

# 中国矿业大学 计算机科学与技术学院

## 2019 级本科生课程设计报告

课程名称： 网络系统与安全实践

班 级： 信息安全 19-1 班

姓 名： 朱公澳、周炯超、江一川、邹凯蓄

报告时间： 2022.6.28

任课教师： 王虎

## 分 工

姓名	完成工作情况
朱公澳	拓扑设计、搭建拓扑、配置路由器、测试验证、撰写实验报告
周炯超	拓扑设计、物理连线、配置二层交换机、测试验证、撰写实验报告
江一川	拓扑设计、拓扑绘制、配置三层交换机、测试验证、撰写实验报告
邹凯蓄	拓扑设计、记录情况、配置 PC 机、测试验证、撰写实验报告

**2021-2022 学年第二学期**  
**《网络系统与安全实践》课程评分表**  
(小组成员每人单独一页)

姓名 朱公澳 学号 08193035 班级 信息安全 19-1 班

编号	课程教学目标	考查方式及考查点	占比	得分
1	(3.4) 目标 1: 根据指定场景, 独立设计网络互联与网络安全解决方案。	方案答辩; 解决方案的合理性、可行性和完备性, 文档规范性;	30%	
2	(4.2) 目标 2: 掌握网络互联设备网络安全设备的配置和使用, 能综合运用网络与安全的基本原理和技术, 解决网络互联与网络安全的具体实践问题, 具备设计与实施网络互联与网络安全工程的能力。	现场检查实验结果和配置文件, 问题提问;	40%	
3	(9.1) 目标 3: 通过分组实验, 锻炼合作团队意识, 锻炼组长的统筹协调能力。	分工情况是否合理, 各组员所完成任务情况;	10%	
4	(10.3) 目标 4: 能够就课程方案设计的过程撰写较为规范的课程报告, 在课程报告审查和方案答辩过程中清晰表达设计思路、出现的问题及解决方案。	课程报告; 课程报告的规范性和正确性;	20%	
	总分			

评阅人: \_\_\_\_\_

# 2021-2022 学年第二学期

## 《网络系统与安全实践》课程评分表

（小组成员每人单独一页）

姓名 周炯超 学号 08193028 班级 信息安全 19-1 班

编号	课程教学目标	考查方式及考查点	占比	得分
1	(3.4) 目标 1: 根据指定场景, 独立设计网络互联与网络安全解决方案。	方案答辩; 解决方案的合理性、可行性和完备性, 文档规范性;	30%	
2	(4.2) 目标 2: 掌握网络互联设备网络安全设备的配置和使用, 能综合运用网络与安全的基本原理和技术, 解决网络互联与网络安全的具体实践问题, 具备设计与实施网络互联与网络安全工程的能力。	现场检查实验结果和配置文件, 问题提问;	40%	
3	(9.1) 目标 3: 通过分组实验, 锻炼合作团队意识, 锻炼组长的统筹协调能力。	分工情况是否合理, 各组员所完成任务情况;	10%	
4	(10.3) 目标 4: 能够就课程方案设计的过程撰写较为规范的课程报告, 在课程报告审查和方案答辩过程中清晰表达设计思路、出现的问题及解决方案。	课程报告; 课程报告的规范性和正确性;	20%	
	总分			

评阅人: \_\_\_\_\_

**2021-2022 学年第二学期**  
**《网络系统与安全实践》课程评分表**  
(小组成员每人单独一页)

姓名 江一川 学号 08193041 班级 信息安全 19-1 班

编号	课程教学目标	考查方式及考查点	占比	得分
1	(3.4) 目标 1: 根据指定场景, 独立设计网络互联与网络安全解决方案。	方案答辩; 解决方案的合理性、可行性和完备性, 文档规范性;	30%	
2	(4.2) 目标 2: 掌握网络互联设备网络安全设备的配置和使用, 能综合运用网络与安全的基本原理和技术, 解决网络互联与网络安全的具体实践问题, 具备设计与实施网络互联与网络安全工程的能力。	现场检查实验结果和配置文件, 问题提问;	40%	
3	(9.1) 目标 3: 通过分组实验, 锻炼合作团队意识, 锻炼组长的统筹协调能力。	分工情况是否合理, 各组员所完成任务情况;	10%	
4	(10.3) 目标 4: 能够就课程方案设计的过程撰写较为规范的课程报告, 在课程报告审查和方案答辩过程中清晰表达设计思路、出现的问题及解决方案。	课程报告; 课程报告的规范性和正确性;	20%	
	总分			

评阅人: \_\_\_\_\_

**2021-2022 学年第二学期**  
**《网络系统与安全实践》课程评分表**  
(小组成员每人单独一页)

姓名 邹凯蓄 学号 08192945 班级 信息安全 19-1 班

编号	课程教学目标	考查方式及考查点	占比	得分
1	(3.4) 目标 1: 根据指定场景, 独立设计网络互联与网络安全解决方案。	方案答辩; 解决方案的合理性、可行性和完备性, 文档规范性;	30%	
2	(4.2) 目标 2: 掌握网络互联设备网络安全设备的配置和使用, 能综合运用网络与安全的基本原理和技术, 解决网络互联与网络安全的具体实践问题, 具备设计与实施网络互联与网络安全工程的能力。	现场检查实验结果和配置文件, 问题提问;	40%	
3	(9.1) 目标 3: 通过分组实验, 锻炼合作团队意识, 锻炼组长的统筹协调能力。	分工情况是否合理, 各组员所完成任务情况;	10%	
4	(10.3) 目标 4: 能够就课程方案设计的过程撰写较为规范的课程报告, 在课程报告审查和方案答辩过程中清晰表达设计思路、出现的问题及解决方案。	课程报告; 课程报告的规范性和正确性;	20%	
	总分			

评阅人: \_\_\_\_\_

# 目录

1 实验设备 .....	9
1.1 路由器 .....	9
1.2 二层交换机 .....	9
1.3 三层交换机 .....	9
1.4 PC 机 .....	9
2 实验背景 .....	10
3 网络连通性要求 .....	10
3.1 Stie1 .....	10
3.2 Stie2 .....	10
3.3 Tunnel .....	10
3.4 NATP .....	10
3.5 ACL .....	11
4 网络安全性要求 .....	11
4.1 VRRP .....	11
4.2 IPSec .....	11
4.3 MAC 地址绑定 .....	12
5 网络拓扑 .....	12
5.1 拓扑构建 .....	12
5.2 拓扑仿真 .....	12
6 网络配置 .....	13
6.1 连通性配置 .....	13
6.1.1 R1 .....	13
6.1.2 R2 .....	14
6.1.3 SW1 .....	15
6.1.4 SW2 .....	16
6.1.5 SW3 .....	17
6.1.6 SW11 .....	18
6.1.7 SW12 .....	18
6.2 安全性配置 .....	19
6.2.1 VRRP .....	19
6.2.2 IPSec .....	20
6.2.3 MAC 地址绑定 .....	21
6.2.4 开启用户远程登陆 .....	21

