
浙江工业大学

计算机网络课程设计报告



学 号 _____

姓 名 _____

班 级 _____

组 号 _____

提交日期 _____ 2024 年 1 月 3 日 _____

目录

.....	1
课程的目的和任务	6
1.1 课程目标.....	6
1.2 课程任务	6
课程的基本内容和要求	6
2.1 常用网络命令	6
2.2 交换机与路由器	24
2.2.1 交换机配置与交换机 VLAN 设计	24
2.2.2 路由配置与静态路由设计	28
2.2.3 动态路由设计	32
2.2.4 网络集成设计（RIP、OSPF 混合动态路由）	37
2.3 网络编程.....	46
2.3.1 开发环境及关键配置	47
2.3.2 设计思路	48
2.3.3 工作流程	49
2.3.4 运行截图及结果说明	51
2.3.5 CPP 源码	52
实验总结与心得体会	54

图目录

图 1-1 IPCONFIG	7
图 1-2 IPCONFIG /?	8
图 1-3 IPCONFIG/ALL（显示完整配置信息）	9
图 1-4 IPCONFIG /RELEASE（释放指定适配器的 IPV4 地址）	10
图 1-5 IPCONFIG /RENEW（更新指定适配器的 IPV4 地址）	11
图 1-6 IPCONFIG /DISPLAYDNS（显示 DNS 解析程序缓存的内容）	12
图 1-7 IPCONFIG /FLUSHDNS（清除 DNS 解析程序缓存）	12
图 1-8 PING /?（显示帮助信息）	13
图 1-9 PING	13
图 1-10 PING -T（持续 PING）	14
图 1-11 PING -N（要发送的回显请求数）	14
图 1-12 NETSTAT	15
图 1-13 NETSTAT /?（显示帮助信息）	16
图 1-14 NETSTAT -A（显示所有连接和侦听端口）	16
图 1-15 NETSTAT -ES（显示网络协议的统计信息）	17
图 1-16 NETSTAT -VB（显示每个连接或监听端口的可执行文件信息和模块信息）	19
图 1-17 NETSTAT -R（显示路由表）	19
图 1-18 NETSTAT -F（显示外部地址的完全限定域名）	20
图 1-19 TRACERT /?	21
图 1-20 TRACERT	21
图 1-21 TRACERT -H（搜索目标的最大跃点数）	21
图 1-22 ARP	22

图 1-24 TELNET	24
图 1-25 TELNET /?.....	24
图 2-1 交换机拓扑图.....	25
图 2-2 交换机配置（命令行）	25
图 2-3 交换机配置结果.....	26
图 2-4 使用 PING 测试	27
图 2-5 创建复杂 PDU	28
图 2-6 数据报交付	28
图 2-7 静态路由拓扑图.....	29
图 2-8 路由器配置（命令行）	30
图 2-9 该路由器配置好后的路由表	30
图 2-10 主机的配置	31
图 2-11 主机 ZYK67_PC1(192.168.0.3) PING 主机 ZYK67_PC3(192.168.1.3)	31
图 2-12 主机 ZYK67_PC1(192.168.0.3) TRACERT 主机 ZYK67_PC3(192.168.1.3)	32
图 2-13 RIP 动态路由配置拓扑图.....	33
图 2-14 路由器端口配置	34
图 2-15 路由器启用 RIP 动态路由模式	34
图 2-16 路由器配置好后的路由表	35
图 2-17 路由器配置好后的路由协议信息	35
图 2-18 主机的配置	36
图 2-19 主机 0 PING TRACERT 主机 1	36
图 2-20 路由器配置好后的路由表（更新后）	37
图 2-21 路由器配置好后的路由协议信息（更新后）	37
图 2-22 RIP、OSPF 混合动态路由配置拓扑图	38

图 2-23 RIP 动态路由端口配置	39
图 2-24 RIP 动态路由模式配置	39
图 2-25 OSPF 动态路由端口配置	40
图 2-26 OSPF 动态路由模式配置	40
图 2-27 混合路由器端口配置.....	41
图 2-28 混合路由器路由模式配置	41
图 2-29 混合路由器跨规则模式配置	41
图 2-30 主机配置	42
图 2-31 主机 0 PING TRACERT 主机 11.....	43
图 2-32 RIP 路由器路由表.....	44
图 2-33 OSPF 路由器路由表.....	45
图 2-34 混合路由器路由表	46
图 2-35 IP 数据报格式.....	47
图 2-36 自定义运输层协议格式.....	47
图 2-37 监听端工作流程.....	49
图 2-38 发送端工作流程	51
图 2-39 发送端结果截图	51
图 2-40 监听端结果截图	51
图 2-41 源代码截图	54

表目录

表 2-1 交换机常用命令	25
表 2-2 路由器配置常用命令	29
表 2-3 路由器配置常用命令（RIP）	32
表 2-4 路由器配置指令（OSPF）	38

课程的目的和任务

1.1 课程目标

课程教学目标 1： 理解计算机网络体系结构和工作原理，掌握网络测试与故障检测的基本方法。

课程教学目标 2： 通过自学熟练掌握 Cisco Packet Tracer 等现代主流网络仿真工具，实现基于仿真工具的协议数据分析，认识网络技术和工具的发展现状。

课程教学目标 3： 针对网络工程问题设计实验方案，能够按照实验方案实施仿真实验，采集和整理数据，并用所学网络知识对实验数据进行分析、处理和解释的能力。

1.2 课程任务

1. 掌握计算机网络相关命令原理及应用，能够运用网络命令进行网络故障检测分析。在仿真环境下，掌握网络设备交换机和路由器的基本操作，掌握现代网络工具的现状与发展。
2. 掌握交换机 VLAN 设置及验证方法。
3. 掌握在路由器上实现静态路由配置方法
4. 掌握在路由器上实现 RIP 或 OSPF 协议配置方法。
5. 基于网络工程任务需求，构建系统方案、包括子网划分、交换机、路由器的配置等，能够实施实验、处理数据并分析实验结果。
6. 基于网络编程的特定任务需求进行 socket 编程，设计处理流程、数据结构、实施实验并分析实验结果

课程的基本内容和要求

2.1 常用网络命令

(1) ipconfig

ipconfig 是一个命令行工具，用于查看和管理计算机的网络配置。它可以显示网络接口的详细信息，比如 IP 地址、子网掩码和默认网关。除了查看信息外，ipconfig 还能执行一些网络配置操作，比如：

- 更新 IP 地址：通过获取新的 IP 地址来重新连接网络（通常用于 DHCP 环境）。
- 释放 IP 地址：断开当前网络连接，释放 DHCP 分配的 IP 地址。
- 刷新 DNS 缓存：清除计算机存储的 DNS 解析记录，解决某些网络连接问题。

这个工具常用于排查网络问题，帮助管理员诊断和解决连接问题。

```

C:\Users\zhaoy>ipconfig

Windows IP 配置

无线局域网适配器 本地连接* 1:

    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :

无线局域网适配器 本地连接* 2:

    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :

无线局域网适配器 WLAN:

    连接特定的 DNS 后缀 . . . . . :
    本地链接 IPv6 地址 . . . . . : fe80::36fb:45cb:d2df:2a8%20
    IPv4 地址 . . . . . : 192.168.31.179
    子网掩码 . . . . . : 255.255.255.0
    默认网关 . . . . . : 192.168.31.1

以太网适配器 VMware Network Adapter VMnet1:

    连接特定的 DNS 后缀 . . . . . :
    本地链接 IPv6 地址 . . . . . : fe80::14f4:5080:ae4b:62a6%4
    IPv4 地址 . . . . . : 192.168.126.1
    子网掩码 . . . . . : 255.255.255.0
    默认网关 . . . . . :

以太网适配器 VMware Network Adapter VMnet8:

    连接特定的 DNS 后缀 . . . . . :
    本地链接 IPv6 地址 . . . . . : fe80::b6db:daec:75b4:74ae%3
    IPv4 地址 . . . . . : 192.168.72.1
    子网掩码 . . . . . : 255.255.255.0
    默认网关 . . . . . :

以太网适配器 蓝牙网络连接:

    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :

以太网适配器 以太网:

    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :

```

图 1-1 ipconfig


```
C:\Users\zhaoy>ipconfig /?
```

用法:

```
ipconfig [/allcompartments] [/? | /all |  
/renew [adapter] | /release [adapter] |  
/renew6 [adapter] | /release6 [adapter] |  
/flushdns | /displaydns | /registerdns |  
/showclassid adapter |  
/setclassid adapter [classid] |  
/showclassid6 adapter |  
/setclassid6 adapter [classid] ]
```

其中

adapter 连接名称
 (允许使用通配符 * 和 ?, 参见示例)

选项:

/?	显示此帮助消息
/all	显示完整配置信息。
/release	释放指定适配器的 IPv4 地址。
/release6	释放指定适配器的 IPv6 地址。
/renew	更新指定适配器的 IPv4 地址。
/renew6	更新指定适配器的 IPv6 地址。
/flushdns	清除 DNS 解析程序缓存。
/registerdns	刷新所有 DHCP 租用并重新注册 DNS 名称
/displaydns	显示 DNS 解析程序缓存的内容。
/showclassid	显示适配器允许的所有 DHCP 类 ID。
/setclassid	修改 DHCP 类 ID。
/showclassid6	显示适配器允许的所有 IPv6 DHCP 类 ID。
/setclassid6	修改 IPv6 DHCP 类 ID。

默认情况下, 仅显示绑定到 TCP/IP 的每个适配器的 IP 地址、子网掩码和默认网关。

对于 Release 和 Renew, 如果未指定适配器名称, 则会释放或更新所有绑定到 TCP/IP 的适配器的 IP 地址租用。

对于 Setclassid 和 Setclassid6, 如果未指定 ClassId, 则会删除 ClassId。

示例:

> ipconfig	... 显示信息
> ipconfig /all	... 显示详细信息
> ipconfig /renew	... 更新所有适配器
> ipconfig /renew EL*	... 更新所有名称以 EL 开头的连接
> ipconfig /release *Con*	... 释放所有匹配的连接, 例如“有线以太网连接 1”或 “有线以太网连接 2”
> ipconfig /allcompartments	... 显示有关所有隔离舱的信息
> ipconfig /allcompartments /all	... 显示有关所有隔离舱的信息

图 1-2 ipconfig /?

```

C:\Users\zhaoy>ipconfig /all

Windows IP 配置

   主机名 . . . . . : ZYK_1218
   主 DNS 后缀 . . . . . : 
   节点类型 . . . . . : 混合
   IP 路由已启用 . . . . . : 否
   WINS 代理已启用 . . . . . : 否

无线局域网适配器 本地连接* 1:

   媒体状态 . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . : 
   描述 . . . . . : Microsoft Wi-Fi Direct Virtual Adapter
   物理地址 . . . . . : DC-46-28-12-FC-97
   DHCP 已启用 . . . . . : 是
   自动配置已启用 . . . . . : 是

无线局域网适配器 本地连接* 2:

   媒体状态 . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . : 
   描述 . . . . . : Microsoft Wi-Fi Direct Virtual Adapter #2
   物理地址 . . . . . : DE-46-28-12-FC-96
   DHCP 已启用 . . . . . : 是
   自动配置已启用 . . . . . : 是

无线局域网适配器 WLAN:

   连接特定的 DNS 后缀 . . . . . : 
   描述 . . . . . : Intel(R) Wi-Fi 6 AX201 160MHz
   物理地址 . . . . . : DC-46-28-12-FC-96
   DHCP 已启用 . . . . . : 是
   自动配置已启用 . . . . . : 是
   本地链接 IPv6 地址 . . . . . : fe80::36fb:45cb:d2df:2a8%28(首选)
   IPv4 地址 . . . . . : 192.168.31.179(首选)
   子网掩码 . . . . . : 255.255.255.0
   获得租约的时间 . . . . . : 2024年10月15日 12:17:36
   租约过期的时间 . . . . . : 2024年10月16日 0:17:32
   默认网关 . . . . . : 192.168.31.1
   DHCP 服务器 . . . . . : 192.168.31.1
   DHCPv6 IAID . . . . . : 182288848
   DHCPv6 客户端 DUID . . . . . : 08-01-08-01-2C-28-05-8E-04-8F-1B-6F-94-6C
   DNS 服务器 . . . . . : 192.168.31.1
   TCP/IP 上的 NetBIOS . . . . . : 已启用

以太网适配器 VMware Network Adapter VMnet1:

   连接特定的 DNS 后缀 . . . . . : 
   描述 . . . . . : VMware Virtual Ethernet Adapter for VMnet1
   物理地址 . . . . . : 08-50-56-C8-00-01
   DHCP 已启用 . . . . . : 是
   自动配置已启用 . . . . . : 是
   本地链接 IPv6 地址 . . . . . : fe80::14f4:5880:ae4b:62a6%4(首选)
   IPv4 地址 . . . . . : 192.168.126.1(首选)
   子网掩码 . . . . . : 255.255.255.0
   获得租约的时间 . . . . . : 2024年10月15日 12:17:28
   租约过期的时间 . . . . . : 2024年10月15日 16:34:51
   默认网关 . . . . . : 
   DHCP 服务器 . . . . . : 192.168.126.254
   DHCPv6 IAID . . . . . : 973899894
   DHCPv6 客户端 DUID . . . . . : 08-01-08-01-2C-28-05-8E-04-8F-1B-6F-94-6C
   TCP/IP 上的 NetBIOS . . . . . : 已启用

以太网适配器 VMware Network Adapter VMnet8:

   连接特定的 DNS 后缀 . . . . . : 
   描述 . . . . . : VMware Virtual Ethernet Adapter for VMnet8
   物理地址 . . . . . : 08-50-56-C8-00-08
   DHCP 已启用 . . . . . : 是
   自动配置已启用 . . . . . : 是
   本地链接 IPv6 地址 . . . . . : fe80::b6db:daec:75b4:74ae%3(首选)
   IPv4 地址 . . . . . : 192.168.72.1(首选)
   子网掩码 . . . . . : 255.255.255.0
   获得租约的时间 . . . . . : 2024年10月15日 12:17:28
   租约过期的时间 . . . . . : 2024年10月15日 16:34:51
   默认网关 . . . . . : 
   DHCP 服务器 . . . . . : 192.168.72.254
   DHCPv6 IAID . . . . . : 1006653526
   DHCPv6 客户端 DUID . . . . . : 08-01-08-01-2C-28-05-8E-04-8F-1B-6F-94-6C
   主 WINS 服务器 . . . . . : 192.168.72.2
   TCP/IP 上的 NetBIOS . . . . . : 已启用

以太网适配器 蓝牙网络连接:

   媒体状态 . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . : 
   描述 . . . . . : Bluetooth Device (Personal Area Network)
   物理地址 . . . . . : DC-46-28-12-FC-9A
   DHCP 已启用 . . . . . : 是
   自动配置已启用 . . . . . : 是

以太网适配器 以太网:

   媒体状态 . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . : 
   描述 . . . . . : Realtek PCIe GbE Family Controller
   物理地址 . . . . . : 8E-BF-1B-6F-94-6C
   DHCP 已启用 . . . . . : 是
   自动配置已启用 . . . . . : 是

```

图 1-3 ipconfig/all（显示完整配置信息）

```

C:\Users\zhaoy>ipconfig /release

Windows IP 配置

不能在 本地连接* 1 上执行任何操作，它已断开媒体连接。
不能在 本地连接* 2 上执行任何操作，它已断开媒体连接。
不能在 蓝牙网络连接 上执行任何操作，它已断开媒体连接。
不能在 以太网 上执行任何操作，它已断开媒体连接。

无线局域网适配器 本地连接* 1:

    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :

无线局域网适配器 本地连接* 2:

    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :

无线局域网适配器 WLAN:

    连接特定的 DNS 后缀 . . . . . :
    本地链接 IPv6 地址 . . . . . : fe80::36fb:45cb:d2df:2a8%20
    默认网关 . . . . . :

以太网适配器 VMware Network Adapter VMnet1:

    连接特定的 DNS 后缀 . . . . . :
    本地链接 IPv6 地址 . . . . . : fe80::14f4:5080:ae4b:62a6%4
    默认网关 . . . . . :

以太网适配器 VMware Network Adapter VMnet8:

    连接特定的 DNS 后缀 . . . . . :
    本地链接 IPv6 地址 . . . . . : fe80::b6db:daec:75b4:74ae%3
    默认网关 . . . . . :

以太网适配器 蓝牙网络连接:

    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :

以太网适配器 以太网:

    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :

```

图 1-4 ipconfig /release (释放指定适配器的 IPv4 地址)

```

C:\Users\zhaoy>ipconfig /renew

Windows IP 配置

不能在 本地连接* 1 上执行任何操作，它已断开媒体连接。
不能在 本地连接* 2 上执行任何操作，它已断开媒体连接。
不能在 蓝牙网络连接 上执行任何操作，它已断开媒体连接。
不能在 以太网 上执行任何操作，它已断开媒体连接。

无线局域网适配器 本地连接* 1:

    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :

无线局域网适配器 本地连接* 2:

    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :

无线局域网适配器 WLAN:

    连接特定的 DNS 后缀 . . . . . :
    本地链接 IPv6 地址 . . . . . : fe80::36fb:45cb:d2df:2a8%20
    IPv4 地址 . . . . . : 192.168.31.179
    子网掩码 . . . . . : 255.255.255.0
    默认网关 . . . . . : 192.168.31.1

以太网适配器 VMware Network Adapter VMnet1:

    连接特定的 DNS 后缀 . . . . . :
    本地链接 IPv6 地址 . . . . . : fe80::14f4:5080:ae4b:62a6%4
    IPv4 地址 . . . . . : 192.168.126.1
    子网掩码 . . . . . : 255.255.255.0
    默认网关 . . . . . :

以太网适配器 VMware Network Adapter VMnet8:

    连接特定的 DNS 后缀 . . . . . :
    本地链接 IPv6 地址 . . . . . : fe80::b6db:daec:75b4:74ae%3
    IPv4 地址 . . . . . : 192.168.72.1
    子网掩码 . . . . . : 255.255.255.0
    默认网关 . . . . . :

以太网适配器 蓝牙网络连接:

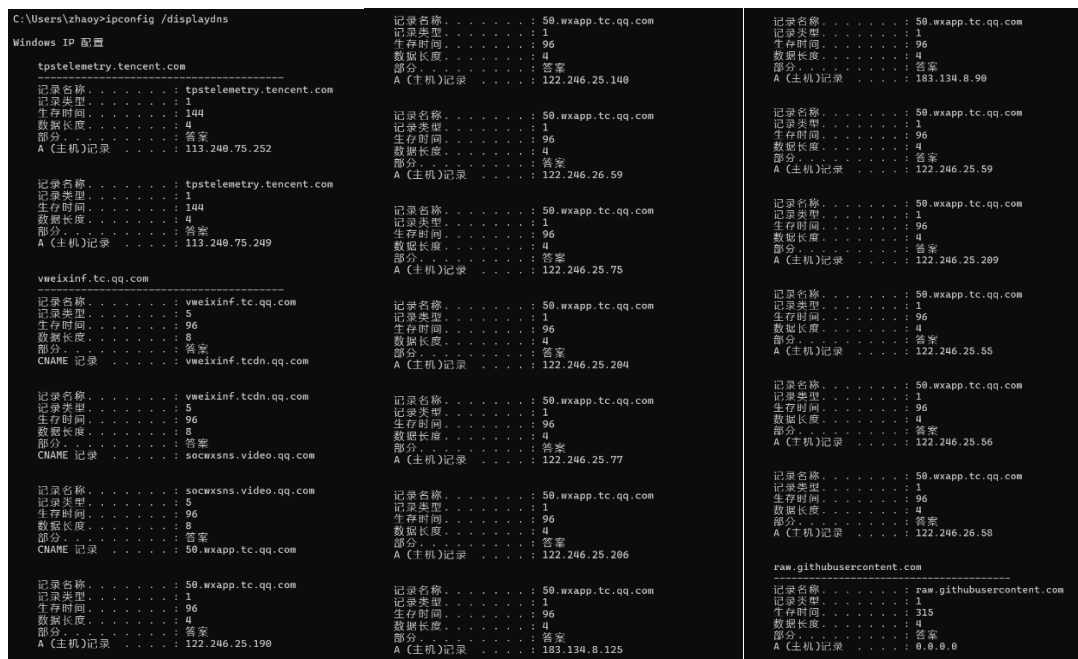
    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :

以太网适配器 以太网:

    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :

```

图 1-5 ipconfig /renew (更新指定适配器的 IPv4 地址)



(2) ping

在执行 ping 时，计算机会发送一个请求包给指定的 IP 地址或域名，并记录从发送到收到响应的的时间。如果目标主机能成功接收并回应请求，ping 会显示每次响应的时延。通过多次测试，可以计算网络连接的丢包率和延迟情况，帮助判断网络质量。

- i 参数：设置每个请求的 TTL（生存时间），限制数据包在网络中经过的跳数（设置数据包在网络中传输时所能经过的路由器数量上限）。
- n 参数：指定发送数据包的次数，用于多次测试连通性和稳定性。

```

C:\Users\zhaoy>ping /?

用法: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
        [-r count] [-s count] [[-j host-list] | [-k host-list]]
        [-w timeout] [-R] [-S srcaddr] [-c compartment] [-p]
        [-4] [-6] target_name

选项:
    -t                Ping 指定的主机, 直到停止。
                    若要查看统计信息并继续操作, 请键入 Ctrl+Break;
                    若要停止, 请键入 Ctrl+C。
    -a                将地址解析为主机名。
    -n count          要发送的回显请求数。
    -l size           发送缓冲区大小。
    -f               在数据包中设置“不分段”标记(仅适用于 IPv4)。
    -i TTL           生存时间。
    -v TOS            服务类型(仅适用于 IPv4。该设置已被弃用,
                    对 IP 标头中的服务类型字段没有任何
                    影响)。
    -r count          记录计数跃点的路由(仅适用于 IPv4)。
    -s count          计数跃点的时间戳(仅适用于 IPv4)。
    -j host-list      与主机列表一起使用的松散源路由(仅适用于 IPv4)。
    -k host-list      与主机列表一起使用的严格源路由(仅适用于 IPv4)。
    -w timeout        等待每次回复的超时时间(毫秒)。
    -R               同样使用路由标头测试反向路由(仅适用于 IPv6)。
                    根据 RFC 5095, 已弃用此路由标头。
                    如果使用此标头, 某些系统可能丢弃
                    回显请求。
    -S srcaddr        要使用的源地址。
    -c compartment    路由隔离舱标识符。
    -p               Ping Hyper-V 网络虚拟化提供程序地址。
    -4               强制使用 IPv4。
    -6               强制使用 IPv6。

```

图 1-8 ping /? (显示帮助信息)

```

C:\Users\zhaoy>ping baidu.com

正在 Ping baidu.com [110.242.68.66] 具有 32 字节的数据:
来自 110.242.68.66 的回复: 字节=32 时间=38ms TTL=51
来自 110.242.68.66 的回复: 字节=32 时间=38ms TTL=51
来自 110.242.68.66 的回复: 字节=32 时间=38ms TTL=51
来自 110.242.68.66 的回复: 字节=32 时间=38ms TTL=51

110.242.68.66 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 = 38ms, 最长 = 38ms, 平均 = 38ms

```

图 1-9 ping

```
C:\Users\zhaoy>ping -t qianqianzyk.top
```

```
正在 Ping qianqianzyk.top [47.96.78.173] 具有 32 字节的数据:
```

```
来自 47.96.78.173 的回复: 字节=32 时间=7ms TTL=54
来自 47.96.78.173 的回复: 字节=32 时间=7ms TTL=54
来自 47.96.78.173 的回复: 字节=32 时间=8ms TTL=54
来自 47.96.78.173 的回复: 字节=32 时间=7ms TTL=54
来自 47.96.78.173 的回复: 字节=32 时间=7ms TTL=54
来自 47.96.78.173 的回复: 字节=32 时间=7ms TTL=54
来自 47.96.78.173 的回复: 字节=32 时间=7ms TTL=54
来自 47.96.78.173 的回复: 字节=32 时间=7ms TTL=54
来自 47.96.78.173 的回复: 字节=32 时间=8ms TTL=54
来自 47.96.78.173 的回复: 字节=32 时间=8ms TTL=54
来自 47.96.78.173 的回复: 字节=32 时间=8ms TTL=54
来自 47.96.78.173 的回复: 字节=32 时间=8ms TTL=54
来自 47.96.78.173 的回复: 字节=32 时间=7ms TTL=54
```

```
47.96.78.173 的 Ping 统计信息:
```

```
数据包: 已发送 = 13, 已接收 = 13, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
```

```
最短 = 7ms, 最长 = 8ms, 平均 = 7ms
```

```
Control-C
```

```
^C
```

图 1-10 ping -t (持续 ping)

```
C:\Users\zhaoy>ping -n 6 qianqianzyk.top
```

```
正在 Ping qianqianzyk.top [47.96.78.173] 具有 32 字节的数据:
```

```
来自 47.96.78.173 的回复: 字节=32 时间=7ms TTL=54
来自 47.96.78.173 的回复: 字节=32 时间=7ms TTL=54
来自 47.96.78.173 的回复: 字节=32 时间=7ms TTL=54
来自 47.96.78.173 的回复: 字节=32 时间=8ms TTL=54
来自 47.96.78.173 的回复: 字节=32 时间=7ms TTL=54
来自 47.96.78.173 的回复: 字节=32 时间=8ms TTL=54
```

```
47.96.78.173 的 Ping 统计信息:
```

```
数据包: 已发送 = 6, 已接收 = 6, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
```

```
最短 = 7ms, 最长 = 8ms, 平均 = 7ms
```

图 1-11 ping -n (要发送的回显请求数)

(3) netstat

netstat 是一个用于查看计算机网络连接状态的命令行工具。它可以显示网络通信的详细信息,比如当前的网络连接、监听的端口、路由信息、以及不同网络协议(如 TCP、UDP)的统计数据。通过使用 netstat,能够了解系统的网络活动情况,并帮助诊断网络问题。netstat 常用于网络故障排查、监控网络流量、检测异常的网络连接,以及管理和优化网络配置。

```
C:\Users\zhaoy>netstat
```

活动连接

协议	本地地址	外部地址	状态
TCP	127.0.0.1:7890	ZYK_1218:49693	TIME_WAIT
TCP	127.0.0.1:7890	ZYK_1218:49758	ESTABLISHED
TCP	127.0.0.1:7890	ZYK_1218:49791	ESTABLISHED
TCP	127.0.0.1:7890	ZYK_1218:50092	ESTABLISHED
TCP	127.0.0.1:7890	ZYK_1218:50455	TIME_WAIT
TCP	127.0.0.1:7890	ZYK_1218:50632	TIME_WAIT
TCP	127.0.0.1:7890	ZYK_1218:50677	ESTABLISHED
TCP	127.0.0.1:7890	ZYK_1218:50710	ESTABLISHED
TCP	127.0.0.1:7890	ZYK_1218:50715	ESTABLISHED
TCP	127.0.0.1:7890	ZYK_1218:50720	TIME_WAIT
TCP	127.0.0.1:7890	ZYK_1218:50721	TIME_WAIT
TCP	127.0.0.1:7890	ZYK_1218:50723	ESTABLISHED
TCP	127.0.0.1:7890	ZYK_1218:50724	ESTABLISHED
TCP	127.0.0.1:7890	ZYK_1218:50739	ESTABLISHED
TCP	127.0.0.1:7890	ZYK_1218:50743	ESTABLISHED
TCP	127.0.0.1:7890	ZYK_1218:50745	TIME_WAIT
TCP	127.0.0.1:7890	ZYK_1218:50748	TIME_WAIT
TCP	127.0.0.1:7890	ZYK_1218:50752	TIME_WAIT
TCP	127.0.0.1:7890	ZYK_1218:50754	ESTABLISHED
TCP	127.0.0.1:7890	ZYK_1218:50759	TIME_WAIT
TCP	127.0.0.1:7890	ZYK_1218:50766	ESTABLISHED
TCP	127.0.0.1:7890	ZYK_1218:50771	ESTABLISHED
TCP	127.0.0.1:7890	ZYK_1218:50777	TIME_WAIT
TCP	127.0.0.1:7890	ZYK_1218:50785	ESTABLISHED
TCP	127.0.0.1:7890	ZYK_1218:50790	TIME_WAIT

图 1-12 netstat


```
C:\Users\zhaoy>netstat /?

显示协议统计信息和当前 TCP/IP 网络连接。

NETSTAT [-a] [-b] [-e] [-f] [-i] [-n] [-o] [-p proto] [-r] [-s] [-t] [-x] [-y] [interval]

-a          显示所有连接和侦听端口。
-b          显示在创建每个连接或侦听端口时涉及的
            可执行文件。在某些情况下，已知可执行文件托管
            多个独立的组件，此时会
            显示创建连接或侦听端口时
            涉及的组件序列。在此情况下，可执行文件的
            名称位于底部 [] 中，它调用的组件位于顶部，
            直至达到 TCP/IP。注意，此选项
            可能很耗时，并且可能因为你没有足够的
            权限而失败。
-e          显示以太网统计信息。此选项可以与 -s 选项
            结合使用。
-f          显示外部地址的完全限定
            域名(FQDN)。
-i          显示 TCP 连接在当前状态所花费的时间。
-n          以数字形式显示地址和端口号。
-o          显示拥有的与每个连接关联的进程 ID。
-p proto    显示 proto 指定的协议的连接；proto
            可以是下列任何一个：TCP、UDP、TCPv6 或 UDPv6。如果与 -s
            选项一起用来显示每个协议的统计信息，proto 可以是下列任何一个：
            IP、IPv6、ICMP、ICMPv6、TCP、TCPv6、UDP 或 UDPv6。
-q          显示所有连接、侦听端口和绑定的
            非侦听 TCP 端口。绑定的非侦听端口
            不一定与活动连接相关联。
-r          显示路由表。
-s          显示每个协议的统计信息。默认情况下，
            显示 IP、IPv6、ICMP、ICMPv6、TCP、TCPv6、UDP 和 UDPv6 的统计信息；
            -p 选项可用于指定默认的子网。
-t          显示当前连接卸载状态。
-x          显示 NetworkDirect 连接、侦听器 and 共享
            终结点。
-y          显示所有连接的 TCP 连接模板。
            无法与其他选项结合使用。
interval    重新显示选定的统计信息，各个显示间暂停的
            间隔秒数。按 CTRL+C 停止重新显示
            统计信息。如果省略，则 netstat 将打印当前的
            配置信息一次。
```

图 1-13 netstat /? (显示帮助信息)

C:\Users\zhaoy>netstat -a				TCP	127.0.0.1:7890	ZYK_1218:49791	ESTABLISHED
活动连接				TCP	127.0.0.1:7890	ZYK_1218:50771	ESTABLISHED
				TCP	127.0.0.1:7890	ZYK_1218:51030	TIME_WAIT
				TCP	127.0.0.1:7890	ZYK_1218:51150	ESTABLISHED
				TCP	127.0.0.1:7890	ZYK_1218:51167	ESTABLISHED
				TCP	127.0.0.1:7890	ZYK_1218:51184	ESTABLISHED
				TCP	127.0.0.1:7890	ZYK_1218:62376	ESTABLISHED
协议	本地地址	外部地址	状态	TCP	127.0.0.1:8440	ZYK_1218:0	LISTENING
TCP	0.0.0.0:135	ZYK_1218:0	LISTENING	TCP	127.0.0.1:8680	ZYK_1218:0	LISTENING
TCP	0.0.0.0:445	ZYK_1218:0	LISTENING	TCP	127.0.0.1:8884	ZYK_1218:0	LISTENING
TCP	0.0.0.0:902	ZYK_1218:0	LISTENING	TCP	127.0.0.1:9010	ZYK_1218:0	LISTENING
TCP	0.0.0.0:912	ZYK_1218:0	LISTENING	TCP	127.0.0.1:9010	ZYK_1218:60568	ESTABLISHED
TCP	0.0.0.0:3306	ZYK_1218:0	LISTENING	TCP	127.0.0.1:9080	ZYK_1218:0	LISTENING
TCP	0.0.0.0:5040	ZYK_1218:0	LISTENING	TCP	127.0.0.1:9100	ZYK_1218:0	LISTENING
TCP	0.0.0.0:7000	ZYK_1218:0	LISTENING	TCP	127.0.0.1:9100	ZYK_1218:60572	ESTABLISHED
TCP	0.0.0.0:7680	ZYK_1218:0	LISTENING	TCP	127.0.0.1:9180	ZYK_1218:0	LISTENING
TCP	0.0.0.0:7890	ZYK_1218:0	LISTENING	TCP	127.0.0.1:9210	ZYK_1218:0	LISTENING
TCP	0.0.0.0:8000	ZYK_1218:0	LISTENING	TCP	127.0.0.1:9210	ZYK_1218:51119	TIME_WAIT
TCP	0.0.0.0:11200	ZYK_1218:0	LISTENING	TCP	127.0.0.1:9210	ZYK_1218:51120	TIME_WAIT
TCP	0.0.0.0:16422	ZYK_1218:0	LISTENING	TCP	127.0.0.1:9210	ZYK_1218:51172	TIME_WAIT
TCP	0.0.0.0:33060	ZYK_1218:0	LISTENING	TCP	127.0.0.1:9210	ZYK_1218:51173	TIME_WAIT
TCP	0.0.0.0:49664	ZYK_1218:0	LISTENING	TCP	127.0.0.1:9210	ZYK_1218:51219	TIME_WAIT
TCP	0.0.0.0:49665	ZYK_1218:0	LISTENING	TCP	127.0.0.1:9210	ZYK_1218:51220	TIME_WAIT
TCP	0.0.0.0:49668	ZYK_1218:0	LISTENING	TCP	127.0.0.1:9210	ZYK_1218:51252	TIME_WAIT
TCP	0.0.0.0:49669	ZYK_1218:0	LISTENING	TCP	127.0.0.1:9210	ZYK_1218:51253	TIME_WAIT
TCP	0.0.0.0:49672	ZYK_1218:0	LISTENING	TCP	127.0.0.1:10000	ZYK_1218:0	LISTENING
TCP	0.0.0.0:49694	ZYK_1218:0	LISTENING	TCP	127.0.0.1:13013	ZYK_1218:0	LISTENING
TCP	127.0.0.1:4301	ZYK_1218:0	LISTENING	TCP	127.0.0.1:13016	ZYK_1218:0	LISTENING
TCP	127.0.0.1:4310	ZYK_1218:0	LISTENING	TCP	127.0.0.1:21440	ZYK_1218:0	LISTENING
TCP	127.0.0.1:5700	ZYK_1218:0	LISTENING	TCP	127.0.0.1:21441	ZYK_1218:0	LISTENING
TCP	127.0.0.1:7890	ZYK_1218:49758	ESTABLISHED	TCP	127.0.0.1:21446	ZYK_1218:0	LISTENING
TCP	127.0.0.1:7890	ZYK_1218:49791	ESTABLISHED	TCP	127.0.0.1:42950	ZYK_1218:0	LISTENING
TCP	127.0.0.1:7890	ZYK_1218:50771	ESTABLISHED				

