



警示

- 1.实验报告如有雷同，雷同各方当次实验成绩均以 0 分计。
- 2.当次小组成员成绩只计学号、姓名登录在下表中的。
- 3.在规定时间内未上交实验报告的，不得以其他方式补交，当次成绩按 0 分计。
- 4.实验报告文件以 PDF 格式提交。

院系	计算机学院		班 级	计科2班	组长	刘晓丹、林隽哲	
学号	22336158	22336107	22336124	22336155	21312450	22365043	22302056
学生	刘晓丹	赖玟慈	李馨媛	刘苇嘉	林隽哲	江颢怡	刘彦凤

期末综合实验

【实验题目、拓扑】

综合实验的拓扑图如图 1 所示，简单地模拟了真实场景中用户内网与外网（或称公网）的网络互联。用户内网包括一台路由器 R1 和三台交换机。这三台交换机形成了一个二级级联结构，其中交换机 S0（模拟核心交换机），用于连接内网中不同部门的交换机（模拟接入交换机）。计算机科学系的交换机 S1 提供网络连接服务给计算机系的老师和学生，而电气工程系的交换机 S2 则为电气工程系的师生提供网络连接服务。此外，外网中设有一台 Web 服务器。

用户内网中的机器使用内网地址，当需要访问外网资源时，内网和外网之间的路由器负责执行地址转换。为了便于管理，用户内网还需设置 VLAN 和访问控制列表等功能。

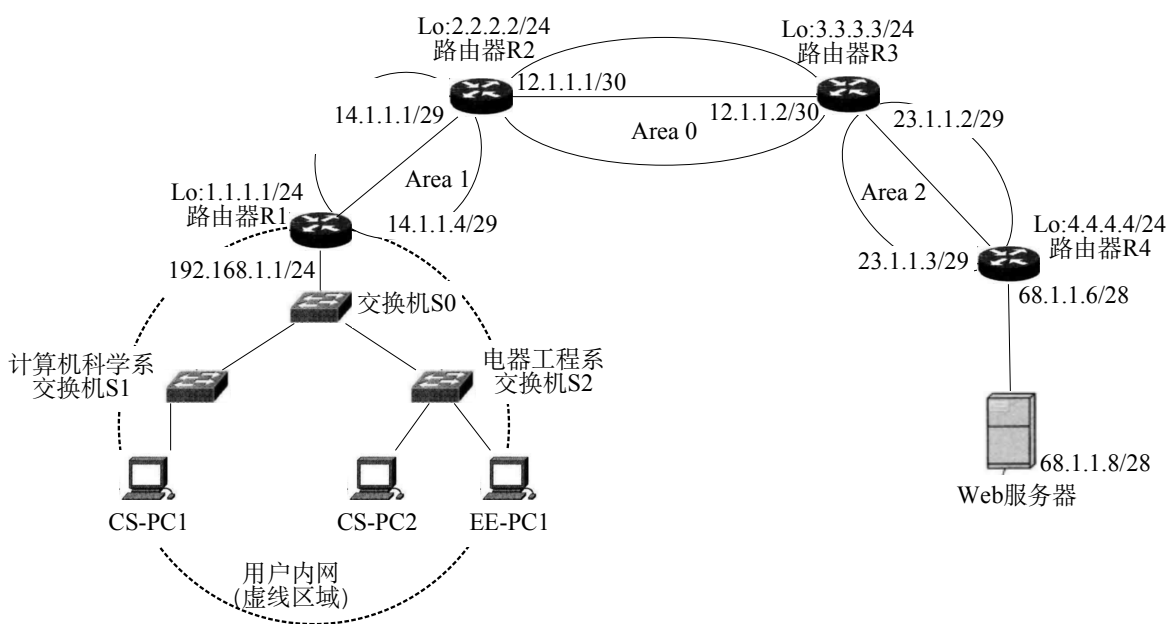


图 1 综合实验拓扑图



【实验步骤】

1. 实验网络环境搭建

- 1) 按图 1 搭建网络环境，配置所有设备网络接口的 IP 地址（没有给定 IP 地址的请自行设置 IP 地址）

由于此图需要两个小组以及四台路由器和交换机来搭建，因此我们此处将拓扑图又细分为以下的拓扑图：（图中接口为实验中实际接口）

- CS-PC1 : 192.168.1.2
- CS-PC2 : 192.168.1.3
- EE-PC1 : 192.168.1.4
- 默认网关都为：192.168.1.1

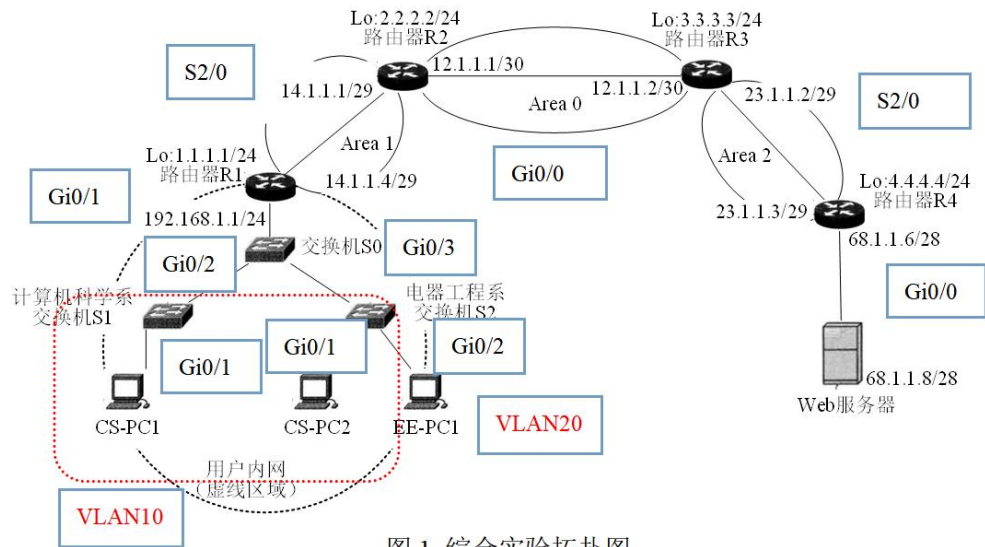


图 1 综合实验拓扑图

Router1 配置：（通过 show ip int br 验证配置）

```
1 R1#configure terminal
2 R1(config)#interface serial 2/0
3 R1(config-if)#ip add 14.1.1.1 255.255.255.248
4 R1(config-if)#no shut
5 R1(config-if)#exit
6 R1(config)#interface gigabitethernet 0/1
7 R1(config-if)#ip add 192.168.1.1 255.255.255.0
8 R1(config-if)#no shut
9 R1(config-if)#exit
10 R1(config)#interface loopback 1
11 R1(config-if)#ip add 1.1.1.1 255.255.255.0
12 R1(config-if)#end
```

R1 配置结果如下：

Interface	IP-Address(Pri)	IP-Address(Sec)	Status	Protocol
Serial 2/0	14.1.1.4/29	no address	up	up
Serial 2/1	no address	no address	down	down
GigabitEthernet 0/0	no address	no address	down	down
GigabitEthernet 0/1	192.168.1.1/24	no address	up	up
GigabitEthernet 0/2	no address	no address	down	down
GigabitEthernet 0/3	no address	no address	down	down
Loopback 1	1.1.1.1/24	no address	up	up
VLAN 1	no address	no address	up	down



