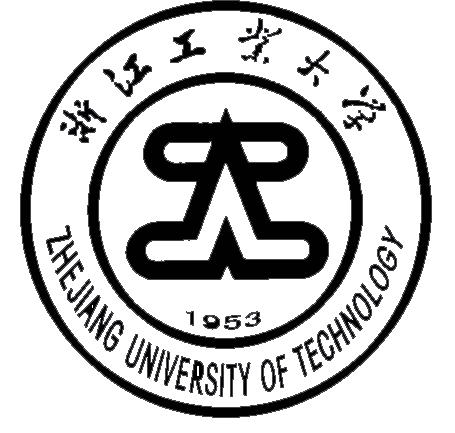


**计算机网络课程设计报告**



学 号 \_

姓 名

班 级

组 号

提交日期

**目 录**

[一． 课程的目的和任务 1](#_Toc28200)

[1.1课程目标 1](#_Toc3939)

[1.2 课程任务 1](#_Toc29929)

[二． 课程的基本内容和要求 1](#_Toc27361)

[2.1常用网络命令 1](#_Toc3526)

[2.2交换机与路由器 7](#_Toc7439)

[2.2.1 交换机配置与交换机VLAN设计 10](#_Toc26430)

[2.2.2 路由配置与静态路由设计 15](#_Toc16577)

[2.2.3 动态路由设计 17](#_Toc27508)

[2.2.4 网络集成设计 19](#_Toc26454)

[2.3网络编程 24](#_Toc23216)

[2.3.1 开发环境及关键配置 24](#_Toc31438)

[2.3.2 设计思路 24](#_Toc24167)

[2.3.3 工作流程 25](#_Toc4211)

[2.3.4 运行截图及结果说明 30](#_Toc10677)

[三． 实验总结与心得体会 31](#_Toc15855)

1. **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及发展。

1. 掌握交换机VLAN设置及验证方法。
2. 掌握在路由器上实现静态路由配置方法
3. 掌握在路由器上实现RIP或OSPF协议配置方法。
4. 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。
5. 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
6. **课程的基本内容和要求**

###### 2.1常用网络命令

1. ipconfig

|  |  |
| --- | --- |
| 参数 | 描述 |
| /all | 显示所有网络适配器的完整 TCP/IP 配置。 |
| /displaydns | 显示 DNS 客户端缓存内容，包括 Hosts 文件和最近查询的记录。 |
| /flushdns | 清空并重置 DNS 客户端缓存，用于解决 DNS 问题。 |
| /registerdns | 手动注册 DNS 名称和 IP 地址，解决动态更新问题。 |
| /release [适配器] | 释放 DHCP 配置，丢弃当前 IP 地址（可指定适配器）。 |
| /release6 [适配器] | 释放 DHCPv6 配置，丢弃当前 IPv6 地址（可指定适配器）。 |
| /renew [适配器] | 续订 DHCP 配置，获取新的 IP 地址（可指定适配器）。 |
| /renew6 [适配器] | 续订 DHCPv6 配置，获取新的 IPv6 地址（可指定适配器）。 |
| /setclassid 适配器[类ID] | 设置适配器的 DHCP 类 ID，为所有适配器可用“\*”通配符。 |
| /showclassid 适配器 | 显示适配器的 DHCP 类 ID，可使用“\*”查看所有适配器。 |
| /? | 显示命令帮助信息。 |

表 1 ipconfig参数

ipconfig 是 Windows 系统中一个用于查看和管理网络配置的命令行工具。它可以显示网络适配器的 IP 地址、子网掩码和默认网关等信息，还能用于刷新和重新注册网络配置。

工作原理：ipconfig 从操作系统的网络栈中获取信息，直接与内核中的网络驱动程序和配置模块交互，以读取网络适配器的状态和设置。

图 1 ipconfig命令截图

1. ping

|  |  |  |
| --- | --- | --- |
| 选项 | 功能说明 | 适用范围 |
| -t | 持续 Ping 指定主机，直到手动停止。按 Ctrl+Break 查看统计信息，按 Ctrl+C 停止。 | IPv4 和 IPv6 |
| -a | 将 IP 地址解析为主机名。 | IPv4 和 IPv6 |
| -n count | 指定要发送的回显请求数。默认值为 4。 | IPv4 和 IPv6 |
| -l size | 指定发送缓冲区的大小（字节）。默认值为 32，最大为 65,500。 | IPv4 和 IPv6 |
| -f | 设置“不分段”标记，用于排查路径 MTU 问题（仅适用于 IPv4）。 | IPv4 |
| -i TTL | 设置生存时间 (TTL) 的值。范围为 1 到 255，默认值为系统的默认 TTL。 | IPv4 和 IPv6 |
| -v TOS | 设置服务类型 (TOS)，该功能已被弃用，对 IP 标头中的服务类型字段无影响（仅适用于 IPv4）。 | IPv4 |
| -r count | 记录数据包通过的路径中的最多 count 跃点（仅适用于 IPv4）。 | IPv4 |
| -s count | 记录每个跃点的时间戳，最多为 count（仅适用于 IPv4）。 | IPv4 |
| -j host-list | 使用松散源路由选项，指定中间路由的主机列表（最多 9 个，仅适用于 IPv4）。 | IPv4 |
| -k host-list | 使用严格源路由选项，指定中间路由的主机列表（最多 9 个，仅适用于 IPv4）。 | IPv4 |
| -w timeout | 指定每次回复的超时时间（毫秒）。默认值为 4000 毫秒（4 秒）。 | IPv4 和 IPv6 |
| -R | 测试反向路由功能（仅适用于 IPv6），已被弃用，某些系统可能丢弃带此标头的回显请求。 | IPv6 |
| -S srcaddr | 指定发送数据包的源地址（仅适用于 IPv6）。 | IPv6 |
| -c compartment | 指定路由隔离舱标识符。 | IPv4 和 IPv6 |
| -p | Ping Hyper-V 网络虚拟化提供程序地址。 | IPv4 和 IPv6 |
| -4 | 强制使用 IPv4 协议进行 Ping 操作。 | IPv4 |
| -6 | 强制使用 IPv6 协议进行 Ping 操作。 | IPv6 |

表 2 ping参数

ping通过发送互联网控制消息协议 (ICMP) 回显请求消息，验证与另一台 TCP/IP 计算机在 IP 层面的连接性。收到对应的回显回复消息后，会显示回复内容以及往返时间。ping 是用于排查连接性、可达性和名称解析问题的主要 TCP/IP 命令。



图 2 ping命令截图

1. netstat

|  |  |
| --- | --- |
| 参数 | 功能说明 |
| -a | 显示所有活动的连接和监听端口。 |
| -n | 以数字格式显示地址和端口（不进行名称解析）。 |
| -o | 显示每个连接的相关进程 ID (PID)。 |
| -p <protocol> | 显示指定协议（TCP、UDP、TCPv6、UDPv6）的连接信息。 |
| -r | 显示当前主机的路由表。 |
| -s | 按协议显示统计信息（TCP、UDP、ICMP、IP）。 |
| -e | 显示网络接口统计信息，包括传输的字节数和数据包数。 |
| -b | 显示每个连接的可执行文件（需要管理员权限）。 |
| -f | 显示外部地址的完全限定域名（FQDN）。 |
| interval | 设置刷新间隔（以秒为单位），持续显示统计信息，直到按 Ctrl+C 退出。 |

表 3 netstat参数

显示活动的 TCP 连接、计算机正在监听的端口、以太网统计信息、IP 路由表、IPv4 协议（包括 IP、ICMP、TCP 和 UDP 协议）的统计信息，以及 IPv6 协议（包括 IPv6、ICMPv6、TCP over IPv6 和 UDP over IPv6 协议）的统计信息。

图 3 netstat命令截图

## tracert

|  |  |
| --- | --- |
| 参数 | 功能说明 |
| -d | 不将地址解析为主机名，直接显示 IP 地址。 |
| -h maximum\_hops | 搜索目标的最大跳数，默认值为 30。 |
| -j host-list | 使用松散源路由与指定的主机列表（仅适用于 IPv4）。 |
| -w timeout | 指定每次回复的超时时间（毫秒），默认值为 4000 毫秒（4 秒）。 |
| -R | 跟踪往返路径（仅适用于 IPv6）。 |
| -S srcaddr | 指定源地址发送请求（仅适用于 IPv6）。 |
| -4 | 强制使用 IPv4 进行跟踪。 |
| -6 | 强制使用 IPv6 进行跟踪。 |

表 4 tracert参数

通过发送互联网控制消息协议 (ICMP) 回显请求消息或 ICMPv6 消息，并逐步增加生存时间 (TTL) 字段的值，确定到目标的路径。路径中的每个路由器在转发 IP 数据包之前必须至少将 TTL 减 1。实际上，TTL 是一个最大跳数计数器。当数据包的 TTL 变为 0 时，路由器需要返回一个 ICMP 超时消息 (Time Exceeded) 给源计算机。

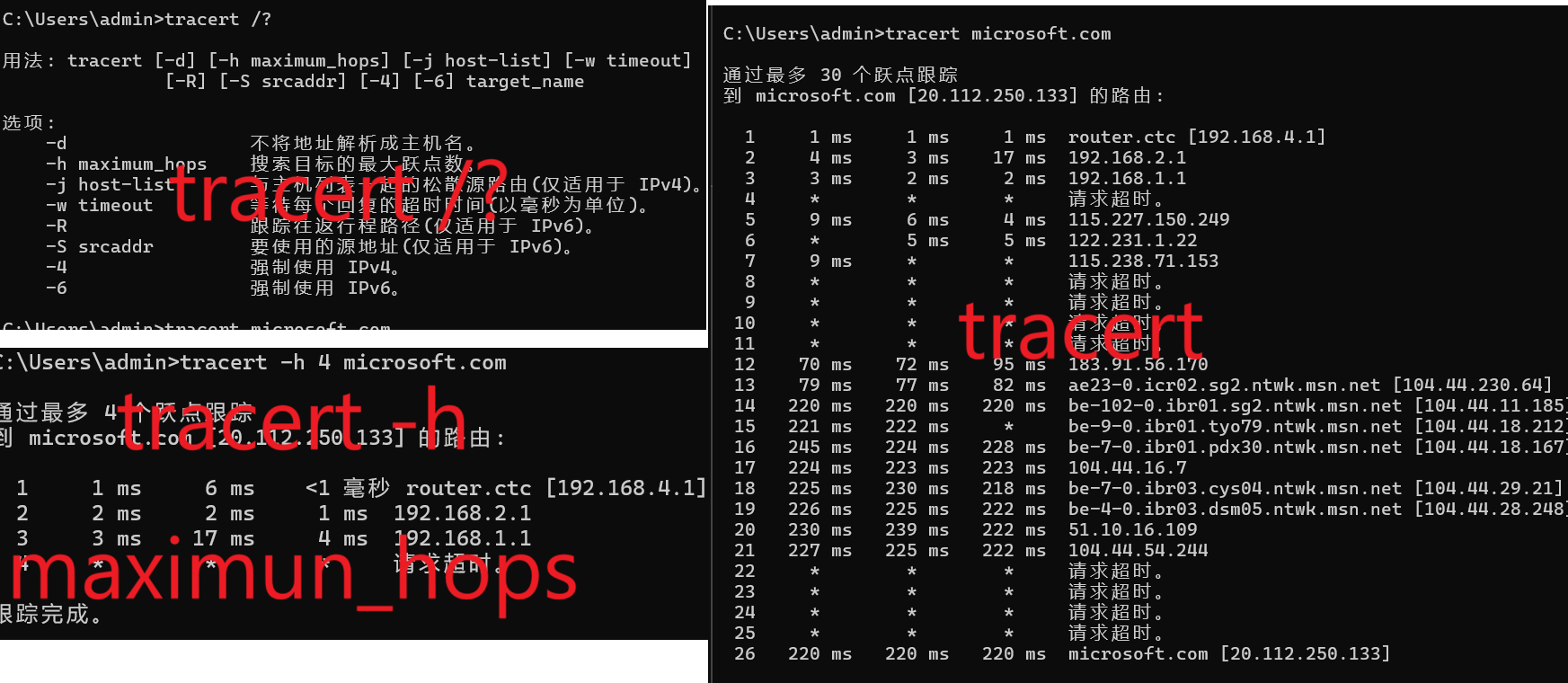
通过发送第一个 TTL 为 1 的回显请求消息，并在每次后续传输中将 TTL 增加 1 来确定路径，直到目标响应或达到最大跳数为止。默认的最大跳数为 30，可以使用 -h 参数指定。