7 结果验证.....	22
7.1 验证 PC1 与各关键节点互通.....	22
7.2 验证 PC2 与各关键节点互通.....	24
7.3 验证 PC3 与各关键节点互通.....	25
7.4 验证 PC4 与各关键节点互通.....	27
7.5 IPsec Tunnel 验证.....	29



# 1 实验设备

## 1.1 路由器

RSR20 路由器：两台

功能特点：

路由器用于连接两个或多个网络的硬件设备，在网络间起网关的作用，是读取每一个数据包中的地址然后决定如何传送的专用智能性的网络设备。它能够理解不同的协议，分析各种不同类型网络传来的数据包的目的地址，把非 TCP/IP 网络的地址转换成 TCP/IP 地址，再根据选定的路由算法把各数据包按最佳路线传送到指定位置。

## 1.2 二层交换机

S2628G-I 二层交换机：2 台。

功能特点：

二层交换机工作于 OSI 模型的第 2 层（数据链路层），可以识别数据帧中的 MAC 地址信息，根据 MAC 地址进行转发，并将这些 MAC 地址与对应的端口记录在自己内部的一个地址表中。

## 1.3 三层交换机

S3760E 三层交换机：3 台。

功能特点：

三层交换机是具有部分路由器功能的交换机，工作在 OSI 网络标准模型的第三层：网络层。三层交换机的最重要目的是加快大型局域网内部的数据交换，所具有的路由功能也是为这目的服务的，能够做到一次路由，多次转发。

对于数据包转发等规律性的过程由硬件高速实现，而像路由信息更新、路由表维护、路由计算、路由确定等功能，由软件实现。

## 1.4 PC 机

Windows PC 机：4 台

功能特点：

PC 机指个人计算机，是指一种大小、价格和性能适用于个人使用的多用途计算机。台式机、笔记本电脑到小型笔记本电脑和平板电脑以及超级本等都属于个人计算机。

## 2 实验背景

某公司内部网络由母公司 Site 1 和子公司 Site 2 组成，其中母公司 Site 1 下包含部门 Office 1 和 Office 2，子公司 Site 2 下包含部门 Office 3 和 Office 4，总部与分公司间网络通过 Tunnel 打通路由。

## 3 网络连通性要求

### 3.1 Site1

母公司 Site 1 的部门 Office 1 和 Office 2 分别隶属于 VLAN 10 和 VLAN20，网关分别指向 SW1 的 SVI 10、SVI 20 接口。SW1 和边界路由器 R1 之间启用动态路由协议 OSPF，并在 area 0 中宣告所有本地路由。其中 SW1 开启快速生成树协议（RSTP），用于提高网络的可靠性、避免环路，实现 STP 的快速收敛。

配置后：位于不同部门的 PC1、PC2 可以相互通信，同时 R1 与 SW1 建立路由邻居并收到 vlan10、20 的路由明细。

### 3.2 Site2

子公司 Site 2 的部门 Office 3 和 Office 4 分别隶属于 vlan30、vlan40。SW2 和 SW3 之间启用 Trunk 放行 VLAN，并分别与边界路由器 R2 建立 OSPF 邻居，在区域 0 中宣告所有本地直连路由。

配置后：位于不同部门的 PC3、PC4 可以相互通信，R2 与 SW2、SW3 建立 OSPF 邻居并收到 VLAN30、VLAN40 的路由明细。

### 3.3 Tunnel

在 R1、R2 上起 Tunnel，源目的地址分别为自己和对端的串口，并通过 Tunnel 隧道建立 OSPF 邻居。

配置后：Tunnel 口创建成功，R1、R2 成功建立 OSPF 邻居，Site 1、Site 2 互通路由明细，PC1、PC2、PC3、PC4 四个部门间可以相互通信。

### 3.4 NATP

利用 NAT 技术，能够转换 IP 包中的 IP 地址，对 IP 包中 TCP 和 UDP 的 Port 进行转换，使多台私有网主机利用 1 个 NAT 公共 IP 实现同时和公网进行通信。

在 R2 的 lo 0 接口模拟公网 IP：8.8.8.8，R1 作为 Site 1 唯一网络出口，

默认路由指向外网接口 s2/0，并下发默认路由。R1 的 s2/0 口上开启端口复用 NAT 对所有来自 Site 1 内部访问外网（8.8.8.8）的流量进行地址转换。

配置后：从 Site 1 去往外部网络的流量均实现了 NAT 地址转换。

### 3.5 ACL

访问控制列表(ACL)是一种基于包过滤的访问控制技术，可以根据设定的条件对接口上的数据包进行过滤，允许其通过或丢弃。通过借助于访问控制列表，可以有效地控制用户对网络的访问。

编写标准 ACL 在 SW2 的入方向放行 PC3 到所有目标地址的流量。编写拓展 ACL 接口下调用在 SW3 入方向只拒绝 PC4 访问外网 8.8.8.8 的流量。

配置后：所有的 PC 间可以相互通信；除 PC4 外，均能访问公网地址 8.8.8.8。

## 4 网络安全性要求

### 4.1 VRRP

vrrp 协议的作用是提供了局域网上的设备备份机制。vrrp 协议是一种容错协议，它保证当主机的下一跳路由器坏掉时，可以及时由另一台路由器来替代，从而保证通讯的连续性和可靠性。vrrp 是指虚拟路由冗余协议，是由 IETF 提出的解决局域网中，配置静态网关出现单点失效现象的路由协议，广泛应用在边缘网络中。

在 SW3 和 SW4 上配置 VRRP（虚拟路由冗余网关），vlan30 的主虚拟网关位于 SW3，vlan40 的主虚拟网关位于 SW4。当交换机检测上行链路转发故障时自动降低本地 vrrp 进程优先级，虚拟网关身份切换到 peer 端。

### 4.2 IPSec

IPSec 是 IETF 提出的使用密码学保护 IP 层通信的安全保密架构，是一个协议簇，通过对 IP 协议的分组进行加密和认证来保护 IP 协议的网络传输协议簇。

IPSec 主要由以下协议组成：

(1)认证头（AH），为 IP 数据报提供无连接数据完整性、消息认证以及防重放攻击保护；

(2)封装安全载荷（ESP），提供机密性、数据源认证、无连接完整性、防重放和有限的传输流（traffic-flow）机密性；

(3)安全关联（SA），提供算法和数据包，提供 AH、ESP 操作所需的参数。

(4)密钥协议（IKE），提供对称密码的密钥的生存和交换。

在企业网络中，为了保护通信安全，用 IPSEC 加密 Tunnel 隧道，模式为隧道模式。规定 IKE 第一阶段 采用预共享密钥的方式建立安全关联，IKE 第二阶段采用 256 位 aes 加密数据、sha 用于数据哈希校验。