图 1-14 netstat -a (显示所有连接和侦听端口)

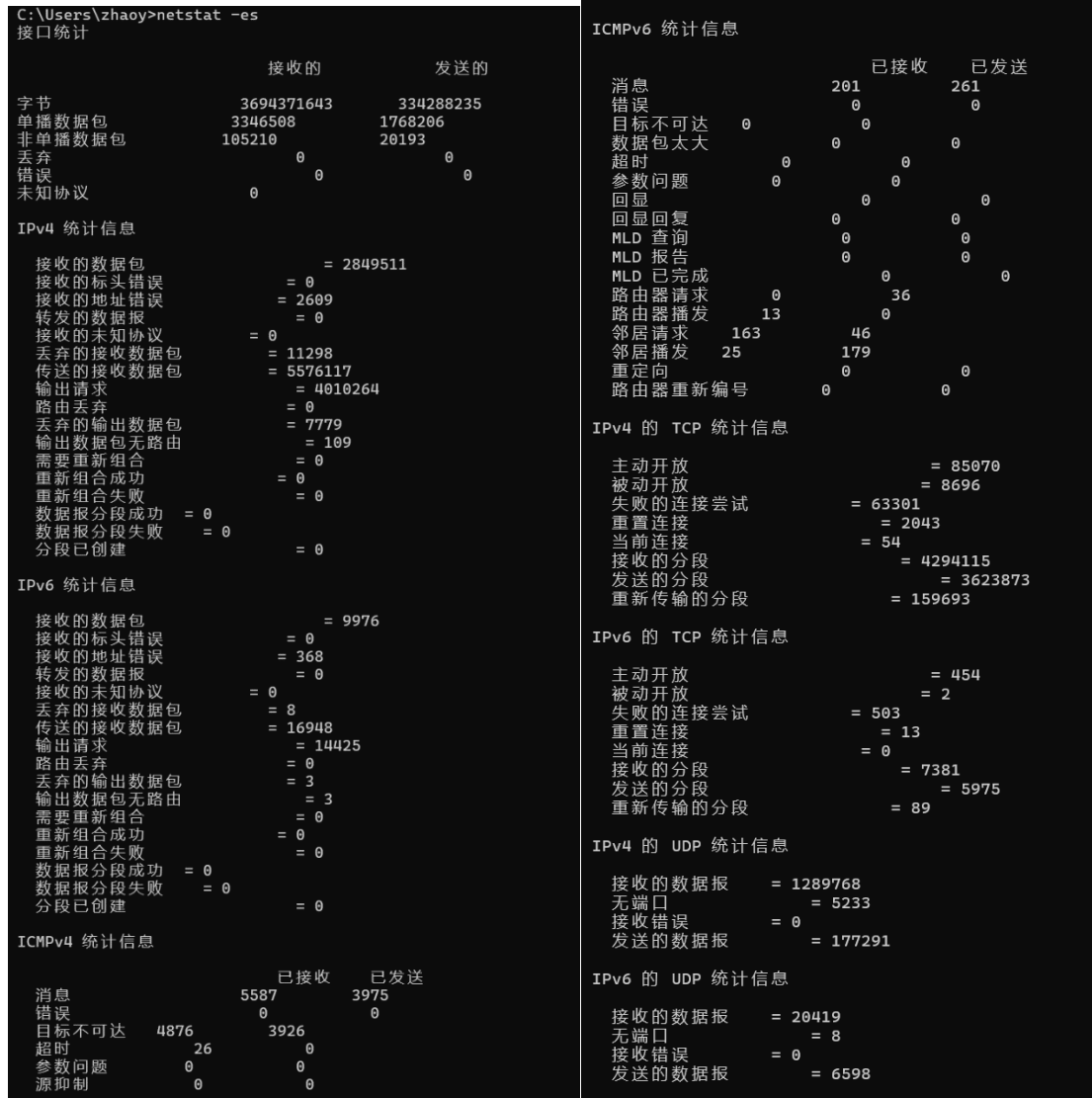


图 1-15 netstat -es (显示网络协议的统计信息)

C:\Windows\System32>netstat -vb

活动连接

协议	本地地址	外部地址	状态
TCP	127.0.0.1:7890 [clash-win64.exe]	ZYK_1218:49758	ESTABLISHED
TCP	127.0.0.1:7890 [clash-win64.exe]	ZYK_1218:49791	ESTABLISHED
TCP	127.0.0.1:7890 [clash-win64.exe]	ZYK_1218:51783	ESTABLISHED
TCP	127.0.0.1:7890 [clash-win64.exe]	ZYK_1218:51789	ESTABLISHED
TCP	127.0.0.1:7890	ZYK_1218:51874	TIME_WAIT
TCP	127.0.0.1:7890 [clash-win64.exe]	ZYK_1218:51970	ESTABLISHED
TCP	127.0.0.1:7890 [clash-win64.exe]	ZYK_1218:51980	ESTABLISHED
TCP	127.0.0.1:7890 [clash-win64.exe]	ZYK_1218:62376	ESTABLISHED
TCP	127.0.0.1:9010 [lghub_agent.exe]	ZYK_1218:60568	ESTABLISHED
TCP	127.0.0.1:9100 [lghub_updater.exe]	ZYK_1218:60572	ESTABLISHED
TCP	127.0.0.1:9210	ZYK_1218:51859	TIME_WAIT
TCP	127.0.0.1:9210	ZYK_1218:51860	TIME_WAIT
TCP	127.0.0.1:9210	ZYK_1218:51903	TIME_WAIT
TCP	127.0.0.1:9210	ZYK_1218:51904	TIME_WAIT
TCP	127.0.0.1:9210	ZYK_1218:51942	TIME_WAIT
TCP	127.0.0.1:9210	ZYK_1218:51943	TIME_WAIT
TCP	127.0.0.1:9210	ZYK_1218:52008	TIME_WAIT
TCP	127.0.0.1:9210	ZYK_1218:52009	TIME_WAIT
TCP	127.0.0.1:49684 [mysqld.exe]	ZYK_1218:49685	ESTABLISHED
TCP	127.0.0.1:49685 [mysqld.exe]	ZYK_1218:49684	ESTABLISHED
TCP	127.0.0.1:49686 [mysqld.exe]	ZYK_1218:49687	ESTABLISHED
TCP	127.0.0.1:49687 [mysqld.exe]	ZYK_1218:49686	ESTABLISHED
TCP	127.0.0.1:49728 [WUDFHost.exe]	ZYK_1218:49729	ESTABLISHED
TCP	127.0.0.1:49729 [WUDFHost.exe]	ZYK_1218:49728	ESTABLISHED
TCP	127.0.0.1:49730 [ipfsvc.exe]	ZYK_1218:49731	ESTABLISHED
TCP	127.0.0.1:49731 [ipfsvc.exe]	ZYK_1218:49730	ESTABLISHED
TCP	127.0.0.1:49733 [WUDFHost.exe]	ZYK_1218:49734	ESTABLISHED
TCP	127.0.0.1:49734 [WUDFHost.exe]	ZYK_1218:49733	ESTABLISHED
TCP	127.0.0.1:49735 [NVDisplay.Container.exe]	ZYK_1218:49736	ESTABLISHED
TCP	127.0.0.1:49736 [NVDisplay.Container.exe]	ZYK_1218:49735	ESTABLISHED
TCP	127.0.0.1:49758 [msedge.exe]	ZYK_1218:7890	ESTABLISHED
TCP	127.0.0.1:49791 [msedge.exe]	ZYK_1218:7890	ESTABLISHED
TCP	127.0.0.1:49933 [KuGou.exe]	ZYK_1218:49934	ESTABLISHED
TCP	127.0.0.1:49934 [KuGou.exe]	ZYK_1218:49933	ESTABLISHED
TCP	127.0.0.1:49935 [KuGou.exe]	ZYK_1218:49936	ESTABLISHED
TCP	127.0.0.1:49936 [KuGou.exe]	ZYK_1218:49935	ESTABLISHED
TCP	127.0.0.1:51783 [msedge.exe]	ZYK_1218:7890	ESTABLISHED
TCP	127.0.0.1:51789	ZYK_1218:7890	ESTABLISHED

图 1-16 netstat -vb (显示每个连接或监听端口的可执行文件信息和模块信息)

```
C:\Users\zhaoy>netstat -r
=====
接口列表
 6...dc 46 28 12 fc 97 .....Microsoft Wi-Fi Direct Virtual Adapter
11...de 46 28 12 fc 96 .....Microsoft Wi-Fi Direct Virtual Adapter #2
20...dc 46 28 12 fc 96 .....Intel(R) Wi-Fi 6 AX201 160MHz
 4...00 50 56 c0 00 01 .....VMware Virtual Ethernet Adapter for VMnet1
 3...00 50 56 c0 00 08 .....VMware Virtual Ethernet Adapter for VMnet8
 7...dc 46 28 12 fc 9a .....Bluetooth Device (Personal Area Network)
19...04 bf 1b 6f 94 6c .....Realtek PCIe GbE Family Controller
 1.....Software Loopback Interface 1
=====

IPv4 路由表
=====
活动路由:
网络目标      网络掩码      网关      接口      跃点数
0.0.0.0        0.0.0.0        192.168.31.1  192.168.31.179  30
127.0.0.0      255.0.0.0
127.0.0.1      255.255.255.255  在链路上      127.0.0.1  331
127.255.255.255 255.255.255.255 在链路上      127.0.0.1  331
192.168.31.0    255.255.255.0    在链路上      192.168.31.179  286
192.168.31.179  255.255.255.255 在链路上      192.168.31.179  286
192.168.31.255  255.255.255.255 在链路上      192.168.31.179  286
192.168.72.0    255.255.255.0    在链路上      192.168.72.1  291
192.168.72.1    255.255.255.255 在链路上      192.168.72.1  291
192.168.72.255  255.255.255.255 在链路上      192.168.72.1  291
192.168.126.0   255.255.255.0    在链路上      192.168.126.1  291
192.168.126.1   255.255.255.255 在链路上      192.168.126.1  291
192.168.126.255 255.255.255.255 在链路上      192.168.126.1  291
224.0.0.0       240.0.0.0         在链路上      127.0.0.1  331
224.0.0.0       240.0.0.0         在链路上      192.168.31.179  286
224.0.0.0       240.0.0.0         在链路上      192.168.72.1  291
224.0.0.0       240.0.0.0         在链路上      192.168.126.1  291
255.255.255.255 255.255.255.255 在链路上      127.0.0.1  331
255.255.255.255 255.255.255.255 在链路上      192.168.31.179  286
255.255.255.255 255.255.255.255 在链路上      192.168.72.1  291
255.255.255.255 255.255.255.255 在链路上      192.168.126.1  291
=====
永久路由:
无

IPv6 路由表
=====
活动路由:
接口跃点数网络目标      网关
1 331 ::1/128 在链路上
20 286 fe80::/64 在链路上
3 291 fe80::/64 在链路上
4 291 fe80::/64 在链路上
4 291 fe80::14f4:5080:ae4b:62a6/128 在链路上
20 286 fe80::36fb:45cb:d2df:2a8/128 在链路上
3 291 fe80::b6db:daec:75b4:74ae/128 在链路上
1 331 ff00::/8 在链路上
20 286 ff00::/8 在链路上
3 291 ff00::/8 在链路上
4 291 ff00::/8 在链路上
=====
永久路由:
无
```

图 1-17 netstat -r (显示路由表)

```
C:\Users\zhaoy>netstat -f
```

活动连接

协议	本地地址	外部地址	状态
TCP	127.0.0.1:7890	ZYK_1218:49758	ESTABLISHED
TCP	127.0.0.1:7890	ZYK_1218:49791	ESTABLISHED
TCP	127.0.0.1:7890	ZYK_1218:51980	TIME_WAIT
TCP	127.0.0.1:7890	ZYK_1218:52229	ESTABLISHED
TCP	127.0.0.1:7890	ZYK_1218:52272	ESTABLISHED
TCP	127.0.0.1:7890	ZYK_1218:62376	ESTABLISHED
TCP	127.0.0.1:9010	ZYK_1218:60568	ESTABLISHED
TCP	127.0.0.1:9100	ZYK_1218:60572	ESTABLISHED
TCP	127.0.0.1:9210	ZYK_1218:52237	TIME_WAIT
TCP	127.0.0.1:9210	ZYK_1218:52238	TIME_WAIT
TCP	127.0.0.1:9210	ZYK_1218:52281	TIME_WAIT
TCP	127.0.0.1:9210	ZYK_1218:52282	TIME_WAIT
TCP	127.0.0.1:9210	ZYK_1218:52314	TIME_WAIT
TCP	127.0.0.1:9210	ZYK_1218:52315	TIME_WAIT
TCP	127.0.0.1:9210	ZYK_1218:52344	TIME_WAIT
TCP	127.0.0.1:9210	ZYK_1218:52345	TIME_WAIT
TCP	127.0.0.1:49684	ZYK_1218:49685	ESTABLISHED
TCP	127.0.0.1:49685	ZYK_1218:49684	ESTABLISHED
TCP	127.0.0.1:49686	ZYK_1218:49687	ESTABLISHED
TCP	127.0.0.1:49687	ZYK_1218:49686	ESTABLISHED
TCP	127.0.0.1:49728	ZYK_1218:49729	ESTABLISHED
TCP	127.0.0.1:49729	ZYK_1218:49728	ESTABLISHED
TCP	127.0.0.1:49730	ZYK_1218:49731	ESTABLISHED
TCP	127.0.0.1:49731	ZYK_1218:49730	ESTABLISHED
TCP	127.0.0.1:49733	ZYK_1218:49734	ESTABLISHED
TCP	127.0.0.1:49734	ZYK_1218:49733	ESTABLISHED
TCP	127.0.0.1:49735	ZYK_1218:49736	ESTABLISHED
TCP	127.0.0.1:49736	ZYK_1218:49735	ESTABLISHED
TCP	127.0.0.1:49758	ZYK_1218:7890	ESTABLISHED
TCP	127.0.0.1:49791	ZYK_1218:7890	ESTABLISHED
TCP	127.0.0.1:49933	ZYK_1218:49934	ESTABLISHED
TCP	127.0.0.1:49934	ZYK_1218:49933	ESTABLISHED
TCP	127.0.0.1:49935	ZYK_1218:49936	ESTABLISHED
TCP	127.0.0.1:49936	ZYK_1218:49935	ESTABLISHED
TCP	127.0.0.1:52229	ZYK_1218:7890	ESTABLISHED
TCP	127.0.0.1:52262	ZYK_1218:7890	TIME_WAIT
TCP	127.0.0.1:52272	ZYK_1218:7890	ESTABLISHED
TCP	127.0.0.1:52373	ZYK_1218:6463	SYN_SENT
TCP	127.0.0.1:52374	ZYK_1218:28194	SYN_SENT
TCP	127.0.0.1:56996	ZYK_1218:62366	ESTABLISHED
TCP	127.0.0.1:56997	ZYK_1218:62366	ESTABLISHED
TCP	127.0.0.1:60568	ZYK_1218:9010	ESTABLISHED
TCP	127.0.0.1:60572	ZYK_1218:9100	ESTABLISHED
TCP	127.0.0.1:61556	ZYK_1218:61557	ESTABLISHED
TCP	127.0.0.1:61557	ZYK_1218:61556	ESTABLISHED
TCP	127.0.0.1:62366	ZYK_1218:56996	ESTABLISHED
TCP	127.0.0.1:62366	ZYK_1218:56997	ESTABLISHED
TCP	127.0.0.1:62376	ZYK_1218:7890	ESTABLISHED