Router2 配置:

```
1 R2#configure terminal
2 R2(config)#interface serial 2/0
3 R2(config-if)#ip add 14.1.1.1 255.255.255.248
4 R2(config-if)#no shut
5 R2(config-if)#exit
6 R2(config)#interface gigabitethernet 0/0
7 R2(config-if)#ip add 12.1.1.1 255.255.255.252
8 R2(config-if)#no shut
9 R2(config-if)#exit
10 R2(config)#interface loopback 1
11 R2(config-if)#ip add 2.2.2.2 255.255.255.0
12 R2(config-if)#end
```

R2 配置结果如下:

```
25-RSR20-2(config-if-Serial 2/1)#show ip int br
Interface                               IP-Address(Pri)    IP-Address(Sec)    Status    Protocol
Serial 2/0                             14.1.1.1/29        no address          up         up
Serial 2/1                             no address         no address          down       down
GigabitEthernet 0/0                    12.1.1.1/30        no address          up         up
GigabitEthernet 0/1                    no address         no address          down       down
GigabitEthernet 0/2                    no address         no address          down       down
GigabitEthernet 0/3                    no address         no address          down       down
```

Router3 配置:

```
1 R3#configure terminal
2 R3(config)#interface gigabitethernet 0/0
3 R3(config-if)#ip add 12.1.1.1 255.255.255.252
4 R3(config-if)#no shut
5 R3(config-if)#exit
6 R3(config)#interface serial 2/0
7 R3(config-if)#ip add 23.1.1.3 255.255.255.248
8 R3(config-if)#no shut
9 R3(config-if)#exit
10 R3(config)#interface loopback 1
11 R3(config-if)#ip add 3.3.3.3 255.255.255.0
12 R3(config-if)#end
```

R3 配置结果如下:

```
26-RSR20-1(config)#sh ip int b
Interface                               IP-Address(Pri)    IP-Address(Sec)    Status    Protocol
Serial 2/0                             23.1.1.2/29        no address          up         up
Serial 2/1                             no address         no address          down       down
GigabitEthernet 0/0                    12.1.1.2/30        no address          up         up
GigabitEthernet 0/1                    no address         no address          down       down
GigabitEthernet 0/2                    no address         no address          down       down
GigabitEthernet 0/3                    no address         no address          down       down
Loopback 1                             3.3.3.3/24         no address          up         up
```

Router4 配置:

```
1 R4#configure terminal
2 R4(config)#interface serial 2/0
3 R4(config-if)#ip add 23.1.1.3 255.255.255.248
4 R4(config-if)#no shut
5 R4(config-if)#exit
6 R4(config)#interface gigabitethernet 0/0
7 R4(config-if)#ip add 68.1.1.6 255.255.255.240
8 R4(config-if)#no shut
9 R4(config-if)#exit
10 R4(config)#interface loopback 1
11 R4(config-if)#ip add 4.4.4.4 255.255.255.0
12 R4(config-if)#end
```

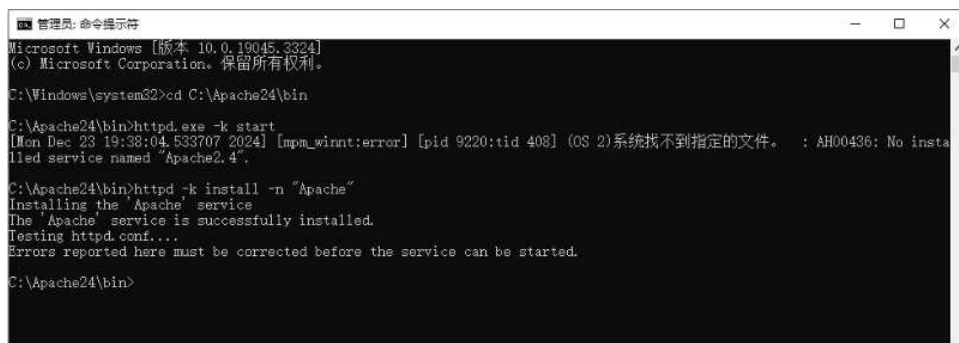


R4 配置后显示如下:

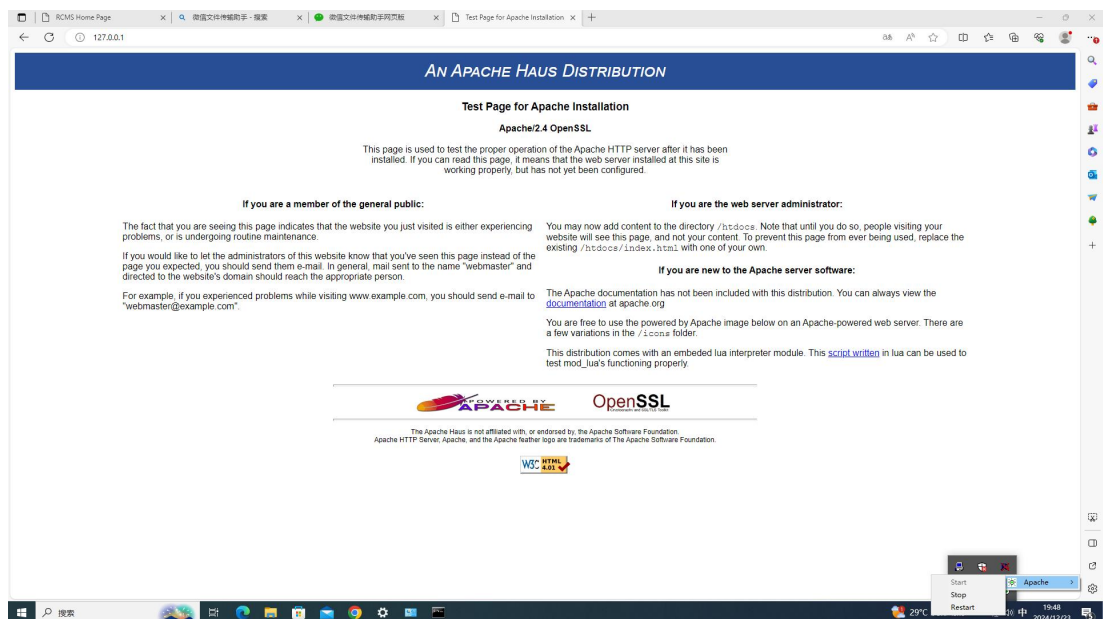
```
26-RSR20-2(config-if-GigabitEthernet 0/0)#show ip int br
Interface                               IP-Address(Pri)      IP-Address(Sec)      Statu
s
Serial 2/0                             23.1.1.3/29          no address            up
                                     up
Serial 2/1                             no address            no address            down
                                     down
GigabitEthernet 0/0                    68.1.1.6/28          no address            up
                                     up
GigabitEthernet 0/1                    no address            no address            down
                                     down
GigabitEthernet 0/2                    no address            no address            down
                                     down
GigabitEthernet 0/3                    no address            no address            down
                                     down
Loopback 1                             4.4.4.4/24          no address            up
                                     up
```

