

# 电气与信息工程学院

## 计算机系

课程名称：网络技术综合实训

项目名称：企业网方案设计与实现

班 级：计 211

学 号：202102010224

姓 名：詹子阳

2024 年 6 月 26 日

# 目录

项目简介.....	1
实验拓扑.....	2
编址规划.....	3
实验设备.....	6
功能要求.....	7
项目实现.....	7
项目整体测试.....	41
实训总结.....	42

## 项目简介

某企业是北京市海淀区高科技企业之一。通过建设一个高速、安全、可靠、可扩充的网络系统，实现企业内信息的共享和传递，并建立出口信道，实现与 Internet 的互联。根据企业规模和业务需求，企业网采用接入层、汇聚层和核心层的三层网络拓扑结构。接入层用来实现用户接入，直接面向用户连接。汇聚层连接接入层和核心层，是接入层交换机的汇聚点。核心层是企业网的骨干部分，提供高速、可靠的数据传输服务。核心层采用双核心的冗余结构，保证网络的稳定性。

企业包括办公室(VLAN10)、人事部(VLAN20)、财务部(VLAN30)、研发部(VLAN40)、市场部(VLAN50)、生产部(VLAN60)和信息中心(VLAN70)等七个部门。

汇聚层交换机 MS1 与核心层交换机 Core1 的网段组成 VLAN100,与核心层交换机 Core2 的网段组成 VLAN110。

汇聚层交换机 MS2 与核心层交换机 Core1 的网段组成 VLAN120,与核心层交换机 Core2 的网段组成 VLAN130。

核心层交换机 Core1 与核心层交换机 Core2 的网段组成 VLAN140。

核心层交换机 Core1 与核心层交换机 Core2 分别与路由器 R1 的网段组成 VLAN150 和 VLAN160。在互联网中，PCa 所在的网段属于 VLAN10，PCb 所在的网段属于 VLAN20，WWW2 服务器所在的网段属于 VLAN30。

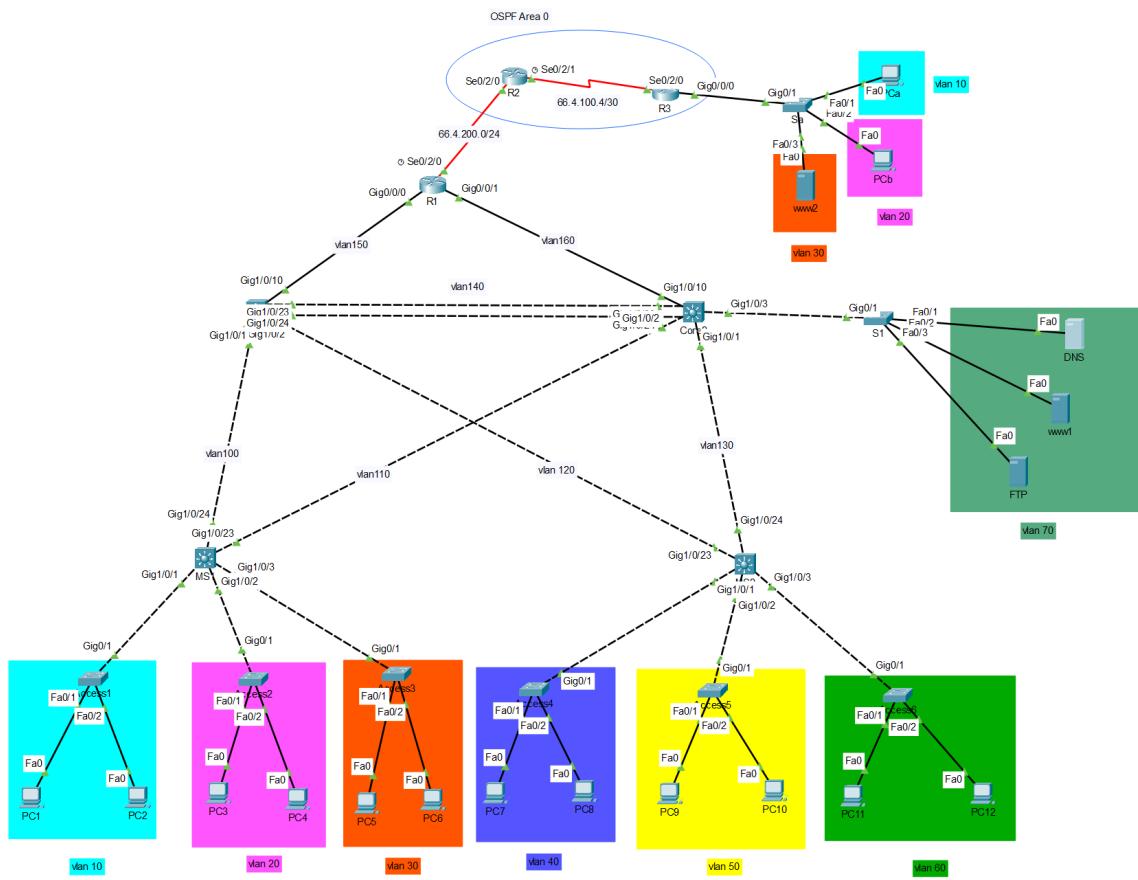
企业网内部主机的网络地址是 10.4.0.0/16,企业网设备与设备之间的互连链路的网络地址是 192.168.4.0/24。互联网的 IP 网络地址是 66.4.0.0/16。

合理规划 IP 地址，要求：

- (1)接入层的 VLAN 子网要使用网络掩码位数为 24 位的 IP 地址。
- (2)设备与设备之间的互连链路要使用网络掩码位数为 30 位的 IP 地址。

(3)对于路由器 R1 与 R2 之间的链路，可以使用网络掩码位数小于 30 位的地址该企业采用现代网络技术，将企业的各种计算机、服务器和局域网连接起来，并接入互联网，能够在网发布公司的信息并获取互联网上的信息资源，从而形成一个结构合理内外连通的企业网络系统。

实验拓扑



## 编址规划

终端设备编址如下

终端设备	所属 VLAN	IP 地址	备注
PC1	VLAN10	10.4.10.1	所有终端设备均处于掩码为 24 位的子网中，PC1-PC6 的 DNS 均设置为 DNS_Server 的 IP 地址 10.4.70.1
PC2		10.4.10.2	
Pca		66.4.10.1	
PC3	VLAN20	10.4.20.1	所有终端设备均处于掩码为 24 位的子网中，PC1-PC6 的 DNS 均设置为 DNS_Server 的 IP 地址 10.4.70.1
PC4		10.4.20.2	
PCb		66.4.20.1	
PC5	VLAN30	10.4.30.1	所有终端设备均处于掩码为 24 位的子网中，PC1-PC6 的 DNS 均设置为 DNS_Server 的 IP 地址 10.4.70.1
PC6		10.4.30.2	
WWW2_Server		66.4.30.1	
PC7	VLAN40	10.4.40.1	所有终端设备均处于掩码为 24 位的子网中，PC1-PC6 的 DNS 均设置为 DNS_Server 的 IP 地址 10.4.70.1
PC8		10.4.40.2	
PC9	VLAN50	10.4.50.1	所有终端设备均处于掩码为 24 位的子网中，PC1-PC6 的 DNS 均设置为 DNS_Server 的 IP 地址 10.4.70.1
PC10		10.4.50.2	
PC11	VLAN60	10.4.60.1	所有终端设备均处于掩码为 24 位的子网中，PC1-PC6 的 DNS 均设置为 DNS_Server 的 IP 地址 10.4.70.1
PC12		10.4.60.2	
DNS_Server	VLAN70	10.4.70.1	所有终端设备均处于掩码为 24 位的子网中，PC1-PC6 的 DNS 均设置为 DNS_Server 的 IP 地址 10.4.70.1
WWW1_Server		10.4.70.2	
FTP_Server		10.4.70.3	

所有终端设备均使用 VLAN 逻辑接口作为默认网关

VLAN 逻辑接口如下

VLAN 类型	VLAN 号	逻辑接口
接入 VLAN	VLAN10	10.4.10.254/24
	VLAN20	10.4.20.254/24
	VLAN30	10.4.30.254/24
	VLAN40	10.4.40.254/24
	VLAN50	10.4.50.254/24
	VLAN60	10.4.60.254/24
	VLAN70	10.4.70.254/24
互联 VLAN	VLAN100	192.168.4.101/30
		192.168.4.102/30
	VLAN110	192.168.4.113/30
		192.168.4.114/30
	VLAN120	192.168.4.121/30
		192.168.4.122/30
	VLAN130	192.168.4.133/30
		192.168.4.134/30
	VLAN140	192.168.4.141/30
		192.168.4.142/30
	VLAN150	192.168.4.153/30
		192.168.4.154/30
	VLAN160	192.168.4.161/30
		192.168.4.162/30