图 4 tracert命令截图

1. arp

|  |  |
| --- | --- |
| 参数 | 功能说明 |
| -a | 显示当前 ARP 项。如果指定 inet\_addr，仅显示该 IP 的 ARP 项。多个接口时，显示每个接口的 ARP 表。 |
| -g | 与 -a 相同。 |
| -v | 详细模式显示当前 ARP 项，包括无效项和环回接口上的项。 |
| inet\_addr | 指定目标 IP 地址。 |
| -N if\_addr | 显示 if\_addr 指定的网络接口的 ARP 项。 |
| -d | 删除指定 IP 地址 (inet\_addr) 的 ARP 项。可使用 \* 删除所有项。 |
| -s | 添加静态 ARP 项，将 IP 地址 (inet\_addr) 与物理地址 (eth\_addr) 关联。该项为永久项。 |
| eth\_addr | 指定物理地址（MAC 地址），用连字符 - 分隔的 6 个十六进制字节。 |
| if\_addr | 指定网络接口的 IP 地址。如果未指定，则默认使用第一个适用接口。 |

表 5 arp参数

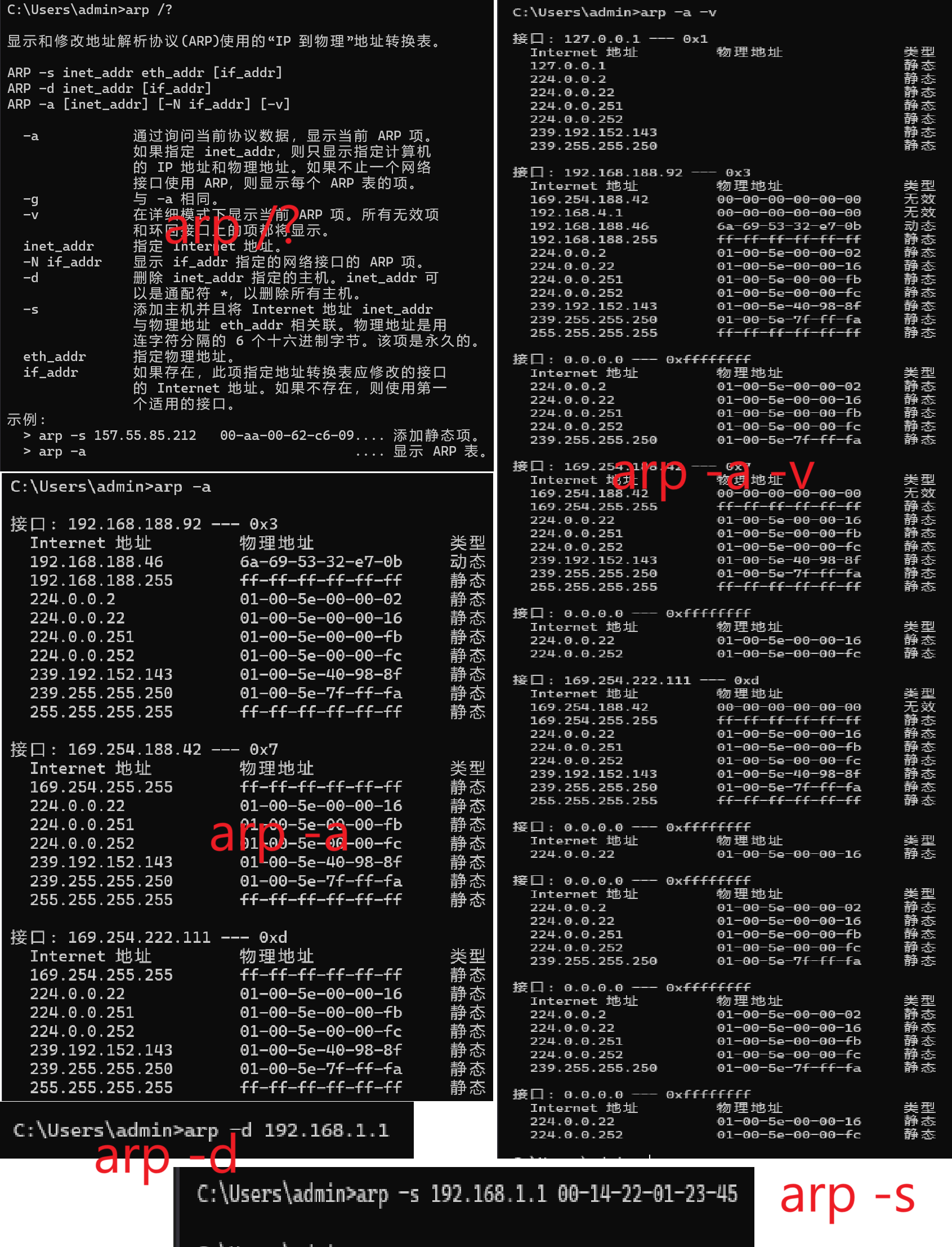
显示和修改地址解析协议 (ARP) 缓存中的条目。ARP 缓存包含一个或多个表，用于存储IP地址及其解析后的以太网或令牌环物理地址。每个安装在计算机上的以太网或令牌环网络适配器都有一个独立的表。

图 5 arp命令截图

1. telnet

|  |  |
| --- | --- |
| 命令 | 功能说明 |
| open | 连接到远程主机，例如 open 192.168.1.1 23。 |
| close | 断开当前连接。 |
| display | 显示当前设置。 |
| set | 设置 Telnet 参数，例如 set localecho 开启本地回显。 |
| unset | 取消设置，例如 unset localecho 关闭本地回显。 |
| status | 显示当前 Telnet 客户端的状态信息。 |
| quit | 退出 Telnet 客户端。 |

表 6 telnet命令

**Microsoft Telnet** 是微软提供的一款命令行工具，用于通过 **Telnet协议**（传输控制协议/网际协议 TCP/IP 的一部分）连接到远程计算机或设备。Telnet 提供一种文本模式的远程登录方式，主要用于远程管理和通信。

图 6 telnet命令截图

## 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. 安装packet tracer，在packet tracer仿真环境下，熟悉交换机命令、交换机初始化配置，熟悉路由器命令及路由器的初始化配置。