2) 在 Web 服务器上安装并运行 Web Server

解压 Apache24 文件夹后将其放置在 C 盘根目录。以管理员身份运行 cmd, 进入 C:/Apache24/bin, 运行命令 `httpd -k install -n "Apache"`。运行文件夹中的 A...Monitor.exe 打开桌面左下角的图形界面, 点击 start。



验证是否安装 web 服务器成功: 浏览器地址栏: <http://127.0.0.1>, 有页面代表成功。



3) 为路由器 R1-R4 配置动态路由协议 OSPF, 自动学习不同网段的路由信息, 实现多区域间网络的互联互通

我们使用以下配置命令为 R1-R4 配置 OSPF 协议 (可以在上述路由器端口地址看到我们的 loopback 已经在配置地址时一起配置了):



Router1:

```
1 R1(config)# router ospf 1
2 R1(config-ospf)#net 192.168.1.0 0.0.0.255 area 1
3 R1(config-ospf)#net 14.1.1.0 0.0.0.15 area 1
4 R1(config-ospf)#net 1,1.1.0 0.0.0.255 area 1
```

Router2:

```
1 R2(config)#router ospf 2
2 R2(config-ospf)#net 14.1.1.0 0.0.0.15 area 1
3 R2(config-ospf)#net 12.1.1.0 0.0.0.3 area 0
4 R2(config-ospf)#net 2.2.2.0 0.0.0.255 area 0
```

Router3:

```
1 R2(config)#router ospf 2
2 R2(config-ospf)#net 14.1.1.0 0.0.0.15 area 1
3 R2(config-ospf)#net 12.1.1.0 0.0.0.3 area 0
4 R2(config-ospf)#net 2.2.2.0 0.0.0.255 area 0
```

Router4:

```
1 R3(config)#router ospf 3
2 R3(config-ospf)#net 12.1.1.0 0.0.0.3 area 0
3 R3(config-ospf)#net 23.1.1.0 0.0.0.7 area 2
4 R3(config-ospf)#net 3.3.3.0 0.0.0.255 area 2
```

- 4) 在路由器 R1-R4 上分别执行 show ip route 命令，记录各路由器的路由表信息
每个路由器 R1-R4 可通过以下指令，查看路由器的路由表信息

```
1 // 查看路由表
2 R# show ip route
3 // 查看路由器的邻居信息
4 R# show ip ospf nei
```

Router1:

- 我们可以看到，配置 OSPF 后，R1 的路由表中出现了 O 条目，这些条目是从邻居 2.2.2.2 中学习到的路由条目；
- 根据拓扑中我们可以看出，通过 OSPF 的学习，R1 学习到了其他路由的条目，包括 Web 服务器的路由条目。

25-RSR20-1(config)#show ip route

```
Codes: C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default
```

```
Gateway of last resort is no set
C    1.1.1.0/24 is directly connected, Loopback 1
C    1.1.1.1/32 is local host.
O IA 2.2.2.2/32 [110/50] via 14.1.1.1, 00:58:58, Serial 2/0
O IA 3.3.3.3/32 [110/51] via 14.1.1.1, 00:46:23, Serial 2/0
O IA 4.4.4.4/32 [110/101] via 14.1.1.1, 00:46:46, Serial 2/0
O IA 12.1.1.0/30 [110/51] via 14.1.1.1, 00:50:55, Serial 2/0
C    14.1.1.0/29 is directly connected, Serial 2/0
C    14.1.1.4/32 is local host.
O IA 23.1.1.0/29 [110/101] via 14.1.1.1, 00:47:05, Serial 2/0
O IA 68.1.1.0/28 [110/102] via 14.1.1.1, 00:46:46, Serial 2/0
C    192.168.1.0/24 is directly connected, GigabitEthernet 0/1
C    192.168.1.1/32 is local host.
C    192.168.10.0/24 is directly connected, GigabitEthernet 0/1.10
```



```
25-RSR20-1(config)#show ip ospf nei
```

```
OSPF process 1, 1 Neighbors, 1 is Full:
Neighbor ID    Pri   State           BFD State  Dead Time   Address
Interface
2.2.2.2        1    Full/ -         -          00:00:33    14.1.1.1
Serial 2/0
```

Router2:

- 我们可以看到，配置完 OSPF 之后，R2 的邻居显示有两个，一个是 R1 (1.1.1.1)；一个是 R3 (3.3.3.3)，这符合我们的拓扑图设置；
- 同时我们也可以看到，R2 路由表出现了 6 条 O 条目，这是因为拓扑图中存在除 R2 外的其他 3 台路由器，这两台路由器每台有两个接口需要 R2 学习，于是通过 OSPF 由 R2 的邻居传递给它。

```
25-RSR20-2#show ip ospf nei
```

```
OSPF process 2, 2 Neighbors, 2 is Full:
Neighbor ID    Pri   State           BFD State  Dead Time   Address      Interface
3.3.3.3        1    Full/BDR        -          00:00:39    12.1.1.2    GigabitEthernet 0/0
1.1.1.1        1    Full/ -         -          00:00:40    14.1.1.4    Serial 2/0
```

```
25-RSR20-2#show ip route
```

```
Codes: C - connected, S - static, R - RIP, B - BGP
O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default
```

```
Gateway of last resort is no set
O 1.1.1.1/32 [110/50] via 14.1.1.4, 00:11:30, Serial 2/0
C 2.2.2.0/24 is directly connected, Loopback 1
C 2.2.2.2/32 is local host.
O IA 3.3.3.3/32 [110/1] via 12.1.1.2, 00:00:21, GigabitEthernet 0/0
O IA 4.4.4.4/32 [110/51] via 12.1.1.2, 00:00:43, GigabitEthernet 0/0
C 12.1.1.0/30 is directly connected, GigabitEthernet 0/0
C 12.1.1.1/32 is local host.
C 14.1.1.0/29 is directly connected, Serial 2/0
C 14.1.1.1/32 is local host.
O IA 23.1.1.0/29 [110/51] via 12.1.1.2, 00:01:02, GigabitEthernet 0/0
O IA 68.1.1.0/28 [110/52] via 12.1.1.2, 00:00:43, GigabitEthernet 0/0
O 192.168.1.0/24 [110/51] via 14.1.1.4, 00:12:56, Serial 2/0
```

Router3:

- 我们可以看到，R3 显示的邻居有两个，分别为 R2 (2.2.2.2) 和 R4 (4.4.4.4)，符合我们拓扑图；
- 同时可以看到 R3 的路由表，从其邻居中学习到了额外的 6 条路由信息，通过 OSPF 协议为路由表中加入 6 条 O 条目。

```
26-RSR20-1(config)#show ip route
```

```
Codes: C - connected, S - static, R - RIP, B - BGP
O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default
```

```
Gateway of last resort is no set
O IA 1.1.1.1/32 [110/51] via 12.1.1.1, 00:01:05, GigabitEthernet 0/0
O 2.2.2.2/32 [110/1] via 12.1.1.1, 00:01:05, GigabitEthernet 0/0
C 3.3.3.0/24 is directly connected, Loopback 1
C 3.3.3.3/32 is local host.
O 4.4.4.4/32 [110/50] via 23.1.1.3, 00:00:32, Serial 2/0
C 12.1.1.0/30 is directly connected, GigabitEthernet 0/0
C 12.1.1.2/32 is local host.
O IA 14.1.1.0/29 [110/51] via 12.1.1.1, 00:01:05, GigabitEthernet 0/0
C 23.1.1.0/29 is directly connected, Serial 2/0
C 23.1.1.2/32 is local host.
O 68.1.1.0/28 [110/51] via 23.1.1.3, 00:00:32, Serial 2/0
O IA 192.168.1.0/24 [110/52] via 12.1.1.1, 00:01:05, GigabitEthernet 0/0
```

```
26-RSR20-1(config)#show ip ospf nei
```

```
OSPF process 3, 2 Neighbors, 2 is Full:
Neighbor ID    Pri   State           BFD State  Dead Time   Address      Interface
2.2.2.2        1    Full/DR        -          00:00:32    12.1.1.1    GigabitEthernet 0/0
4.4.4.4        1    Full/ -         -          00:00:37    23.1.1.3    Serial 2/0
```



Router4:

- 可以看到，R4 的邻居只有 R3 (3.3.3.3)，通过 S2/0 连接，符合我们拓扑图所示；
- 同时在路由表中，R4 此时通过其邻居 R3 学习到了 6 条 O 条目。

```
26-RSR20-2(config-if-GigabitEthernet 0/0)#show ip ospf nei

OSPF process 4, 1 Neighbors, 1 is Full:
Neighbor ID    Pri  State           BFD State  Dead Time   Address      Interface
3.3.3.3        1    Full/ -         -          00:00:38    23.1.1.2     Serial 2/0

26-RSR20-2(config-if-GigabitEthernet 0/0)#show ip route

Codes: C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
O IA 1.1.1.1/32 [110/101] via 23.1.1.2, 00:10:56, Serial 2/0
O IA 2.2.2.2/32 [110/51] via 23.1.1.2, 00:10:56, Serial 2/0
O   3.3.3.3/32 [110/50] via 23.1.1.2, 00:10:26, Serial 2/0
C   4.4.4.0/24 is directly connected, Loopback 1
C   4.4.4.4/32 is local host.
O IA 12.1.1.0/30 [110/51] via 23.1.1.2, 00:10:56, Serial 2/0
O IA 14.1.1.0/29 [110/101] via 23.1.1.2, 00:10:56, Serial 2/0
C   23.1.1.0/29 is directly connected, Serial 2/0
C   23.1.1.3/32 is local host.
C   68.1.1.0/28 is directly connected, GigabitEthernet 0/0
C   68.1.1.6/32 is local host.
O IA 192.168.1.0/24 [110/102] via 23.1.1.2, 00:10:56, Serial 2/0
```

总结分析:

经过以上的 OSPF 学习，我们可以看到 R1-R4 的路由表条目中都存在各个路由器接口的地址，局域网中的 PC 将数据包发向默认网关 R1，R1 通过路由表转发出去，其余路由器也通过路由表转发数据包，最终数据包可以到达 Web 服务器。

- 5) 在 CS-PC1 上执行 traceroute Web 服务器，分析执行结果。
在 CS-PC1 的终端执行指令 traceroute

```
1 | tracert 68.1.1.8
```

其结果如下:

```
C:\Users\D502>tracert 68.1.1.8

通过最多 30 个跃点跟踪
到 D52_78 [68.1.1.8] 的路由:

 1  <1 毫秒  <1 毫秒  <1 毫秒  192.168.1.1
 2  43 ms    40 ms    40 ms    14.1.1.1
 3  42 ms    40 ms    40 ms    12.1.1.2
 4  82 ms    80 ms    80 ms    23.1.1.3
 5  92 ms    92 ms    92 ms    D52_78 [68.1.1.8]

跟踪完成。
```

分析:

- 我们可以看到，从 CS-PC1 只需 5 跳就可到达 Web 服务器；
- **第一跳:** 数据包从主机首先转发到默认网关 R1 (192.168.1.1) 上，随后 R1 会查询其路由表，发现目的 IP 68.1.1.8 符合 68.1.1.0/28 的 OSPF 路由条目，将从端口 S2/0 转发出去；
- **第二跳:** 数据包从 R1 的 S2/0 端口到达 R2 (14.1.1.1)；随后 R2 查询路由表，发现其路由表中存在 OSPF 条目与 68.1.1.8 符合，根据路由表将数据包从端口 Giga0/0 转发出去



- **第三跳：**数据包从 R2 的 Giga0/0 端口到达 R3 (12.1.1.2)；随后 R3 查询路由表，发现其路由表中存在 OSPF 条目与 68.1.1.8 符合，根据路由表将数据包从端口 S2/0 转发出去
- **第四跳：**数据包从 R3 的 S2/0 端口到达 R4 (23.1.1.3)；随后 R4 查询路由表，发现其路由表中存在 OSPF 条目与 68.1.1.8 符合，根据路由表将数据包从端口 Giga0/0 转发出去
- **第五跳：**数据包从 R4 的 Giga0/0 端口到达该端口所连接的 Web 服务器 (68.1.1.8)，也就是我们的目的 IP 地址，此时数据包到达目的地，结束跟踪。
- **综上所述，根据 OSPF 所学习到的条目，我们的路由器可以将我们拓扑图连通，使得用户内网可与外网连通，同时跟踪符合我们预期，也符合拓扑图中对应的 5 跳。**

2. 用户内网的 VLAN 设置

为便于用户内网管理、方便移动办公，在计算机科学系交换机 S1 和电器工程系交换机 S2 上设置基于端口的 VLAN，使得 CS-PC1 和 CS-PC2 属于同一个 VLAN，EE-PC1 属于另一个 VLAN。验证所设置的 VLAN 是否达到该目标。

交换机 S1 配置：

```
1 Switch1#configure terminal
2 Switch1(config)#vlan 10
3 Switch1(config-vlan)#exit
4 Switch1(config)#interface gigabitethernet 0/2
5 Switch1(config-if)#switchport mode trunk
6 Switch1(config)#interface gigabitethernet 0/1
7 Switch1(config-if)#switchport access vlan 10
8 Switch1(config)#exit
```

分析：

- 在 S1 配置中，首先创建 VLAN 10。
- 将端口 Gi0/2 设置为 trunk 模式，此模式允许该端口传输多个 VLAN 的数据，常用于连接其他交换机或路由器，确保不同 VLAN 间的通信可通过此端口进行中继。
- 端口 Gi0/1 被配置为 access vlan 10，这意味着该端口仅允许 VLAN 10 的数据通过，将连接在此端口的设备（如 CS - PC1 或 CS - PC2）划分到 VLAN 10 中，实现了基于端口的 VLAN 划分，使同一 VLAN 内设备可在二层网络中直接通信，而不同 VLAN 间的通信则需借助三层设备（如路由器）进行转发。

交换机 S2 配置：

```
Switch2#configure terminal
Switch2(config)#vlan 10
Switch2(config-vlan)#exit
Switch2(config)#vlan 20
Switch2(config-vlan)#exit
Switch2(config)#interface gigabitethernet 0/1
Switch2(config-if)#switchport access vlan 10
Switch2(config)#interface gigabitethernet 0/2
Switch2(config-if)#switchport access vlan 20
Switch2(config)#interface gigabitethernet 0/3
Switch2(config-if)#switchport mode trunk
Switch2(config)#exit
```