图 1-18 netstat -f (显示外部地址的完全限定域名)

(4) tracert

tracert 是一个用来追踪数据包在网络上传输路径的命令行工具。它能够显示数据包从本地计算机到达目标地址过程中所经过的每个路由器节点。通过逐步增加数据包的 TTL (生存时间) 值, tracert 可以记录和显示数据包每次跳跃的路由器 IP 地址以及每一跳所耗费的时间。

```
C:\Users\zhaoy>tracert /?
```

用法: tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout] [-R] [-S srcaddr] [-4] [-6] target_name

选项:

-d	不将地址解析成主机名。
-h maximum_hops	搜索目标的最大跃点数。
-j host-list	与主机列表一起的松散源路由(仅适用于 IPv4)。
-w timeout	等待每个回复的超时时间(以毫秒为单位)。
-R	跟踪往返行程路径(仅适用于 IPv6)。
-S srcaddr	要使用的源地址(仅适用于 IPv6)。
-4	强制使用 IPv4。
-6	强制使用 IPv6。

图 1-19 tracert /?

```
C:\Users\zhaoy>tracert qianqianzyk.top
```

通过最多 30 个跃点跟踪
到 qianqianzyk.top [47.96.78.173] 的路由:

1	<1 毫秒	<1 毫秒	<1 毫秒	xiaoqiang [192.168.31.1]
2	2 ms	2 ms	2 ms	115.200.0.1
3	2 ms	*	*	61.130.126.170
4	*	*	*	请求超时。
5	8 ms	8 ms	8 ms	115.233.23.142
6	5 ms	5 ms	5 ms	122.224.214.70
7	*	*	*	请求超时。
8	*	*	*	请求超时。
9	*	*	*	请求超时。
10	*	*	*	请求超时。
11	7 ms	7 ms	9 ms	47.96.78.173

跟踪完成。

图 1-20 tracert

```
C:\Users\zhaoy>tracert -h 3 qianqianzyk.top
```

通过最多 3 个跃点跟踪
到 qianqianzyk.top [47.96.78.173] 的路由:

1	2 ms	2 ms	4 ms	xiaoqiang [192.168.31.1]
2	126 ms	4 ms	23 ms	115.200.0.1
3	*	*	*	请求超时。

跟踪完成。

图 1-21 tracert -h (搜索目标的最大跃点数)

(5) arp

arp 是一个用于管理和查看地址解析协议 (ARP) 缓存的命令行工具。ARP 的

主要作用是在局域网中将 IP 地址转换为物理地址（MAC 地址），以便计算机能够通过数据链路层进行通信。使用 arp 命令，可以查看当前的 ARP 缓存，添加或删除条目。

```
C:\Users\zhaoy>arp
```

显示和修改地址解析协议(ARP)使用的“IP 到物理”地址转换表。

```
ARP -s inet_addr eth_addr [if_addr]
```

```
ARP -d inet_addr [if_addr]
```

```
ARP -a [inet_addr] [-N if_addr] [-v]
```

-a	通过询问当前协议数据，显示当前 ARP 项。如果指定 inet_addr，则只显示指定计算机的 IP 地址和物理地址。如果不止一个网络接口使用 ARP，则显示每个 ARP 表的项。
-g	与 -a 相同。
-v	在详细模式下显示当前 ARP 项。所有无效项和环回接口上的项都将显示。
inet_addr	指定 Internet 地址。
-N if_addr	显示 if_addr 指定的网络接口的 ARP 项。
-d	删除 inet_addr 指定的主机。inet_addr 可以是通配符 *，以删除所有主机。
-s	添加主机并且将 Internet 地址 inet_addr 与物理地址 eth_addr 相关联。物理地址是用连字符分隔的 6 个十六进制字节。该项是永久的。
eth_addr	指定物理地址。
if_addr	如果存在，此项指定地址转换表应修改的接口的 Internet 地址。如果不存在，则使用第一个适用的接口。

示例：

```
> arp -s 157.55.85.212 00-aa-00-62-c6-09.... 添加静态项。  
> arp -a .... 显示 ARP 表。
```

图 1-22 arp

```
C:\Users\zhaoy>arp -a

接口: 192.168.72.1 --- 0x3
Internet 地址      物理地址      类型
192.168.72.254     00-50-56-e5-43-92 动态
192.168.72.255     ff-ff-ff-ff-ff-ff 静态
224.0.0.22         01-00-5e-00-00-16 静态
224.0.0.251        01-00-5e-00-00-fb 静态
224.0.0.252        01-00-5e-00-00-fc 静态
238.238.238.238    01-00-5e-6e-ee-ee 静态
239.238.237.236    01-00-5e-6e-ed-ec 静态
239.255.255.250    01-00-5e-7f-ff-fa 静态
255.255.255.255    ff-ff-ff-ff-ff-ff 静态

接口: 192.168.126.1 --- 0x4
Internet 地址      物理地址      类型
192.168.126.254    00-50-56-e7-8e-ef 动态
192.168.126.255    ff-ff-ff-ff-ff-ff 静态
224.0.0.22         01-00-5e-00-00-16 静态
224.0.0.251        01-00-5e-00-00-fb 静态
224.0.0.252        01-00-5e-00-00-fc 静态
238.238.238.238    01-00-5e-6e-ee-ee 静态
239.238.237.236    01-00-5e-6e-ed-ec 静态
239.255.255.250    01-00-5e-7f-ff-fa 静态
255.255.255.255    ff-ff-ff-ff-ff-ff 静态

接口: 192.168.31.179 --- 0x14
Internet 地址      物理地址      类型
192.168.31.1       44-f7-70-14-87-94 动态
192.168.31.93      62-1c-a0-71-c6-ab 动态
192.168.31.255     ff-ff-ff-ff-ff-ff 静态
224.0.0.22         01-00-5e-00-00-16 静态
224.0.0.251        01-00-5e-00-00-fb 静态
224.0.0.252        01-00-5e-00-00-fc 静态
238.238.238.238    01-00-5e-6e-ee-ee 静态
239.238.237.236    01-00-5e-6e-ed-ec 静态
239.255.255.250    01-00-5e-7f-ff-fa 静态
255.255.255.255    ff-ff-ff-ff-ff-ff 静态
```

图 1-23 arp -a (显示当前 ARP 项)

(6) telnet

telnet 是一种应用层协议。它允许用户通过网络连接到远程主机，并与其进行交互，像是在本地计算机上一样执行命令。

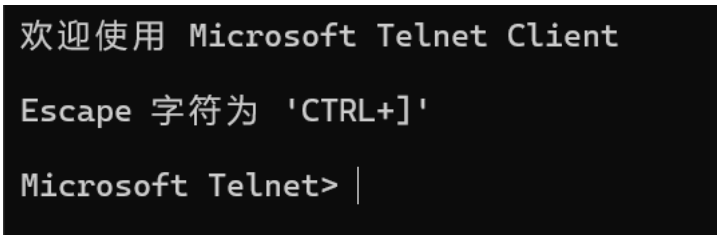


图 1-24 telnet

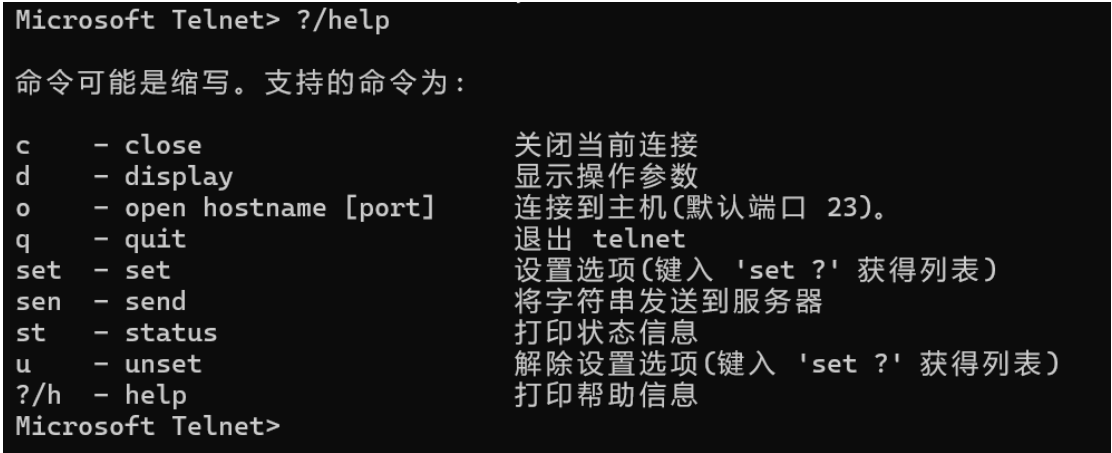


图 1-25 telnet /?

2.2 交换机与路由器

2.2.1 交换机配置与交换机 VLAN 设计

(1) 交换机命令与初始化配置

命令	作用	模式（要求）
enable(en)	进入特权模式	普通模式
config terminal(conf t)	进入全局配置模式	特权模式
hostname <hostname>	修改当前用户名	全局配置模式
vlan <n>	创建编号为 n 的局域网	全局配置模式
interface fastethernet x/y (int fa x/y)	进入某个快速以太网接口的配置模式	全局配置模式
switchport mode access	设置端口为访问模式	接口配置模式
switchport access vlan <n>	给某个接口指定子网	接口配置模式
exit	退出当前模式	任意配置模式
show vlan	查看当前局域网与接口信息	普通模式
end	退出配置模式并返回特权模式	任意配置模式

表 2-1 交换机常用命令

(2) 交换机拓扑图

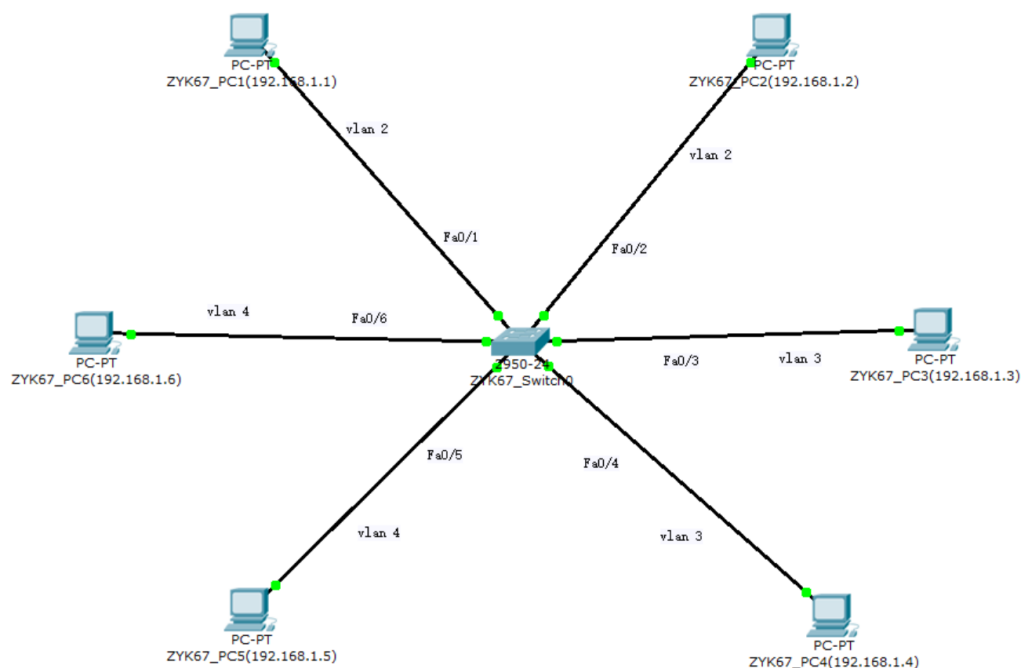


图 2-1 交换机拓扑图

(3) 交换机配置过程（命令行）

```
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#hostname ZYK67
ZYK67(config)#vlan 2
ZYK67(config-vlan)#name VLAN2
ZYK67(config-vlan)#exit
ZYK67(config)#vlan 3
ZYK67(config-vlan)#name VLAN3
ZYK67(config-vlan)#exit
ZYK67(config)#vlan 4
ZYK67(config-vlan)#name VLAN4
ZYK67(config-vlan)#exit
ZYK67(config)#interface range fastethernet 0/1 - 2
ZYK67(config-if-range)#switchport access vlan 2
ZYK67(config-if-range)#exit
ZYK67(config)#interface range fastethernet 0/3 - 4
ZYK67(config-if-range)#switchport access vlan 3
ZYK67(config-if-range)#exit
ZYK67(config)#interface range fastethernet 0/5 - 6
ZYK67(config-if-range)#switchport access vlan 4
ZYK67(config-if-range)#exit
```

图 2-2 交换机配置（命令行）

命令说明（选择一个局域网的配置过程）

- en 进入特权模式，获取更高的权限来执行配置命令
- conf d 进入全局配置模式，允许对设备进行全面的配置
- hostname ZYK67 将交换机的主机名设置为“ZYK67”，方便设备的识别与管理
- vlan 2 创建编号为 2 的 VLAN
- name VLAN2 为 VLAN 2 设置名称为“VLAN2”
- interface range fastethernet 0/1 - 2 进入接口范围配置模式，对 fastethernet 0/1 到 0/2 的接口进行批量配置
- switchport access vlan 2 将选定的接口分配到 VLAN 2，接口被设置为访问模式，可以将接入的设备加入 VLAN 2

(4) 交换机配置结果

```
ZYK67(config)#exit
ZYK67#
%SYS-5-CONFIG_I: Configured from console by console
show vlan
```

VLAN	Name	Status	Ports
1	default	active	Fa0/7, Fa0/8, Fa0/9, Fa0/10 Fa0/11, Fa0/12, Fa0/13, Fa0/14 Fa0/15, Fa0/16, Fa0/17, Fa0/18 Fa0/19, Fa0/20, Fa0/21, Fa0/22 Fa0/23, Fa0/24
2	VLAN2	active	Fa0/1, Fa0/2
3	VLAN3	active	Fa0/3, Fa0/4
4	VLAN4	active	Fa0/5, Fa0/6
1002	fddi-default	act/unsup	
1003	token-ring-default	act/unsup	
1004	fddinet-default	act/unsup	
1005	trnet-default	act/unsup	

VLAN	Type	SAID	MTU	Parent	RingNo	BridgeNo	Stp	BrdgMode	Trans1	Trans2
1	enet	100001	1500	-	-	-	-	-	0	0
2	enet	100002	1500	-	-	-	-	-	0	0
3	enet	100003	1500	-	-	-	-	-	0	0
4	enet	100004	1500	-	-	-	-	-	0	0