### 4.3 MAC地址绑定

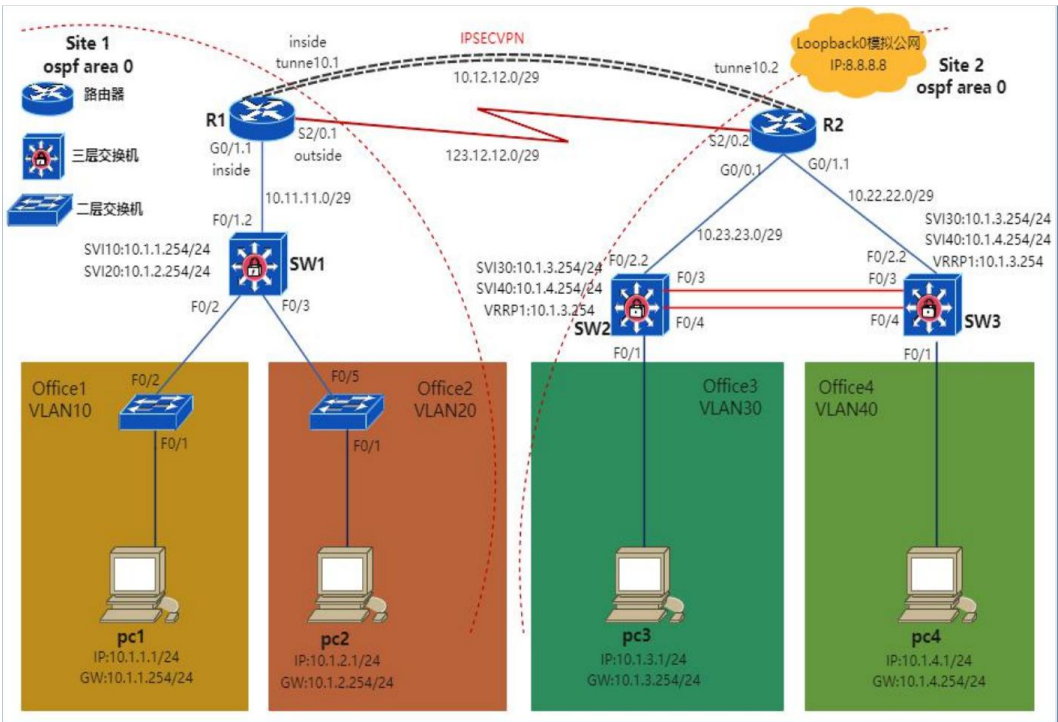
MAC 地址绑定就是利用三层交换机的安全控制列表将交换机上的端口与所对应的 MAC 地址进行捆绑。由于每个网络适配卡具有唯一的 MAC 地址，为了有效防止非法用户盗用网络资源，MAC 地址绑定可以有效的规避非法用户的接入。以进行网络物理层面的安全保护。

企业网络的配置中，在 SW2\3 交换口上启用 mac 地址绑定，如果检测到主机 MAC 改动立即关闭端口，从而保护了网络，避免身份伪造请求。

## 5 网络拓扑

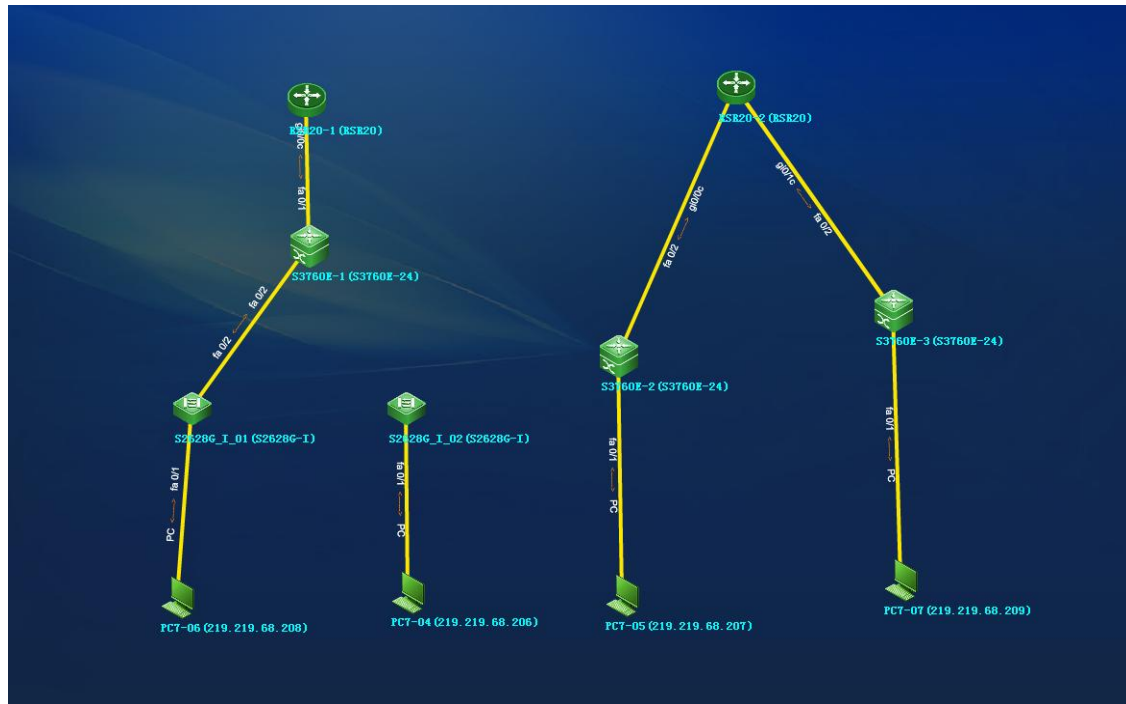
### 5.1 拓扑构建

根据企业的要求，构建了如下的拓扑图：



### 5.2 拓扑仿真

在锐捷仿真实验平台上绘制的拓扑图如下：



除了以上连接通过仿真实验平台连接，下面的线路通过物理线路连接：

- (1) RSR20-1 的 S2/0 口与 RSR20-2 的 S2/0 口；
- (2) S3760E-1 的 f0/3 口与 S2620G-1-02 的 f0/5 口；
- (3) S3760E-2 的 f0/3 口与 S3760E-3 的 f0/3 口；
- (4) S3760E-2 的 f0/4 口与 S3760E-3 的 f0/4 口。

## 6 网络配置

### 6.1 连通性配置

#### 6.1.1 R1

```
enable
```

```
configure terminal
```

```
hostname R1
```

```
interface gi0/1 //给接口配置 ip
```

```
ip address 10.11.11.1 255.255.255.248
```

```
no shutdown
```

```
interface s2/0
```

```
ip address 123.12.12.1 255.255.255.248
```

```
no shutdown
```

```
interface tunnel 0 //
```