所有物理接口、IP 地址及对应 VLAN 关系如下，其中颜色相同的为一对互联的接口

设备类型	设备	物理接口	逻辑接口	接口类型	所属 VLAN
弱三层交换机	Access1	F0/1	*	Access	VLAN10
		F0/2	*	Access	
		G0/1	*	Access	
	Access2	F0/1	*	Access	VLAN20
		F0/2	*	Access	
		G0/1	*	Access	
	Access3	F0/1	*	Access	VLAN30
		F0/2	*	Access	
		G0/1	*	Access	
	Access4	F0/1	*	Access	VLAN40
		F0/2	*	Access	
		G0/1	*	Access	
	Access5	F0/1	*	Access	VLAN50
		F0/2	*	Access	
		G0/1	*	Access	
	Access6	F0/1	*	Access	VLAN60
		F0/2	*	Access	
		G0/1	*	Access	
三层交换机	Sa	F0/1	*	Access	VLAN10
		F0/2	*	Access	VLAN20
		F0/3	*	Access	VLAN30
		G0/1	*	Trunk	*
	S1	F0/1	*	Access	VLAN 70
		F0/2	*	Access	
		F0/3	*	Access	
		G0/1	*	Access	
三层交换机	MS1	G1/0/1	10.4.10.254/24	Access	VLAN10
		G1/0/2	10.4.20.254/24	Access	VLAN20
		G1/0/3	10.4.30.254/24	Access	VLAN30
		G1/0/23	192.168.4.113/30	Access	VLAN110
		G1/0/24	192.168.4.101/30	Access	VLAN100

设备类型	设备	物理接口	逻辑接口	接口类型	所属 VLAN
路由器	MS2	G1/0/1	10.4.40.254/24	Access	VLAN40
		G1/0/2	10.4.50.254/24	Access	VLAN50
		G1/0/3	10.4.60.254/24	Access	VLAN60
		G1/0/23	192.168.4.121/30	Access	VLAN120
		G1/0/24	192.168.4.133/30	Access	VLAN130
	Core1	G1/0/1	192.168.4.102/30	Access	VLAN100
		G1/0/2	192.168.4.122/30	Access	VLAN120
		G1/0/23	192.168.4.141/30	Access	VLAN140
		G1/0/24		Access	
		G1/0/10	192.168.4.153/30	Access	VLAN150
	Core2	G1/0/1	192.168.4.134/30	Access	VLAN130
		G1/0/2	192.168.4.114/30	Access	VLAN110
		G1/0/23	192.168.4.142/30	Access	VLAN140
		G1/0/24		Access	
		G1/0/3	10.4.70.254/24	Access	VLAN70
		G1/0/10	192.168.4.161/30	Access	VLAN160
	R1	G0/0/0	192.168.4.154/30	三层接口	
		G0/0/1	192.168.4.162/30	三层接口	
		S0/2/0	66.4.200.1/24	三层接口	
	R2	S0/2/0	66.4.200.254/24	三层接口	
		S0/2/1	66.4.100.5/30	三层接口	
	R3	S0/2/0	66.4.100.6/30	三层接口	
		G0/0/0	66.4.10.254/24	三层接口	VLAN10
			66.4.20.254/24	三层接口	VLAN20
			66.4.30.254/24	三层接口	VLAN30

## 实验设备

路由器 3 台：型号 4331

三层交换机 4 台：型号 3650-24PS

二层交换机 8 台：型号 2960

PC 机 14 台：命名从 PC1 到 PC12、PCA、PCB

服务器 4 台：WWW1\_Server、WWW2\_Server、FTP\_Server、DNS\_Server

## 功能要求

- 1、根据要求，创建企业网内部各个 VLAN，并完成 VLANIF 地址的配置。
- 2、采用链路聚合技术提高核心层交换机之间的链路带宽。
- 3、在企业网内部配置 RIPv2 协议。要求在汇聚层交换机 MS1 和 MS2，核心层交换机 Core1 和 Core2，路由器 R1 上运行 RIPv2 协议。
- 4、在路由器 R1 上配置默认路由，以接入互联网。同时，向 RIP 网络中注入默认路由。
- 5、在交换机 Sa 上创建各个 VLAN，在路由器 R3 上使用单臂路由技术实现 VLAN 之间的通信。
- 6、配置路由器 R2 和路由器 R3 运行 OSPF 协议。
- 7、路由器 R1 与 R2 之间封装 PPP 协议，使用 CHAP 认证，密码为姓名全拼。
- 8、完成 DNS 服务器的配置。www1 服务器的域名为 www1.姓名全拼.com，www2 服务器的域名为 www2.姓名全拼.com。
- 9、配置 NAT。在路由器 R1 上配置 NAPT，使企业用户可以访问互联网资源。在路由器 R1 上配置静态 NAT，使互联网用户可以访问企业的 WWWI 服务器。
- 10、在汇聚层交换机 MS1 上配置扩展 ACL，使财务部(VLAN30)的主机不能访问 FTP 服务器，其它服务不受影响。在汇聚层交换机 MS2 上配置扩展 ACL，使生产部(VLAN60)的主机不能访问 www2 服务器，其它服务不受影响。

## 项目实现

根据需求先取出 14 台 PC，4 台 Server，8 台强二层（弱三层）交换机，4 台三层交换机，3 台路由器，并且根据规划连接好对应线缆。

连接端口时应该注意满足达线速转发。例如，当接入交换机下接两个 FastEthernet 端口即 100M 端口时，要满足两台下接设备达线速转发，上联口应该选用 GE 端口即 1000M 端口，才能满足 200M 的转发速率。

- 1、根据要求，创建企业网内部各个 VLAN，并完成 VLANIF 地址的配置。

对于 Access1 到 Access6 配置几乎完全相同

配置 Access1

Switch>

Switch>en

Switch#conf t

Switch(config)#hostname Access1

Access1(config)#vlan 10

```
Access1(config-vlan)#ex
Access1(config)#int range f0/1-2, g0/1
Access1(config-if-range)#sw acc vlan 10
Access1(config-if-range)#end
Access1#
Access1#wr
```

验证 Access1 配置正确

```
Access1#show vlan id 10
Access1#show vlan id 10
VLAN Name Status Ports
---- -----
10 VLAN0010 active Fa0/1, Fa0/2, Gig0/1
```

配置 Access2

```
Switch>
Switch>en
Switch#conf t
Switch(config)#hostname Access2
Access2(config)#vlan 20
Access2(config-vlan)#ex
Access2(config)#int range f0/1-2, g0/1
Access2(config-if-range)#sw acc vlan 20
Access2(config-if-range)#end
Access2#
Access2#wr
```

验证 Access2 配置正确

```
Access2#show vlan id 20
Access2#show vlan id 20
VLAN Name Status Ports
---- -----
20 VLAN0020 active Fa0/1, Fa0/2, Gig0/1
```

配置 Access3

```
Switch>
Switch>en
Switch#conf t
Switch(config)#hostname Access3
Access3(config)#vlan 30
Access3(config-vlan)#ex
Access3(config)#int range f0/1-2, g0/1
Access3(config-if-range)#sw acc vlan 30
Access3(config-if-range)#end
Access3#
Access3#wr
```

验证 Access3 配置正确

```
Access3#show vlan id 30
Access3#show vlan id 30
+-----+-----+-----+
| VLAN Name | Status | Ports |
+-----+-----+-----+
| 30   VLAN0030 | active | Fa0/1, Fa0/2, Gig0/1 |
```

配置 Access4

```
Switch>
Switch>en
Switch#conf t
Switch(config)#hostname Access4
Access4(config)#vlan 40
Access4(config-vlan)#ex
Access4(config)#int range f0/1-2, g0/1
Access4(config-if-range)#sw acc vlan 40
Access4(config-if-range)#end
Access4#
Access4#wr
```

验证 Access4 配置正确

```
Access4#show vlan id 40
```

```

Access4#show vlan id 40
VLAN Name                               Status      Ports
---- -----
40    VLAN0040                           active     Fa0/1, Fa0/2, Gig0/1

```

配置 Access5

```

Switch>
Switch>en
Switch#conf t
Switch(config)#hostname Access5
Access5(config)#vlan 50
Access5(config-vlan)#ex
Access5(config)#int range f0/1-2, g0/1
Access5(config-if-range)#sw acc vlan 50
Access5(config-if-range)#end
Access5#wr

```

验证 Access5 配置正确

```

Access5#show vlan id 50
VLAN Name                               Status      Ports
---- -----
50    VLAN0050                           active     Fa0/1, Fa0/2, Gig0/1

```

配置 Access6

```

Switch>
Switch>en
Switch#conf t
Switch(config)#hostname Access6
Access6(config)#vlan 60
Access6(config-vlan)#ex
Access6(config)#int range f0/1-2, g0/1
Access6(config-if-range)#sw acc vlan 60
Access6(config-if-range)#end
Access6#
Access6#wr

```

验证 Access6 配置正确

Access6#show vlan id 60

```
Access6#show vlan id 60
VLAN Name Status Ports
---- -- -- --
60 VLAN0060 active Fa0/1, Fa0/2, Gig0/1
```

验证同部门内终端可以互访

```
C:\>ping 10.4.10.2
Pinging 10.4.10.2 with 32 bytes of data:
Reply from 10.4.10.2: bytes=32 time=13ms TTL=128
Reply from 10.4.10.2: bytes=32 time<1ms TTL=128
Reply from 10.4.10.2: bytes=32 time<1ms TTL=128
Reply from 10.4.10.2: bytes=32 time<1ms TTL=128
```