分析:

- S2 配置类似但更为复杂, 创建了 VLAN 10 和 VLAN 20。
- 端口 Gi0/1 接入 VLAN 10, 可连接计算机科学系相关设备, Gi0/2 接入 VLAN 20 用于电器工程系的 EE-PC1。
- 端口 Gi0/3 设置为 trunk 模式, 用于连接核心交换机 S0 或其他设备, 实现不同 VLAN 数据的传输与交互, 确保网络的扩展性和灵活性, 满足多部门不同 VLAN 设备的连接需求。

交换机 S0 配置:

```
Switch0#configure terminal
Switch0(config)#vlan 10
Switch0(config-vlan)#exit
Switch0(config)#vlan 20
Switch0(config-vlan)#exit
Switch0(config)#interface gigabitethernet 0/1
Switch0(config-if)#switchport mode trunk
Switch0(config)#interface gigabitethernet 0/2
Switch0(config-if)#switchport mode trunk
Switch0(config)#interface gigabitethernet 0/3
Switch0(config-if)#switchport mode trunk
Switch0(config)#exit
```

分析:

- 在交换机 S0 的配置中, 首先进入全局配置模式。
- 接着创建了 VLAN 10 和 VLAN 20, 用于划分不同的虚拟局域网。
- 然后, 将 GigabitEthernet 0/1、0/2 和 0/3 接口都配置为 trunk 模式。
- 这种配置方式使得这些接口能够同时传输多个 VLAN 的数据, 确保不同 VLAN 间的数据可以在交换机 S0 上进行交换。

验证 VLAN 设置:

1) CS-PC1 连接测试:

- Ping CS-PC2

```
C:\Users\D502>ping 192.168.1.20

正在 Ping 192.168.1.20 具有 32 字节的数据:
来自 192.168.1.20 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.1.20 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.1.20 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.1.20 的回复: 字节=32 时间<1ms TTL=128

192.168.1.20 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 0ms, 最长 = 0ms, 平均 = 0ms
```

- Ping EE-PC1

```
C:\Users\D502>ping 192.168.1.30

正在 Ping 192.168.1.30 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。

192.168.1.30 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),
```



2) CS-PC2 连接测试:

- Ping CS-PC1

```
正在 Ping 192.168.1.10 具有 32 字节的数据:  
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=128  
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=128  
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=128  
来自 192.168.1.10 的回复: 字节=32 时间<1ms TTL=128  
  
192.168.1.10 的 Ping 统计信息:  
数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),  
往返行程的估计时间(以毫秒为单位):  
最短 = 0ms, 最长 = 0ms, 平均 = 0ms
```

- Ping EE-PC1

```
C:\Users\D502>ping 192.168.1.30  
  
正在 Ping 192.168.1.30 具有 32 字节的数据:  
请求超时。  
请求超时。  
请求超时。  
请求超时。  
  
192.168.1.30 的 Ping 统计信息:  
数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),
```

3) EE-PC1 连接测试:

- Ping CS-PC1

```
C:\Users\D502>ping 192.168.1.10  
  
正在 Ping 192.168.1.10 具有 32 字节的数据:  
请求超时。  
来自 192.168.1.30 的回复: 无法访问目标主机。  
来自 192.168.1.30 的回复: 无法访问目标主机。  
来自 192.168.1.30 的回复: 无法访问目标主机。  
  
192.168.1.10 的 Ping 统计信息:  
数据包: 已发送 = 4, 已接收 = 3, 丢失 = 1 (25% 丢失),
```

- Ping CS-PC2

```
C:\Users\D502>ping 192.168.1.20  
  
正在 Ping 192.168.1.20 具有 32 字节的数据:  
来自 192.168.1.30 的回复: 无法访问目标主机。  
来自 192.168.1.30 的回复: 无法访问目标主机。  
来自 192.168.1.30 的回复: 无法访问目标主机。  
来自 192.168.1.30 的回复: 无法访问目标主机。  
  
192.168.1.20 的 Ping 统计信息:  
数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
```

总结:

- 连接测试结果表明, CS-PC1 和 CS-PC2 能相互 ping 通, 但 CS-PC1 或 CS-PC2 与 EE-PC1 都不能相互 ping 通, 和预期结果相符, 这说明我们的 vlan 配置正确。
- 实验中, CS-PC1 和 CS-PC2 属于 vlan10, 它们同属于 vlan10,能相互 ping 通。
- EE-PC1 属于 vlan20, 与 CS-PC1 或 CS-PC2 所处的 vlan 不同, EE-PC1 不能 ping 通 CS-PC1 或 CS-PC2。



额外步骤:

实验过程发现, 按照上述步骤配置完 VLAN 之后, PC 机无法 PING 通路由器 R1 了, 这会造成我们无法与外网通信, 影响后续实验进程, 我们还需要对路由器 R1 进行配置子接口的操作:

路由器 R1 配置:

```
1 R1(config)#interface gigabitethernet 0/1.10
2 R1(config-subif)#description valn 10
3 R1(config-subif)#encapsulation dot1q 10
4 R1(config-subif)#ip address 192.168.10.1 255.255.255.0
5 R1(config-subif)#exit
6
7 R1(config)#interface gigabitethernet 0/1.20
8 R1(config-subif)#description valn 20
9 R1(config-subif)#encapsulation dot1q 20
10 R1(config-subif)#ip address 192.168.20.1 255.255.255.0
11 R1(config-subif)#exit
12
13 #通告子接口网络
14 R1(config)#router ospf 1
15 R1(config - router)#network 192.168.10.1 0.0.0.255 area1
16 R1(config - router)#network 192.168.20.1 0.0.0.255 area1
```

分析:

- 路由器 R1 通过配置子接口实现对不同 VLAN 流量的处理。创建子接口 Gi0/1.10 和 Gi0/1.20, 分别对应 VLAN 10 和 VLAN 20。
- encapsulation dot1q 命令用于指定子接口的 VLAN 封装协议, 使路由器能够识别并剥离相应 VLAN 的数据帧。
- 为子接口配置 IP 地址后, 路由器子接口的地址可作为不同 VLAN 的网关。

配完子接口后的路由器接口的 IP 和状态表:

Interface	IP-Address(Pri)	IP-Address(Sec)	Status	Protocol
Serial 2/0	14.1.1.4/29	no address	up	up
Serial 2/1	no address	no address	down	down
GigabitEthernet 0/0	no address	no address	down	down
GigabitEthernet 0/1.20	192.168.20.1/24	no address	up	up
GigabitEthernet 0/1.10	192.168.10.1/24	no address	up	up
GigabitEthernet 0/1	192.168.1.1/24	no address	up	up
GigabitEthernet 0/2	no address	no address	down	down
GigabitEthernet 0/3	no address	no address	down	down
Loopback 1	1.1.1.1/24	no address	up	up
VLAN 1	no address	no address	up	down