配置 tunnel 口，设置模式、协议、IP 地址、源目

```
tunnel mode gre ip
tunnel source 123.12.12.1
tunnel destination 123.12.12.2
ip address 10.12.12.1 255.255.255.248
no shutdown
router ospf 1 //ospf进程 1
network 10.11.11.0 0.0.0.7 area 0 //宣告接口
network 10.12.12.0 0.0.0.7 area 0
default-info originate //给邻居下发默认路由
ip route 0.0.0.0 0.0.0.0 ser2/0 //配置静态默认路由
ip access-list extend NAT //拓展 ACL NAT
permit ip 10.1.0.0 0.0.255.255 hostnamet 8.8.8.8 //
允许源自 10.1.0.0/16 的 ip 层流量访问主机 8.8.8.8
exit //退出
ip nat inside source list NAT interface s2/0 overload //
动态 nat 在 s2/0 接口端口复用
interface s2/0
ip nat outside //nat 流量为出方向
interface tunnel0
ip nat inside //nat 流量进方向
interface gi0/1
ip nat inside //nat 流量进方向
```

## 6.1.2 R2

```
enable
configure terminal
hostname R2
interface gi0/0 //打开接口配置 ip
ip address 10.22.22.1 255.255.255.248
no shutdown
interface gi0/1
ip address 10.23.23.1 255.255.255.248
no shutdown
interface s2/0
ip address 123.12.12.2 255.255.255.248
```

```
no shutdown
interface tunnel 0 //进入 tunnel 口 0
tunnel mode gre ip //tunnel 模式为 gre, ip 支持 ipv4
tunnel source 123.12.12.2 //设置 tunnel 源
tunnel destination 123.12.12.1 //设置 tunnel 目的
ip address 10.12.12.2 255.255.255.248 //给 tunnel 口配置 ip 地址
no shutdown //开启接口
interface lo 0 //进入环回接口 loopback0
ip address 8.8.8.8 255.255.255.255 //配置 ip
router ospf 1 //ospf 进程 1
network 10.22.22.0 0.0.0.7 area 0 //在 area 0 宣告路由
network 10.23.23.0 0.0.0.7 area 0
network 10.12.12.0 0.0.0.7 area 0
```

### 6.1.3 SW1

```
enable //修改主机名
configure terminal
hostname switch1
spanning-tree enable //开启生成树
spanning-tree mode rstp
```

```
vlan 10 //创建 vlan
vlan 20
interface f0/2 //划分 vlan
switch mode access
switch access vlan 10
no shutdown
interface f0/3
switch mode access
switch access vlan 20
no shutdown
interface vlan 10 //进入 svi 口
ip address 10.1.1.254 255.255.255.0 //设置 svi 的 ip 地址
no shutdown //打开接口
interface vlan 20 //设置 svi 口
```

```
ip address 10.1.2.254 255.255.255.0
no shutdown
interface f0/1 //进入接口
no switch //关闭交换功能（打开路由功能）
ip address 10.11.11.2 255.255.255.248 //配置 ip
no shutdown //开启接口
router ospf 1 //开启 ospf 进程 1
network 10.1.1.0 0.0.0.255 area 0 //在 area 0 中宣告网段 10.1.1.0/24
network 10.1.2.0 0.0.0.255 area 0 //宣告网段 10.1.1.0/24
network 10.11.11.0 0.0.0.7 area 0 //宣告网段 10.11.11.0/29
```

#### 6.1.4 SW2

```
enable //修改主机名
configure terminal
hostname switch2
vlan 30 //创建 vlan
vlan 40
interface vlan 30
ip address 10.1.3.254 255.255.255.0
no shutdown
interface r f0/3-4
switch mode trunk
no shutdown
interface f0/1 //vlan 划分
switch mode access
switch access vlan 30
no shutdown
spanning-tree //开启生成树
spanning-tree mode mst //生成树模式 mst
spanning-tree mst conf //配置 mst
instance 1 vlan 30 //划分 vlan30 到 mst 实例 1
instance 2 vlan 40
spanning-tree mst 1 prio 0 //配置实例 1 优先级（本地最高）
spanning-tree mst 2 prio 4096 //配置实例 2 优先级
interface f0/2 //关闭交换功能配置三层 ip
```



```
no switch
ip address 10.22.22.2 255.255.255.248
no shutdown
router ospf 1 //开启 ospf 进程并在 area 0 中宣告路由
network 10.22.22.0 0.0.0.7 area 0
network 10.1.3.0 0.0.0.255 area 0
network 10.1.4.0 0.0.0.255 area 0
ip access-list stand 10 //标准的访问控制列表 10
permit host 10.1.3.1 //放行源地址是 10.1.3.1 的所有流量
interface f0/1 //进入接口
ip access-group 10 in //将 ACL 10 接口下调用在接口的入方向
```

### 6.1.5 SW3

```
enable //修改主机名
configure terminal
hostname switch3
vlan 30 //
vlan 40 //创建 vlan 40 并设置 svi 40 接口
interface vlan 40
ip address 10.1.4.254 255.255.255.0
no shutdown
interface r f0/3-4
switch mode trunk
no shutdown
interface f0/1 //vlan 划分
switch mode access
switch access vlan 40
no shutdown
spanning-tree //配置 mst 生成树
spanning-tree mode mst
spanning-tree mst conf
instance 2 vlan 40
instance 1 vlan 30
spanning-tree mst 2 prio 0
spanning-tree mst 1 prio 4096
```

```
interface f0/2 //关闭交换功能，打开路由功能
no switch
ip address 10.23.23.2 255.255.255.248
no shutdown
router ospf 1 //开启 ospf 进程 1 并宣告网段
network 10.23.23.0 0.0.0.7 area 0
network 10.1.4.0 0.0.0.255 area 0
network 10.1.3.0 0.0.0.255 area 0
ip access-list extended 100 //拓展访问控制列表 100
deny ip host 10.1.4.1 host 8.8.8.8
//拒绝主机 10.1.4.1 访问主机 8.8.8.8
permit ip any any //放行所有流量
interface f0/1 //进入接口 f0/1 并在入方向接口下调用 ACL100
ip access-group 100 in
```

### 6.1.6 SW11

```
enable
configure terminal //特权模式
hostname switch11 //命名
vlan 10 //创建 vlan10
spanning-tree //开启生成树
spanning-tree mode rstp //设置生成树模式 rstp
interface f0/1 //进入接口
switch mode access //设置接口模式
switch access vlan 10 //给接口划分 vlan
no shutdown //打开接口
interface f0/2 //划分 vlan
switch mode access
switch access vlan 10
no shutdown
```

### 6.1.7 SW12

```
enable //进入特权模式修改主机名
configure terminal
hostname switch12
```

```
vlan 20 //创建 vlan
spanning-tree //开启生成树
spanning-tree mode rstp
interface f0/1 //划分 vlan
switch mode access
switch access vlan 20
no shutdown
interface f0/2 //划分 vlan
switch mode access
switch access vlan 20
no shutdown
```