PC1 ping PC2

```
C:\>ping 10.4.30.2
Pinging 10.4.30.2 with 32 bytes of data:
Reply from 10.4.30.2: bytes=32 time<1ms TTL=128
```

PC3 ping PC4

```
C:\>ping 10.4.20.2
Pinging 10.4.20.2 with 32 bytes of data:
Reply from 10.4.20.2: bytes=32 time<1ms TTL=128
```

PC5 ping PC6

```
C:\>ping 10.4.40.2
Pinging 10.4.40.2 with 32 bytes of data:
Reply from 10.4.40.2: bytes=32 time<1ms TTL=128
Reply from 10.4.40.2: bytes=32 time<1ms TTL=128
Reply from 10.4.40.2: bytes=32 time<1ms TTL=128
```

PC7 ping PC8

```
C:\>ping 10.4.50.2
Pinging 10.4.50.2 with 32 bytes of data:
Reply from 10.4.50.2: bytes=32 time<1ms TTL=128
```

PC9 ping PC10

```
C:\>ping 10.4.60.2
Pinging 10.4.60.2 with 32 bytes of data:
Reply from 10.4.60.2: bytes=32 time<1ms TTL=128
```

PC11 ping PC12

在 S1 上创建 VLAN70

Switch>

Switch>en

Switch#conf t

Switch(config)#hostname S1

S1(config)#vlan 70

S1(config-vlan)#int range f0/1-3,g0/1

S1(config-if-range)#sw acc vlan 70

S1(config-if-range)#end

S1#

S1#wr

验证创建成功

S1#show vlan id 70

VLAN	Name	Status	Ports
70	VLAN0070	active	Fa0/1, Fa0/2, Fa0/3, Gig0/1

验证同部门内设备连通性

```
C:\>ping 10.4.70.3

Pinging 10.4.70.3 with 32 bytes of data:

Reply from 10.4.70.3: bytes=32 time<1ms TTL=128
```

DNS\_Server Ping FTP\_Server

在 Sa 上创建 VLAN10, VLAN20, VLAN30

Switch>

Switch>en

Switch#conf t

Switch(config)#hostname Sa

Sa(config)#vlan 10

Sa(config-vlan)#vlan 20

Sa(config-vlan)#vlan 30

Sa(config-vlan)#int f0/1

Sa(config-if)#sw acc vlan 10

Sa(config-if)#int f0/2

Sa(config-if)#sw acc vlan 20

Sa(config-if)#int f0/3

Sa(config-if)#sw acc vlan 30

Sa(config-if)#int g0/1

Sa(config-if)#sw mode tru

Sa(config-if)#sw tru allowed vlan 10

Sa(config-if)#sw tru allowed vlan 20

Sa(config-if)#sw tru allowed vlan 30

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

```
Sa#wr
```

验证 vlan 创建成功

```
Sa#show vlan
```

10	VLAN0010	active	Fa0/1
20	VLAN0020	active	Fa0/2
30	VLAN0030	active	Fa0/3

至此，接入层设备所有 VLAN 创建完毕，然后开始设置接入 VLAN 的网关，即设置逻辑接口。

在汇聚层设备 MS1 中

```
Switch>en
```

```
Switch#conf t
```

```
Switch(config)#hostname MS1
```

```
MS1(config)#vlan 10
```

```
MS1(config-vlan)#vlan 20
```

```
MS1(config-vlan)#vlan 30
```

```
MS1(config-if)#int vlan 10
```

```
MS1(config-if)#ip add 10.4.10.254 255.255.255.0
```

```
MS1(config-if)#int vlan 20
```

```
MS1(config-if)#ip add 10.4.20.254 255.255.255.0
```

```
MS1(config-if)#int vlan 30
```

```
MS1(config-if)#ip add 10.4.30.254 255.255.255.0
```

```
MS1(config-if)#int g0/1
```

```
MS1(config-if)#sw acc vlan 10
```

```
MS1(config-if)#int g0/2
```

```
MS1(config-if)#sw acc vlan 20
```

```
MS1(config-if)#int g0/3
```

```
MS1(config-if)#sw acc vlan 30
```

```
MS1(config-if)#end
```

```
MS1#wr
```

验证创建成功，且部门设备可以 ping 通网关

```
MS1# show ip int bri
```

Vlan10	10.4.10.254	YES manual up
Vlan20	10.4.20.254	YES manual up
Vlan30	10.4.30.254	YES manual up

```
c:\>ping 10.4.10.254
Pinging 10.4.10.254 with 32 bytes of data:
Reply from 10.4.10.254: bytes=32 time<1ms TTL=255
```

VLAN10 设备 ping 网关

```
C:\>ping 10.4.20.254
Pinging 10.4.20.254 with 32 bytes of data:
Reply from 10.4.20.254: bytes=32 time<1ms TTL=255
```

VLAN20 设备 ping 网关

```
C:\>ping 10.4.30.254
Pinging 10.4.30.254 with 32 bytes of data:
Reply from 10.4.30.254: bytes=32 time<1ms TTL=255
```

VLAN30 设备 ping 网关

同理完成 MS2 接入 VLAN 的网关设置

```
Switch>en
Switch#conf t
Switch(config)#hostname MS2
MS2(config)#vlan 40
MS2(config-vlan)#vlan 50
MS2(config-vlan)#vlan 60
MS2(config-vlan)#int vlan 40
MS2(config-if)#ip add 10.4.40.254 255.255.255.0
MS2(config-if)#int vlan 50
```

```
MS2(config-if)#ip add 10.4.50.254 255.255.255.0
MS2(config-if)#int vlan 60
MS2(config-if)#ip add 10.4.60.254 255.255.255.0
MS2(config-if)#int g0/1
MS2(config-if)#sw acc vlan 40
MS2(config-if)#int g0/2
MS2(config-if)#sw acc vlan 50
MS2(config-if)#int g0/3
MS2(config-if)#sw acc vlan 60
MS2(config-if)#end
MS2#wr
```

验证创建成功，且部门设备可以 ping 通网关

```
MS2#show ip int bri
```

Vlan40	10.4.40.254	YES manual up
Vlan50	10.4.50.254	YES manual up
Vlan60	10.4.60.254	YES manual up

```
C:\>ping 10.4.40.254
Pinging 10.4.40.254 with 32 bytes of data:
Reply from 10.4.40.254: bytes=32 time<1ms TTL=255
```

VLAN40 设备 ping 网关

```
C:\>ping 10.4.50.254
Pinging 10.4.50.254 with 32 bytes of data:
Reply from 10.4.50.254: bytes=32 time<1ms TTL=255
```

VLAN50 设备 ping 网关

```
C:\>ping 10.4.60.254
Pinging 10.4.60.254 with 32 bytes of data:
Reply from 10.4.60.254: bytes=32 time<1ms TTL=255
```

VLAN60 设备 ping 网关

VLAN70 的网关设置在核心设备 Core2 上

```
Switch>
Switch>en
Switch#conf t
Switch#hostname Core2
Core2(config)#vlan 70
Core2(config-vlan)#int vlan 70
Core2(config-if)#ip add 10.4.70.254 255.255.255.0
Core2(config-if)#int g1/0/3
Core2(config-if)#sw acc vlan 70
Core2(config-if)#end
Core2#wr
```

验证创建成功且部门设备可以 ping 通网关

```
Core2# show ip int bri
```

Vlan70	10.4.70.254	YES manual up
--------	-------------	---------------

```
C:\>ping 10.4.70.254
Pinging 10.4.70.254 with 32 bytes of data:
Reply from 10.4.70.254: bytes=32 time<1ms TTL=255
Reply from 10.4.70.254: bytes=32 time<1ms TTL=255
Reply from 10.4.70.254: bytes=32 time=9ms TTL=255
Reply from 10.4.70.254: bytes=32 time<1ms TTL=255
```

VLAN 设备 ping 网关

至此完成了所有接入网关的设置

接下来配置互联 VLAN

MS1——Core1 为 VLAN 100

MS1——Core2 为 VLAN 110

MS2——Core1 为 VLAN 120

MS2——Core2 为 VLAN 130

Core1——Core2 为 VLAN140

Core1——R1 为 VLAN 150

Core2——R1 为 VLAN 160

分别在这些设备上创建 VLAN 且设置逻辑接口地址

在 MS1 上配置逻辑接口地址

```
MS1>en
MS1#conf t
MS1(config)#vlan 100
MS1(config-vlan)#vlan 110
MS1(config-vlan)#int vlan 100
MS1(config-if)#ip add 192.168.4.101 255.255.255.252
MS1(config-if)#int vlan 110
MS1(config-if)#ip add 192.168.4.113 255.255.255.252
MS1(config-if)#int g1/0/24
MS1(config-if)#sw acc vlan 100
MS1(config-if)#int g1/0/23
MS1(config-if)#sw acc vlan 110
MS1(config-if)#end
MS1#wr
```