在通告了子接口网络之后, 其余路由器新学习到了有子接口 IP 的 O 条目, 因此 VLAN 数据包可以正常通过子接口转发出去:

```
26-RSR20-2(config-if-GigabitEthernet 0/0)#show ip route

Codes: C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
O IA 1.1.1.1/32 [110/101] via 23.1.1.2, 00:10:56, Serial 2/0
O IA 2.2.2.2/32 [110/51] via 23.1.1.2, 00:10:56, Serial 2/0
O   3.3.3.3/32 [110/50] via 23.1.1.2, 00:10:26, Serial 2/0
C   4.4.4.0/24 is directly connected, Loopback 1
C   4.4.4.4/32 is local host.
O IA 12.1.1.0/30 [110/51] via 23.1.1.2, 00:10:56, Serial 2/0
O IA 14.1.1.0/29 [110/101] via 23.1.1.2, 00:10:56, Serial 2/0
C   23.1.1.0/29 is directly connected, Serial 2/0
C   23.1.1.3/32 is local host.
C   68.1.1.0/28 is directly connected, GigabitEthernet 0/0
C   68.1.1.6/32 is local host.
O IA 192.168.1.0/24 [110/102] via 23.1.1.2, 00:10:56, Serial 2/0
O IA 192.168.10.0/24 [110/102] via 23.1.1.2, 00:01:23, Serial 2/0
O IA 192.168.20.0/24 [110/102] via 23.1.1.2, 00:01:17, Serial 2/0
```



注意:

- 配置完路由器 R1 的子接口之后, 我们需要更新 PC 机的网关, 变成子接口的 IP。
- 同时, 我们也需要更新了 PC 机的 IP, 与路由器 R1 的子接口在同一网段下。

更新后的 CS-PC1 的 IP 和默认网关的 IP:

```
以太网适配器 以太网 3:

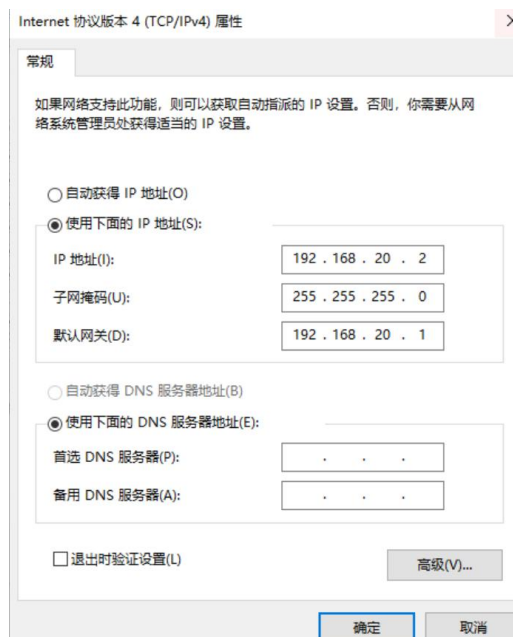
    连接特定的 DNS 后缀 . . . . . : 
    本地链接 IPv6 地址. . . . . : fe80::b199:efe6:7406:3e64%5
    IPv4 地址 . . . . . : 192.168.10.2
    子网掩码 . . . . . : 255.255.255.0
    默认网关. . . . . : 192.168.10.1
```

更新后的 CS-PC2 的 IP 和默认网关的 IP:

```
以太网适配器 以太网 3:

    连接特定的 DNS 后缀 . . . . . : 
    本地链接 IPv6 地址. . . . . : fe80::b59b:5af7:8ee6:4377%5
    IPv4 地址 . . . . . : 192.168.10.3
    子网掩码 . . . . . : 255.255.255.0
    默认网关. . . . . : 192.168.10.1
```

更新后的 EE-PC1 的 IP 和默认网关的 IP:



此时, PC 能正常 ping 通 R1 路由器了, 也能正常 ping 通服务器。

```
C:\Users\D502>ping 68.1.1.8

正在 Ping 68.1.1.8 具有 32 字节的数据:
来自 68.1.1.8 的回复: 字节=32 时间=77ms TTL=124
来自 68.1.1.8 的回复: 字节=32 时间=73ms TTL=124
来自 68.1.1.8 的回复: 字节=32 时间=73ms TTL=124
来自 68.1.1.8 的回复: 字节=32 时间=76ms TTL=124

68.1.1.8 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 73ms, 最长 = 77ms, 平均 = 74ms
```




3. 采用 NAT 技术处理用户内网与外网的连接

在路由器 R1 处配置 NAT, 将用户内网所使用的内网地址 (私有地址) 映射为外网的 IP 地址 14.1.1.4/29, 以便计算机科学系和电器工程系的师生机器可以访问外网。验证 NAT 转换。

注意此时主机 IP 已经更改为子接口网段下的地址。新的主机 IP:

CS-PC1:192.168.10.2

CS-PC2:192.168.10.3

EE-PC1:192.168.20.2

在路由器 R1 上配置:

```
1 // 创建访问控制列表, 允许内网地址访问外部网络
2 Router(config)# access-list 1 permit 192.168.1.0 0.0.0.255
3 Router(config)# access-list 1 permit 192.168.10.0 0.0.0.255
4 Router(config)# access-list 1 permit 192.168.20.0 0.0.0.255
5 // 标记为内部、外部接口
6 Router(config)# interface gigabitethernet 0/1
7 Router(config-if)# ip nat inside
8 Router(config)# interface gigabitethernet 0/1.10
9 Router(config-if)# ip nat inside
10 Router(config)# interface gigabitethernet 0/1.20
11 Router(config-if)# ip nat inside
12 Router(config)# interface serial 2/0
13 Router(config-if)# ip nat outside
14 // 配置 NAT 转换规则, 将内网 IP 地址分别转换为外部 IP 地址
15 Router(config)# ip nat inside source list 1 interface serial 2/0 overload
```

验证结果:

通过以下指令来查看 NAT 表以及内网与外网 IP 地址转换的过程

```
1 // 查看路由器 NAT 表
2 Router# show ip nat translations
3 // 查看路由器的地址转换过程
4 Router# debug ip nat
```

由于 debug ip nat 指令使用时信息呈现具有一定滞后性, 故我们主要分析使用指令 show ip nat translations 后的路由器 NAT 表。以下为 CS-PC1 与 CS-PC2 尝试访问外网时 R1 上的 NAT 表:

```
25-RSR20-1(config)#show ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
icmp14.1.1.4:1         192.168.10.3:1    68.1.1.8           68.1.1.8
25-RSR20-1(config)#
25-RSR20-1(config)#show ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
icmp14.1.1.4:1         192.168.10.2:1    68.1.1.8           68.1.1.8
25-RSR20-1(config)#
```

分析:

- 可以看到为 R1 配置 NAT 后其将来自内网不同主机的 IP 报文翻译为外网合法地址 14.1.1.4, 后经外部网络配置, 实现与 web 服务器 (IP 地址: 68.1.1.8) 间的通信。
- 配置 NAT 后, CS-PC1、CS-PC2 尝试 ping web 服务器 IP 68.1.1.8:



```
C:\Users\D502>ping 68.1.1.8

正在 Ping 68.1.1.8 具有 32 字节的数据:
来自 68.1.1.8 的回复: 字节=32 时间=76ms TTL=124
来自 68.1.1.8 的回复: 字节=32 时间=76ms TTL=124
来自 68.1.1.8 的回复: 字节=32 时间=73ms TTL=124
来自 68.1.1.8 的回复: 字节=32 时间=76ms TTL=124

68.1.1.8 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 73ms, 最长 = 76ms, 平均 = 75ms
```

```
(c) Microsoft Corporation。保留所有权利。

C:\Users\D502>ping 68.1.1.8

正在 Ping 68.1.1.8 具有 32 字节的数据:
来自 68.1.1.8 的回复: 字节=32 时间=73ms TTL=124
来自 68.1.1.8 的回复: 字节=32 时间=76ms TTL=124
来自 68.1.1.8 的回复: 字节=32 时间=74ms TTL=124
来自 68.1.1.8 的回复: 字节=32 时间=76ms TTL=124

68.1.1.8 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 73ms, 最长 = 76ms, 平均 = 74ms
```

可以看到配置完 NAT 后内网主机可以使用同一个转换地址与 web 服务器进行正常通信。

4. 用户内网的 ACL 设置

为便于网络管理，用户内网设置 ACL，允许计算机科学系的两台主机 CS-PC1 和 CS-PC2 在工作时间（9:00-17:00）内访问外网中的 Web 服务器，但禁止电器工程系的计算机 EE-PC1 在工作时间（9:00-17:00）访问外网中的 Web 服务器。在其他未明确指定的时间段内，用户内网所有机器均不得访问 Web 服务器。验证 ACL 的设置是否满足这一要求。

注意此时我们已经完成对于主机 IP 的更换，新的主机 IP：

CS-PC1:192.168.10.2

CS-PC2:192.168.10.3

EE-PC1:192.168.20.2

在路由器 R1 上配置 ACL：

```
1 // 设置上班时间
2 router# config
3 router(config)# time-range work-times
4 router(config-time-range)# periodic weekdays 09:00 to 17:00
5 router(config-time-range)# exit
6 // 设置ACL
7 router(config)#ip access-list extended accessctrl
8 router(config-ext-nacl)#permit tcp 192.168.10.0 0.0.0.255 host 68.1.1.8 eq www time-range work-
times
9 router(config-ext-nacl)#deny tcp 192.168.20.0 0.0.0.255 host 68.1.1.8 eq www time-range work-
times
10 router(config-ext-nacl)#show access-lists
11 router(config-ext-nacl)#exit
12
13 // 应用ACL
14 router(config)# int s2/0
15 router(config-if)#ip access-group accessctrl out
16 router(config-if)#end
17
18 // 设置系统时间
19 router#show clock
20 router#clock set 10:00:45 12 23 2024
```



计算机网络实验报告

根据上述命令定义工作时间，并对相应网段做访问限制以实现题目要求：

此时 ACL 表为：

```
ip access-list extended accessctrl
10 permit tcp 192.168.10.0 0.0.0.255 host 68.1.1.8 eq www time-range work-times
(inactive)
20 deny tcp 192.168.20.0 0.0.0.255 host 68.1.1.8 eq www time-range work-times (
inactive)
```

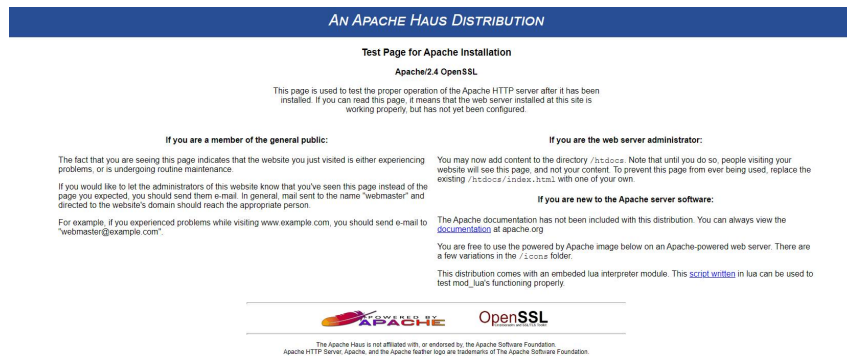
将其应用到相应端口上并激活，至此我们实现了外网服务器的时间访问限制。

验证 ACL 设置效果：（通过浏览器输入 Web 服务器的 IP 地址来验证）

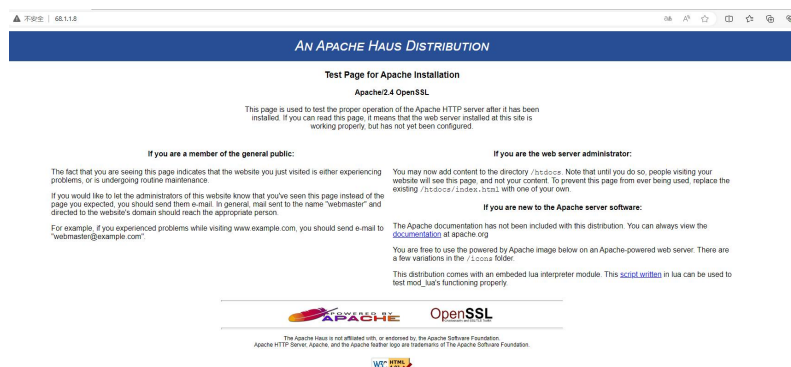
1) 上班时间 (clock set 为 10:00:45 Mon)

```
25-RSR20-1#clock set 10:00:45 12 23 2024
25-RSR20-1#*Dec 23 10:00:45: %SYS-6-CLOCKUPDATE: System clock has been updated to 10:00:45 UTC Mon Dec 23 2024.
```

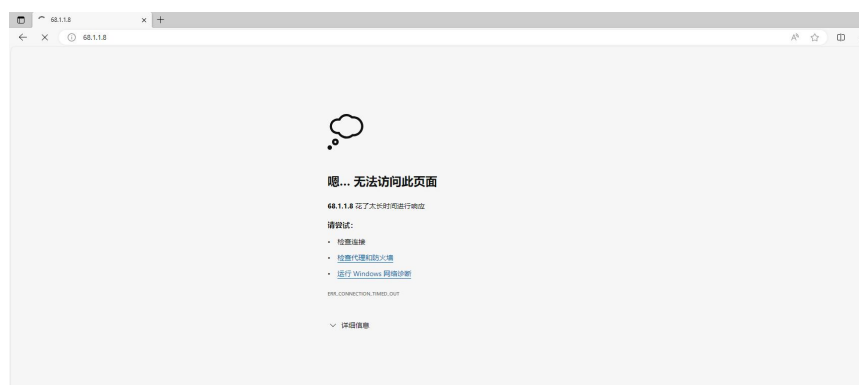
• CS-PC1



• CS-PC2



• EE-PC1





2) 下班时间 (clock set 为 07:30:55 Tue)

```
25-RSR20-1#show clock
```

```
07:30:43 UTC Tue, Oct 29, 2024
```

```
25-RSR20-1#*Oct 29 07:30:55: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2-Serial 2/0 from Full to Init, 1-WayReceived.
```