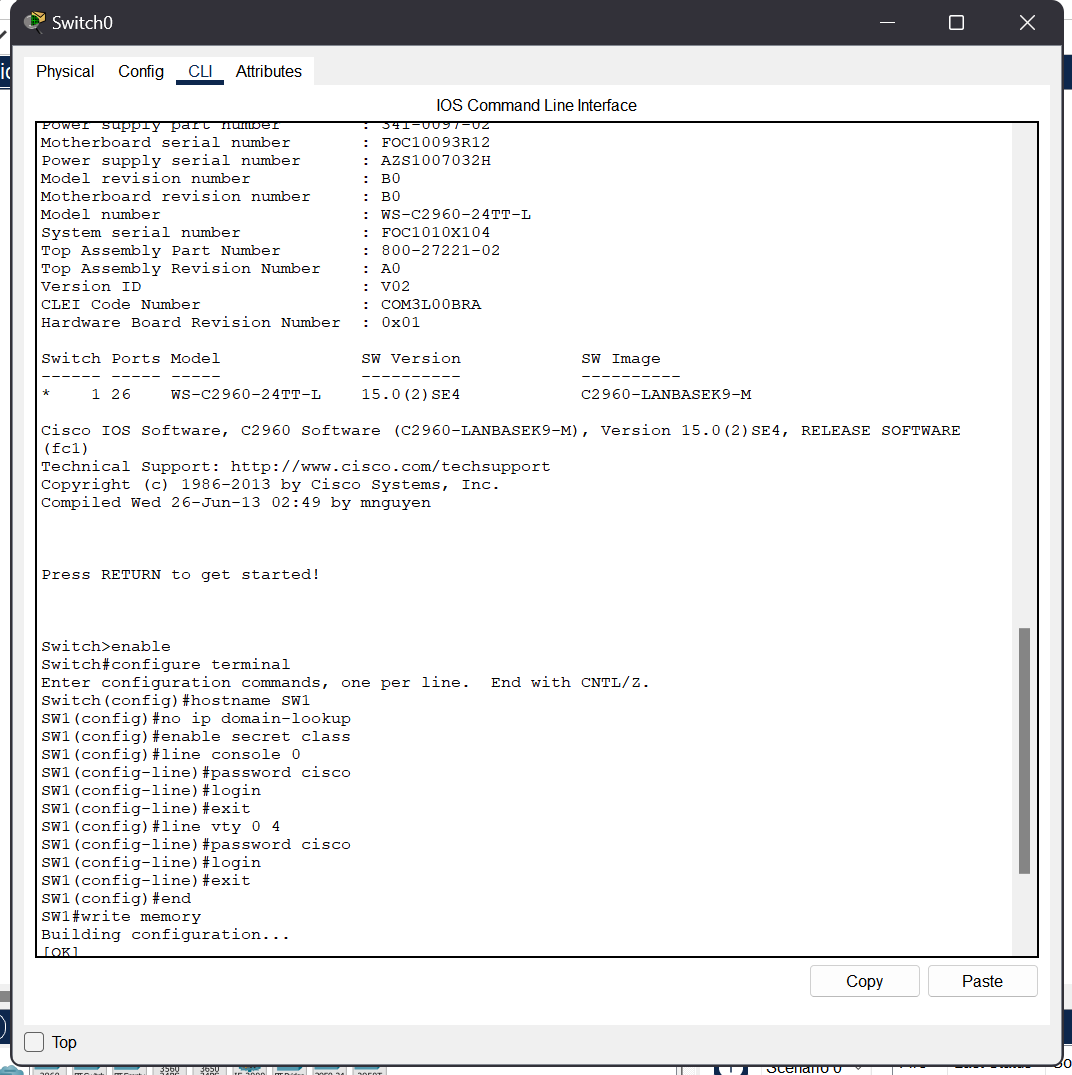


图 7交换机初始化

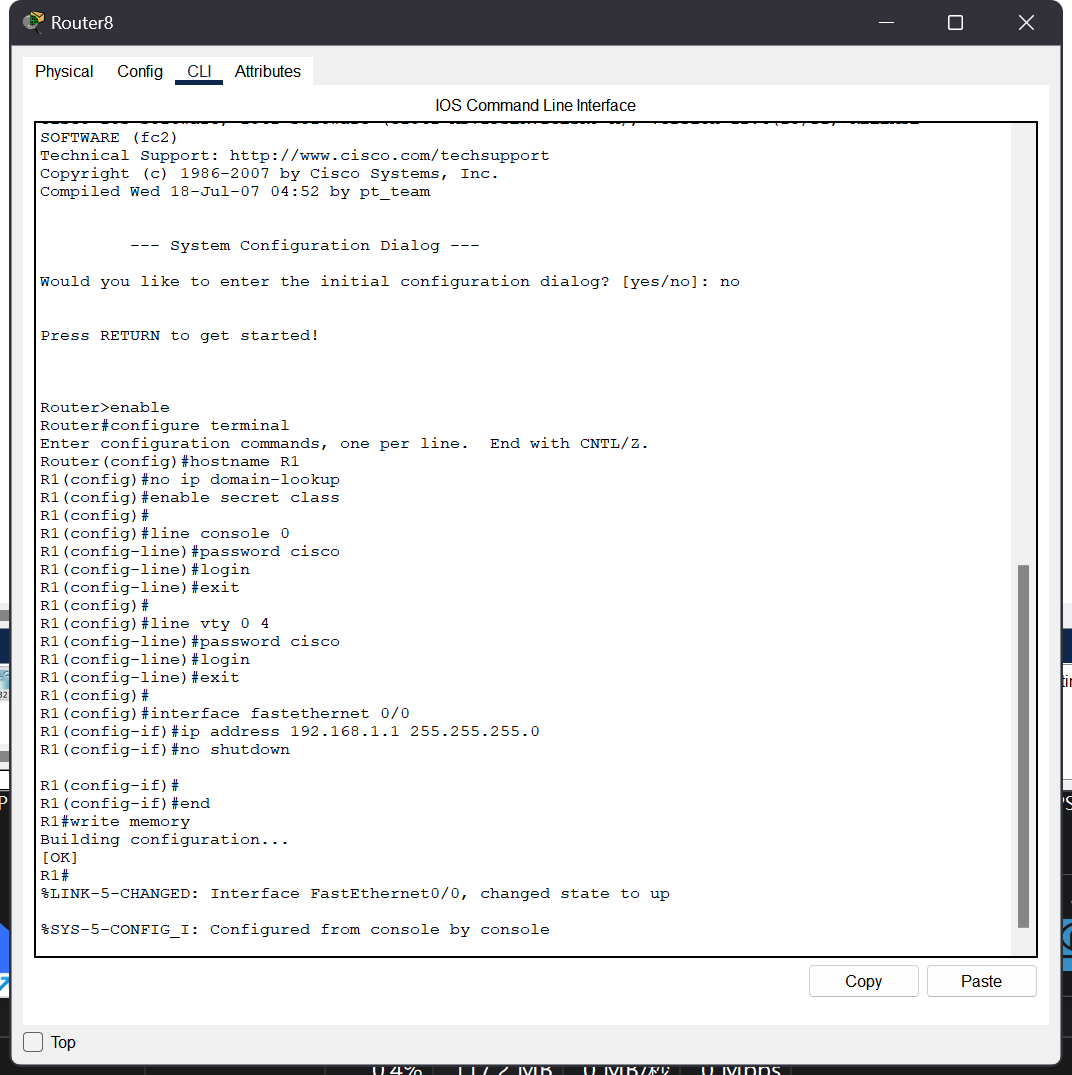


图 8路由器初始化

1. 掌握交换机VLAN设置及验证方法。

在交换机上实现VLAN配置，实现在一个交换机上基于端口的VLAN配置，要求至少创建三个VLAN，在仿真环境下给出网络拓扑结构，网络划分方案，并进行配置、分析验证实验结果。