验证逻辑接口成功创建

```
MS1#show ip int bri
+-----+-----+-----+-----+
|  VLAN |      IP Address     | Status | MTU |
+-----+-----+-----+-----+
|    10  | 10.4.10.254         | YES    | 1500 |
|    20  | 10.4.20.254         | YES    | 1500 |
|    30  | 10.4.30.254         | YES    | 1500 |
|   100  | 192.168.4.101       | YES    | 1500 |
|   110  | 192.168.4.113       | YES    | 1500 |
+-----+-----+-----+-----+
```

在 MS2 上配置逻辑接口地址

```
MS2>en
MS2#conf t
MS2(config-vlan)#vlan 120
MS2(config-vlan)#vlan 130
MS2(config-vlan)#int vlan 120
MS2(config-if)#ip add 192.168.4.121 255.255.255.252
MS2(config-if)#int vlan 130
MS2(config-if)#ip add 192.168.4.133 255.255.255.252
MS2(config-if)#int g1/0/23
MS2(config-if)#sw acc vlan 120
MS2(config-if)#int g1/0/24
MS2(config-if)#sw acc vlan 130
```

```
MS2(config-if)#end
```

```
MS2#
```

```
MS2#wr
```

验证逻辑接口创建成功

```
MS2#show ip int bri
```

Vlan40	10.4.40.254	YES manual up
Vlan50	10.4.50.254	YES manual up
Vlan60	10.4.60.254	YES manual up
Vlan120	192.168.4.121	YES manual up
Vlan130	192.168.4.133	YES manual up
"		

在 Core1 上配置逻辑接口地址

```
Core1>en
```

```
Core1#conf t
```

```
Core1(config)#vlan 100
```

```
Core1(config-vlan)#vlan 120
```

```
Core1(config-vlan)#vlan 140
```

```
Core1(config-vlan)#vlan 150
```

```
Core1(config-vlan)#int vlan 100
```

```
Core1(config-if)#ip add 192.168.4.102 255.255.255.252
```

```
Core1(config-if)#int vlan 120
```

```
Core1(config-if)#ip add 192.168.4.122 255.255.255.252
```

```
Core1(config-if)#int vlan 140
```

```
Core1(config-if)#ip add 192.168.141 255.255.255.252
```

```
Core1(config-if)#int vlan 150
```

```
Core1(config-if)#ip add 192.168.4.153 255.255.255.252
```

```
Core1(config-if)#int g1/0/1
```

```
Core1(config-if)#sw acc vlan 100
```

```
Core1(config-if)#int g1/0/2
```

```
Core1(config-if)#sw acc vlan 120
```

```
Core1(config-if)#int range g1/0/23-g1/0/24
```

```
Core1(config-if-range)#sw acc vlan 140
```

```
Core1(config-if-range)#int g1/0/10
```

```
Core1(config-if)#sw acc vlan 150
```

```
Core1(config-if)#end
```

```
Core1#  
Core1(config-if)#wr
```

验证逻辑接口创建成功

```
Core1#show ip int bri  
      |Vlan100          192.168.4.102    YES manual up  
      |Vlan120          192.168.4.122    YES manual up  
      |Vlan140          192.168.4.141    YES manual up  
      |Vlan150          192.168.4.153    YES manual up
```

在 Core2 上配置逻辑接口地址

```
Core2>  
Core2>en  
Core2#conf t  
Core2(config)#vlan 110  
Core2(config-vlan)#vlan 130  
Core2(config-vlan)#vlan 140  
Core2(config-vlan)#vlan 160  
Core2(config-vlan)#int vlan 110  
Core2(config-if)#ip add 192.168.4.114 255.255.255.252  
Core2(config-if)#int vlan 130  
Core2(config-if)#ip add 192.168.4.134 255.255.255.252  
Core2(config-if)#int vlan 140  
Core2(config-if)#ip add 192.168.142 255.255.255.252  
Core2(config-if)#int vlan 160  
Core2(config-if)#ip add 192.168.4.161 255.255.255.252  
Core2(config-if)#int g1/0/1  
Core2(config-if)#sw acc vlan 130  
Core2(config-if)#int g1/0/2  
Core2(config-if)#sw acc vlan 110  
Core2(config-if)#int range g1/0/23-g1/0/24  
Core2(config-if-range)#sw acc vlan 140  
Core2(config-if-range)#int g1/0/10  
Core2(config-if)#sw acc vlan 160  
Core2(config-if)#end  
Core2#
```

```
Core2#wr
```

验证逻辑接口创建成功

```
Core2#show ip int bri
```

Vlan70	10.4.70.254	YES	manual	up
Vlan110	192.168.4.114	YES	manual	up
Vlan130	192.168.4.134	YES	manual	up
Vlan140	192.168.4.142	YES	manual	up
Vlan160	192.168.4.161	YES	manual	up

在 R1 上配置逻辑接口地址

```
R1>
```

```
R1>en
```

```
R1(config)#conf t
```

```
R1(config)#int g0/0/0
```

```
R1(config-if)#ip add 192.168.4.154 255.255.255.252
```

```
R1(config-if)#int g1/0/1
```

```
R1(config-if)#ip add 192.168.4.162 255.255.255.252
```

```
R1(config-if)#end
```

```
R1#
```

```
R1#wr
```

验证逻辑接口创建成功

```
R1#show ip int bri
```

Interface	IP-Address	OK?	Method	Status
GigabitEthernet0/0/0	192.168.4.154	YES	manual	up
GigabitEthernet0/0/1	192.168.4.162	YES	manual	up
...				

至此，功能 1 要求已完成

## 2、采用链路聚合技术提高核心层交换机之间的链路带宽。

为了保证链路聚合即提升带宽也满足冗余的要求，使用 Static LACP 静态聚合

```
Core1
```

```
Core1>en
```

```
Core1(config)#conf t
```

```
Core1(config)#int range g1/0/23-g1/0/24
```

```
Core1(config-if-range)#channel-group 1 mode on
```

```
Core1(config-if-range)#int port-channel 1
```

```
Core1(config-if)#sw acc vlan 140
```

```
Core1(config-if)#end
```

```
Core1#
```

```
Core1#wr
```

验证聚合端口已经启用

```
Core1#show ip int bri
```

Core1#show ip int bri				
Interface	IP-Address	OK?	Method	Status
Port-channel1	unassigned	YES	unset	up

Core2 的配置与 Core1 完全相同

```
Core2>en
```

```
Core2(config)#conf t
```

```
Core2(config)#int range g1/0/23-g1/0/24
```

```
Core2(config-if-range)#channel-group 1 mode on
```

```
Core2(config-if-range)#int port-channel 1
```

```
Core2(config-if)#sw acc vlan 140
```

```
Core2(config-if)#end
```

```
Core2#
```

```
Core2#wr
```

验证聚合端口已经启用

```
Core1#show ip int bri
```

Core2#show ip int bri				
Interface	IP-Address	OK?	Method	Status
Port-channel1	unassigned	YES	unset	up

验证带宽增加且断开一条后仍然连通

```
Core2#show int port-channel 1
```

```
Core2#show int port-channel 1
Port-channel1 is up, line protocol is up (connected)
  Hardware is EtherChannel, address is 000a.f373.e83b (bia
  000a.f373.e83b)
    MTU 1500 bytes, BW 2000000 Kbit, DLY 1000 usec,
```

可以看到两个 GE 端口聚合成了一个带宽为 2GE 的端口

手动 down 掉一个端口后仍然能互通

```
Core2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Core2(config)#int g1/0/23
Core2(config-if)#shu
Core2(config-if)#shutdown

Core2(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet1/0/23, changed state to
administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet1/0/23, changed state to down
ex
Core2(config)#ex
Core2#
%SYS-5-CONFIG_I: Configured from console by console
ping 192.168.4.141

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.141, timeout is 2
seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 0/0/0 ms
```

3、在企业网内部配置 RIPv2 协议。要求在汇聚层交换机 MS1 和 MS2，核心层交换机 Core1 和 Core2，路由器 R1 上运行 RIPv2 协议。