## 6.2 安全性配置

### 6.2.1 VRRP

#### (1) SW2

```
int vlan 30
ip address 10.1.3.252 255.255.255.0
vrrp 1 version 2 //vrrp 进程 1 版本 2
vrrp 1 ip 10.1.3.254 //虚拟网关 10.1.3.254
vrrp 1 prio 100 //本地进程优先级 100（主）
vrrp 1 preempt //开启抢占，进程优先级高的会抢占成为主设备
vrrp 1 track f0/2 20 //监控 f0/2 状态，如果异常优先级降低 20
Int vlan40
Ip add 10.1.4.252 255.255.255.0
vrrp 2 version 2 //进程 1 版本 2
vrrp 2 ip 10.1.4.254 //虚拟网关 10.1.4.254
vrrp 2 prio 99 //本地进程优先级 99（备）
vrrp 2 preEmpt //开启抢占
vrrp 2 track f0/2 20 /监控 f0/2 口状态，异常降低优先级
```

#### (2) SW3

```
int vlan 30
ip address 10.1.3.253 255.255.255.0
vrrp 1 version 2 //版本
vrrp 1 ip 10.1.3.254 //虚拟网关
```

```
vrrp 1 prio 99 //优先级（备）
vrrp 1 pre //抢占
vrrp 1 track f0/2 20 //监控端口
int vlan 40
ip add 10.1.4.253 255.255.255.0
vrrp 2 version 2 //版本
vrrp 2 ip 10.1.4.254 //虚拟网关
vrrp 2 prio 100 //优先级（主）
vrrp 2 pre //抢占
vrrp 2 track f0/2 20 //监控端口
```

## 6.2.2 IPSec

### (1) R1

```
ip access-list extend 100 //拓展 ACL 抓取加密感兴趣流
per ip 10.0.0.0 0.0.0.255
crypto iskamp police 10 //ike 第一阶段策略 10
encry 3des //加密算 3des
authen preshare //协商方法预共享密钥
group 2 //密钥长度 1024
crypto iskamp key 7 ruijie add 10.12.12.2 //加密的共享密钥 ruijie, 对端 ip10.12.12.2
crypto ipsec transform-set IPSEC esp-aes-256 esp-sha-hmac
//ike 第二阶段 设置传输集 IPSEC, 约定 esp 协议封装数据包、加密算法 256 位
aes、哈希算法 sha
mode tunnel //加密模式位传输
crypto map VPN 1 ipsec-iskamp //配置加密映射表 VPN 策略 1
set transform-set IPSEC //设定传输集 IPSEC
set peer 10.12.12.2 //设置对端 ip10.12.12.2
match add 100 //匹配感兴趣流量
int tunnel0
crypto map VPN //接口下调用加密策略
```

### (2) R2

```
ip access-list extend 100 //同上
per ip 10.0.0.0 0.0.0.255
crypto iskamp police 10
encry 3des
```

```
authen preshare
group 2
crypto iskamp key 7 ruijie add 10.12.12.1
crypto ipsec transform-set IPSEC esp-aes-256 esp-sha-hmac
mode tunnel
crypto map VPN 1 ipsec-iskamp
set transform-set IPSEC
set peer 10.12.12.1
match add 100
int tunnel0
crypto map VPN
```

### 6.2.3 MAC地址绑定

#### (1) SW2

```
interface f0/1
sw port-sec mac-address sticky //端口安全自动绑定 mac
sw port-sec violation shutdown //发生违规自动关闭端口
```

#### (2) SW3

```
interface f0/1
sw port-sec mac-address sticky //端口安全自动绑定 mac
sw port-sec violation shutdown //发生违规自动关闭端口
```

### 6.2.4 开启用户远程登陆

#### (1) SW1

```
aaa new-mode //开启 AAA
radius-server host namest 150.1.1.1 //AAA 服务器 ip
radius-server key ruijie
//用于连接 radius 服务器的密钥 ruijie
aaa authentication login ruijie group radius local
//登录方法认证列表 ruijie，优先采用 radius 组认证其次本地组
aaa local authentication attempts 3
//允许 3 次登录失败
aaa local authentication lockout-time 1
//连续 3 次输错密码锁定账户 1 小时
username admin password ruijie
```

```

//创建本地用户 admin 密码 ruijie
username admin privilege 15
//用户权限 15 级
aaa authostnamerization exec execauth group radius local
//登陆授权列表 execauth，优先采用 radius 组认证其次本地组
aaa authostnamerization commands 15 commauth group radius local
//命令授权列表 commauth，优先采用 radius 组认证其次本地组
aaa accounting exec execaccount start-stop group radius local
//登入登出审计列表 execaccount，优先采用 radius 组认证其次本地组
aaa accounting commands 15 commaccount start-stop group radius local
//命令审计列表 commaccount，优先采用 radius 组认证其次本地组
line vty 0 4
//进入接口 vty
login authentication ruijie
//接口下调用认证列表
login authostnamerization exec execauth
//接口下调用登陆授权列表
login authostnamerization commands commauth
//接口下调用命令授权列表
accounting exec execaccount
//接口下调用登入登出审计列表
accounting commands 15 commaccount
//接口下调用命令登出审计列表

```

## 7 结果验证

### 7.1 验证 PC1 与各关键节点互通

(1) PC1 -> PC2、PC3、PC4



The screenshot shows a Windows command prompt window titled "管理员: C:\Windows\system32\cmd.exe". The window displays the output of a ping command to 10.1.2.1. The output indicates that the ping is successful, with 32 bytes of data being received from 10.1.2.1, and the response time is less than 1ms with a TTL of 128. The output is as follows:

```

Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>ping 10.1.2.1

正在 Ping 10.1.2.1 具有 32 字节的数据:
来自 10.1.2.1 的回复: 字节=32 时间<1ms TTL=128
来自 10.1.2.1 的回复: 字节=32 时间<1ms TTL=128
来自 10.1.2.1 的回复: 字节=32 时间<1ms TTL=128
来自 10.1.2.1 的回复: 字节=32 时间<1ms TTL=128

```

```

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

C:\Users\Administrator>ping 10.1.3.1

正在 Ping 10.1.3.1 具有 32 字节的数据:
来自 10.1.3.1 的回复: 字节=32 时间=3109ms TTL=124
来自 10.1.3.1 的回复: 字节=32 时间=3114ms TTL=124
来自 10.1.3.1 的回复: 字节=32 时间=3126ms TTL=124
请求超时。

10.1.3.1 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 3, 丢失 = 1 (25% 丢失),
往返行程的估计时间<以毫秒为单位>:
    最短 = 3109ms, 最长 = 3126ms, 平均 = 3116ms

C:\Users\Administrator>ping 10.1.4.1

正在 Ping 10.1.4.1 具有 32 字节的数据:
来自 10.1.4.1 的回复: 字节=32 时间=3115ms TTL=124
来自 10.1.4.1 的回复: 字节=32 时间=3115ms TTL=124
来自 10.1.4.1 的回复: 字节=32 时间=3110ms TTL=124
来自 10.1.4.1 的回复: 字节=32 时间=3094ms TTL=124

10.1.4.1 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间<以毫秒为单位>:
    最短 = 3094ms, 最长 = 3115ms, 平均 = 3108ms

C:\Users\Administrator>_