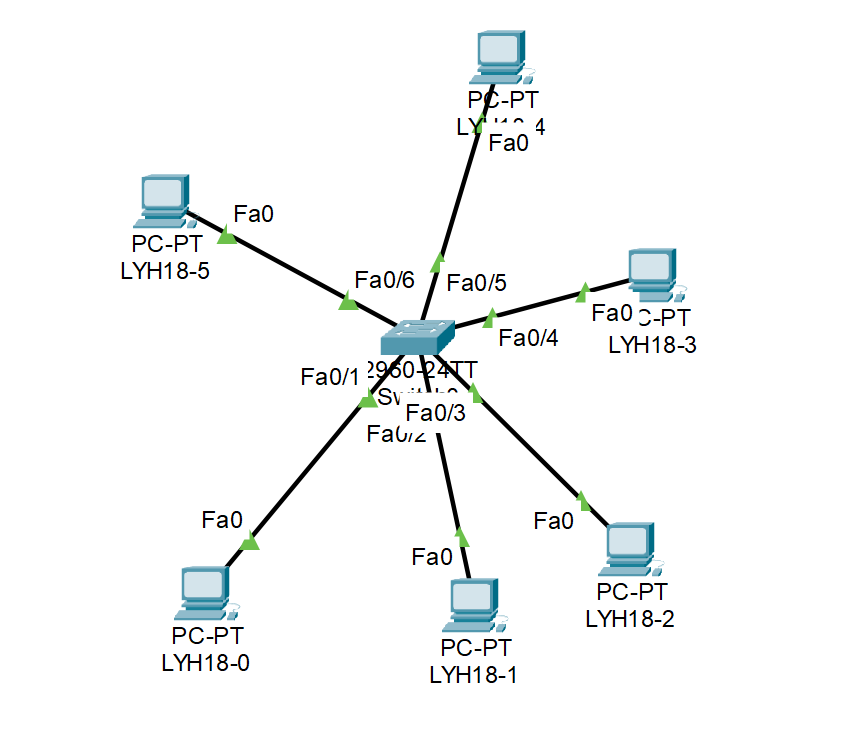


图 9拓扑图

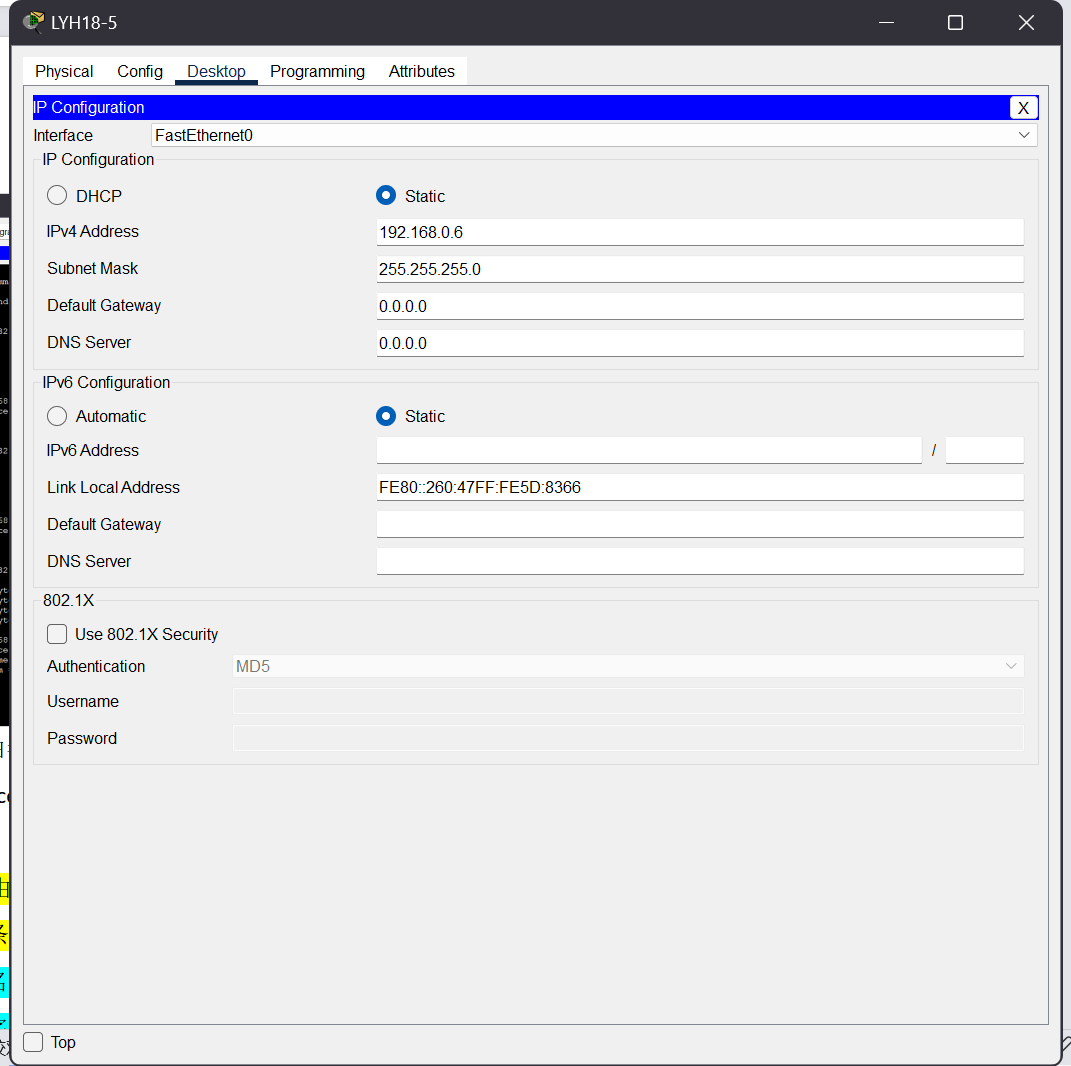


图 10主机静态ip配置

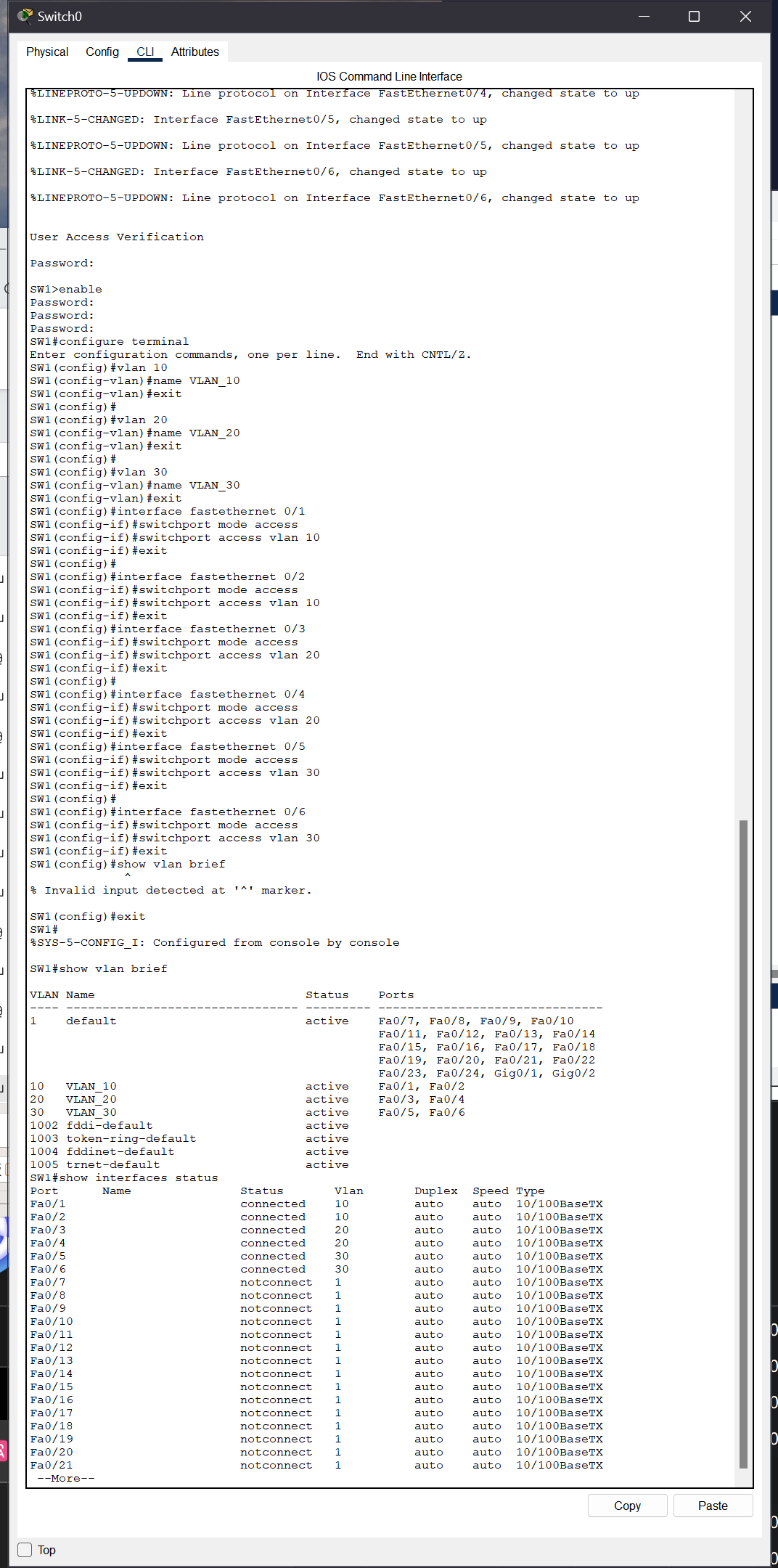


图 11 VLAN配置和结果

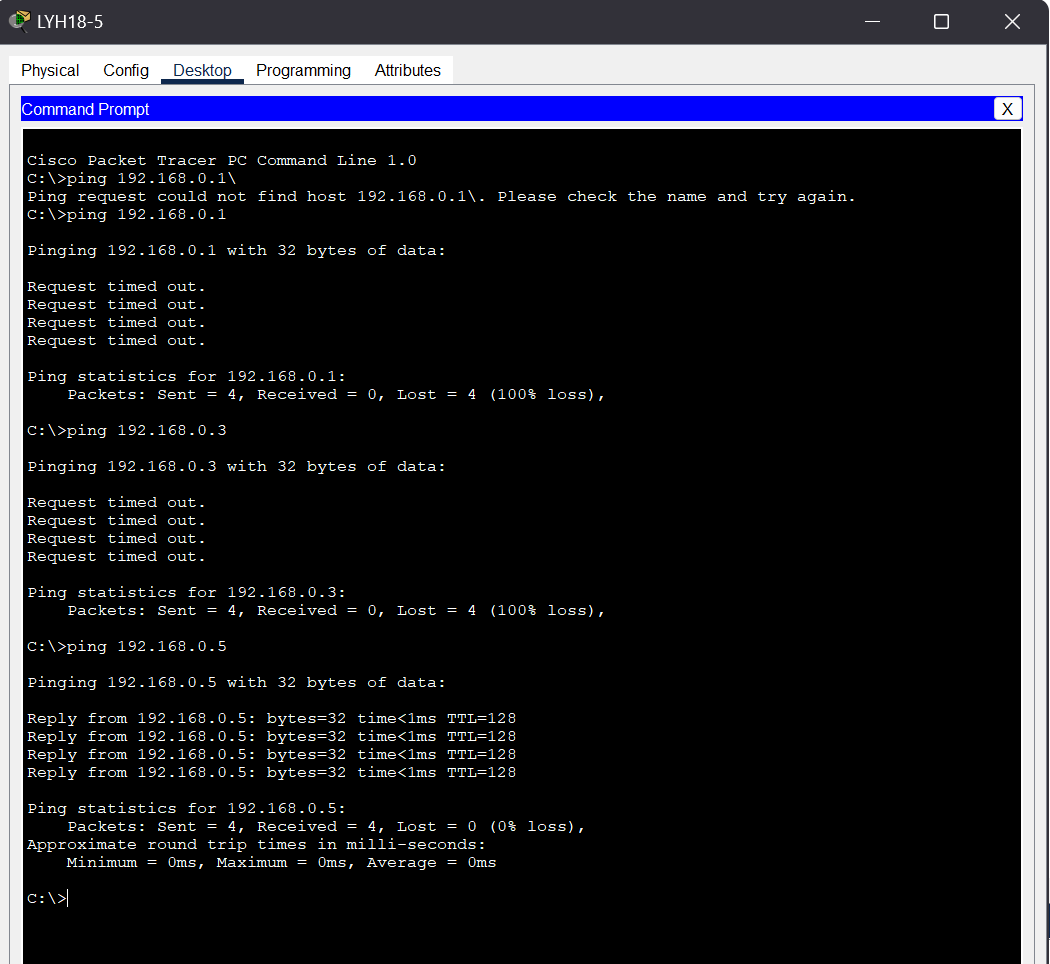


图 12 ping验证

show vlan brief之后检查vlan配置成功，ping测试后发现只有同VLAN可以ping通。说明交换机上跨vlan子网不能直接连通。

1. 基于Console控制台登录配置路由器，学习路由器配置相关命令

图片包含 折线图

描述已自动生成

图 13 Console控制台拓扑结构

文本

描述已自动生成

图 14设置路由器初始化

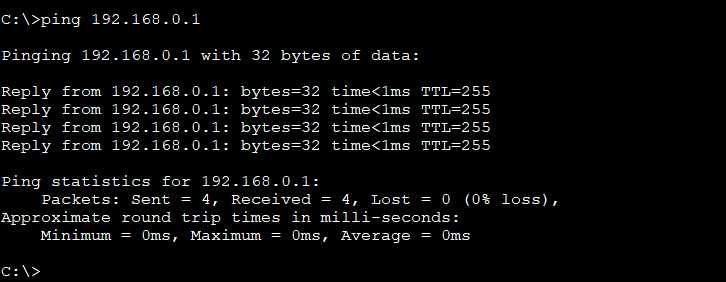


图 15 ping检验