配置 RIPv2 协议需要理清该设备直连的网段

MS1: 10.4.x.0/24, 192.168.4.x/30

MS2: 10.4.x.0/24, 192.168.4.x/30

Core1: 192.168.4.x/30

Core2: 10.4.x.0/24, 192.168.4.x/30

R1: 192.168.4.x/30, 66.4.200.0/24

RIPv2 协议中会将 10.x.x.x 自动配置成 10.0.0.0, MS1, MS2, Core2 完全相同

给 MS1 配置 RIPv2 协议

MS1>

MS1>en

MS1#conf t

MS1(config)#route rip

MS1(config-router)#v 2

MS1(config-router)#no auto

MS1(config-router)#net 10.0.0.0

MS1(config-router)#net 192.168.4.0

MS1(config-router)#end

MS1#

MS1#wr

验证 RIPv2 协议配置成功

MS1#show ip route

R 为通过 RIP 协议学习到的路由

```
Gateway of last resort is 192.168.4.114 to network 0.0.0.0

    10.0.0.0/24 is subnetted, 7 subnets
C        10.4.10.0 is directly connected, Vlan10
C        10.4.20.0 is directly connected, Vlan20
C        10.4.30.0 is directly connected, Vlan30
R        10.4.40.0 [120/2] via 192.168.4.114, 00:00:15, Vlan110
                  [120/2] via 192.168.4.102, 00:00:04, Vlan100
R        10.4.50.0 [120/2] via 192.168.4.114, 00:00:15, Vlan110
                  [120/2] via 192.168.4.102, 00:00:04, Vlan100
R        10.4.60.0 [120/2] via 192.168.4.114, 00:00:15, Vlan110
                  [120/2] via 192.168.4.102, 00:00:04, Vlan100
R        10.4.70.0 [120/1] via 192.168.4.114, 00:00:15, Vlan110
66.0.0.0/24 is subnetted, 1 subnets
R        66.4.200.0 [120/2] via 192.168.4.102, 00:00:04, Vlan100
                  [120/2] via 192.168.4.114, 00:00:15, Vlan110
192.168.4.0/30 is subnetted, 7 subnets
C        192.168.4.100 is directly connected, Vlan100
C        192.168.4.112 is directly connected, Vlan110
R        192.168.4.120 [120/1] via 192.168.4.102, 00:00:04, Vlan100
R        192.168.4.132 [120/1] via 192.168.4.114, 00:00:15, Vlan110
R        192.168.4.140 [120/1] via 192.168.4.102, 00:00:04, Vlan100
                  [120/1] via 192.168.4.114, 00:00:15, Vlan110
R        192.168.4.152 [120/1] via 192.168.4.102, 00:00:04, Vlan100
R        192.168.4.160 [120/1] via 192.168.4.114, 00:00:15, Vlan110
R*       0.0.0.0/0 [120/2] via 192.168.4.114, 00:00:15, Vlan110
                  [120/2] via 192.168.4.102, 00:00:04, Vlan100
```

给 MS2 配置 RIPv2 协议

MS2>

MS2>en

MS2#conf t

MS2(config)#route rip

MS2(config-router)#v 2

MS2(config-router)#no auto

MS2(config-router)#net 10.0.0.0

MS2(config-router)#net 192.168.4.0

MS2(config-router)#end

MS2#

MS2#wr

验证 RIPv2 协议配置成功

MS2#show ip route

R 为通过 RIP 协议学习到的路由

```
Gateway of last resort is 192.168.4.134 to network 0.0.0.0

    10.0.0.0/24 is subnetted, 7 subnets
R      10.4.10.0 [120/2] via 192.168.4.122, 00:00:10, Vlan120
          [120/2] via 192.168.4.134, 00:00:20, Vlan130
R      10.4.20.0 [120/2] via 192.168.4.122, 00:00:10, Vlan120
          [120/2] via 192.168.4.134, 00:00:20, Vlan130
R      10.4.30.0 [120/2] via 192.168.4.122, 00:00:10, Vlan120
          [120/2] via 192.168.4.134, 00:00:20, Vlan130
C      10.4.40.0 is directly connected, Vlan40
C      10.4.50.0 is directly connected, Vlan50
C      10.4.60.0 is directly connected, Vlan60
R      10.4.70.0 [120/1] via 192.168.4.134, 00:00:20, Vlan130
66.0.0.0/24 is subnetted, 1 subnets
R      66.4.200.0 [120/2] via 192.168.4.122, 00:00:10, Vlan120
          [120/2] via 192.168.4.134, 00:00:20, Vlan130
192.168.4.0/30 is subnetted, 7 subnets
R      192.168.4.100 [120/1] via 192.168.4.122, 00:00:10, Vlan120
R      192.168.4.112 [120/1] via 192.168.4.134, 00:00:20, Vlan130
C      192.168.4.120 is directly connected, Vlan120
C      192.168.4.132 is directly connected, Vlan130
R      192.168.4.140 [120/1] via 192.168.4.122, 00:00:10, Vlan120
          [120/1] via 192.168.4.134, 00:00:20, Vlan130
R      192.168.4.152 [120/1] via 192.168.4.122, 00:00:10, Vlan120
R      192.168.4.160 [120/1] via 192.168.4.134, 00:00:20, Vlan130
R*     0.0.0.0/0 [120/2] via 192.168.4.134, 00:00:20, Vlan130
          [120/2] via 192.168.4.122, 00:00:10, Vlan120
```

给 Core1 配置 RIPv2 协议

Core1>

Core1>en

Core1#conf t

Core1(config)#route rip

Core1(config-router)#v 2

Core1(config-router)#no auto

Core1(config-router)#net 192.168.4.0

Core1(config-router)#end

Core1#wr

验证 RIPv2 协议配置成功

Core1#show ip route

R 为通过 RIP 协议学习到的路由

```

Gateway of last resort is 192.168.4.154 to network 0.0.0.0

    10.0.0.0/24 is subnetted, 7 subnets
R      10.4.10.0 [120/1] via 192.168.4.101, 00:00:11, Vlan100
R      10.4.20.0 [120/1] via 192.168.4.101, 00:00:11, Vlan100
R      10.4.30.0 [120/1] via 192.168.4.101, 00:00:11, Vlan100
R      10.4.40.0 [120/1] via 192.168.4.121, 00:00:13, Vlan120
R      10.4.50.0 [120/1] via 192.168.4.121, 00:00:13, Vlan120
R      10.4.60.0 [120/1] via 192.168.4.121, 00:00:13, Vlan120
R      10.4.70.0 [120/1] via 192.168.4.142, 00:00:13, Vlan140
    66.0.0.0/24 is subnetted, 1 subnets
R      66.4.200.0 [120/1] via 192.168.4.154, 00:00:16, Vlan150
    192.168.4.0/30 is subnetted, 7 subnets
C      192.168.4.100 is directly connected, Vlan100
R      192.168.4.112 [120/1] via 192.168.4.142, 00:00:13, Vlan140
                  [120/1] via 192.168.4.101, 00:00:11, Vlan100
C      192.168.4.120 is directly connected, Vlan120
R      192.168.4.132 [120/1] via 192.168.4.142, 00:00:13, Vlan140
                  [120/1] via 192.168.4.121, 00:00:13, Vlan120
C      192.168.4.140 is directly connected, Vlan140
C      192.168.4.152 is directly connected, Vlan150
R      192.168.4.160 [120/1] via 192.168.4.142, 00:00:13, Vlan140
                  [120/1] via 192.168.4.154, 00:00:16, Vlan150
R*     0.0.0.0/0 [120/1] via 192.168.4.154, 00:00:16, Vlan150

```

### 给 Core2 配置 RIPv2 协议

Core2>

Core2>en

Core2(config)#conf t

Core2#route rip

Core2(config-router)#v 2

Core2(config-router)#no auto

Core2(config-router)#net 10.0.0.0

Core2(config-router)#net 192.168.4.0

Core2(config-router)#end

Core2#

Core2#wr

### 验证 RIPv2 协议配置成功

Core2#show ip route

R 为通过 RIP 协议学习到的路由

```

Gateway of last resort is 192.168.4.162 to network 0.0.0.0

      10.0.0.0/24 is subnetted, 7 subnets
R        10.4.10.0 [120/1] via 192.168.4.113, 00:00:19, Vlan110
R        10.4.20.0 [120/1] via 192.168.4.113, 00:00:19, Vlan110
R        10.4.30.0 [120/1] via 192.168.4.113, 00:00:19, Vlan110
R        10.4.40.0 [120/1] via 192.168.4.133, 00:00:20, Vlan130
R        10.4.50.0 [120/1] via 192.168.4.133, 00:00:20, Vlan130
R        10.4.60.0 [120/1] via 192.168.4.133, 00:00:20, Vlan130
C        10.4.70.0 is directly connected, Vlan70
      66.0.0.0/24 is subnetted, 1 subnets
R          66.4.200.0 [120/1] via 192.168.4.162, 00:00:24, Vlan160
      192.168.4.0/30 is subnetted, 7 subnets
R          192.168.4.100 [120/1] via 192.168.4.141, 00:00:19, Vlan140
                           [120/1] via 192.168.4.113, 00:00:19, Vlan110
C          192.168.4.112 is directly connected, Vlan110
R          192.168.4.120 [120/1] via 192.168.4.141, 00:00:19, Vlan140
                           [120/1] via 192.168.4.133, 00:00:20, Vlan130
C          192.168.4.132 is directly connected, Vlan130
C          192.168.4.140 is directly connected, Vlan140
R          192.168.4.152 [120/1] via 192.168.4.141, 00:00:19, Vlan140
                           [120/1] via 192.168.4.162, 00:00:24, Vlan160
C          192.168.4.160 is directly connected, Vlan160
R*        0.0.0.0/0 [120/1] via 192.168.4.162, 00:00:24, Vlan160

