网络中距离矢量型路由选择协议水平分割问题的解决方法:

1. 关闭水平分割
2. 将接口配置成点对点接口

#### 四 帧中继点对点接口

实验拓扑: 与之前实验相同

实验目的: 掌握帧中继点对点接口的配置

实验需求: 在 R1 上启动帧中继点对点接口, 以解决距离矢量型水平分割问题

实验步骤:

步骤一: FR、R2、R3 基本配置与之前实验相同, 在 R1 中重新配置, 配置点对点接口, 将两条 PVC 划分到两个子接口中, 注意, 不同子接口属于不同网段

```
R1(config)#int s0          //先在主接口上配置封装
R1(config-if)#no ip address
R1(config-if)#encapsulation frame-relay ietf
R1(config-if)#frame-relay lmi-type ansi
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#int s0.1 point-to-point
//启动一个点对点接口
R1(config-subif)#frame-relay interface-dlci 102
//设置此子接口的 DLCI 号
R1(config-fr-dlci)#exit
R1(config-subif)#ip add 10.1.1.1 255.255.255.0
R1(config-subif)#no shutdown
R1(config-subif)#exit
R1(config)#int s0.2 point-to-point
R1(config-subif)#frame-relay interface-dlci 103
R1(config-fr-dlci)#exit
R1(config-subif)#ip add 10.1.2.1 255.255.255.0
R1(config-subif)#no shutdown
R1(config-subif)#end
```

然后在不关闭水平分割的情况下, 观察 EIGRP 协议的运行情况。看是否能够学习到路由。

#### 五 帧中继点对多点接口

实验拓扑: 与之前拓扑相同

实验目的: 掌握帧中继点对多点接口的配置

**实验步骤:**

步骤一: **FR、R2、R3** 配置与之前相同

步骤二: 在 **R1** 上启用点对多点子接口

```
R1(config)#int s0
R1(config-if)#encapsulation frame-relay
R1(config-if)#no ip address
R1(config-if)#no shutdown
R1(config)#int s0.1 multipoint
R1(config-subif)#ip add 10.1.1.1 255.255.255.0
R1(config-subif)#frame-relay interface-dlci 102
R1(config-fr-dlci)#exit
R1(config-subif)#frame-relay interface-dlci 103
R1(config-fr-dlci)#exit
```

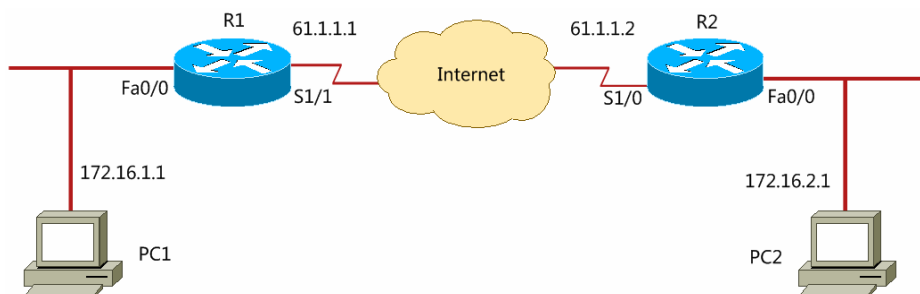
点对多点子接口与一般帧中继接口一样,同样会产生水平分割问题。与点对点子接口的区别还在于,点对点子接口不同子接口属于不同子网,而多点子接口属于相同子网。

**常用 Frame-relay 查看命令:**

Router#show frame-relay map	显示IP与DLCI映射
Router#show frame-relay pvc	显示当前PVC配置状态
Router#show frame-relay lmi	显示LMI状态
Router#clear frame-relay counters	清除所有FR统计信息
Router#clear frame-relay inarp	清除inverse arp缓存

## LAB4-3 IPSec VPN 实验

实验拓扑:



实验目的: 掌握 IPSec VPN 原理  
 掌握 site-to-site VPN 配置  
 IPSec 配置参数:

IKE policy	isakmp key	转换集
加密算法 3DES	cisco	载荷加密算法 esp-3des
哈希算法 MessageDigest 5		载荷散列算法 esp-sha-hmac
认证方式 Pre-Shared Key		认证头 ah-sha-hmac
Diffie-Hellman 组 #2 (1024 bit)		