## 2.2.2 路由配置与静态路由设计

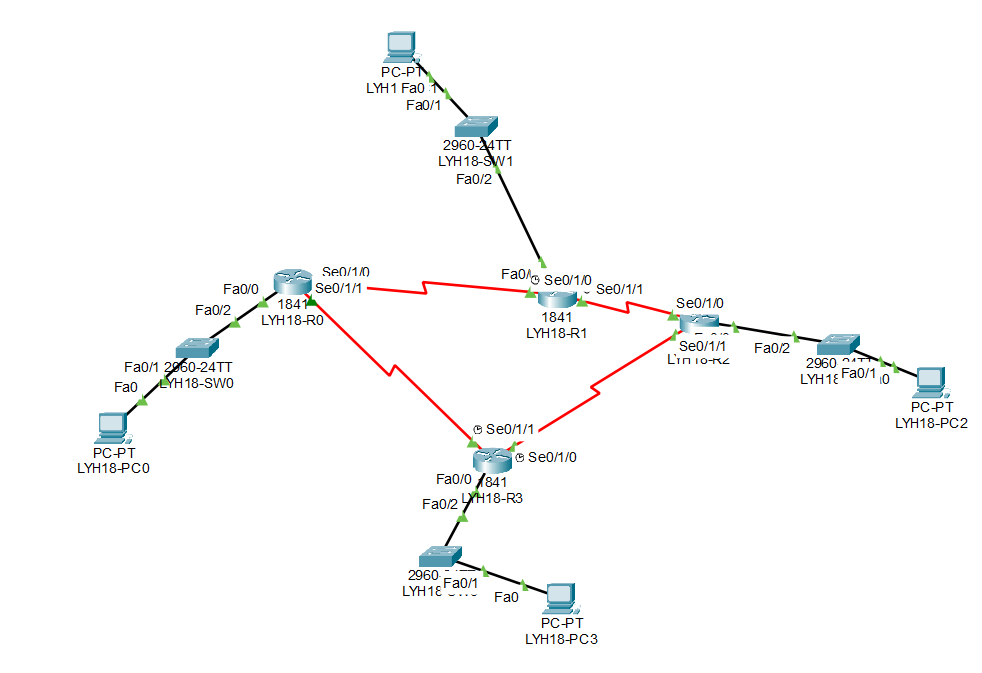


图 16拓扑图

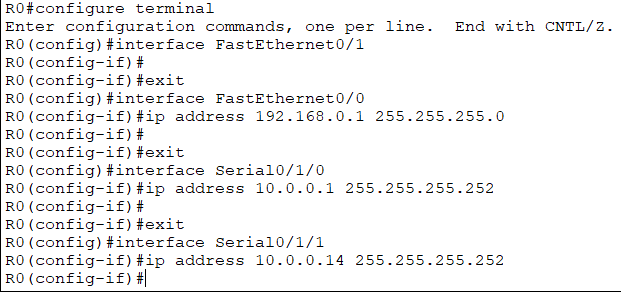


图 17路由端口ip配置

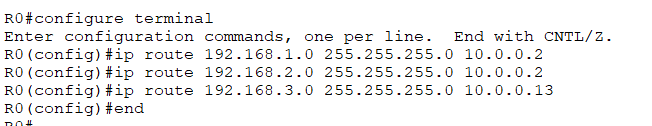


图 18静态路由配置

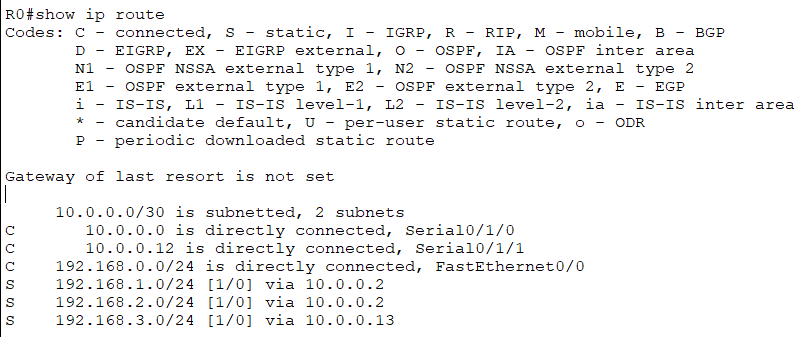


图 19结果查看

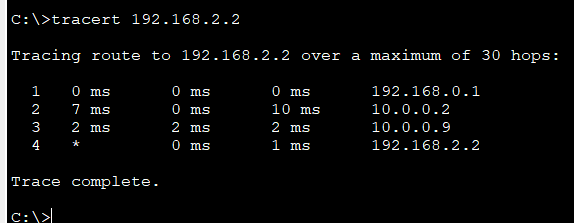


图 20 tracert检测

配置成功

## 2.2.3 动态路由设计

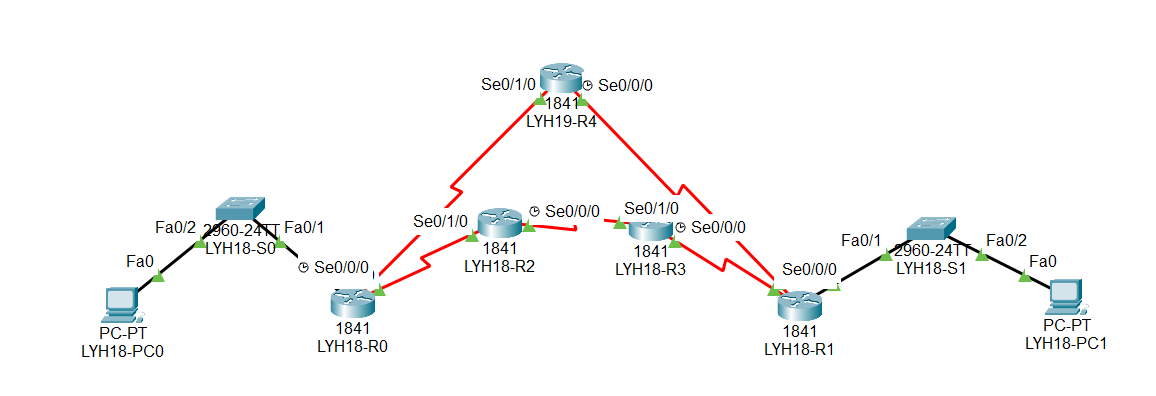


图 21拓扑图



图 22五台路由基础配置

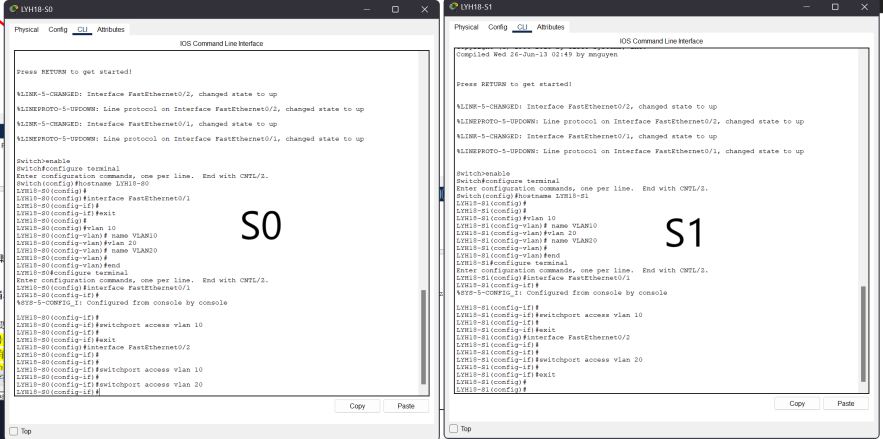
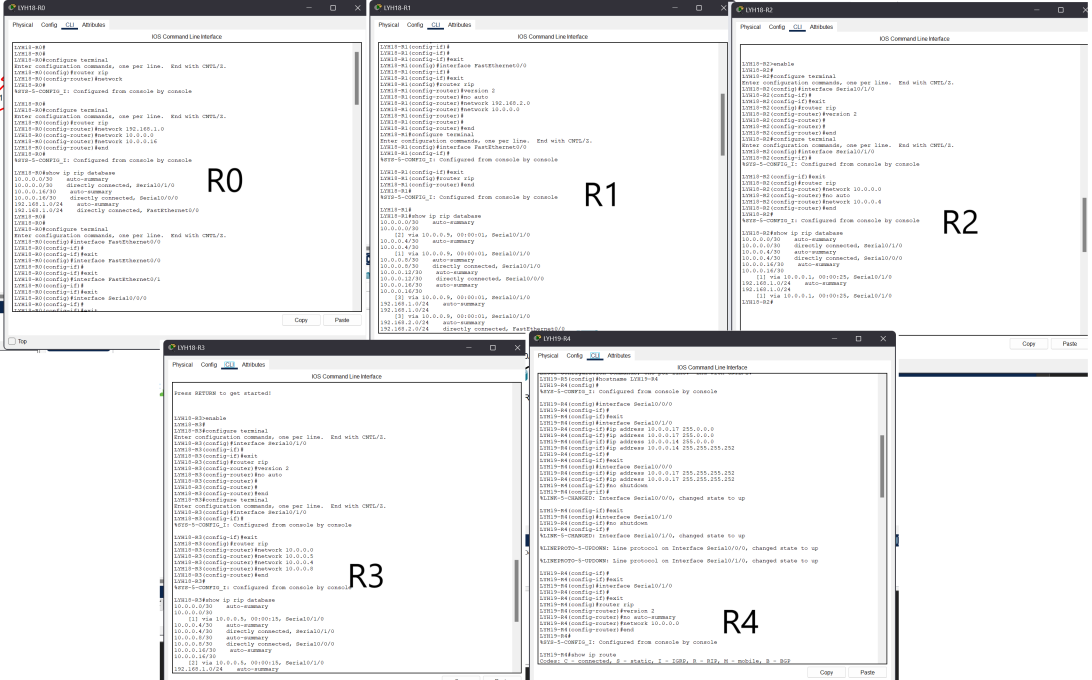