--More--

图 2-3 交换机配置结果

(5) 验证交换机是否配置成功

1. PING

```
PC>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time=29ms TTL=128
Reply from 192.168.1.2: bytes=32 time=16ms TTL=128
Reply from 192.168.1.2: bytes=32 time=17ms TTL=128
Reply from 192.168.1.2: bytes=32 time=13ms TTL=128

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 13ms, Maximum = 29ms, Average = 18ms

PC>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 192.168.1.5

Pinging 192.168.1.5 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.1.5:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

图 2-4 使用 PING 测试

交换机配置完成后测试主机 ZYK67_PC1 是否可以 ping 通三个局域网中的主机（vlan 2/3/4），发现只能 ping 通与自己在同一个局域网下的主机，也就是 vlan 2 下的主机。

2. 自定义数据报

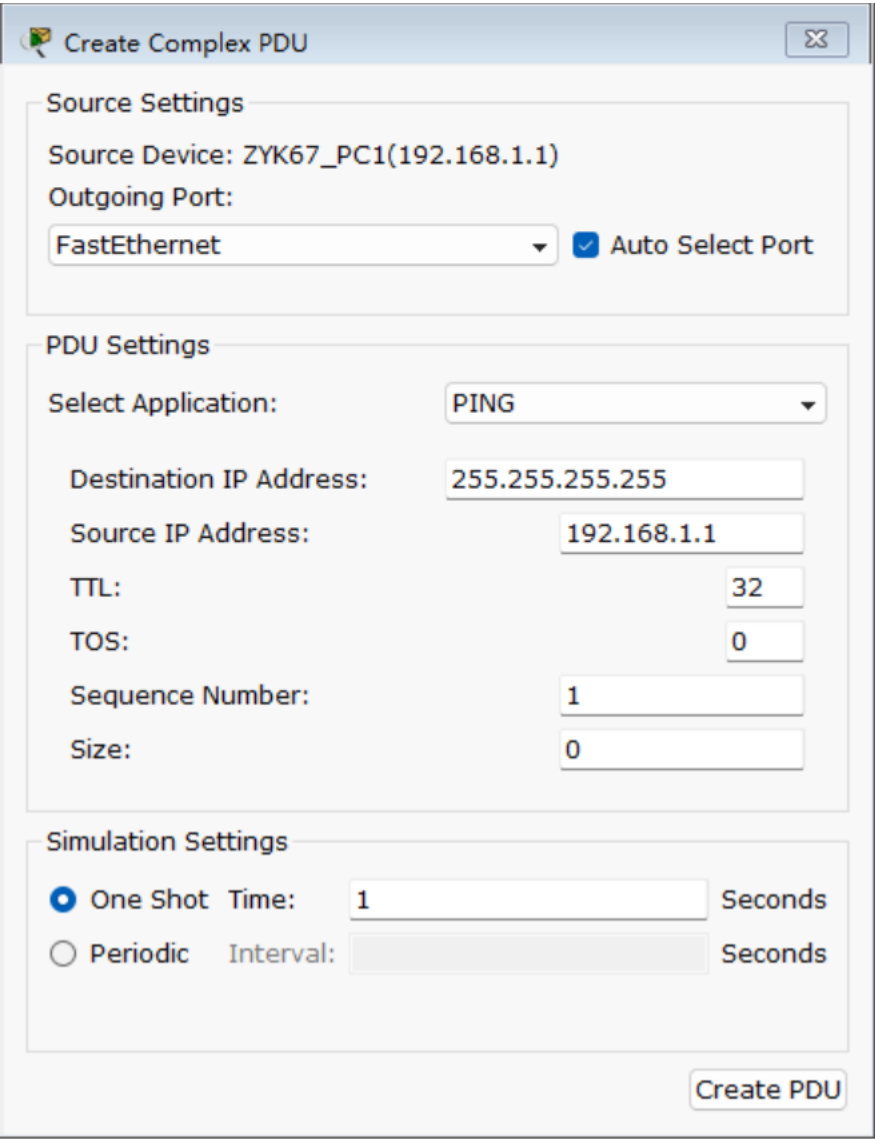


图 2-5 创建复杂 PDU

1.001	ZYK67_PC1(192.168.1.1)	ZYK67_Switch0	ICMP	
1.002	ZYK67_Switch0	ZYK67_PC2(192.168.1.2)	ICMP	
1.003	ZYK67_PC2(192.168.1.2)	ZYK67_Switch0	ICMP	
1.004	ZYK67_Switch0	ZYK67_PC1(192.168.1.1)	ICMP	

图 2-6 数据报交付

可以看到自定义的数据报通过广播的方式发出，只有处在同意局域网下的 ZYK67_PC2 主机接收到了该数据报，其他局域网下的主机均未收到数据报。

2.2.2 路由配置与静态路由设计

(1) 路由器配置指令

命令	作用	模式（要求）
interface Serial x/y(int sex/y)	进入从某个快速串口的配置模式	全局配置模式
ip address <ipv4	设置该接口的 ipv4 地址	接口配置模式

address>(ip add <ipv4 address>)		
no shut(no sh)	开启该接口	接口配置模式
ip route <network> <netmask> <gateway>	设置路由表	全局配置模式
show ip route	显示本机路由表	普通模式
clock rate <bps>	设置该接口的时钟频率	接口配置模式

表 2-2 路由器配置常用命令

(2) 静态路由拓扑图

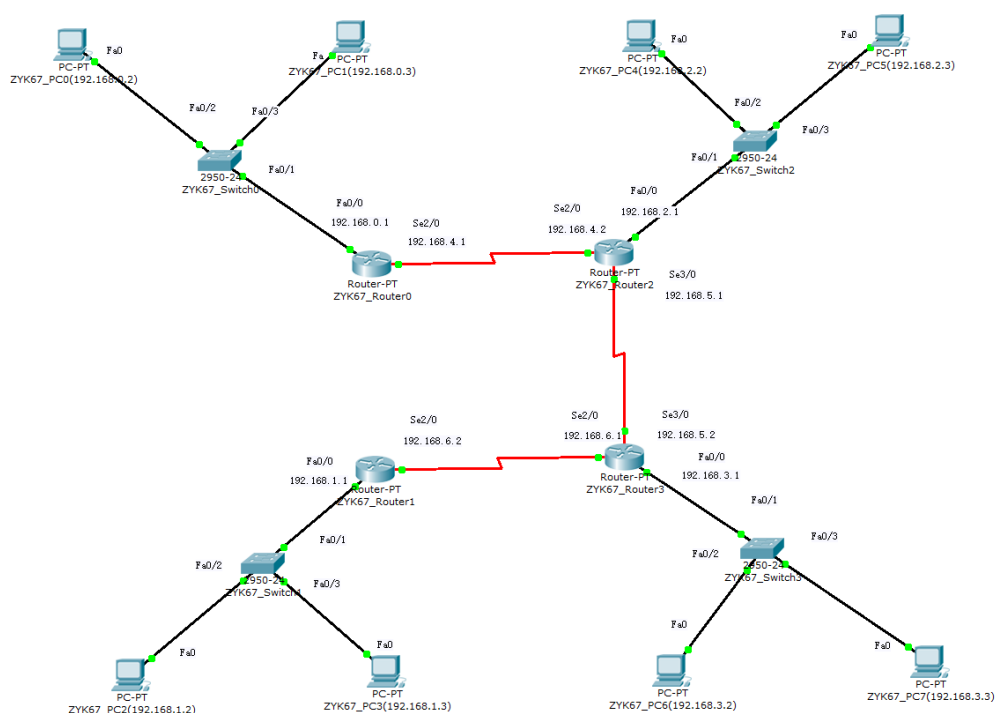


图 2-7 静态路由拓扑图

(3) 路由器配置过程（命令行）

```

Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname ZYK67
ZYK67(config)#interface FastEthernet 0/0
ZYK67(config-if)#ip address 192.168.0.1 255.255.255.0
ZYK67(config-if)#no shutdown

ZYK67(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

ZYK67(config-if)#exit
ZYK67(config)#interface Serial 2/0
ZYK67(config-if)#ip address 192.168.4.1 255.255.255.0
ZYK67(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial2/0, changed state to down
ZYK67(config-if)#exit
ZYK67(config)#end
ZYK67#
%SYS-5-CONFIG_I: Configured from console by console

ZYK67#enable
ZYK67#con t
% Ambiguous command: "con t"
ZYK67#conf t
Enter configuration commands, one per line. End with CNTL/Z.
ZYK67(config)#ip route 192.168.2.0 255.255.255.0 192.168.4.2
ZYK67(config)#ip route 192.168.3.0 255.255.255.0 192.168.4.2
ZYK67(config)#ip route 192.168.1.0 255.255.255.0 192.168.4.2
ZYK67(config)#exit

```

图 2-8 路由器配置（命令行）

```

ZYK67>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, 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, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.0.0/24 is directly connected, FastEthernet0/0
S    192.168.1.0/24 [1/0] via 192.168.4.2
S    192.168.2.0/24 [1/0] via 192.168.4.2
S    192.168.3.0/24 [1/0] via 192.168.4.2
C    192.168.4.0/24 is directly connected, Serial2/0

```

图 2-9 该路由器配置好后的路由表
(其它三台路由器也进行类似的配置)

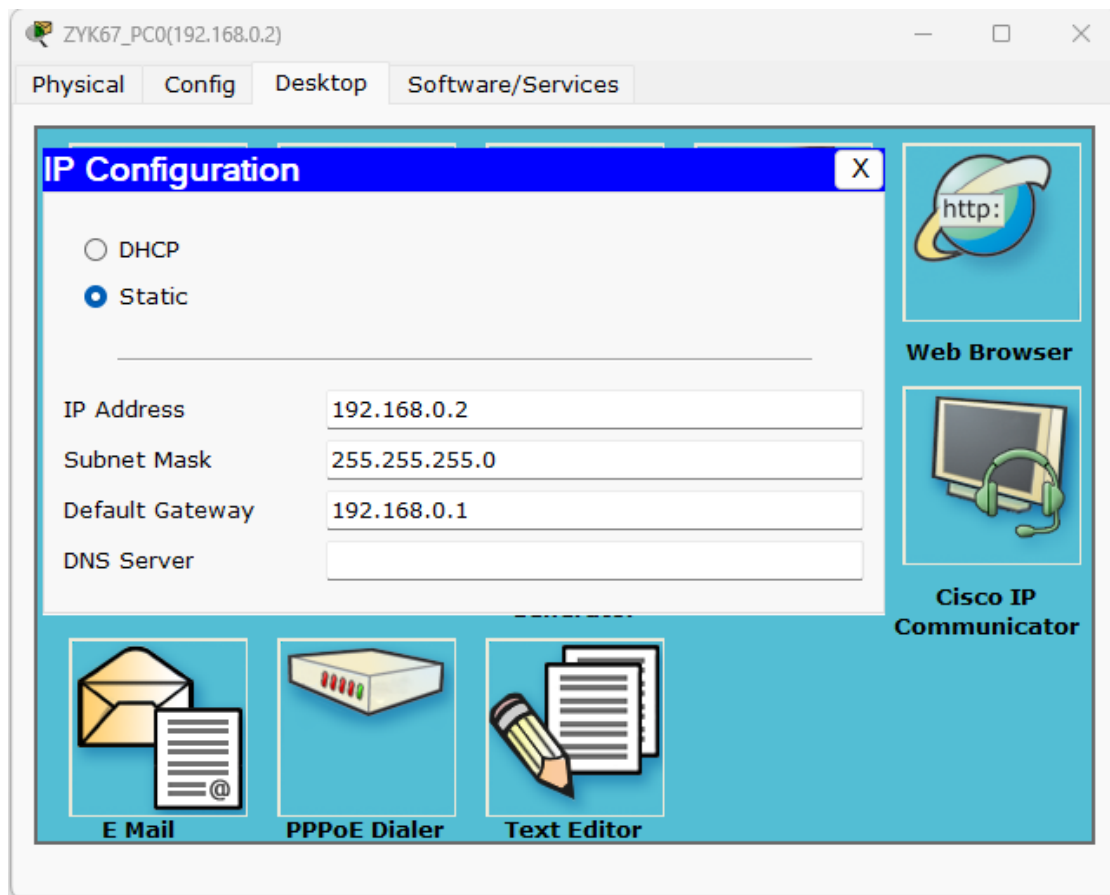


图 2-10 主机的配置

(其它主机也进行类似的配置)

(4) 验证路由器是否配置成功

```
PC>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:

Request timed out.
Reply from 192.168.1.3: bytes=32 time=219ms TTL=124
Reply from 192.168.1.3: bytes=32 time=202ms TTL=124
Reply from 192.168.1.3: bytes=32 time=218ms TTL=124

Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 202ms, Maximum = 219ms, Average = 213ms
```

图 2-11 主机 ZYK67_PC1(192.168.0.3) PING 主机 ZYK67_PC3(192.168.1.3)

第一个 ICMP 请求未收到响应，后续 ICMP 回应表明网络通信正常，目的主机 192.168.1.3 收到了 ICMP 请求，并成功返回了回应。

分析可知第一个数据包丢失可能是由于初始 ARP 表项未建立引起的。后续数据包正常回复说明网络路径整体可用，且静态路由配置无误。


```
PC>tracert 192.168.1.3

Tracing route to 192.168.1.3 over a maximum of 30 hops:

  0  58 ms    49 ms    65 ms    192.168.0.1
  1  98 ms    91 ms    76 ms    192.168.4.2
  2 108 ms   128 ms   123 ms   192.168.5.2
  3 141 ms   151 ms   111 ms   192.168.6.2
  4 190 ms   208 ms   217 ms   192.168.1.3

Trace complete.
```

图 2-12 主机 ZYK67_PC1(192.168.0.3) TRACERT 主机 ZYK67_PC3(192.168.1.3)

分析可知每一跳的 IP 地址对应网络路径中的路由器或主机。路径共 5 跳，符合实验拓扑结构：源主机与目标主机之间通过 4 个路由器转发数据。

2.2.3 动态路由设计

(1) 路由器配置指令 (RIP)

命令	作用	模式 (要求)
route rip	开启 rip 动态路由并进入配置模式	全局配置模式
no auto-summary(no auto)	关闭路由器自动汇总功能	rip 配置模式
network <network>	添加直连网段	rip 配置模式
version <1 2>	配置 rip 版本	rip 配置模式
show ip protocols	显示路由协议与配置	普通模式

表 2-3 路由器配置常用命令 (RIP)

(2) RIP 动态路由配置拓扑图

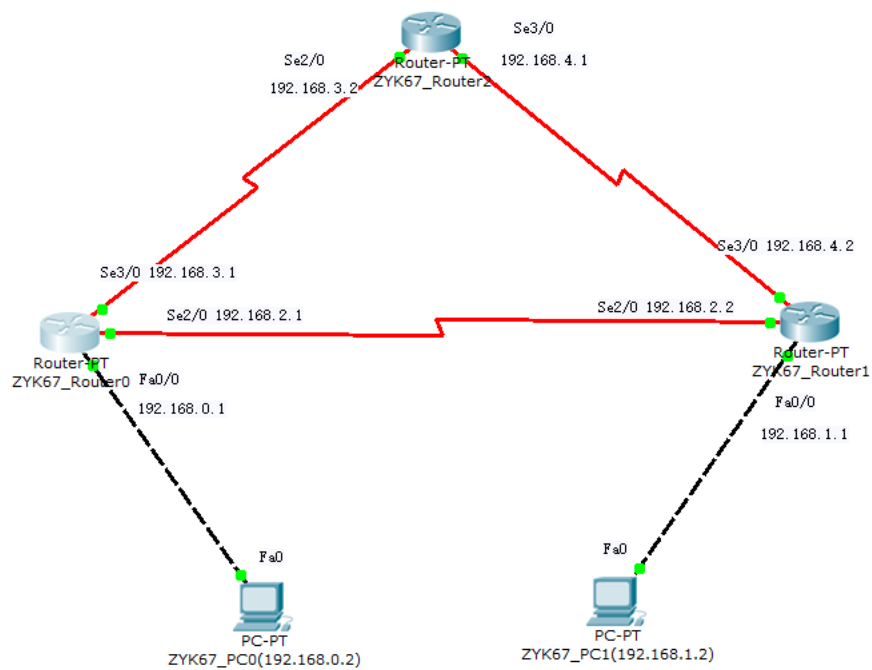


图 2-13 RIP 动态路由配置拓扑图

(3) RIP 动态路由配置过程（命令行）

```

Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname ZYK67
ZYK67(config)#int fa 0/0
ZYK67(config-if)#ip add 192.168.0.1 255.255.255.0
ZYK67(config-if)#no sh

%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state t
o up

ZYK67(config-if)#
ZYK67(config-if)#int se 2/0
ZYK67(config-if)#ip add 192.168.2.1 255.255.255.0
ZYK67(config-if)#
ZYK67(config-if)#exit
ZYK67(config)#interface Serial2/0
ZYK67(config-if)#clock rate 4000000
ZYK67(config-if)#no sh

%LINK-5-CHANGED: Interface Serial2/0, changed state to down
ZYK67(config-if)#
ZYK67(config-if)#exit
ZYK67(config)#int se 3/0
ZYK67(config-if)#ip add 192.168.3.1 255.255.255.0
ZYK67(config-if)#clock rate 4000000
ZYK67(config-if)#no sh