```

### 给 R1 配置 RIPv2 协议

R1>

R1>en

R1#conf t

R1(config)#route rip

R1(config-router)#v 2

R1(config-router)#no auto

R1(config-router)#net 66.4.200.0

R1(config-router)#net 192.168.4.0

R1(config-router)#end

R1#wr

### 验证 RIPv2 协议配置成功

R1#show ip route

R 为通过 RIP 协议学习到的路由

```

Gateway of last resort is 66.4.200.254 to network 0.0.0.0

  10.0.0.0/24 is subnetted, 7 subnets
R    10.4.10.0/24 [120/2] via 192.168.4.153, 00:00:22, GigabitEthernet0/0/0
      [120/2] via 192.168.4.161, 00:00:24, GigabitEthernet0/0/1
R    10.4.20.0/24 [120/2] via 192.168.4.153, 00:00:22, GigabitEthernet0/0/0
      [120/2] via 192.168.4.161, 00:00:24, GigabitEthernet0/0/1
R    10.4.30.0/24 [120/2] via 192.168.4.153, 00:00:22, GigabitEthernet0/0/0
      [120/2] via 192.168.4.161, 00:00:24, GigabitEthernet0/0/1
R    10.4.40.0/24 [120/2] via 192.168.4.161, 00:00:24, GigabitEthernet0/0/1
      [120/2] via 192.168.4.153, 00:00:22, GigabitEthernet0/0/0
R    10.4.50.0/24 [120/2] via 192.168.4.161, 00:00:24, GigabitEthernet0/0/1
      [120/2] via 192.168.4.153, 00:00:22, GigabitEthernet0/0/0
R    10.4.60.0/24 [120/2] via 192.168.4.161, 00:00:24, GigabitEthernet0/0/1
      [120/2] via 192.168.4.153, 00:00:22, GigabitEthernet0/0/0
R    10.4.70.0/24 [120/1] via 192.168.4.161, 00:00:24, GigabitEthernet0/0/1
66.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C      66.4.200.0/24 is directly connected, Serial0/2/0
L      66.4.200.1/32 is directly connected, Serial0/2/0
C      66.4.200.254/32 is directly connected, Serial0/2/0
192.168.4.0/24 is variably subnetted, 9 subnets, 2 masks
R      192.168.4.100/30 [120/1] via 192.168.4.153, 00:00:22, GigabitEthernet0/0/0
R      192.168.4.112/30 [120/1] via 192.168.4.161, 00:00:24, GigabitEthernet0/0/1
R      192.168.4.120/30 [120/1] via 192.168.4.153, 00:00:22, GigabitEthernet0/0/0
R      192.168.4.132/30 [120/1] via 192.168.4.161, 00:00:24, GigabitEthernet0/0/1
R      192.168.4.140/30 [120/1] via 192.168.4.161, 00:00:24, GigabitEthernet0/0/1
      [120/1] via 192.168.4.153, 00:00:22, GigabitEthernet0/0/0
C      192.168.4.152/30 is directly connected, GigabitEthernet0/0/0
L      192.168.4.154/32 is directly connected, GigabitEthernet0/0/0
C      192.168.4.160/30 is directly connected, GigabitEthernet0/0/1
L      192.168.4.162/32 is directly connected, GigabitEthernet0/0/1
S*    0.0.0.0/0 [1/0] via 66.4.200.254

```

4、在路由器 R1 上配置默认路由，以接入互联网。同时，向 RIP 网络中注入默认路由。

```

R1>
R1>en
R1#conf t

```

```
R1(config)#ip route 0.0.0.0 0.0.0.0 66.4.200.254
```

```
R1(config)#route rip
```

```
R1(config-router)#redistribute static
```

```
R1(config-router)#end
```

```
R1#
```

```
R1#wr
```

验证默认路由设置成功，并且 RIP 中也注入了默认路由

S 表示 Static，静态路由，R\*表示向 RIP 网络中的默认路由

```
R1#show ip route
```

```

Gateway of last resort is 66.4.200.254 to network 0.0.0.0

      10.0.0.0/24 is subnetted, 7 subnets
R        10.4.10.0/24 [120/2] via 192.168.4.153, 00:00:10, GigabitEthernet0/0/0
                  [120/2] via 192.168.4.161, 00:00:13, GigabitEthernet0/0/1
R        10.4.20.0/24 [120/2] via 192.168.4.153, 00:00:10, GigabitEthernet0/0/0
                  [120/2] via 192.168.4.161, 00:00:13, GigabitEthernet0/0/1
R        10.4.30.0/24 [120/2] via 192.168.4.153, 00:00:10, GigabitEthernet0/0/0
                  [120/2] via 192.168.4.161, 00:00:13, GigabitEthernet0/0/1
R        10.4.40.0/24 [120/2] via 192.168.4.161, 00:00:13, GigabitEthernet0/0/1
                  [120/2] via 192.168.4.153, 00:00:10, GigabitEthernet0/0/0
R        10.4.50.0/24 [120/2] via 192.168.4.161, 00:00:13, GigabitEthernet0/0/1
                  [120/2] via 192.168.4.153, 00:00:10, GigabitEthernet0/0/0
R        10.4.60.0/24 [120/2] via 192.168.4.161, 00:00:13, GigabitEthernet0/0/1
                  [120/2] via 192.168.4.153, 00:00:10, GigabitEthernet0/0/0
R        10.4.70.0/24 [120/1] via 192.168.4.161, 00:00:13, GigabitEthernet0/0/1
66.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C          66.4.200.0/24 is directly connected, Serial0/2/0
L          66.4.200.1/32 is directly connected, Serial0/2/0
C          66.4.200.254/32 is directly connected, Serial0/2/0
192.168.4.0/24 is variably subnetted, 9 subnets, 2 masks
R            192.168.4.100/30 [120/1] via 192.168.4.153, 00:00:10, GigabitEthernet0/0/0
R            192.168.4.112/30 [120/1] via 192.168.4.161, 00:00:13, GigabitEthernet0/0/1
R            192.168.4.120/30 [120/1] via 192.168.4.153, 00:00:10, GigabitEthernet0/0/0
R            192.168.4.132/30 [120/1] via 192.168.4.161, 00:00:13, GigabitEthernet0/0/1
R            192.168.4.140/30 [120/1] via 192.168.4.161, 00:00:13, GigabitEthernet0/0/1
                  [120/1] via 192.168.4.153, 00:00:10, GigabitEthernet0/0/0
C            192.168.4.152/30 is directly connected, GigabitEthernet0/0/0
L            192.168.4.154/32 is directly connected, GigabitEthernet0/0/0
C            192.168.4.160/30 is directly connected, GigabitEthernet0/0/1
L            192.168.4.162/32 is directly connected, GigabitEthernet0/0/1
S*          0.0.0.0/0 [1/0] via 66.4.200.254

```

MS1#show ip route

```

Gateway of last resort is 192.168.4.114 to network 0.0.0.0

      10.0.0.0/24 is subnetted, 7 subnets
C        10.4.10.0 is directly connected, Vlan10
C        10.4.20.0 is directly connected, Vlan20
C        10.4.30.0 is directly connected, Vlan30
R        10.4.40.0 [120/2] via 192.168.4.114, 00:00:15, Vlan110
                  [120/2] via 192.168.4.102, 00:00:04, Vlan100
R        10.4.50.0 [120/2] via 192.168.4.114, 00:00:15, Vlan110
                  [120/2] via 192.168.4.102, 00:00:04, Vlan100
R        10.4.60.0 [120/2] via 192.168.4.114, 00:00:15, Vlan110
                  [120/2] via 192.168.4.102, 00:00:04, Vlan100
R        10.4.70.0 [120/1] via 192.168.4.114, 00:00:15, Vlan110
66.0.0.0/24 is subnetted, 1 subnets
R        66.4.200.0 [120/2] via 192.168.4.102, 00:00:04, Vlan100
                  [120/2] via 192.168.4.114, 00:00:15, Vlan110
192.168.4.0/30 is subnetted, 7 subnets
C        192.168.4.100 is directly connected, Vlan100
C        192.168.4.112 is directly connected, Vlan110
R        192.168.4.120 [120/1] via 192.168.4.102, 00:00:04, Vlan100
R        192.168.4.132 [120/1] via 192.168.4.114, 00:00:15, Vlan110
R        192.168.4.140 [120/1] via 192.168.4.102, 00:00:04, Vlan100
                  [120/1] via 192.168.4.114, 00:00:15, Vlan110
R        192.168.4.152 [120/1] via 192.168.4.102, 00:00:04, Vlan100
R        192.168.4.160 [120/1] via 192.168.4.114, 00:00:15, Vlan110
R*        0.0.0.0/0 [120/2] via 192.168.4.114, 00:00:15, Vlan110
                  [120/2] via 192.168.4.102, 00:00:04, Vlan100

```

5、在交换机 Sa 上创建各个 VLAN，在路由器 R3 上使用单臂路由技术实现 VLAN 之间的通信。

Sa 中接入 VLAN 已在第一步中创建完成，在 R3 上实现单臂路由

R3>

R3>en

```
R3#conf t
R3(config)#int g0/0/0
R3(config-if)#no ip add
R3(config-if)#int g0/0/0.10
R3(config-subif)#encap dot1q 10
R3(config-subif)#ip add 66.4.10.254 255.255.255.0
R3(config-subif)#int g0/0/0.20
R3(config-subif)#encap dot1q 20
R3(config-subif)#ip add 66.4.10.254 255.255.255.0
R3(config-subif)#int g0/0/0.30
R3(config-subif)#encap dot1q 30
R3(config-subif)#ip add 66.4.30.254 255.255.255.0
R3(config-subif)#end
R3#
R3#wr
```