图 23两台交换机配置

图 24 rip配置

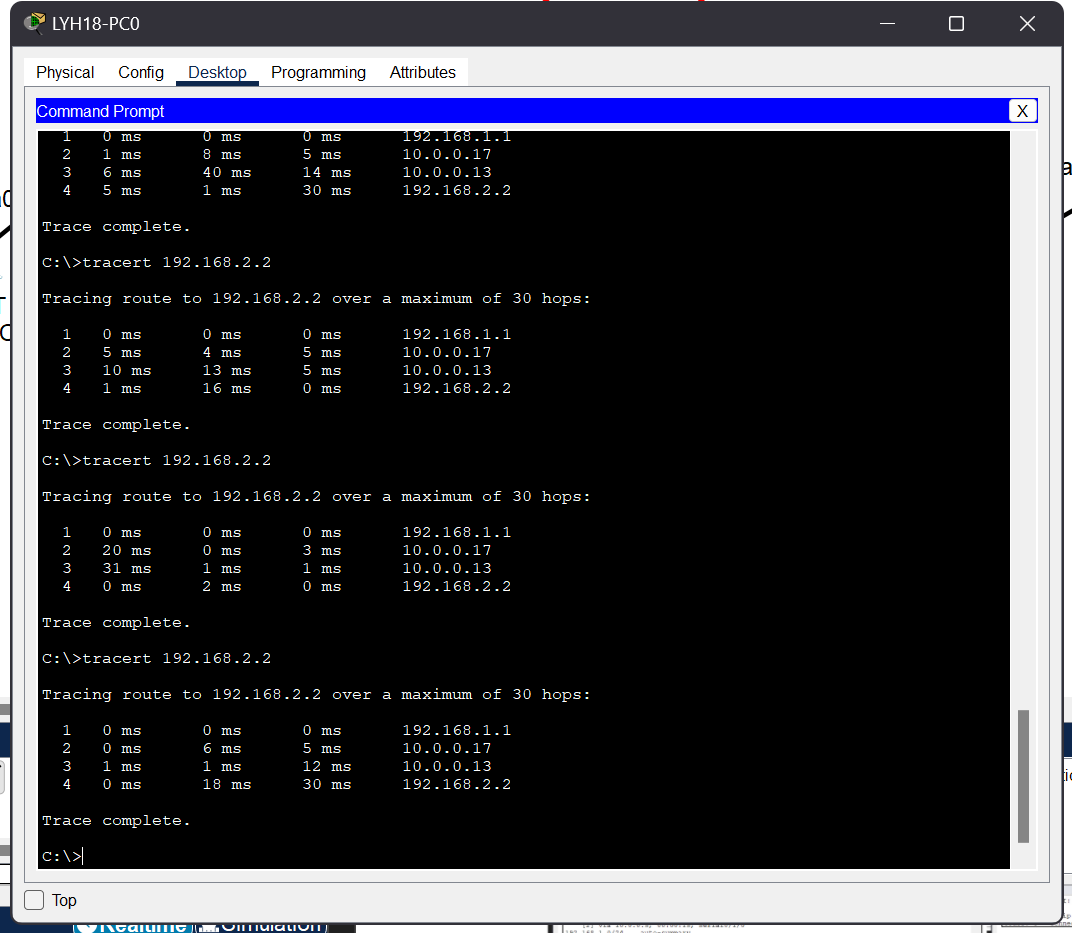


图 25 tracert检测

tracert检测，发现选择较短路径,说明rip生效，优先选择较短路径

## 2.2.4 网络集成设计

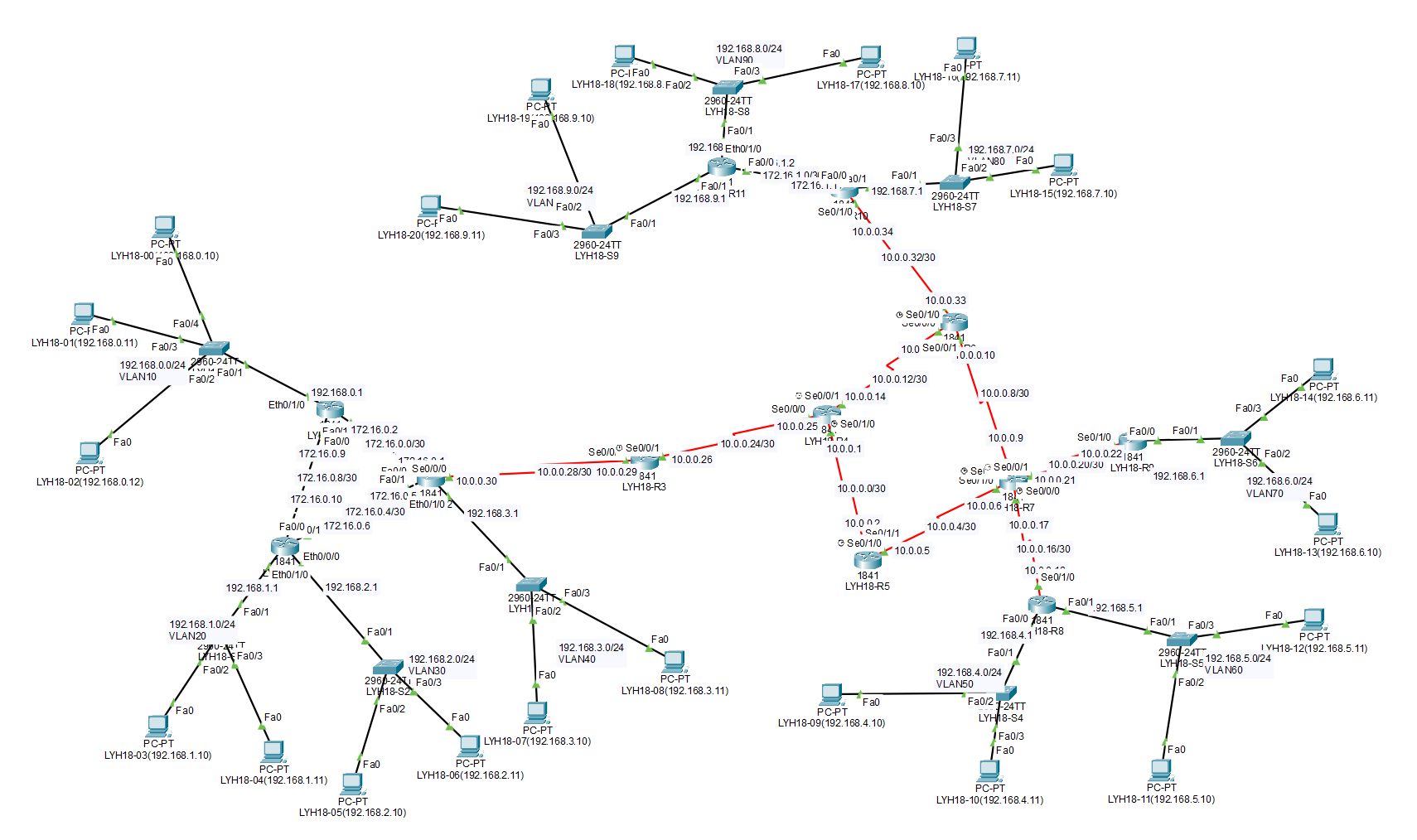


图 26拓扑图

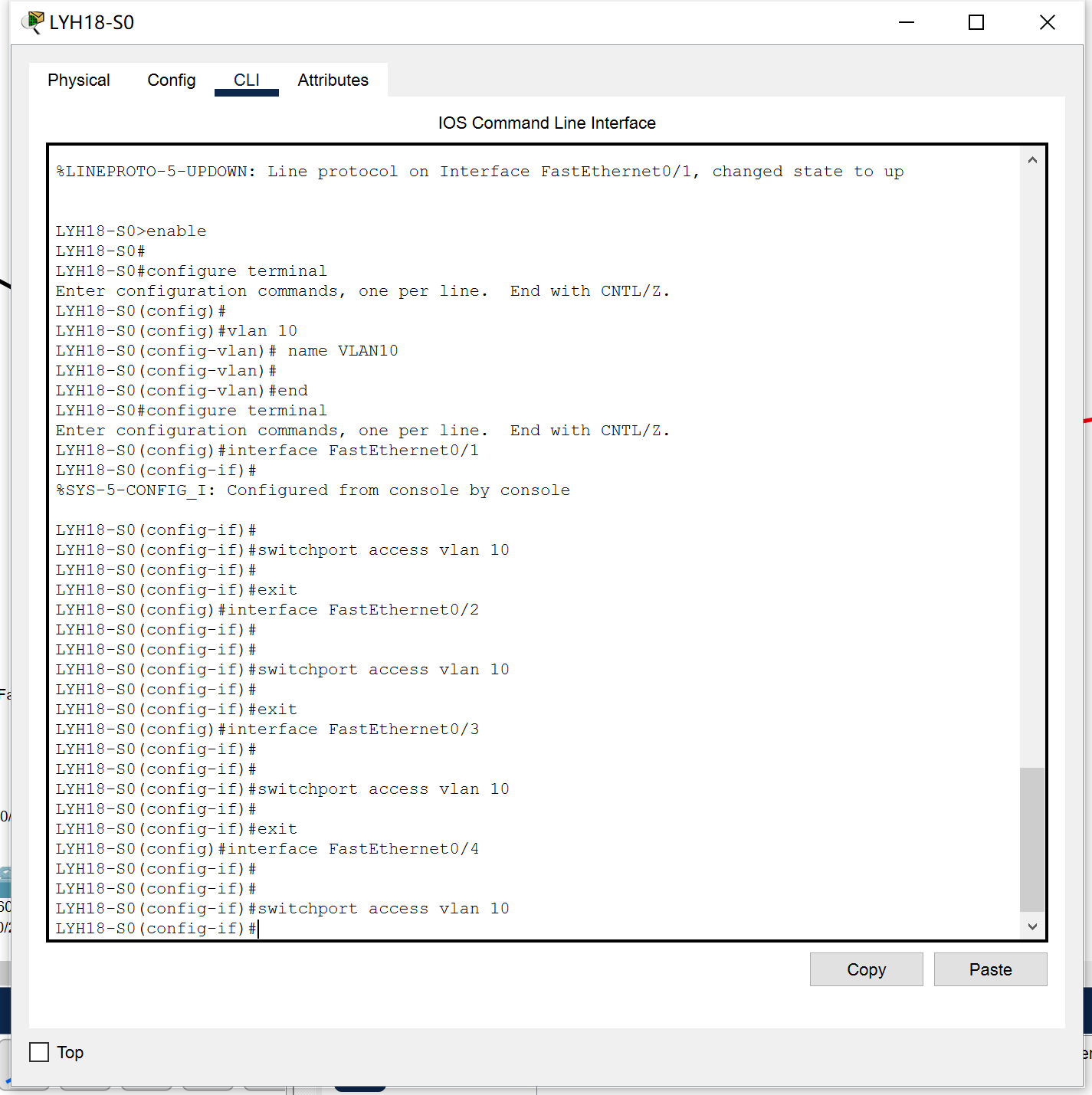


图 27交换机配置

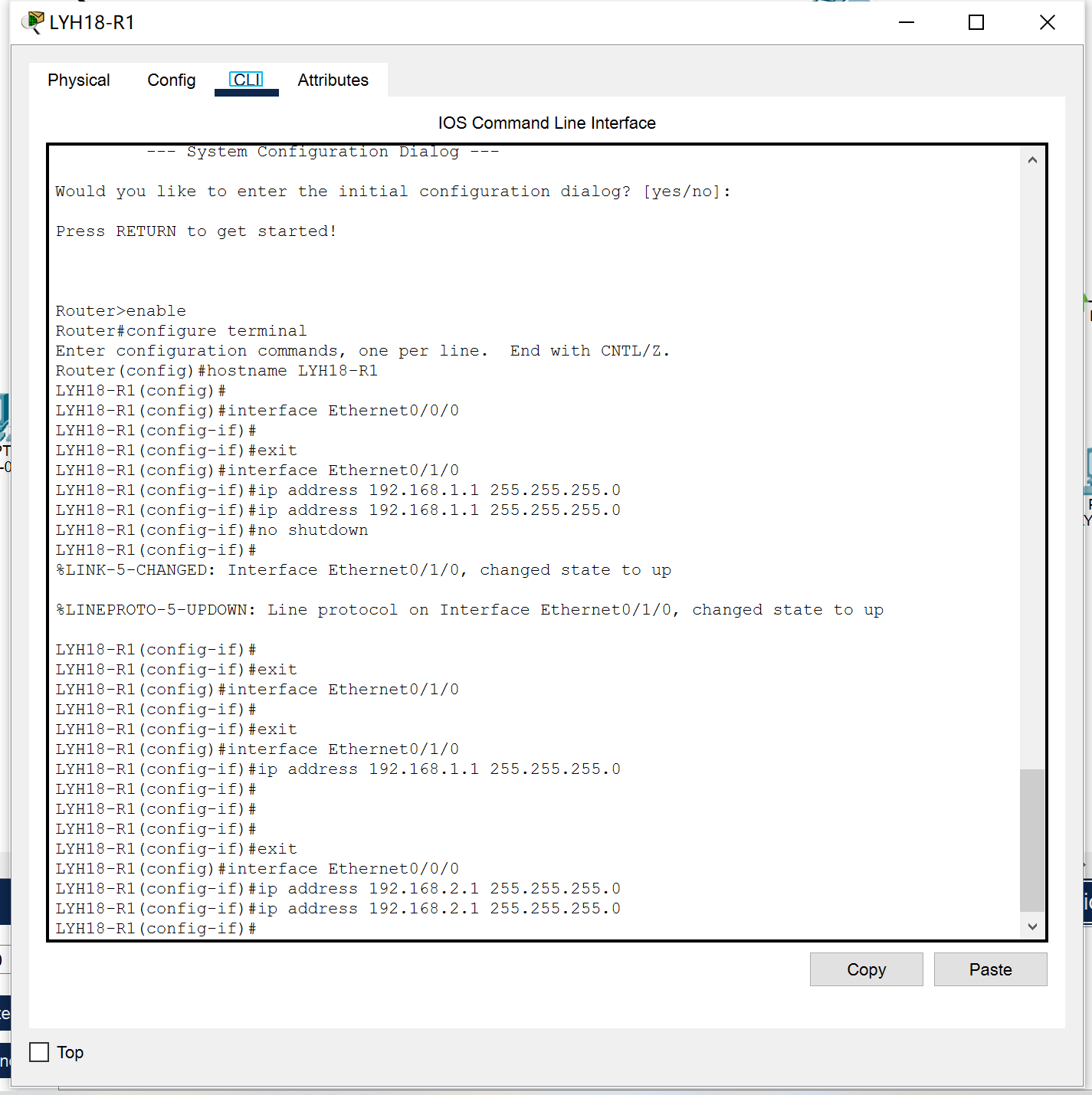


图 28路由基础配置

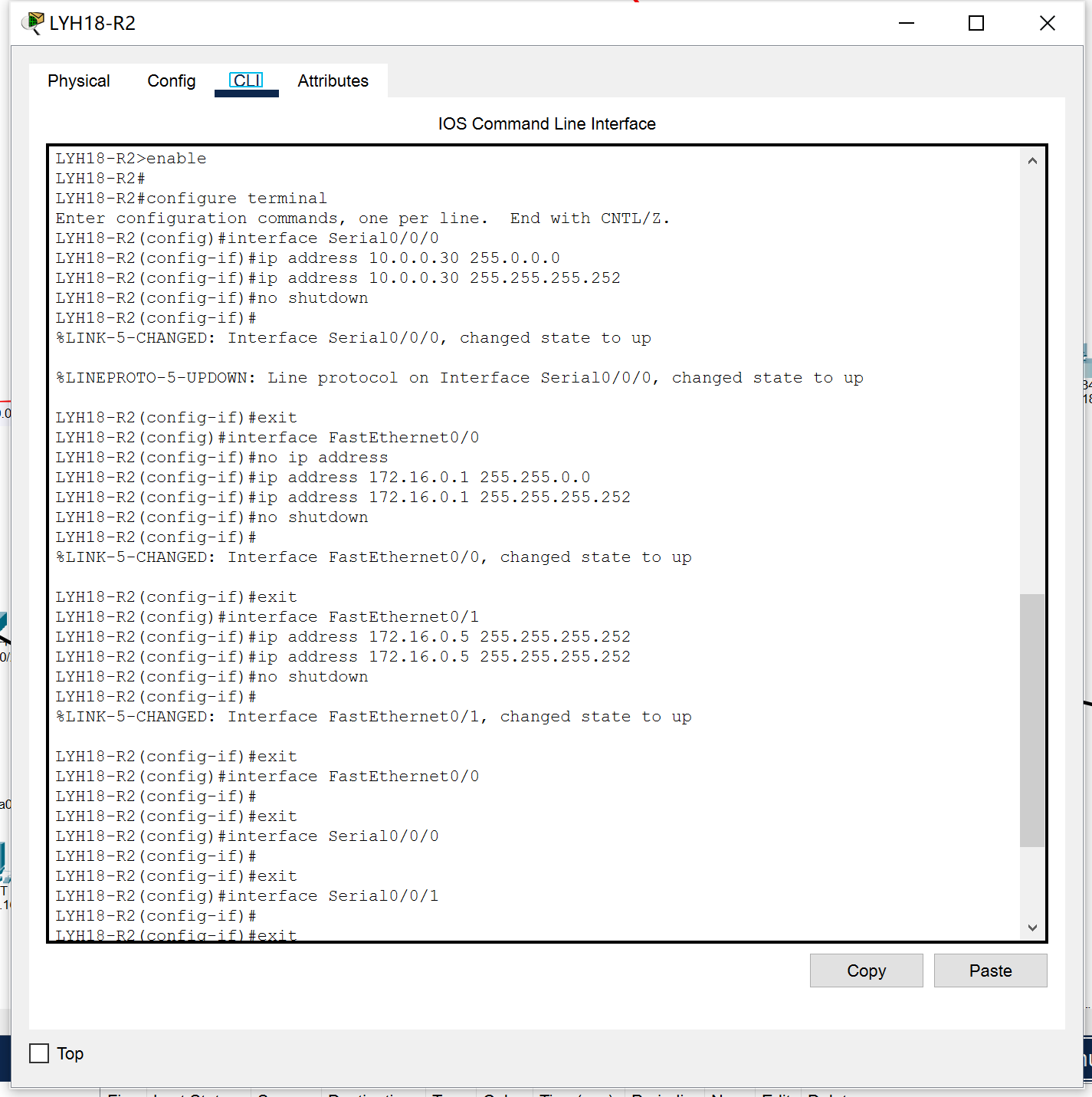


图 29中转路由接口配置

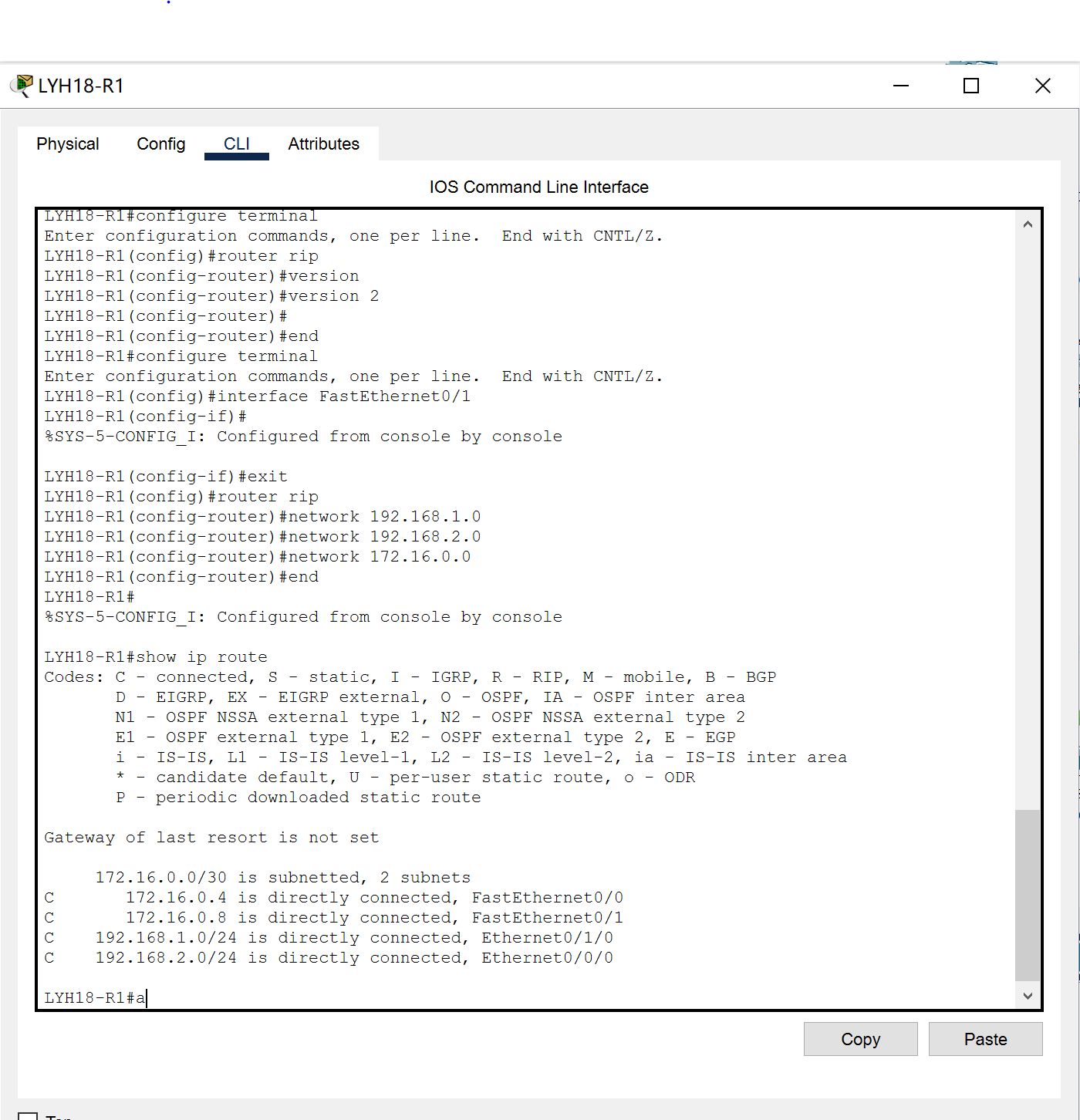


图 30路由rip配置

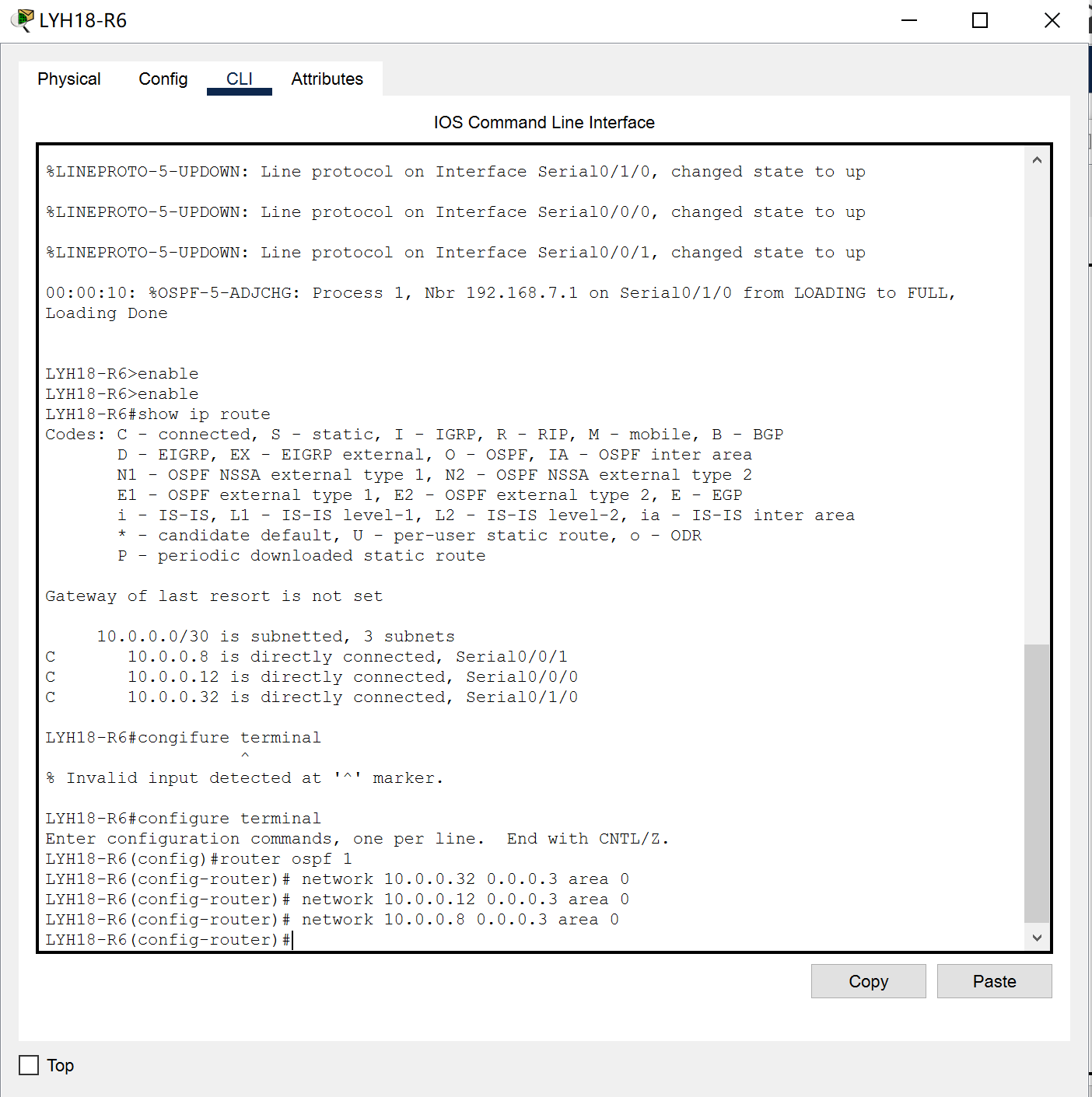


图 31路由ospf设置

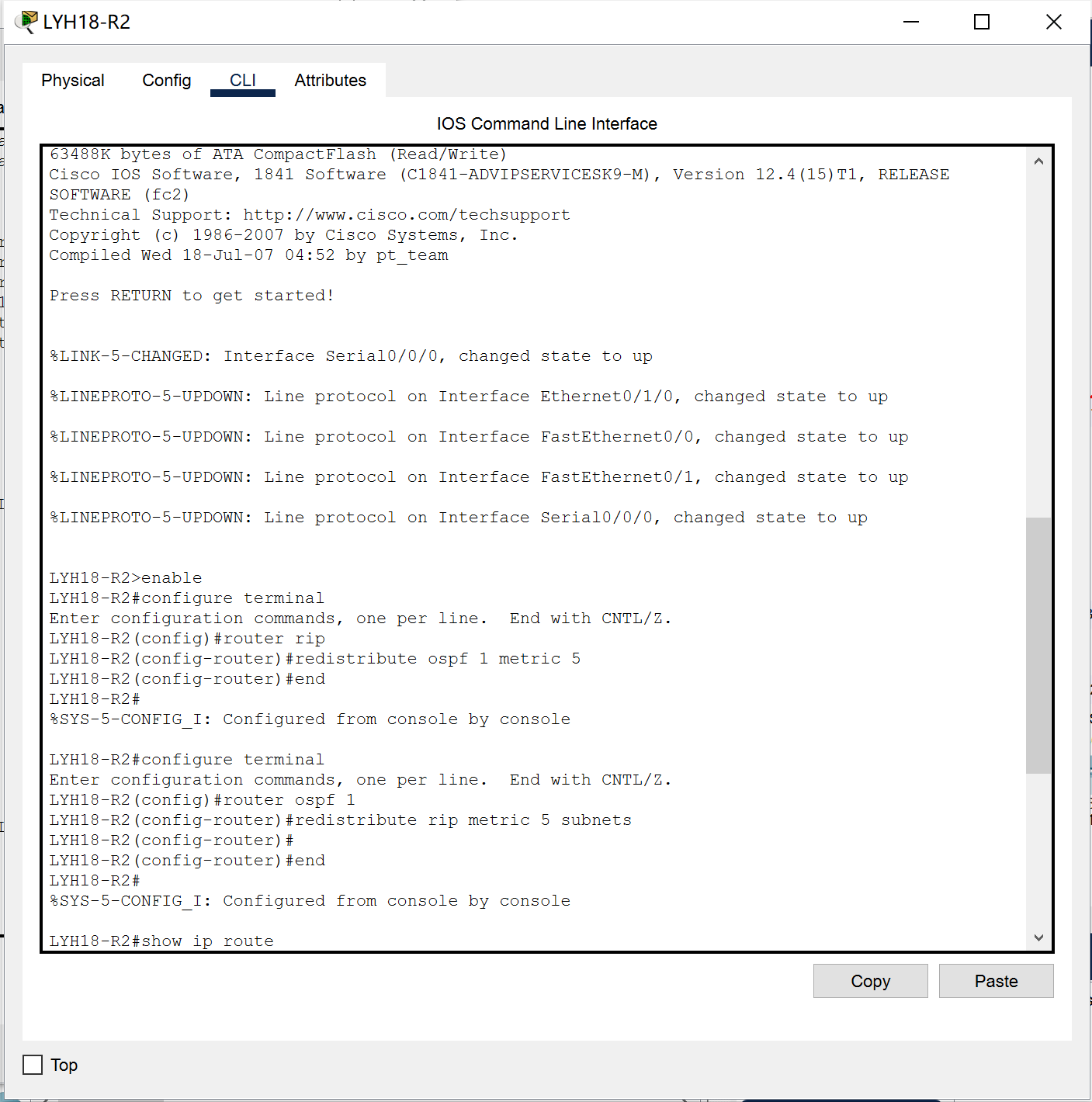


图 32中转路由转发设置

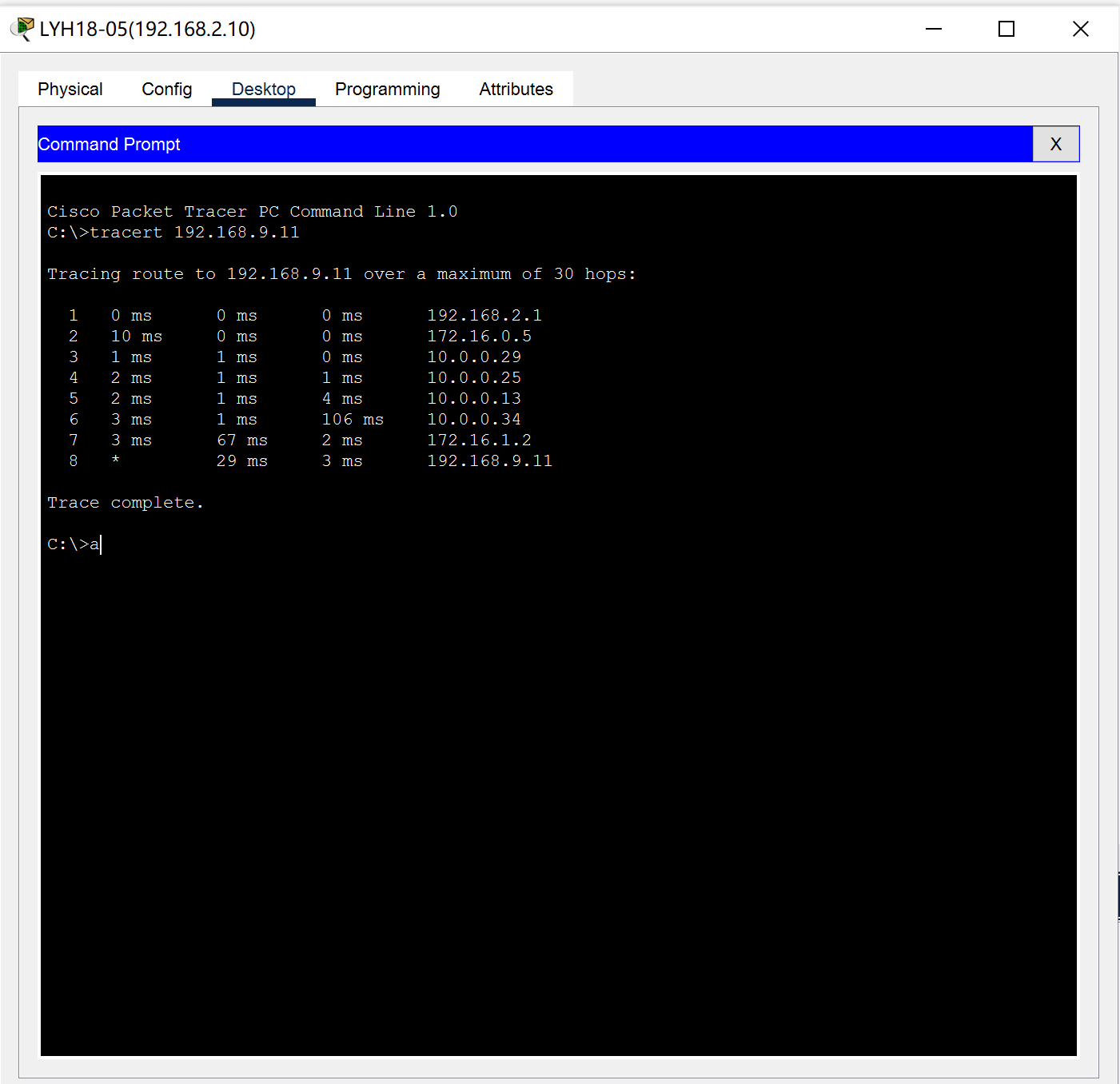


图 33 tracert测试

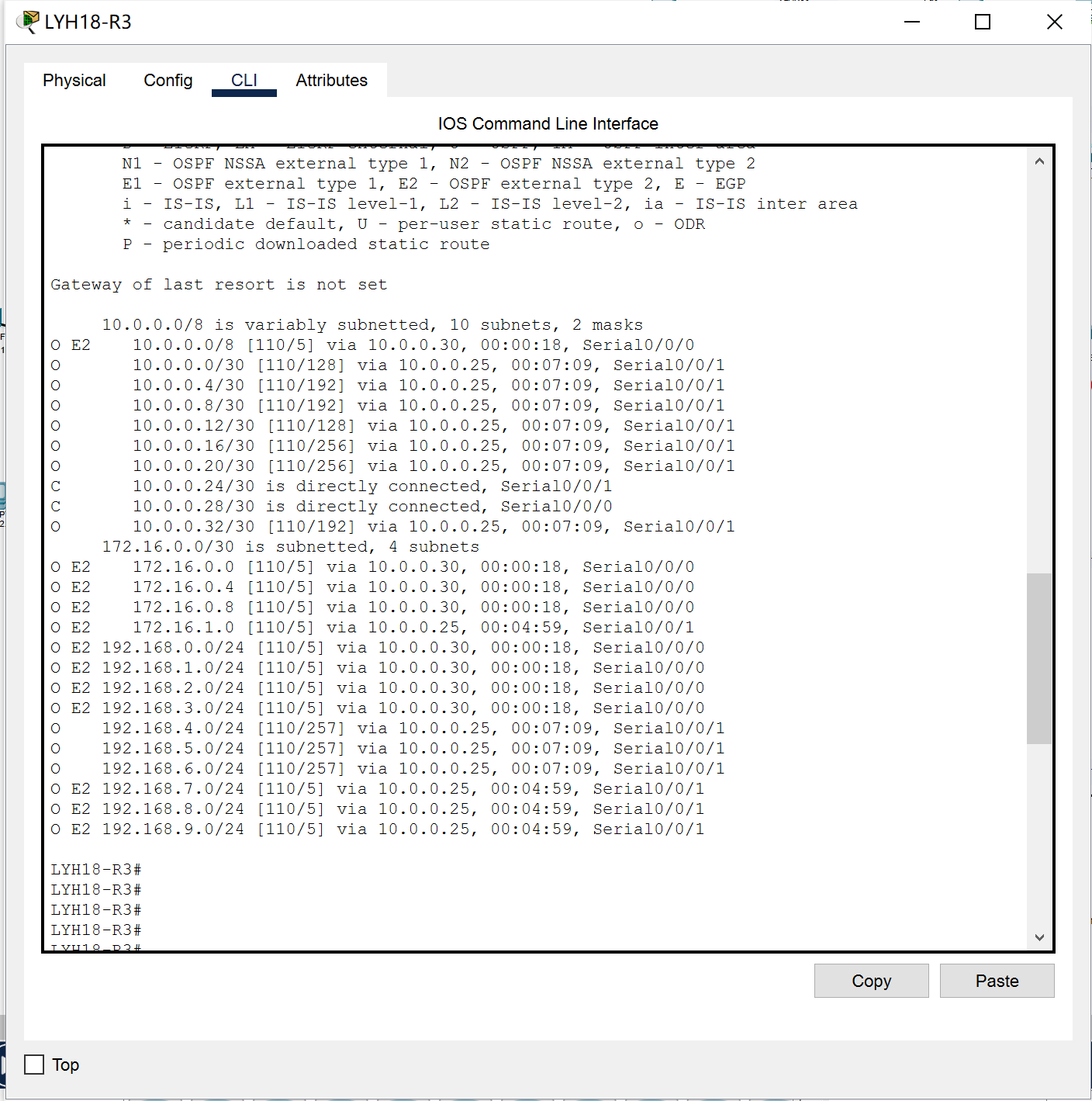


图 34 show ip route测试

配置中，红线（Serial DCE）使用ospf协议模拟远距离传输，虚线（Copper Cross-Over）使用rip协议模拟小规模子网近距离传输。其间路由为中转路由，使用两种协议并在两者之间转发信息。

## 2.3网络编程

## 2.3.1 开发环境及关键配置

**开发平台**

操作系统: Windows 11 家庭版

集成开发环境 (IDE): Microsoft Visual Studio 2022

**必备库及工具**

1.Npcap

本程序基于 pcap 库捕获网络数据包，需要安装以下：

Npcap: 使用最新版的 Npcap，它兼容 WinPcap，并在 Windows 10/11 上运行良好。

2.WinSock2

用于网络编程的 Windows 套接字库，提供底层网络访问功能。

3.PCAP 库

需要安装以下库文件（由 Npcap 提供）：

wpcap.lib

packet.lib

## 2.3.2 设计思路

### 1. ****功能目标****

本程序旨在实现对网络数据包的实时捕获，并解析以太网帧和 IP 数据包的头部信息。主要功能包括：

* 列出系统中可用的网络适配器。