```

## (2) PC1->SW1

```

C:\Users\Administrator>ping 10.1.1.254

正在 Ping 10.1.1.254 具有 32 字节的数据:
来自 10.1.1.254 的回复: 字节=32 时间=4ms TTL=64
来自 10.1.1.254 的回复: 字节=32 时间=3ms TTL=64
来自 10.1.1.254 的回复: 字节=32 时间=3ms TTL=64
来自 10.1.1.254 的回复: 字节=32 时间=7ms TTL=64

10.1.1.254 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 = 3ms, 最长 = 7ms, 平均 = 4ms

```

## (3) PC1->R1

```

C:\Users\Administrator>ping 10.11.11.1

正在 Ping 10.11.11.1 具有 32 字节的数据:
来自 10.11.11.1 的回复: 字节=32 时间<1ms TTL=63
来自 10.11.11.1 的回复: 字节=32 时间<1ms TTL=63
来自 10.11.11.1 的回复: 字节=32 时间<1ms TTL=63
来自 10.11.11.1 的回复: 字节=32 时间<1ms TTL=63

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

```

#### (4) PC1->8.8.8.8

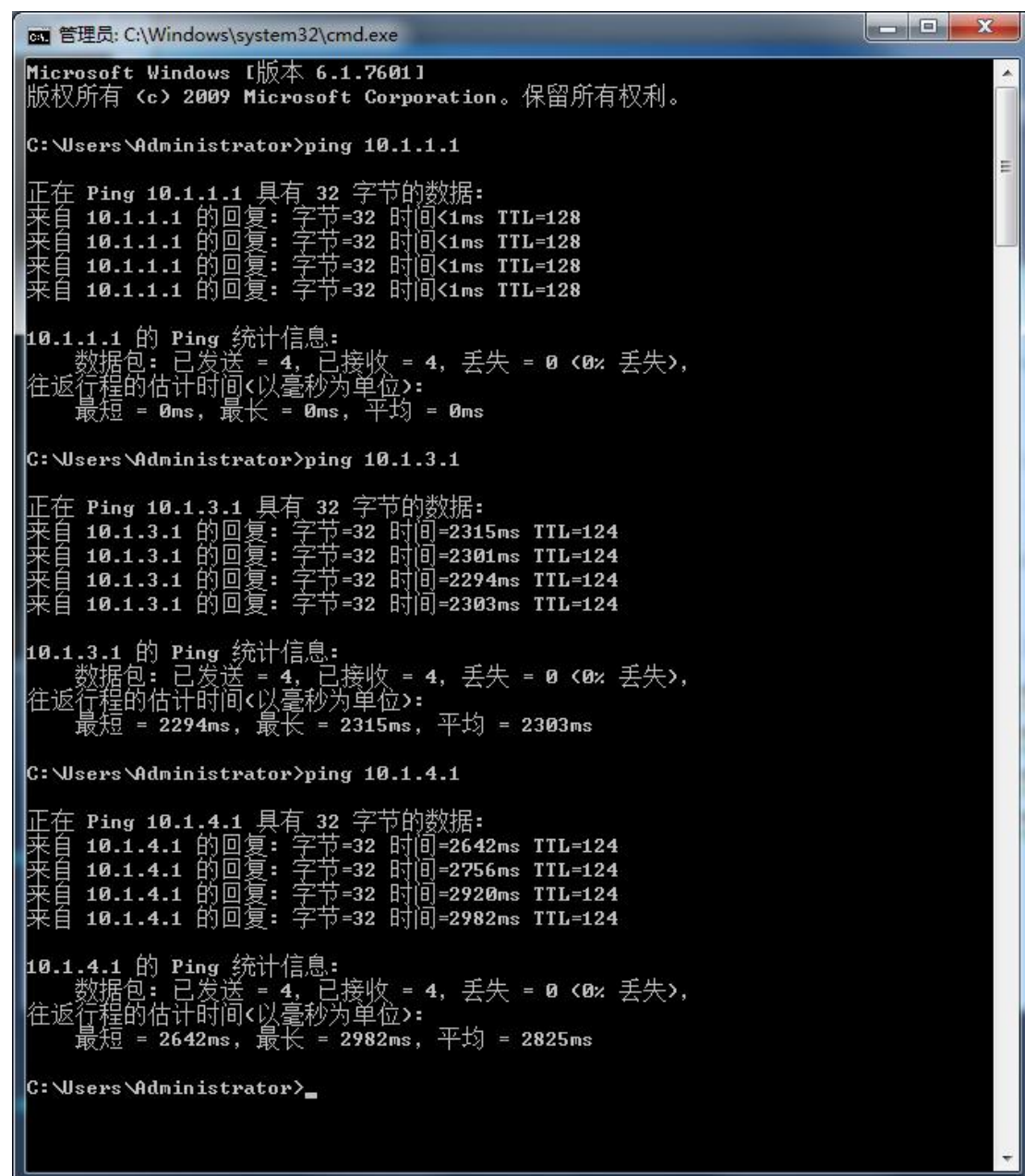
```
C:\Users\Administrator>ping 8.8.8.8

正在 Ping 8.8.8.8 具有 32 字节的数据:
来自 8.8.8.8 的回复: 字节=32 时间=507ms TTL=62
来自 8.8.8.8 的回复: 字节=32 时间=511ms TTL=62
来自 8.8.8.8 的回复: 字节=32 时间=554ms TTL=62
来自 8.8.8.8 的回复: 字节=32 时间=504ms TTL=62

8.8.8.8 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 = 504ms, 最长 = 554ms, 平均 = 519ms
```

## 7.2 验证 PC2 与各关键节点互通

#### (1) PC2->PC1、PC3、PC4



```
ca. 管理员: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>ping 10.1.1.1

正在 Ping 10.1.1.1 具有 32 字节的数据:
来自 10.1.1.1 的回复: 字节=32 时间<1ms TTL=128
来自 10.1.1.1 的回复: 字节=32 时间<1ms TTL=128
来自 10.1.1.1 的回复: 字节=32 时间<1ms TTL=128
来自 10.1.1.1 的回复: 字节=32 时间<1ms TTL=128

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

C:\Users\Administrator>ping 10.1.3.1

正在 Ping 10.1.3.1 具有 32 字节的数据:
来自 10.1.3.1 的回复: 字节=32 时间=2315ms TTL=124
来自 10.1.3.1 的回复: 字节=32 时间=2301ms TTL=124
来自 10.1.3.1 的回复: 字节=32 时间=2294ms TTL=124
来自 10.1.3.1 的回复: 字节=32 时间=2303ms TTL=124

10.1.3.1 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间<以毫秒为单位>:
    最短 = 2294ms, 最长 = 2315ms, 平均 = 2303ms

C:\Users\Administrator>ping 10.1.4.1

正在 Ping 10.1.4.1 具有 32 字节的数据:
来自 10.1.4.1 的回复: 字节=32 时间=2642ms TTL=124
来自 10.1.4.1 的回复: 字节=32 时间=2756ms TTL=124
来自 10.1.4.1 的回复: 字节=32 时间=2920ms TTL=124
来自 10.1.4.1 的回复: 字节=32 时间=2982ms TTL=124

10.1.4.1 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间<以毫秒为单位>:
    最短 = 2642ms, 最长 = 2982ms, 平均 = 2825ms

C:\Users\Administrator>
```