验证子接口配置成功

```
R3#show ip int bri
R3#show ip int bri
  Interface          IP-Address      OK? Method Status
  Protocol
  GigabitEthernet0/0/0  unassigned    YES unset   up
  GigabitEthernet0/0/0.10 66.4.10.254 YES manual  up
  GigabitEthernet0/0/0.20 66.4.20.254 YES manual  up
  GigabitEthernet0/0/0.30 66.4.30.254 YES manual  up
```

测试连通性

```
C:\>ping 66.4.10.254

Pinging 66.4.10.254 with 32 bytes of data:

Reply from 66.4.10.254: bytes=32 time<1ms TTL=255
```

PC ping 网关

```
C:\>ping 66.4.20.1
Pinging 66.4.20.1 with 32 bytes of data:
Request timed out.
Reply from 66.4.20.1: bytes=32 time<1ms TTL=127
Reply from 66.4.20.1: bytes=32 time<1ms TTL=127
Reply from 66.4.20.1: bytes=32 time<1ms TTL=127
```

PCa ping PCb

## 6、配置路由器 R2 和路由器 R3 运行 OSPF 协议。

先实现 R2 与 R3 连通

packet tracer 8.0 版本中自动配置 clock rate 2000000

配置 R2 与 R3 互联的端口

R2>

R2>en

R2#conf t

R2(config)#int s0/2/1

R2(config-if)#ip add 66.4.100.5 255.255.255.252

R2(config-if)#end

R2#

R2#wr

验证端口配置成功且开启

R2#show ip int bri

Serial0/2/0	66.4.200.254	YES	NVRAM	up
Serial0/2/1	66.4.100.5	YES	NVRAM	up

配置 R3 与 R2 相连的端口

R3>

R3>en

R3#conf t

R3(config)#int s0/2/0

R3(config-if)#ip add 66.4.100.6 255.255.255.252

R3(config-if)#end

R3#

R3#wr

验证端口配置成功且开启

R3#show ip int bri

Serial0/2/0	66.4.100.6	YES manual up
-------------	------------	---------------

验证 R2 与 R3 之间的连通性

```
R2#ping 66.4.100.6
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 66.4.100.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 6/17/24 ms
```

R2 Ping R3

给 R2 配置 OSPF 协议

R2>

R2>en

R2#conf t

R2(config)#router ospf 1

R2(config-router)#network 66.4.100.4 0.0.0.3 area 0

R2(config-router)#network 66.4.200.0 0.0.0.255 area 0

R2(config-router)#end

R2#wr

给 R3 配置 OSPF 协议

R3>

R3>en

R3#conf t

R3(config)#router ospf 1

R3(config-router)#network 66.4.100.4 0.0.0.3 area 0

R3(config-router)#network 66.4.10.0 0.0.0.255 area 0

R3(config-router)#network 66.4.20.0 0.0.0.255 area 0

R3(config-router)#network 66.4.30.0 0.0.0.255 area 0

R3(config-router)#end

R3#wr

验证 OSPF 协议生效，O 表示通过 OSPF 学习到的路由

R2#show ip route

```

Gateway of last resort is not set

  66.0.0.0/8 is variably subnetted, 8 subnets, 3 masks
O   66.4.10.0/24 [110/65] via 66.4.100.6, 02:04:32, Serial0/2/1
O   66.4.20.0/24 [110/65] via 66.4.100.6, 02:04:32, Serial0/2/1
O   66.4.30.0/24 [110/65] via 66.4.100.6, 02:04:32, Serial0/2/1
C   66.4.100.4/30 is directly connected, Serial0/2/1
L   66.4.100.5/32 is directly connected, Serial0/2/1
C   66.4.200.0/24 is directly connected, Serial0/2/0
C   66.4.200.1/32 is directly connected, Serial0/2/0
L   66.4.200.254/32 is directly connected, Serial0/2/0

```

R3#show ip route

```

Gateway of last resort is not set

  66.0.0.0/8 is variably subnetted, 9 subnets, 3 masks
C   66.4.10.0/24 is directly connected, GigabitEthernet0/0/0.10
L   66.4.10.254/32 is directly connected, GigabitEthernet0/0/0.10
C   66.4.20.0/24 is directly connected, GigabitEthernet0/0/0.20
L   66.4.20.254/32 is directly connected, GigabitEthernet0/0/0.20
C   66.4.30.0/24 is directly connected, GigabitEthernet0/0/0.30
L   66.4.30.254/32 is directly connected, GigabitEthernet0/0/0.30
C   66.4.100.4/30 is directly connected, Serial0/2/0
L   66.4.100.6/32 is directly connected, Serial0/2/0
O   66.4.200.0/24 [110/128] via 66.4.100.5, 02:05:09, Serial0/2/0

```

7、路由器 R1 与 R2 之间封装 PPP 协议，使用 CHAP 认证，密码为姓名全拼。

先实现 R1 与 R2 互通

R1>

R1>en

R1#conf t

R1(config)#int s0/2/0

R1(config-if)#ip add 66.4.200.1 255.255.255.0

R1(config-if)#end

R1#

R1#wr

验证端口配置成功

R1#show ip int bri

Serial0/2/0	66.4.200.1	YES	NVRAM	up
-------------	------------	-----	-------	----

R2>

R2>en

R2#conf t

```
R2(config)#int s0/2/0
R2(config-if)#ip add 66.4.200.254 255.255.255.0
R2(config-if)#end
R2#
R2#wr
```

验证端口配置成功

```
R2#show ip int bri
| Serial0/2/0          66.4.200.254      YES  NVRAM   up
```

给 R1 配置 PPP 协议

```
R1>
R1>en
R1#conf t
R1(config)#username R2 password zhanziyang
R1(config)#int s0/2/0
R1(config-if)#encap ppp
R1(config-if)#ppp auth chap
R1(config-if)#end
R1#
R1#wr
```

验证 PPP 协议配置成功

```
R1#show int s0/2/0
Serial0/2/0 is up, line protocol is up (connected)
  Hardware is HD64570
  Internet address is 66.4.200.1/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set, keepalive set (10 sec)
    TCP over
```

给 R2 配置 PPP 协议

```
R2>
R2>en
R2#conf t
R2(config)#username R1 password zhanziyang
```

```

R2(config)#int s0/2/0
R2(config-if)#encap ppp
R2(config-if)#ppp auth chap
R2(config-if)#end
R2#
R2#wr

```

验证 PPP 协议配置成功

```

R2#show int s0/2/0
R2#show int s0/2/0
Serial0/2/0 is up, line protocol is up (connected)
  Hardware is HD64570
  Internet address is 66.4.200.254/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set, keepalive set (10 sec)
  LCP Open

```

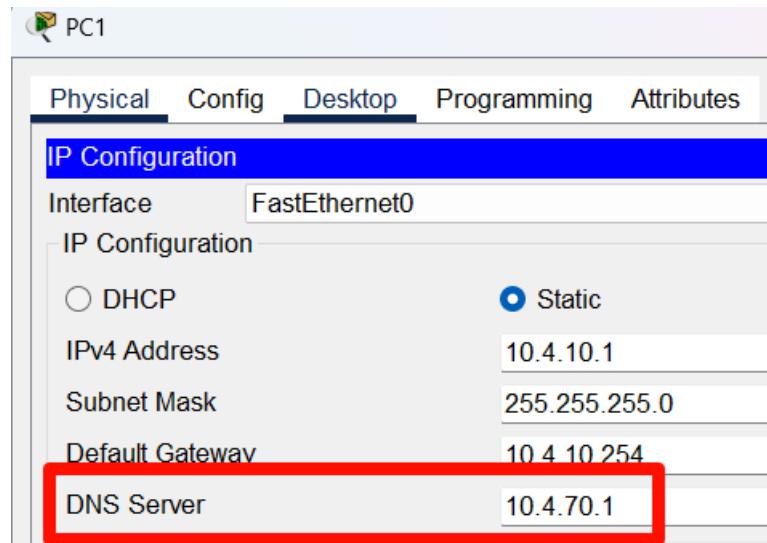
8、完成 DNS 服务器的配置。www1 服务器的域名为 www1.姓名全拼.com，www2 服务器的域名为 www2.姓名全拼.com。

进入服务器设置界面，添加 A 记录，A 记录对应 IPv4 解析，AAAA 对应 IPv6 解析  
 www1.zhanziyang.com 10.4.70.2  
 www2.zhanziyang.com 66.4.30.1

DNS			
DNS Service	<input checked="" type="radio"/> On	<input type="radio"/> Off	
Resource Records			
Name	<input type="text"/>		Type <input type="text" value="A Record"/>
Address	<input type="text"/>		
<input type="button" value="Add"/>		<input type="button" value="Save"/>	<input type="button" value="Remove"/>
No.	Name	Type	Detail
0	www1.zhanziyang.com	A Record	10.4.70.2
1	www2.zhanziyang.com	A Record	66.4.30.1