* 捕获用户选择的适配器上的实时网络数据包。
* 解析数据包的以太网头部和 IP 头部信息，并打印关键信息（如 IP 地址、协议类型等）。

### 2. ****模块划分****

程序分为以下几个模块，各部分职责明确：

#### (1) 网络适配器管理模块

* 功能：获取和显示系统中所有可用的网络适配器，供用户选择。
* 设计思路：
  + 使用 pcap\_findalldevs 获取网络适配器列表。
  + 遍历链表结构 pcap\_if\_t，提取并显示适配器的名称和描述信息。

#### (2) 数据包捕获模块

* 功能：通过选定的网络适配器实时捕获数据包。
* 设计思路：
  + 使用 pcap\_open\_live 打开选定的网络适配器。
  + 使用 pcap\_loop 持续监听数据包。
  + 设置回调函数 packet\_handler 处理每个捕获到的数据包。

#### (3) 数据包解析模块

* 功能：解析捕获的数据包，提取并打印以太网帧和 IP 数据包的头部信息。
* 设计思路：
  + 以太网头部解析：
    - 从数据包的前 14 字节提取以太网头部，判断帧类型。
    - 仅处理 IPv4 数据包（以太网类型为 0x0800）。
  + IP 数据包头部解析：
    - 根据 IHL 提取 IP 头部长度。
    - 解析 IP 数据包的版本、服务类型、总长度、标志/片偏移等字段。
    - 使用 InetNtopA 将源和目的 IP 地址转换为可读的字符串格式。
  + 根据协议字段值，标识常见协议（如 ICMP、TCP、UDP）。

### 3. ****关键设计点****

#### (1) 数据结构设计

* **以太网头部结构**：ethernet\_header
  + 包括目的地址、源地址和类型字段。
* **IP 头部结构**：ip\_header
  + 包括版本、头部长度、服务类型、标志、TTL、协议、校验和等字段。

#### (2) 用户交互设计

* 允许用户选择适配器的编号，并提供输入验证以避免错误操作。

#### (3) 错误处理

* 包括以下场景的处理：
  + 未找到网络适配器时，提示用户检查安装环境。
  + 适配器打开失败时，提供错误信息。
  + 数据包捕获失败时，及时清理资源并退出。

### 4. ****未来扩展****

* 支持更多协议的解析（如 ARP、IPv6 等）。
* 增加数据包统计功能，计算传输速率等指标。
* 实现数据包过滤功能，仅捕获指定类型的数据包

## 2.3.3 工作流程

代码截图如下：



图 35 代码截图1

文本

描述已自动生成

图 36 代码截图2