%LINK-5-CHANGED: Interface Serial3/0, changed state to down
ZYK67(config-if)#exit
ZYK67(config)#end
ZYK67#
%SYS-5-CONFIG_I: Configured from console by console

```

图 2-14 路由器端口配置

```

ZYK67#conf t
Enter configuration commands, one per line. End with CNTL/Z.
ZYK67(config)#route rip
ZYK67(config-router)#version 2
ZYK67(config-router)#no auto
ZYK67(config-router)#network 192.168.0.0
ZYK67(config-router)#network 192.168.3.0
ZYK67(config-router)#network 192.168.2.0
ZYK67(config-router)#end
ZYK67#
%SYS-5-CONFIG_I: Configured from console by console

```

图 2-15 路由器启用 RIP 动态路由模式

```

ZYK67>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, 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, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.0.0/24 is directly connected, FastEthernet0/0
C    192.168.2.0/24 is directly connected, Serial2/0
C    192.168.3.0/24 is directly connected, Serial3/0

```

图 2-16 路由器配置好后的路由表

```

ZYK67>show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 9 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Redistributing: rip
  Default version control: send version 2, receive 2
    Interface          Send  Recv  Triggered RIP  Key-chain
    FastEthernet0/0      2      2
    Serial3/0            2      2
    Serial2/0            2      2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    192.168.0.0
    192.168.2.0
    192.168.3.0
  Passive Interface(s):
  Routing Information Sources:
    Gateway         Distance      Last Update
  Distance: (default is 120)

```

图 2-17 路由器配置好后的路由协议信息

（其它两台路由器也进行类似的配置）

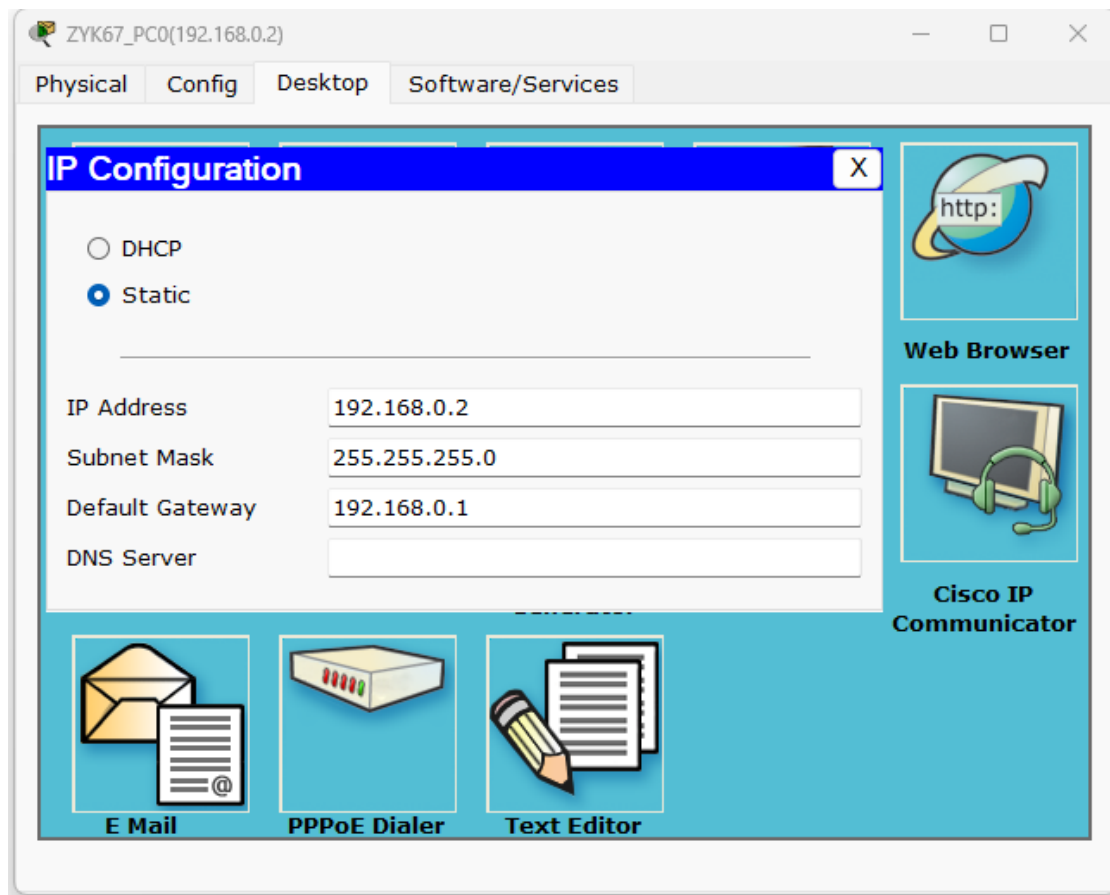


图 2-18 主机的配置

(另一台主机也进行类似的配置)

(4) 验证 RIP 动态路由是否配置成功

```

PC>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Request timed out.
Reply from 192.168.1.2: bytes=32 time=89ms TTL=126
Reply from 192.168.1.2: bytes=32 time=77ms TTL=126
Reply from 192.168.1.2: bytes=32 time=76ms TTL=126

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 76ms, Maximum = 89ms, Average = 80ms

PC>tracert 192.168.1.2

Tracing route to 192.168.1.2 over a maximum of 30 hops:

  0  28 ms    32 ms    30 ms    192.168.0.1
  1  62 ms    47 ms    31 ms    192.168.2.2
  2  93 ms    93 ms    108 ms   192.168.1.2

Trace complete.

```

图 2-19 主机 0 PING TRACERT 主机 1

可以看到主机 1 已经成功接收到了 ICMP 请求。由于配置了 RIP 动态路由，故选择最短路径（跳数最少的路径）。从 tracert 可以看出选择了 192.168.0.1->192.168.2.2->192.168.1.2（共三跳），而不是 192.168.0.1->192.168.3.2->192.168.4.2->192.168.1.2（共四跳），同样也可以验证 RIP 动态路由配置成功了。

```
ZYK67>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, 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, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.0.0/24 is directly connected, FastEthernet0/0
R    192.168.1.0/24 [120/1] via 192.168.2.2, 00:00:14, Serial2/0
C    192.168.2.0/24 is directly connected, Serial2/0
C    192.168.3.0/24 is directly connected, Serial3/0
R    192.168.4.0/24 [120/1] via 192.168.3.2, 00:00:09, Serial3/0
                  [120/1] via 192.168.2.2, 00:00:14, Serial2/0
```

图 2-20 路由器配置好后的路由表（更新后）

```
ZYK67>show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 19 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Redistributing: rip
  Default version control: send version 2, receive 2
    Interface          Send  Recv  Triggered RIP  Key-chain
    FastEthernet0/0      2      2
    Serial3/0            2      2
    Serial2/0            2      2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    192.168.0.0
    192.168.2.0
    192.168.3.0
  Passive Interface(s):
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.2.2           120          00:00:25
    192.168.3.2           120          00:00:19
```

图 2-21 路由器配置好后的路由协议信息（更新后）

从新增的记录可以看到，到达 192.168.4.0/24 该网段有两个路径，也是 RIP 动态路由更新路由表后的结果。

2.2.4 网络集成设计 (RIP、OSPF 混合动态路由)

(1) 路由器配置指令 (OSPF)

命令	作用	模式（要求）
----	----	--------

router ospf <process-id>	配置一个 ospf 路由进程并且进入配置模式	全局配置模式
network <network> <wildcard-mask> area <n>	定义运行 ospf 的接口以及这些接口的区域 id	ospf 配置模式
redistribute rip metric 200 subnets	将 rip 网络的路由重发布到 ospf 的网络中	ospf 配置模式
redistribute ospf 1 metric 10	将 ospf 网络的路由重发布到 rip 的网络中	rip 配置模式

表 2-4 路由器配置指令（OSPF）

(2) RIP、OSPF 混合动态路由配置拓扑图

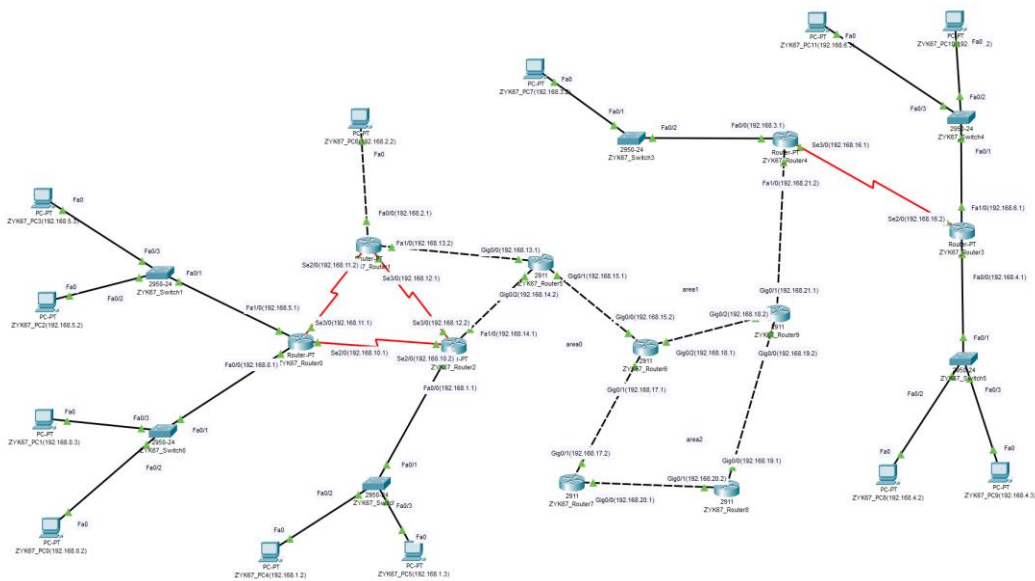


图 2-22 RIP、OSPF 混合动态路由配置拓扑图

(3) 混合动态路由配置过程（命令行）

```

Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname ZYK67
ZYK67(config)#int fa 0/0
ZYK67(config-if)#ip add 192.168.0.1 255.255.255.0
ZYK67(config-if)#no sh

ZYK67(config-if)#
ZYK67(config-if)#int fa 0/1
%Invalid interface type and number
ZYK67(config)#int fa 0/1
%Invalid interface type and number
ZYK67(config)#exit
ZYK67#conf t
Enter configuration commands, one per line. End with CNTL/Z.
ZYK67(config)#int fa 1/0
ZYK67(config-if)#exit
ZYK67(config)#interface FastEthernet1/0
ZYK67(config-if)#ip add 192.168.5.1 255.255.255.0
ZYK67(config-if)#no sh

ZYK67(config-if)#exit
ZYK67(config)#int se 2/0
ZYK67(config-if)#ip add 192.168.10.1 255.255.255.0
ZYK67(config-if)#no sh

%LINK-5-CHANGED: Interface Serial2/0, changed state to down
ZYK67(config-if)#exit
ZYK67(config)#int se 3/0
ZYK67(config-if)#ip add 192.168.11.1 255.255.255.0
ZYK67(config-if)#no sh

%LINK-5-CHANGED: Interface Serial3/0, changed state to down
ZYK67(config-if)#exit
ZYK67(config)#exit
ZYK67#

```

图 2-23 RIP 动态路由端口配置

```

ZYK67>en
ZYK67#conf t
Enter configuration commands, one per line. End with CNTL/Z.
ZYK67(config)#route rip
ZYK67(config-router)#network 192.168.0.0
ZYK67(config-router)#network 192.168.5.0
ZYK67(config-router)#network 192.168.11.0
ZYK67(config-router)#network 192.168.10.0
ZYK67(config-router)#version 2
ZYK67(config-router)#no auto
ZYK67(config-router)#end
ZYK67#
%SYS-5-CONFIG_I: Configured from console by console

```

图 2-24 RIP 动态路由模式配置


```

Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname ZYK67
ZYK67(config)#int gig 0/0
ZYK67(config-if)#ip add 192.168.15.2 255.255.255.0
ZYK67(config-if)#no sh

ZYK67(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up

ZYK67(config-if)#exit
ZYK67(config)#int gig 0/1
ZYK67(config-if)#ip add 192.168.17.1 255.255.255.0
ZYK67(config-if)#no sh

ZYK67(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up

ZYK67(config-if)#exit
ZYK67(config)#int gig 0/2
ZYK67(config-if)#ip add 192.168.10.1 255.255.255.0
ZYK67(config-if)#no sh

ZYK67(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/2, changed state to up

```

图 2-25 OSPF 动态路由端口配置

```

ZYK67(config)#route ospf 1
ZYK67(config-router)#network 192.168.15.0 0.0.0.255 area 0
ZYK67(config-router)#network 192.168.17.0
00:56:46: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.15.1 on GigabitEthernet0/0 from LOADING
to FULL, Loading Done

% Incomplete command.
ZYK67(config-router)#network 192.168.17.0 0.0.0.255 area 0
ZYK67(config-router)#network 192.168.18.0 0.0.0.255 area 1
ZYK67(config-router)#end
ZYK67#
%SYS-5-CONFIG_I: Configured from console by console

```

图 2-26 OSPF 动态路由模式配置

```

Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname ZYK67
ZYK67(config)#int gig 0/0
ZYK67(config-if)#ip add 192.168.13.1 255.255.255.0
ZYK67(config-if)#no sh

ZYK67(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up

ZYK67(config-if)#exit
ZYK67(config)#int gig 0/2
ZYK67(config-if)#ip add 192.168.14.2 255.255.255.0
ZYK67(config-if)#no sh

ZYK67(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/2, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2, changed state to up

ZYK67(config-if)#exit
ZYK67(config)#int gig 0/1
ZYK67(config-if)#ip add 192.168.15.1 255.255.255.0
ZYK67(config-if)#no sh

ZYK67(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up

ZYK67(config-if)#exit
ZYK67(config)#

```

图 2-27 混合路由器端口配置

```

ZYK67(config)#route rip
ZYK67(config-router)#version 2
ZYK67(config-router)#no auto
ZYK67(config-router)#network 192.168.13.0
ZYK67(config-router)#network 192.168.14.0
ZYK67(config-router)#end
ZYK67#
%SYS-5-CONFIG_I: Configured from console by console

ZYK67#conf t
Enter configuration commands, one per line. End with CNTL/Z.
ZYK67(config)#route ospf 1
ZYK67(config-router)#network 192.168.15.0 0.0.0.255 area 0
ZYK67(config-router)#end
ZYK67#
%SYS-5-CONFIG_I: Configured from console by console

```

图 2-28 混合路由器路由模式配置