实验需求: 在 R1、R2 间配置 site-to-site VPN, 对 172.16.1.0/24 和 172.16.2.0/24 网段数据进行加密

实验步骤:

步骤一: 按照实验拓扑, 对路由器、PC 机进行基本配置。保证底层网络互通。

步骤二: 对 R1 进行 IPSec 配置

A- 配置密钥交换策略

```
R1(config)#crypto isakmp policy 10
R1(config-isakmp)#authentication pre-share
R1(config-isakmp)#hash md5
R1(config-isakmp)#group 2
R1(config-isakmp)#encryption 3des
R1(config-isakmp)#exit
```

B- 配置预共享密钥

```
R1(config)#crypto isakmp key 6 cisco address 61.1.1.2
```

C- 配置加密转换集 myset

```
R1(config)#crypto ipsec transform-set myset esp-3des
esp-sha-hmac ah-sha-hmac
```

D- 配置访问控制列表, 定义兴趣流量, 控制对 172.16.1.0/24 到 172.16.2.0/24 网络数据进行加密

```
R1(config)#access-list 100 permit ip 172.16.1.0
0.0.0.255 172.16.2.0 0.0.0.255
```

E- 配置加密映射图, 绑定接口



```

R1(config)#crypto map mymap 10 ipsec-isakmp
R1(config-crypto-map)#match address 100
R1(config-crypto-map)#set transform-set myset
R1(config-crypto-map)#set peer 61.1.1.2
R1(config-crypto-map)#end
R1(config)#int s1/1
R1(config-if)#crypto map mymap

```

### 步骤三：对 R2 进行 IPSec 配置

#### A- 配置密钥交换策略

```

R2(config)#crypto isakmp policy 10
R2(config-isakmp)#authentication pre-share
R2(config-isakmp)#hash md5
R2(config-isakmp)#group 2
R2(config-isakmp)#encryption 3des
R2(config-isakmp)#exit

```

#### B- 配置预共享密钥

```

R2(config)#crypto isakmp key 6 cisco address 61.1.1.1

```

#### C- 配置加密转换集 myset

```

R2(config)#crypto ipsec transform-set myset esp-3des
esp-sha-hmac ah-sha-hmac

```

#### D- 配置访问控制列表，定义兴趣流量，控制对 172.16.2.0/24 到 172.16.1.0/24 网络数据进行加密

```

R2(config)#access-list 100 permit ip 172.16.2.0
0.0.0.255 172.16.1.0 0.0.0.255

```

#### E- 配置加密映射图，绑定接口

```

R2(config)#crypto map mymap 10 ipsec-isakmp
R2(config-crypto-map)#match address 100
R2(config-crypto-map)#set transform-set myset
R2(config-crypto-map)#set peer 61.1.1.1
R2(config-crypto-map)#exit
R2(config)#int s1/0
R2(config-if)#crypto map mymap

```

步骤三：使用 ping 命令从 PC1 ping PC2，测试连通性。Ping 通后使用命令 show crypto isakmp peers 查看建立的对等体连接。

### IPSec VPN 常用查看命令：

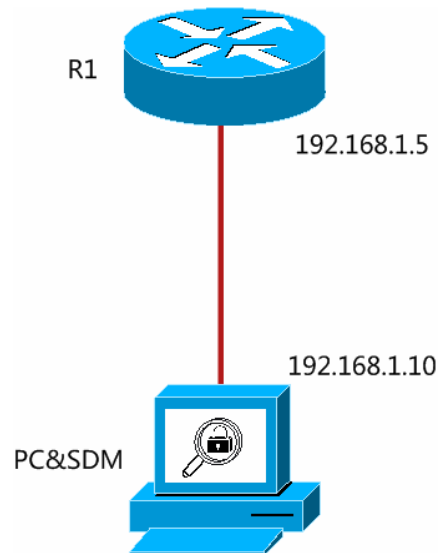
show crypto map	查看加密映射图
show crypto isakmp policy	查看密钥交换策略
show crypto isakmp key	查看当前密钥交换方式所使用的密钥
show crypto isakmp peers	查看已建立的对等体