### ****工作流程****

#### (1) 初始化

* 调用 WSAStartup 初始化 WinSock。
* 使用 pcap\_findalldevs 获取适配器列表，并通过控制台输出。

#### (2) 选择适配器

* 用户输入适配器编号。
* 程序检查输入有效性，跳转到指定适配器。

#### (3) 数据包捕获

* 使用 pcap\_open\_live 打开选定的适配器。
* 调用 pcap\_loop 开始捕获数据包。

#### (4) 数据包解析

* 每捕获一个数据包，调用 packet\_handler 进行解析。
* 解析以太网头部和 IP 头部信息，并输出到控制台。

#### (5) 释放资源

* 停止捕获后，释放适配器列表 (pcap\_freealldevs) 和捕获句柄 (pcap\_close)。
* 调用 WSACleanup 释放 WinSock 资源。

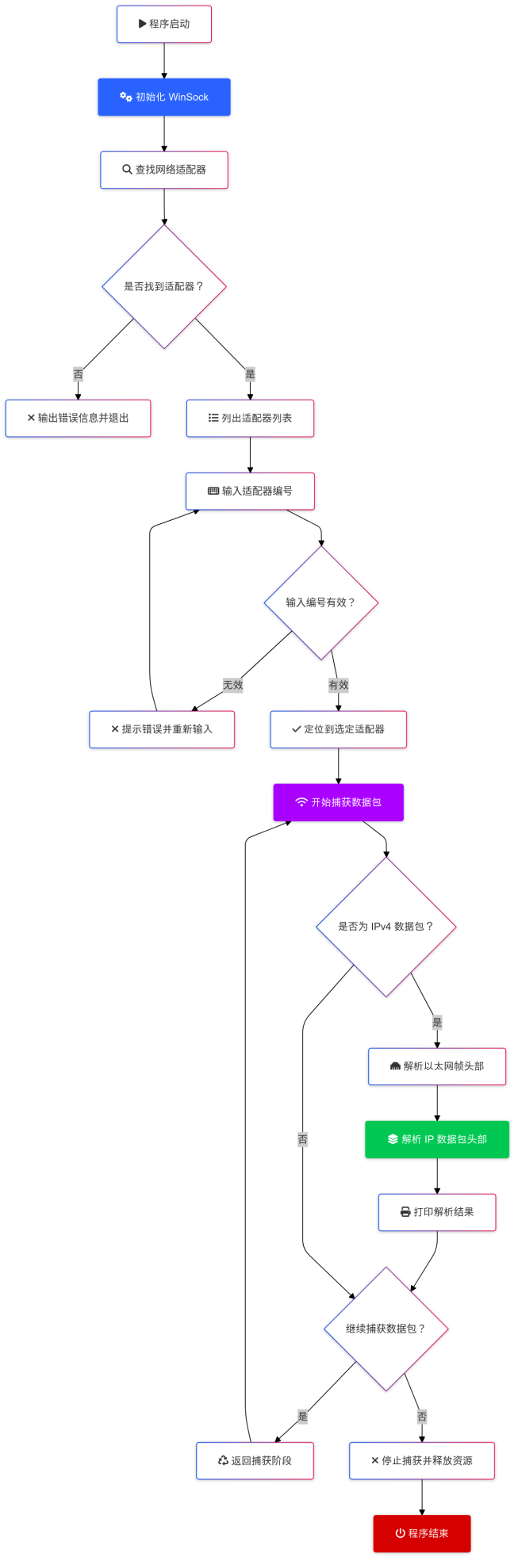


图 37流程图

## 2.3.4 运行截图及结果说明

图 38程序输出

以图中中间输出的数据包为例。该数据包是一个 IPv4 数据包，协议为 UDP（协议号 17）。其 IP 头部长度为 5 DWORDS（20 字节），TOS（服务类型）为 4，表示服务质量（QoS）需求，这里表示需求吞吐量。数据包的总长度为 962 字节，标识符为 61990，表示该数据包在可能的分片情况下的标识符。该数据包没有设置分片标志，且片偏移为 0，说明它是一个完整的数据包，没有被分片。

TTL（生存时间）为 127，表示该数据包最多可以经过 127 个路由器，如果超过这个数量，它将被丢弃。源 IP 地址为 192.168.113.1，目的 IP 地址为 10.136.9.152，表示发送和接受的IPv4地址。校验和为 0x37，表示 IP 头部的完整性校验值。

1. **实验总结与心得体会**

通过这次计算机网络课程设计实验，我加深了对计算机网络原理的理解，并提升了实操能力。主要收获包括：

1. 网络命令应用：熟练掌握了 ipconfig、ping、netstat 等命令，能快速进行网络诊断和配置检查。
2. 交换机与路由器配置：通过 Packet Tracer 仿真，掌握了 VLAN 配置、静态路由和动态路由的设置，并成功验证了配置效果。
3. 网络编程实践：实现了数据包捕获和解析，熟悉了 c++捕获网络信息的流程，熟悉了ip首部的结构。
4. 解决问题能力：实验中遇到的配置和调试问题通过查阅资料和讨论解决，锻炼了我的问题分析和解决能力。

实验加深了我对网络技术的理解，提高了我的实践能力和动手能力。