## (2) PC2->SW1

```
C:\Users\Administrator>ping 10.1.2.254

正在 Ping 10.1.2.254 具有 32 字节的数据:
来自 10.1.2.254 的回复: 字节=32 时间=1ms TTL=64
来自 10.1.2.254 的回复: 字节=32 时间=1ms TTL=64
来自 10.1.2.254 的回复: 字节=32 时间=1ms TTL=64
来自 10.1.2.254 的回复: 字节=32 时间=1ms TTL=64

10.1.2.254 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 = 1ms, 最长 = 1ms, 平均 = 1ms
```

## (3) PC2->R1

```
C:\Users\Administrator>ping 10.11.11.1

正在 Ping 10.11.11.1 具有 32 字节的数据:
来自 10.11.11.1 的回复: 字节=32 时间<1ms TTL=63
来自 10.11.11.1 的回复: 字节=32 时间<1ms TTL=63
来自 10.11.11.1 的回复: 字节=32 时间<1ms TTL=63
来自 10.11.11.1 的回复: 字节=32 时间<1ms TTL=63

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

## (4) PC2->8.8.8.8

```
C:\Users\Administrator>ping 8.8.8.8

正在 Ping 8.8.8.8 具有 32 字节的数据:
来自 8.8.8.8 的回复: 字节=32 时间=507ms TTL=62
来自 8.8.8.8 的回复: 字节=32 时间=511ms TTL=62
来自 8.8.8.8 的回复: 字节=32 时间=554ms TTL=62
来自 8.8.8.8 的回复: 字节=32 时间=504ms TTL=62

8.8.8.8 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 = 504ms, 最长 = 554ms, 平均 = 519ms
```

## 7.3 验证 PC3 与各关键节点互通

### (1) PC3->PC1、PC2、PC4

```
管理员: C:\Windows\system32\cmd.exe

Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>ping 10.1.1.1

正在 Ping 10.1.1.1 具有 32 字节的数据:
来自 10.1.1.1 的回复: 字节=32 时间=2490ms TTL=124
来自 10.1.1.1 的回复: 字节=32 时间=2504ms TTL=124
来自 10.1.1.1 的回复: 字节=32 时间=2477ms TTL=124
来自 10.1.1.1 的回复: 字节=32 时间=2529ms TTL=124
```

```
10.1.1.1 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间<以毫秒为单位>:
    最短 = 2477ms, 最长 = 2529ms, 平均 = 2500ms
```

```
C:\Users\Administrator>ping 10.1.2.1
```

```
正在 Ping 10.1.2.1 具有 32 字节的数据:
来自 10.1.2.1 的回复: 字节=32 时间=1951ms TTL=124
来自 10.1.2.1 的回复: 字节=32 时间=2201ms TTL=124
来自 10.1.2.1 的回复: 字节=32 时间=2517ms TTL=124
来自 10.1.2.1 的回复: 字节=32 时间=2677ms TTL=124
```

```
10.1.2.1 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间<以毫秒为单位>:
    最短 = 1951ms, 最长 = 2677ms, 平均 = 2336ms
```

```
C:\Users\Administrator>ping 10.1.4.1
```

```
正在 Ping 10.1.4.1 具有 32 字节的数据:
来自 10.1.4.1 的回复: 字节=32 时间<1ms TTL=127
来自 10.1.4.1 的回复: 字节=32 时间<1ms TTL=127
来自 10.1.4.1 的回复: 字节=32 时间<1ms TTL=127
来自 10.1.4.1 的回复: 字节=32 时间<1ms TTL=127
```

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

```
C:\Users\Administrator>
```

(2) PC3 → SW2

```
C:\Users\Administrator>ping 10.1.3.254
```

```
正在 Ping 10.1.3.254 具有 32 字节的数据:
来自 10.1.3.254 的回复: 字节=32 时间=1ms TTL=64
来自 10.1.3.254 的回复: 字节=32 时间=1ms TTL=64
来自 10.1.3.254 的回复: 字节=32 时间=1ms TTL=64
来自 10.1.3.254 的回复: 字节=32 时间=2ms TTL=64
```

```
10.1.3.254 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 = 1ms, 最长 = 2ms, 平均 = 1ms
```

```
C:\Users\Administrator>
```

(3) PC3→R2

```
C:\Users\Administrator>ping 10.23.23.1
```

```
正在 Ping 10.23.23.1 具有 32 字节的数据:
来自 10.23.23.1 的回复: 字节=32 时间<1ms TTL=63
来自 10.23.23.1 的回复: 字节=32 时间<1ms TTL=63
来自 10.23.23.1 的回复: 字节=32 时间<1ms TTL=63
来自 10.23.23.1 的回复: 字节=32 时间<1ms TTL=63
```

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

#### (4) PC3->8.8.8.8

```
C:\Users\Administrator>ping 8.8.8.8

正在 Ping 8.8.8.8 具有 32 字节的数据:
来自 8.8.8.8 的回复: 字节=32 时间=507ms TTL=62
来自 8.8.8.8 的回复: 字节=32 时间=511ms TTL=62
来自 8.8.8.8 的回复: 字节=32 时间=554ms TTL=62
来自 8.8.8.8 的回复: 字节=32 时间=504ms TTL=62

8.8.8.8 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 = 504ms, 最长 = 554ms, 平均 = 519ms
```

### 7.4 验证 PC4 与各关键节点互通

#### (1) PC4->PC1、PC2、PC3

```
C:\Users\Administrator>ping 10.1.3.1

正在 Ping 10.1.3.1 具有 32 字节的数据:
来自 10.1.3.1 的回复: 字节=32 时间<1ms TTL=127
来自 10.1.3.1 的回复: 字节=32 时间<1ms TTL=127
来自 10.1.3.1 的回复: 字节=32 时间<1ms TTL=127
来自 10.1.3.1 的回复: 字节=32 时间<1ms TTL=127

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

C:\Users\Administrator>ping 10.1.1.1

正在 Ping 10.1.1.1 具有 32 字节的数据:
来自 10.1.1.1 的回复: 字节=32 时间=3194ms TTL=124
来自 10.1.1.1 的回复: 字节=32 时间=3150ms TTL=124
来自 10.1.1.1 的回复: 字节=32 时间=3145ms TTL=124
来自 10.1.1.1 的回复: 字节=32 时间=3141ms TTL=124