验证 DNS 生效，由于尚未连通互联网，先验证企业网内 WWW1\_Server 的 DNS 是否生效

设置 PC1 的 DNS 为 10.4.70.1



```
C:\>ping www1.zhanziyang.com
Pinging 10.4.70.2 with 32 bytes of data:
Request timed out.
Reply from 10.4.70.2: bytes=32 time<1ms TTL=126
Reply from 10.4.70.2: bytes=32 time<1ms TTL=126
Reply from 10.4.70.2: bytes=32 time<1ms TTL=126
```

PC1 Ping WWW1\_Server

9、配置 NAT。在路由器 R1 上配置 NAPT，使企业用户可以访问互联网资源。在路由器 R1 上配置静态 NAT，使互联网用户可以访问企业的 WWW1 服务器。

给 R1 配置 NAPT

R1>

R1>en

R1#conf t

R1(config)#int range g0/0/0-g0/0/1

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

R1(config-if-range)#int s0/2/0

R1(config-if)#ip nat out

R1(config-if)#ip nat pool zzy 66.4.200.2 66.4.200.100 netmask 255.255.255.0

R1(config)#

R1(config)#access-list 1 permit any

R1(config)#ip nat inside source list 1 pool zzy overload

```
R1(config)#ex
```

```
R1#wr
```

验证从企业网访问互联网，同时使用域名访问以验证上一步的 DNS 设置的正确性

```
C:\>ping www2.zhanziyang.com  
Pinging 66.4.30.1 with 32 bytes of data:  
Reply from 66.4.30.1: bytes=32 time=3ms TTL=123  
Reply from 66.4.30.1: bytes=32 time=2ms TTL=123  
Reply from 66.4.30.1: bytes=32 time=2ms TTL=123  
Reply from 66.4.30.1: bytes=32 time=4ms TTL=123
```

PC1 ping WWW2\_Server

配置 Static NAT

```
R1>
```

```
R1>en
```

```
R1#conf t
```

```
R1(config)#ip nat inside source static tcp 10.4.70.2 80 66.4.200.1 80
```

```
R1(config)#end
```

```
R1#
```

```
R1#wr
```

验证从互联网访问服务器



访问 NAT 后的 WWW1\_Server

10、在汇聚层交换机 MS1 上配置扩展 ACL，使财务部(VLAN30)的主机不能访问 FTP 服务器，其它服务不受影响。在汇聚层交换机 MS2 上配置扩展 ACL，使生产部(VLAN60)的主机不能访问 www2 服务器，其它服务不受影响。

先配置 VLAN30 不能访问 FTP\_Server

配置前进行测试，访问 FTP\_Server 的 TCP 21 端口，由图可见配置 ACL 前可以 Ping 通且可以访问 21 端口

```
C:\>ping 10.4.70.3
Pinging 10.4.70.3 with 32 bytes of data:
Request timed out.
Reply from 10.4.70.3: bytes=32 time<1ms TTL=126
Reply from 10.4.70.3: bytes=32 time<1ms TTL=126
Reply from 10.4.70.3: bytes=32 time<1ms TTL=126
```

Ping

```
C:\>ftp 10.4.70.3
Trying to connect...10.4.70.3
Connected to 10.4.70.3
220- Welcome to PT Ftp server
Username:zzy
331- Username ok, need password
Password:
230- Logged in
(passive mode on)
ftp>|
```

访问 FTP 服务

给 MS1 配置扩展 ACL

MS1>

MS1>en

MS1#conf t

MS1(config)#access-list 100 deny tcp 10.4.30.0 0.0.0.255 host 10.4.70.3 eq ftp

MS1(config)#access-list 100 permit ip any any

MS1(config-if)#int vlan 30

MS1(config-if)#ip access-group 100 in

MS1(config-if)#end

MS1#wr

验证 ACL 是否配置成功

MS1#show access-list 100

```
MS1#show access-lists 100
Extended IP access list 100
    deny tcp 10.4.30.0 0.0.0.255 host 10.4.70.3 eq ftp
    permit ip any any
```

验证是否 ACL 是否生效，由图可见配置 ACL 后可以 Ping 通 FTP\_Server，但是无法访问 FTP 服务所在的 TCP 21 端口

```
C:\>ping 10.4.70.3
Pinging 10.4.70.3 with 32 bytes of data:
Reply from 10.4.70.3: bytes=32 time<1ms TTL=126
```

Ping

```
C:\>ftp 10.4.70.3
Trying to connect...10.4.70.3
%Error opening ftp://10.4.70.3/ (Timed out)
.
```

访问 FTP 服务

配置 VLAN60 不能访问 WWW2\_Server

配置 ACL 前进行验证能否访问

```
C:\>ping 66.4.30.1
Pinging 66.4.30.1 with 32 bytes of data:
Reply from 66.4.30.1: bytes=32 time=2ms TTL=123
```

Ping



## Cisco Packet Tracer

Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide Open!

Quick Links:

[A small page](#)

[Copyrights](#)

http 80 端口



## Cisco Packet Tracer

Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide Open!

Quick Links:

[A small page](#)

[Copyrights](#)

https 443 端口

给 MS2 配置 ACL

MS2>

MS2>en

MS2#conf t

MS2(config)#access-list 100 deny tcp 10.4.60.0 0.0.0.255 host 66.4.30.1 eq www

MS2(config)#access-list 100 deny tcp 10.4.60.0 0.0.0.255 host 66.4.30.1 eq 443

MS2(config)#access-list 100 permit ip any any

MS2(config)#int vlan 60

MS2(config-if)#ip access-group 100 in

MS2(config-if)#end

MS2#wr

验证 ACL 是否生效



http 80 端口无法访问



https 443 端口无法访问

```
C:\>ping 66.4.30.1  
Pinging 66.4.30.1 with 32 bytes of data:  
Reply from 66.4.30.1: bytes=32 time=2ms TTL=123  
Reply from 66.4.30.1: bytes=32 time=2ms TTL=123  
Reply from 66.4.30.1: bytes=32 time=2ms TTL=123  
Reply from 66.4.30.1: bytes=32 time=2ms TTL=123
```

VLAN60 终端 Ping WWW2\_Server

至此，所有功能要求均已实现

## 项目整体测试

首先测试企业网内的 VLAN10 终端与 VLAN40 终端的连通性，因为 VLAN10 和 VLAN40 分别出于不同的接入交换机和汇聚交换，测试他们测连通性可以测试出汇聚层和核心层的连通性

```
C:\>ping 10.4.40.1  
Pinging 10.4.40.1 with 32 bytes of data:  
Request timed out.  
Reply from 10.4.40.1: bytes=32 time<1ms TTL=125  
Reply from 10.4.40.1: bytes=32 time<1ms TTL=125  
Reply from 10.4.40.1: bytes=32 time<1ms TTL=125
```

PC1 Ping PC7

然后测试企业网内 VLAN10 终端与 VLAN70 终端的连通性，因为 VLAN70 的网关设置在核心层交换机 Core2 中，这样可以测试出接入层——汇聚层——核心层的连通性。同时使用域名进行访问，以验证 DNS 生效。

```
C:\>ping www1.zhanziyang.com  
Pinging 10.4.70.2 with 32 bytes of data:  
Reply from 10.4.70.2: bytes=32 time<1ms TTL=126  
Reply from 10.4.70.2: bytes=32 time=10ms TTL=126  
Reply from 10.4.70.2: bytes=32 time<1ms TTL=126  
Reply from 10.4.70.2: bytes=32 time<1ms TTL=126
```

企业网到互联网的连通性在进行 NAT 配置的时候已经得到验证；企业网 VLAN 内连通性再接入层配置时已经验证，对这两个部分不再进行重复验证。

## 实训总结

通过这次实训回顾了上学期所学的计算机网络知识。从 OSI 的七层模型到每一层的协议。ARP 协议是二层间的通信协议，二层交换机维护一张 ARP 表来实现通信。三层设备，包括三层交换机、路由器等维护一张路由表来实现通信。而路由表的维护又引申出静态路由协议和动态路由协议。经典动态路由协议包括 RIP 和 OSPF，通过宣告自身直连网段可以和相邻路由器维护同一张路由表。VLAN 技术涉及到了 IEEE802.1Q 技术。在三层交换机到路由器时，需要进行 VLAN 帧的封装或者叫做 VLAN 终结来保证路由的三层接口可以正确收取数据包。

大型网络的构建往往涉及到控制广播风暴，链路冗余等问题。在这次实训中通过划分 VLAN 来实现广播域的控制，保证部门间的广播包不会互相影响，同时保证了连通性。冗余部分采用了双机热备、Static LACP 以及 OSPF。设置两台核心交换机保证任意一台核心 Down 的时候不影响企业网的使用。Static LACP 即提升了带宽也保证了链路可靠性。OSPF 协议的运用保证了在有任意一台设备 Down 的时候依旧能访问到出口路由 R1，不影响企业网内部使用互联网。扩展 ACL 的运用可以控制每个部门的访问权限，再细分可以控制不同时段不同部门指定设备的访问权限。

本次实训提升了我的实操能力，以及排除问题的能力，同时锻炼了大型网络构建的思维和提升我对大型网络构建的信心，这有助于在今后的工作中发挥我的能力。