---

show crypto isakmp sa	查看安全关联
show crypto ipsec transform-set	查看 IPSec 加密转换集

## LAB5-1 使用 cisco SDM 管理路由器

实验拓扑：



实验目的：掌握如何通过 SDM 对路由器进行管理

实验需求：设置 R1 使其能够通过 SDM 进行管理

实验步骤：

步骤一：配置基本 IP 地址

步骤二：在 R1 上进行设置，使其能够通过 SDM 连接

```
R1(config)#username stsd privilege 15 secret cisco
```

//创建 15 级用户

```
R1(config)#ip http server
```

//打开 R1 的 http 服务器

```
R1(config)#ip http authentication local
```

//将 http 认证设置为使用本地认证数据库

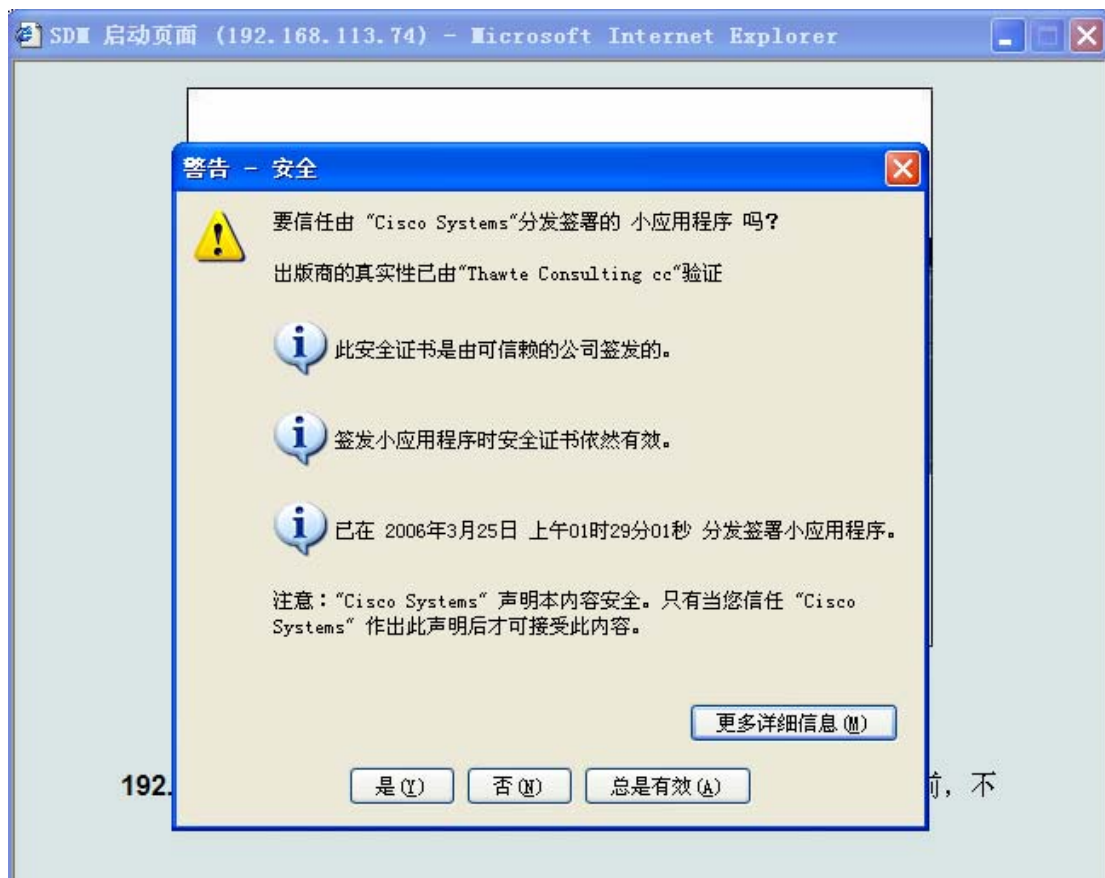
步骤三：通过 SDM 连接 R1



在地址栏中输入 R1 的 IP 地址。



输入在 R1 中创建的 15 级的用户名与密码



同意 java 安全警告



在输入一次用户名密码



连接完成，现在就可以通过 SDM 对 R1 进行管理了。