```

ZYK67>en
ZYK67#conf t
Enter configuration commands, one per line. End with CNTL/Z.
ZYK67(config)#route ospf 1
ZYK67(config-router)#redistribute rip metric 200 subnets
ZYK67(config-router)#exit
ZYK67(config)#route rip
ZYK67(config-router)#redistribute ospf 1 metric 10
ZYK67(config-router)#exit
ZYK67(config)#

```

图 2-29 混合路由器跨规则模式配置

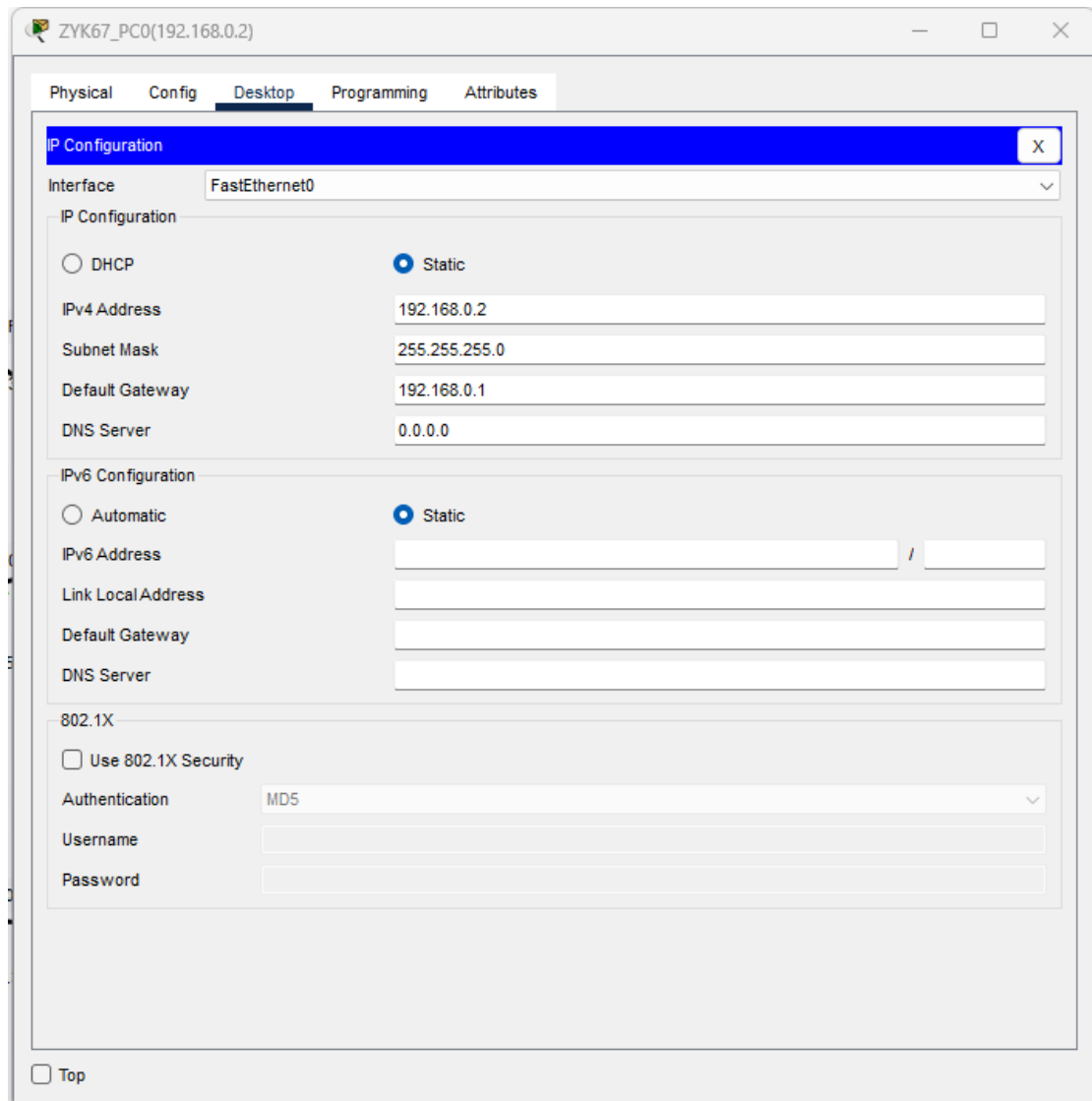


图 2-30 主机配置

(其它路由器，主机也作类似配置)

(4) 验证混合动态路由是否配置成功

```
C:\>ping 192.168.6.3

Pinging 192.168.6.3 with 32 bytes of data:

Reply from 192.168.6.3: bytes=32 time=27ms TTL=119
Reply from 192.168.6.3: bytes=32 time=21ms TTL=121
Reply from 192.168.6.3: bytes=32 time=24ms TTL=119
Reply from 192.168.6.3: bytes=32 time=26ms TTL=121

Ping statistics for 192.168.6.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 21ms, Maximum = 27ms, Average = 24ms

C:\>tracert 192.168.6.3

Tracing route to 192.168.6.3 over a maximum of 30 hops:

  0  0 ms    0 ms    0 ms    192.168.0.1
  1  13 ms   0 ms    0 ms    192.168.11.2
  2  1 ms    12 ms   0 ms    192.168.13.1
  3  0 ms    0 ms    0 ms    192.168.15.2
  4  0 ms    7 ms    11 ms   192.168.18.2
  5  1 ms    0 ms    0 ms    192.168.20.2
  6  0 ms    0 ms    0 ms    192.168.18.2
  7  1 ms    6 ms    6 ms    192.168.21.2

Trace complete.
```

图 2-31 主机 0 PING TRACERT 主机 11

```

ZYK67>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, 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, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.0.0/24 is directly connected, FastEthernet0/0
R    192.168.1.0/24 [120/1] via 192.168.10.2, 00:00:24, Serial2/0
R    192.168.2.0/24 [120/1] via 192.168.11.2, 00:00:25, Serial3/0
R    192.168.3.0/24 [120/11] via 192.168.10.2, 00:00:24, Serial2/0
        [120/11] via 192.168.11.2, 00:00:25, Serial3/0
R    192.168.4.0/24 [120/11] via 192.168.10.2, 00:00:24, Serial2/0
        [120/11] via 192.168.11.2, 00:00:25, Serial3/0
C    192.168.5.0/24 is directly connected, FastEthernet1/0
R    192.168.6.0/24 [120/11] via 192.168.11.2, 00:00:25, Serial3/0
        [120/11] via 192.168.10.2, 00:00:24, Serial2/0
C    192.168.10.0/24 is directly connected, Serial2/0
C    192.168.11.0/24 is directly connected, Serial3/0
R    192.168.12.0/24 [120/1] via 192.168.11.2, 00:00:25, Serial3/0
        [120/1] via 192.168.10.2, 00:00:24, Serial2/0
R    192.168.13.0/24 [120/1] via 192.168.11.2, 00:00:25, Serial3/0
R    192.168.14.0/24 [120/1] via 192.168.10.2, 00:00:24, Serial2/0
R    192.168.15.0/24 [120/11] via 192.168.11.2, 00:00:25, Serial3/0
        [120/11] via 192.168.10.2, 00:00:24, Serial2/0
R    192.168.16.0/24 [120/11] via 192.168.11.2, 00:00:25, Serial3/0
        [120/11] via 192.168.10.2, 00:00:24, Serial2/0
R    192.168.17.0/24 [120/11] via 192.168.11.2, 00:00:25, Serial3/0
        [120/11] via 192.168.10.2, 00:00:24, Serial2/0
R    192.168.18.0/24 [120/11] via 192.168.10.2, 00:00:24, Serial2/0
        [120/11] via 192.168.11.2, 00:00:25, Serial3/0
R    192.168.19.0/24 [120/11] via 192.168.11.2, 00:00:25, Serial3/0
        [120/11] via 192.168.10.2, 00:00:24, Serial2/0
R    192.168.20.0/24 [120/11] via 192.168.11.2, 00:00:25, Serial3/0
        [120/11] via 192.168.10.2, 00:00:24, Serial2/0
R    192.168.21.0/24 [120/11] via 192.168.10.2, 00:00:24, Serial2/0
        [120/11] via 192.168.11.2, 00:00:25, Serial3/0

```

图 2-32 RIP 路由表

```

ZYK67>show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, 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, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

O E2 192.168.0.0/24 [110/200] via 192.168.15.1, 00:18:03, GigabitEthernet0/0
O E2 192.168.1.0/24 [110/200] via 192.168.15.1, 00:11:26, GigabitEthernet0/0
O E2 192.168.2.0/24 [110/200] via 192.168.15.1, 00:18:03, GigabitEthernet0/0
O E2 192.168.3.0/24 [110/200] via 192.168.18.2, 00:06:19, GigabitEthernet0/2
    [110/200] via 192.168.17.2, 00:05:43, GigabitEthernet0/1
O E2 192.168.4.0/24 [110/200] via 192.168.18.2, 00:06:19, GigabitEthernet0/2
    [110/200] via 192.168.17.2, 00:05:43, GigabitEthernet0/1
O E2 192.168.5.0/24 [110/200] via 192.168.15.1, 00:18:03, GigabitEthernet0/0
O E2 192.168.6.0/24 [110/200] via 192.168.18.2, 00:06:19, GigabitEthernet0/2
    [110/200] via 192.168.17.2, 00:05:43, GigabitEthernet0/1
O E2 192.168.10.0/24 [110/200] via 192.168.15.1, 00:18:03, GigabitEthernet0/0
O E2 192.168.11.0/24 [110/200] via 192.168.15.1, 00:18:03, GigabitEthernet0/0
O E2 192.168.12.0/24 [110/200] via 192.168.15.1, 00:18:03, GigabitEthernet0/0
O E2 192.168.13.0/24 [110/200] via 192.168.15.1, 00:18:03, GigabitEthernet0/0
O E2 192.168.14.0/24 [110/200] via 192.168.15.1, 00:18:03, GigabitEthernet0/0
    192.168.15.0/24 is variably subnetted, 2 subnets, 2 masks
C      192.168.15.0/24 is directly connected, GigabitEthernet0/0
L      192.168.15.2/32 is directly connected, GigabitEthernet0/0
O E2 192.168.16.0/24 [110/200] via 192.168.18.2, 00:06:19, GigabitEthernet0/2
    [110/200] via 192.168.17.2, 00:05:43, GigabitEthernet0/1
    192.168.17.0/24 is variably subnetted, 2 subnets, 2 masks
C      192.168.17.0/24 is directly connected, GigabitEthernet0/1
L      192.168.17.1/32 is directly connected, GigabitEthernet0/1
    192.168.18.0/24 is variably subnetted, 2 subnets, 2 masks
C      192.168.18.0/24 is directly connected, GigabitEthernet0/2
L      192.168.18.1/32 is directly connected, GigabitEthernet0/2
O IA 192.168.19.0/24 [110/3] via 192.168.17.2, 00:17:53, GigabitEthernet0/1
O IA 192.168.20.0/24 [110/2] via 192.168.17.2, 00:17:53, GigabitEthernet0/1
O E2 192.168.21.0/24 [110/200] via 192.168.18.2, 00:06:19, GigabitEthernet0/2
    [110/200] via 192.168.17.2, 00:05:43, GigabitEthernet0/1

```

图 2-33 OSPF 路由器路由表

```

ZYK67>show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, 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, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

R    192.168.0.0/24 [120/2] via 192.168.13.2, 00:00:06, GigabitEthernet0/0
      [120/2] via 192.168.14.1, 00:00:03, GigabitEthernet0/2
R    192.168.1.0/24 [120/1] via 192.168.14.1, 00:00:03, GigabitEthernet0/2
R    192.168.2.0/24 [120/1] via 192.168.13.2, 00:00:06, GigabitEthernet0/0
O E2 192.168.3.0/24 [110/200] via 192.168.15.2, 00:04:20, GigabitEthernet0/1
O E2 192.168.4.0/24 [110/200] via 192.168.15.2, 00:04:20, GigabitEthernet0/1
R    192.168.5.0/24 [120/2] via 192.168.13.2, 00:00:06, GigabitEthernet0/0
      [120/2] via 192.168.14.1, 00:00:03, GigabitEthernet0/2
O E2 192.168.6.0/24 [110/200] via 192.168.15.2, 00:04:20, GigabitEthernet0/1
R    192.168.10.0/24 [120/1] via 192.168.14.1, 00:00:03, GigabitEthernet0/2
R    192.168.11.0/24 [120/1] via 192.168.13.2, 00:00:06, GigabitEthernet0/0
R    192.168.12.0/24 [120/1] via 192.168.13.2, 00:00:06, GigabitEthernet0/0
      [120/1] via 192.168.14.1, 00:00:03, GigabitEthernet0/2
      192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.13.0/24 is directly connected, GigabitEthernet0/0
L    192.168.13.1/32 is directly connected, GigabitEthernet0/0
      192.168.14.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.14.0/24 is directly connected, GigabitEthernet0/2
L    192.168.14.2/32 is directly connected, GigabitEthernet0/2
      192.168.15.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.15.0/24 is directly connected, GigabitEthernet0/1
L    192.168.15.1/32 is directly connected, GigabitEthernet0/1
O E2 192.168.16.0/24 [110/200] via 192.168.15.2, 00:04:20, GigabitEthernet0/1
O    192.168.17.0/24 [110/2] via 192.168.15.2, 00:16:40, GigabitEthernet0/1
O IA 192.168.18.0/24 [110/2] via 192.168.15.2, 00:09:07, GigabitEthernet0/1
O IA 192.168.19.0/24 [110/4] via 192.168.15.2, 00:16:30, GigabitEthernet0/1
O IA 192.168.20.0/24 [110/3] via 192.168.15.2, 00:16:30, GigabitEthernet0/1
O E2 192.168.21.0/24 [110/200] via 192.168.15.2, 00:04:20, GigabitEthernet0/1

```

图 2-34 混合路由器路由表

可以看到主机 0 已经 PING 到了主机 11，同时路由器的路由表也进行了更新，也就是说我们的混合动态路由配置成功。

2.3 网络编程

数据报详细介绍

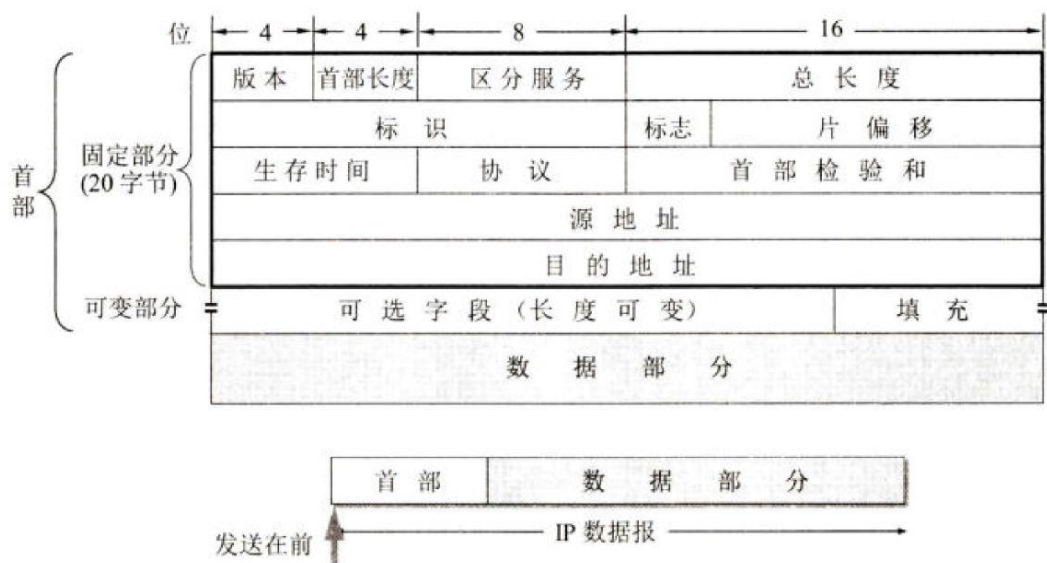


图 2-35 IP 数据报格式

1. 版本：占 4bit，通信双方使用的版本必须一致。对于 IPv4，字段的值是 4。
2. 首部长度：占 4bit，首部长度说明首部有多少个 4 字节。
3. 区分服务：占 6bit，最初被定义为服务类型字段，实际上并未使用。
4. 总长度：占 16bit，定义了报文总长，包含首部和数据，单位为字节。
5. 标识：占 16bit，这个字段主要被用来唯一地标识一个报文的所有分片。
6. 标志：占 3bit，用于控制和识别分片，它们是：
 - 位 0：保留，必须为 0；
 - 位 1：禁止分片（DF），当 DF=0 时才允许分片；
 - 位 2：更多分片（MF），MF=1 代表后面还有分片，MF=0 代表已经是最后一个分片。
7. 片偏移：占 13bit，指明了每个分片相对于原始报文开头的偏移量，以 8 字节作单位。
8. 生存时间：占 8bit。是一个跳数计数器：报文经过的每个路由器都将此字段减 1，当此字段等于 0 时，报文不再向下一跳传送并被丢弃。
9. 协议：占 8bit，这个字段定义了该报文数据区使用的协议。
10. 首部检验和：占 16bit。该检验和字段只对首部查错，不包括数据部分。
11. 源地址：占 32bit，表示源主机的 IPv4 地址。
12. 目的地址：占 32bit，表示目标主机的 IPv4 地址。
13. 可选字段：占 0-320bit，用于控制和测试。

我自定义了一个运输层协议号 222，格式如下

自定义标识（2 字节）	数据长度（2 字节）
数据	

图 2-36 自定义运输层协议格式

2.3.1 开发环境及关键配置

操作系统：Windows11

编程语言：C++

开发软件：Clion

主要使用的库：

1. **winsock2.h**：Windows 网络编程的核心头文件，提供了对 Winsock API 的访问。Winsock 是 Microsoft 对 BSD 套接字 API 的实现，使得开发人员能够在 Windows 系统上进行网络通信操作。该头文件包括了创建和管理套接字、连接、数据传输、设置选项等所需的函数和结构体定义。
2. **ws2tcpip.h**：该头文件包含了 Winsock 2 API 的扩展，用于支持 Internet 协议族的高级功能。它为 IPv4 和 IPv6 地址、主机名解析、DNS 查询、套接字选项等提供了更多功能。

注意：创建原始套接字需要管理员权限。

2.3.2 设计思路

1. 网络通信模型：程序采用原始套接字（Raw Socket）进行通信，允许用户自定义协议和 IP 包的构造和解析。原始套接字允许直接操作 IP 层数据包，而不需要操作高层协议（如 TCP/UDP）的封装。

2. 关键功能：发送自定义数据包（构造一个带有自定义协议的 IP 包并发送）；接收并解析自定义协议的数据包（监听并接收经过本机网络接口的数据包，解析其中自定义协议的数据）

3. 详细设计

发送自定义数据包

创建原始套接字： 使用 `socket(AF_INET, SOCK_RAW, IPPROTO_RAW)` 创建一个原始套接字，并指定协议为 `IPPROTO_RAW`。这样可以自己构造 IP 包的各个部分，而不依赖内核的协议栈。

设置套接字选项： 使用 `setsockopt()` 设置 `IP_HDRINCL` 选项，告诉操作系统将由程序手动构造 IP 头部。

构造 IP 头部： 自定义了一个 `struct ip` 结构体，包含 IP 头部的各个字段，如 IP 版本、头部长度、源 IP 和目的 IP 等。

构造数据部分： 将要传输的数据（payload）附加在协议头之后。

计算校验和： 为了保证数据完整性，使用 `calculate_checksum` 函数计算 IP 头部的校验和。

接收并解析自定义协议的数据包

创建原始套接字： 使用 `socket(AF_INET, SOCK_RAW, CUSTOM_PROTOCOL)` 创建原始套接字，并指定自定义协议号进行数据包捕获。

绑定套接字到网络接口： 通过 `bind()` 函数将套接字绑定到本地 IP 接口，确保接收特定网络接口的数据包。

设置混杂模式： 使用 `WSAIoctl()` 配置原始套接字为混杂模式，捕获所有流经网络接口的数据包，而不仅仅是发送到本机的包。

接收数据包： 使用 `recvfrom()` 函数接收数据包，并根据协议号判断是否为自定义协议包。

解析数据包： 解析 IP 头部，提取源 IP 和目标 IP。如果协议号匹配自定义协议号，进一步解析自定义协议头部和有效载荷，提取数据内容。

输出数据包信息

2.3.3 工作流程

监听端：

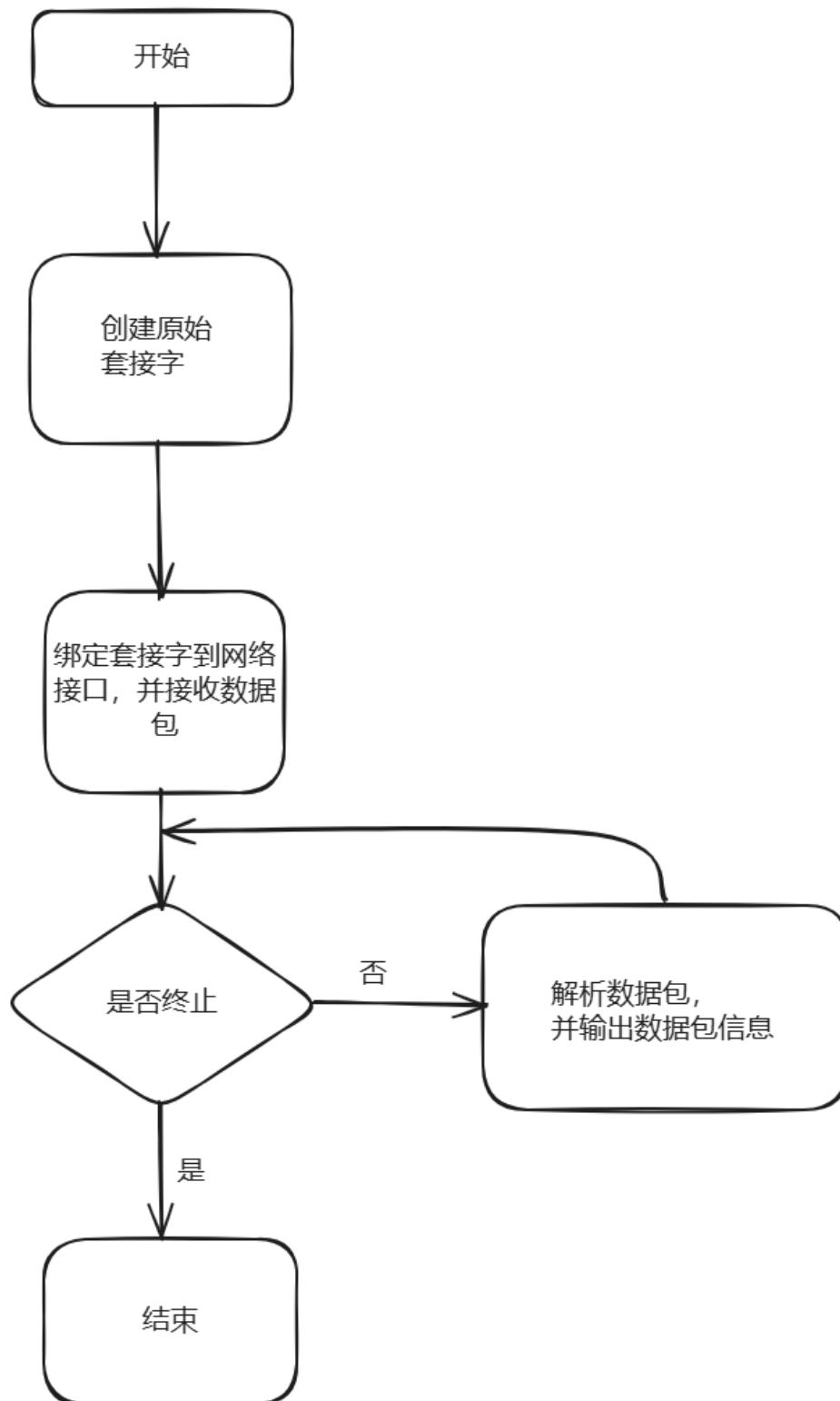


图 2-37 监听端工作流程

发送端：

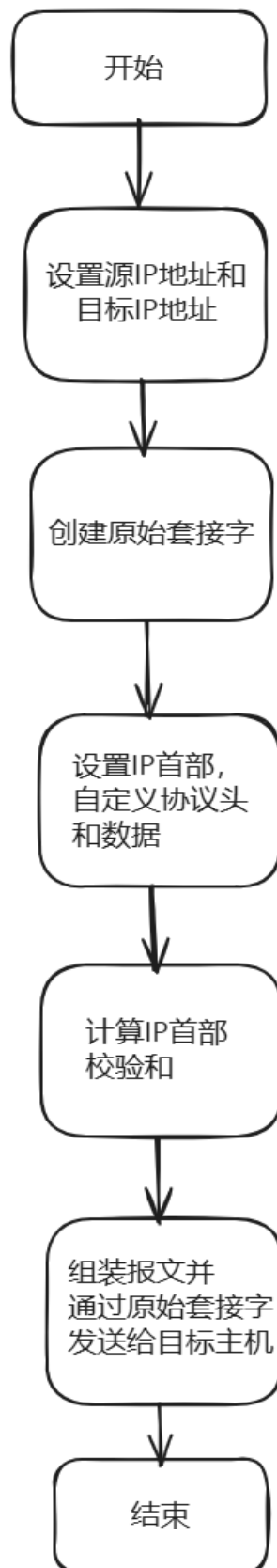


图 2-38 发送端工作流程

2.3.4 运行截图及结果说明