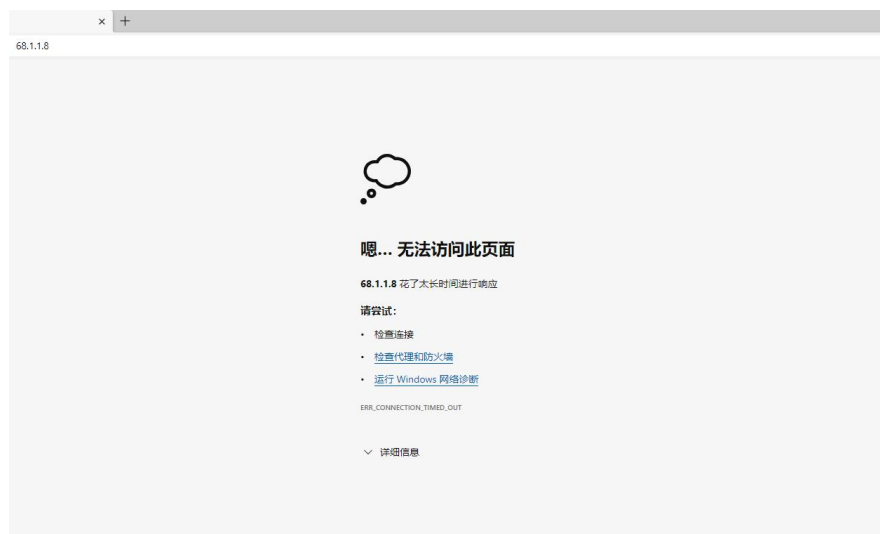
- CS-PC1



- CS-PC2



- EE-PC1



分析:

- 可以看到增加了基于时间的 ACL 表项后, CS-PC1 与 CS-PC2 可以在工作时间访问外网服务器, 其余时间不能访问;



- 而 EE-PC1 一直访问不了外网服务器，即在工作时间内来自 CS-PC1 与 CS-PC2 的流量被放行而来自 EE-PC1 的流量不被允许通过。
- 我们只定义了关于工作时间的 ACL 项，并没有定义下班时间不能访问 Web，但 ACL 仍成功实现了，这是因为 ACL 列表最后隐藏了一条默认拒绝所有流量的表项。
- 这与我们预期结果一致，ACL 配置成功。

【实验问题与解决方法】

1. VLAN 配置问题:

(1) 一开始没有给路由器设置子接口的时候，PC 机 ping 不到路由器

- ① **原因分析:** 在没有配置路由器子接口时，VLAN 间的通信无法实现。从图中可以看到，网络中有不同的 VLAN (VLAN10 和 VLAN20)。路由器在默认情况下，每个物理接口只能属于一个网络，没有配置子接口时，路由器无法识别和转发不同 VLAN 之间的数据包。例如，VLAN10 中的 CS - PC1 和 VLAN20 中的 EE - PC1 在不同的 VLAN 中。没有路由器子接口的配置，路由器不知道如何在这些 VLAN 之间转发数据。
- ② **工作原理:** 当 PC 机 (如 CS - PC1) 尝试 ping 路由器时，它发送的 ICMP 数据包首先到达所在 VLAN (VLAN10) 的交换机 (S1)。交换机根据 VLAN 配置将数据包转发到连接路由器的端口。但是，如果路由器没有配置子接口来处理 VLAN10 的流量，路由器将不知道如何处理这个数据包，可能会直接丢弃它。路由器子接口的作用是允许一个物理接口处理多个 VLAN 的流量。通过配置子接口并分配相应的 VLAN ID，路由器可以识别和转发不同 VLAN 之间的数据包。
- ③ **解决方案:** 为了实现 VLAN 间的通信，需要在路由器 (R1) 上配置子接口。例如，在路由器 R1 的 Gi0/1 接口上配置子接口 Gi0/1.10 (对应 VLAN10) 和 Gi0/1.20 (对应 VLAN20)，并为每个子接口配置相应的 IP 地址和 VLAN 封装。配置完成后，路由器就能够识别和转发 VLAN10 和 VLAN20 之间的数据包，PC 机就可以 ping 通路由器了。

(2) 配置完子接口后，PC 机仍然 ping 不到路由器

- ① **原因分析:** 尽管路由器已成功配置子接口，但 PC 机的默认网关设置并未相应更新，依旧为原来物理接口的 IP 地址。在网络通信过程中，PC 机发送数据包时会依据默认网关进行路由选择，由于默认网关未指向子接口，导致数据包并非通过子接口进行转发，而是尝试从原物理接口传输，从而使子接口的配置未能生效，无法实现预期的 VLAN 间通信。
- ② **解决方案:** 将 PC 机的默认网关地址修改为与其相连的子接口的 IP 地址。具体而言，对于位于 VLAN10 的 CS - PC1 和 CS - PC2，将其默认网关修改为 192.168.10.1；对于处于 VLAN20 的 EE - PC1，将默认网关设置为 192.168.20.1。同时将 PC 的 IP 地址更改为默认网关下的网段地址；完成此项修改后，PC 机能够正确地将数据包发送至路由器的子接口，进而成功实现与 R1 路由器的正常通信，可顺利进行 ping 操作，确保了网络的连通性与数据传输的正确性。

2. ACL 配置问题: (设置 tcp 访问控制，但通过 ping 来验证: UDP、TCP 问题，改为不限制访问方式)

(1) 配置好了 ACL 列表却不生效。



- ① **原因分析:** 我们查看 access-list, 发现存在两个 access 列表 (如下), 其中 standard ACL 列表是我们在 NAT 实验时配置的, 它与 extended ACL 列表应用在同一个端口 S2/0 上, 而由于 standard 配置时间较早, 因此根据 ACL 的生效规则, 先应用在该端口的先起作用, 因此在 standard ACL 时会让几台主机的流量全部通过, 导致所有主机都能访问外网。

```
25-RSR20-1(config-ext-nacl)#show access-lists

ip access-list standard 1
10 permit 192.168.1.0 0.0.0.255
20 permit 192.168.10.0 0.0.0.255
30 permit 192.168.20.0 0.0.0.255

ip access-list extended accessctrl
10 permit tcp 192.168.10.0 0.0.0.255 host 68.1.1.8 eq www time-range work-times
(inactive)
20 deny tcp 192.168.20.0 0.0.0.255 host 68.1.1.8 eq www time-range work-times (
inactive)
```

- ② **解决方案:** 我们将 standard ACL 删除, 只保留后面配置的 extended ACL, 即让 S2/0 端口上只应用后面配置的 extended ACL, 此时端口便会应用 extended ACL 来就行流量筛选, 成功过滤了不符合条件的流量, 实现了上班时间 CS 的 PC 能访问 Web, 其余时间和主机都不能访问 Web。

本次实验完成后, 请根据组员在实验中的贡献, 请实事求是, 自评在实验中应得的分数。(按百分制)

学号	学生	自评分
<u>22336158</u>	刘晓丹	<u>100</u>
<u>22336107</u>	赖玟慈	<u>100</u>
<u>22336124</u>	李馨媛	<u>100</u>
<u>22336155</u>	刘苇嘉	<u>100</u>
<u>21312450</u>	林隽哲	<u>100</u>
<u>22365043</u>	江颢怡	<u>100</u>
<u>22302056</u>	刘彦凤	<u>100</u>