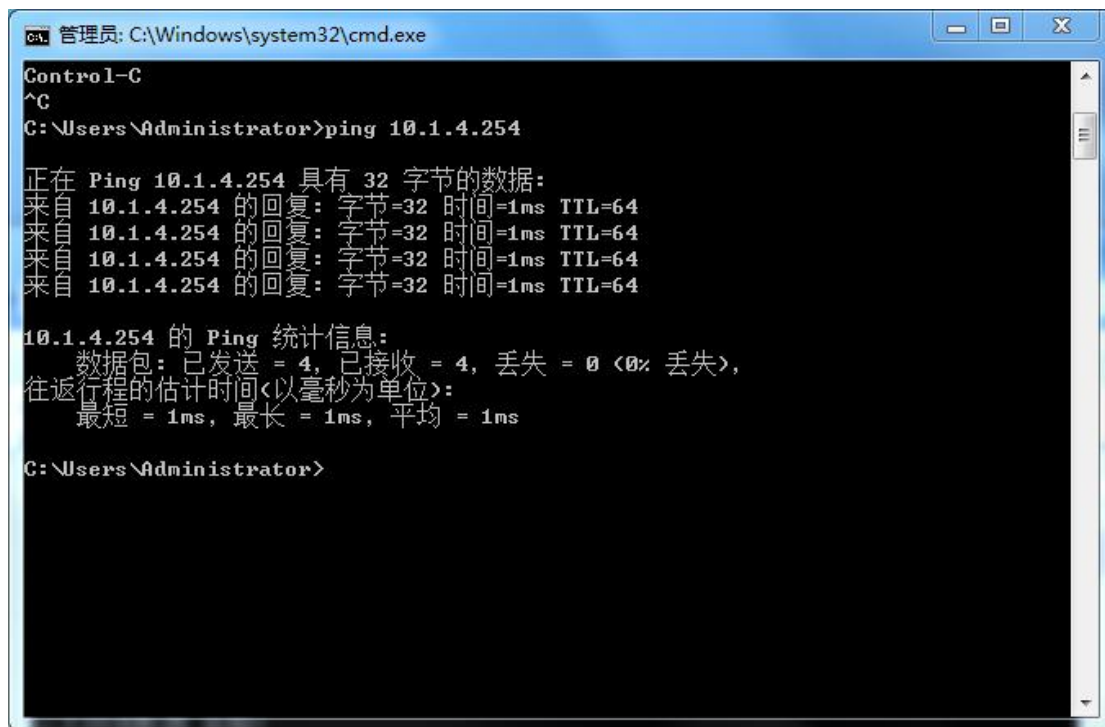
10.1.1.1 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 3141ms, 最长 = 3194ms, 平均 = 3157ms

C:\Users\Administrator>ping 10.1.2.1

正在 Ping 10.1.2.1 具有 32 字节的数据:
请求超时。
来自 10.1.2.1 的回复: 字节=32 时间=2896ms TTL=124
来自 10.1.2.1 的回复: 字节=32 时间=2885ms TTL=124
来自 10.1.2.1 的回复: 字节=32 时间=2909ms TTL=124

10.1.2.1 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 3, 丢失 = 1 (25% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 2885ms, 最长 = 2909ms, 平均 = 2896ms
```

(2) PC4->SW3



```
ca 管理员: C:\Windows\system32\cmd.exe
Control-C
^C
C:\Users\Administrator>ping 10.1.4.254

正在 Ping 10.1.4.254 具有 32 字节的数据:
来自 10.1.4.254 的回复: 字节=32 时间=1ms TTL=64
来自 10.1.4.254 的回复: 字节=32 时间=1ms TTL=64
来自 10.1.4.254 的回复: 字节=32 时间=1ms TTL=64
来自 10.1.4.254 的回复: 字节=32 时间=1ms TTL=64

10.1.4.254 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 = 1ms, 最长 = 1ms, 平均 = 1ms

C:\Users\Administrator>
```

(3) PC4->R2

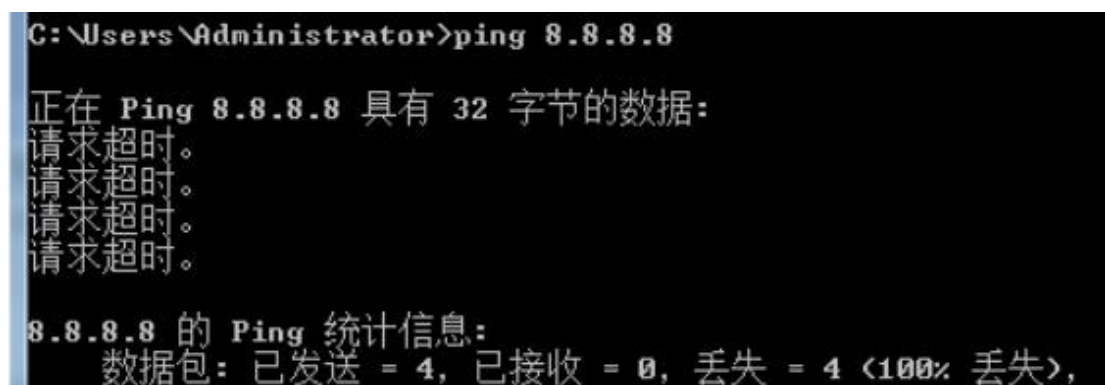


```
C:\Users\Administrator>ping 10.22.22.1

正在 Ping 10.22.22.1 具有 32 字节的数据:
来自 10.22.22.1 的回复: 字节=32 时间=607ms TTL=62
来自 10.22.22.1 的回复: 字节=32 时间=603ms TTL=62
来自 10.22.22.1 的回复: 字节=32 时间=557ms TTL=62
来自 10.22.22.1 的回复: 字节=32 时间=616ms TTL=62

10.22.22.1 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间<以毫秒为单位>:
        最短 = 557ms, 最长 = 616ms, 平均 = 595ms
```

(4) PC4 → 8.8.8.8 (由于配置了 ACL, 所以无 ping 通)



```
C:\Users\Administrator>ping 8.8.8.8

正在 Ping 8.8.8.8 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。

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

## 7.5 IPsec Tunnel 验证

### (1) PC2->PC4

```
C:\Users\Administrator>tracert 10.1.4.1

通过最多 30 个跃点跟踪
到 WL21-16 [10.1.4.1] 的路由:

 1      1 ms      1 ms      1 ms  10.1.2.254
 2     <1 毫秒   <1 毫秒   <1 毫秒 10.11.11.1
 3    2659 ms    2696 ms    2698 ms 10.12.12.2
 4      *        *        3164 ms 10.22.22.2
 5    3047 ms     *        2971 ms WL21-16 [10.1.4.1]

跟踪完成。
```

### (2) PC3->PC1

```
C:\Users\Administrator>tracert 10.1.1.1

通过最多 30 个跃点跟踪
到 WL21-19 [10.1.1.1] 的路由:

 1      7 ms      1 ms      1 ms  10.1.3.252
 2     <1 毫秒   <1 毫秒   <1 毫秒 10.23.23.1
 3    3447 ms     *          *      10.12.12.1
 4    3565 ms     *        3540 ms 10.11.11.2
 5    3451 ms    3328 ms    3464 ms WL21-19 [10.1.1.1]

跟踪完成。
```