```
C:\Users\zhaoy\Desktop\ip\cmake-build-debug\ip.exe
1. Send Custom Packet
2. Receive Custom Packet
Enter your choice: 1
Packet sent successfully!
```

图 2-39 发送端结果截图

```
C:\Users\zhaoy\Desktop\ip\cmake-build-debug\ip.exe
1. Send Custom Packet
2. Receive Custom Packet
Enter your choice: 2
本机可用的 IP 地址:
    1: 192.168.72.1
    2: 192.168.126.1
    3: 192.168.31.179
请选择捕获数据包的接口号: 1
成功绑定到接口 1
开始接受数据...
收到来自 192.168.72.1 的数据包, 长度: 48 字节
源 IP: 192.168.72.1, 目的 IP: 192.168.72.1
版本号: 4
总长度: 48 字节
标识: 54321
标志位: 0
片偏移: 0
协议: 222
自定义协议 ID: 1234, 数据长度: 24
数据内容: qianqianzyk:Hello World!
-----
```

图 2-40 监听端结果截图

从监听端结果截图中可以看出, 已经成功捕获到了自定义 IP 数据报内容。同时版本号、

总长度、标识、标志位、片偏移、协议、源地址和目的地址也已成功输出。

2.3.5 CPP 源码

```
#include "winsock2.h"
#include "stdio.h"
#include <iostream>
#include <string>
#include <stdlib.h>
#include <unistd.h>
#include <thread>

using namespace std;
#pragma comment(lib, "ws2_32.lib")

#define DEFAULT_BUFLEN 65535 // 默认缓冲区大小为 65535 字节
#define CUSTOM_PROTOCOL 222 // 自定义传输协议号
#define DEFAULT_NAMELEN 512 // 自定义名称长度为 512 字节

// 自定义协议头
struct CustomHeader {
    uint16_t id; // 自定义标识
    uint16_t length; // 数据长度
};

// IP 头
struct ip {
    uint8_t ip_vhl : 4, ip_v : 4;
    uint16_t ip_tot;
    uint16_t ip_len;
    uint16_t ip_id;
    uint16_t ip_off;
    uint16_t ip_ttl;
    uint16_t ip_sr;
    uint16_t ip_len2;
    struct in_addr ip_src, ip_dst;
};

// 计算校验和
uint16_t calculate_checksum(unsigned short *buffer, int size)
{
    unsigned long checksum = 0;
    // 逐字节相加
    while (size > 0)
    {
        checksum += *buffer++;
        size -= 2; // 每次取两个字节
    }
    // 处理溢出 (高 16 位和低位 16 位)
    while (checksum >> 16)
    {
        checksum = (checksum & 0xFFFF) + (checksum >> 16);
    }
    // 返回校验和
    return ~checksum;
}

// 创建并发送自定义 IP 包
void send_custom_packet(const char *src_ip, const char *dest_ip, const char *payload)
{
    // 创建套接字 (AF_INET, IPv4 地址族, SOCK_RAW 原始套接字, 允许操作底层的网络包 (包括 IP 头) + IPPROTO_RAW 表示发送数据时不会做校验, 需要设置 IP 包)
    int sockfd = socket(AF_INET, SOCK_RAW, IPPROTO_RAW);
    if (sockfd < 0)
    {
        cout << "Socket creation failed" << endl;
        exit(EXIT_FAILURE);
    }
    // 设置 IP 包头 (HLEN 表示自己构造 IP 包, IPPROTO_IP 表示操作 IP 包, 通常用于 IP 包相关的操作, 如设置 IP 地址、IP 源地址、IP 目标地址等, IP_HLEN 表示自己构造 IP 包, &opt 表示选项, sizeof(opt) 表示选项的大小)
    int opt = 1;
    if (setsockopt(sockfd, IPPROTO_IP, IP_HDRINCL, &opt, sizeof(opt)) < 0)
    {
        cout << "Failed to set IP_HDRINCL" << endl;
        close(sockfd);
        exit(EXIT_FAILURE);
    }
    // 构造 sock_addr_in 结构体变量 dest_addr, 用来存储 IPv4 地址的结构体
    struct sockaddr_in dest_addr;
    // 填充结构体成员, 包括地址族 IPv4 地址
    dest_addr.sin_family = AF_INET;
    // 表示不要求为原始 IP 数据指定源端口号
    dest_addr.sin_port = 0;
    // 将字符串转换为结构体 IP 地址 (通常使用 inet_aton 函数将字符串转换为结构体 IP 地址)
    if (inet_aton(src_ip, &dest_addr.sin_addr) != 1)
    {
        cout << "Invalid address" << endl;
        close(sockfd);
        exit(EXIT_FAILURE);
    }
    // 构造 IP 包
    char packet[DEFAULT_BUFLEN];
    memset(packet, 0, sizeof(packet));
    struct ip ip_header = reinterpret_cast<struct ip*>(packet);
    ip_header->ip_vhl = 5; // IP 包长度 (20 字节)
    ip_header->ip_v = 4; // IPv4 版本
    ip_header->ip_tot = 0; // 服务类型
    ip_header->ip_len = htons((uint16_t)sizeof(struct ip) + sizeof(CustomHeader) + strlen(payload)); // 总长度
    ip_header->ip_id = htons((uint16_t)54321); // 标识
    ip_header->ip_off = 0; // 偏移
    ip_header->ip_ttl = 64; // 生存时间
    ip_header->ip_sr = CUSTOM_PROTOCOL; // 自定义协议号
    inet_ntoa(dest_addr.sin_addr, &src_ip); // 将结构体 IP 地址转换为字符串
    inet_ntoa(dest_addr.sin_addr, &dest_ip); // 将结构体 IP 地址转换为字符串
    ip_header->ip_src = calculate_checksum((uint16_t)reinterpret_cast<unsigned short*>(ip_header), sizeof(ip_header) * 2);
    // 构造自定义协议头
    CustomHeader custom_header = reinterpret_cast<CustomHeader*>(packet + sizeof(struct ip));
    custom_header->id = htons((uint16_t)1234); // 自定义标识
    custom_header->length = htons((uint16_t)strlen(payload)); // 数据长度
```



```

// 创建 IP 头
struct ip_header = reinterpret_cast<struct ip*>(recvbuf);
// INET_ADDRSTRLEN 表示存储 IPv4 地址需要的字符串的容量
char src_ip[INET_ADDRSTRLEN], dest_ip[INET_ADDRSTRLEN];
inet_ntop(AF_INET, &((ip_header->ip_src)), src_ip, sizeof(src_ip));
inet_ntop(AF_INET, &((ip_header->ip_dst)), dest_ip, sizeof(dest_ip));

// 输出 IP 头信息
cout << "源 IP: " << src_ip << ", 目的 IP: " << dest_ip << endl;
cout << "版本号: " << (int)(ip_header->ip_v) << endl;
cout << "总长度: " << ntohs(ntohl(ip_header->ip_len)) << " 字节" << endl;
cout << "序号: " << ntohs(ntohl(ip_header->ip_id)) << endl;
cout << "源端口: " << ((ip_header->ip_off & 0xFFFF) >> 12) << endl; // 高 3 位是标志位
cout << "片偏移: " << (ip_header->ip_off & 0xFFFF) << endl; // 高 13 位是偏移量
cout << "协议: " << (int)(ip_header->ip_p) << endl;

// 检查协议是否由自定义协议
if (ip_header->ip_p == CUSTOM_PROTOCOL) {
    // 解析自定义协议头
    CustomHeader* custom_header = reinterpret_cast<CustomHeader*>(recvbuf + ip_header->ip_hl * 4);
    uint16_t payload_length = ntohs(ntohl(custom_header->length));
    uint32_t id = ntohl(ntohl(custom_header->id));

    cout << "自定义协议 ID: " << id << ", 数据长度: " << payload_length << endl;

    // 解析自定义数据
    char* payload = recvbuf + ip_header->ip_hl * 4 + sizeof(CustomHeader);
    cout << "数据内容: " << string(payload, payload_length) << endl;
} else {
    cout << "数据自定义协议头, 协议号: " << (int)(ip_header->ip_p) << endl;
}

cout << "-----" << endl;
} else {
    cout << "未知头, 错误代码: " << WSAGetLastError() << endl;
}
} while (iResult > 0);

// 关闭套接字
closesocket(sock);
cout << "接收完毕, 套接字已关闭" << endl;
}

int main() {
    WSADATA wsaData;
    if (WSAStartup(WVersionRequested, MAKEWORD(2, 2), &WSADATA) != 0) {
        cout << "WSAStartup failed" << endl;
        return 1;
    }

    int choice;
    cout << "1. Send Custom Packet" << endl;
    cout << "2. Receive Custom Packet" << endl;
    cout << "Enter your choice: ";
    cin >> choice;

    if (choice == 1) {
        send_custom_packet("192.168.72.1", "192.168.72.1", "payload: 'golangryx:hello World!'");
    } else if (choice == 2) {
        receive_custom_packets();
    } else {
        cout << "Invalid choice!" << endl;
    }

    WSACleanup();

    return 0;
}

```

图 2-41 源代码截图

实验总结与心得体会

本次实验让我对计算机的网络原理有了更深的理解。常用网络命令行工具让用户可以很方便的了解到当前主机的各种网络状态。

Cisco packet tracer 可以让我搭建一个属于自己的网络，配置好后可以像平时上网一样使用各种功能。

借助网络编程这一机会，我接触到了原始套接字，同时加深了对数据包传输底层的理解。

通过这次实验，我受益匪浅，更加感受到了计算机网络的魅力。