实验报告二

|  |  |  |
| --- | --- | --- |
| **姓名：彭彦杰** | **时间：2025.05.19** | **评分：** |

# ARP实验

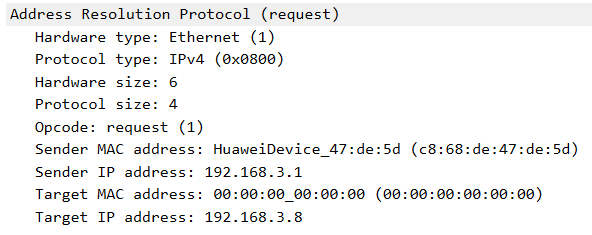
## 在本地电脑搭建环境进行ARP实验，要求能实现查看ARP请求与回应

1. 清空ARP缓存

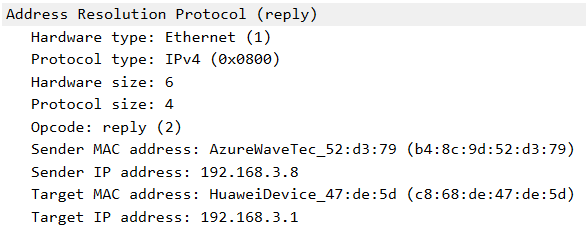
arp -d \* # 清除所有ARP记录

2. 使用Wireshark捕获ARP流量

ARP请求（广播）



ARP响应（单播）



## 分析ARP请求与回应的报文结构

ARP请求字段

|  |  |  |
| --- | --- | --- |
| **字段** | **值** | **说明** |
| Sender MAC | c8:68:de:47:de:5d | 请求方MAC地址 |
| Sender IP | 192.168.3.1 | 请求方IP地址 |
| Target MAC | 00:00:00:00:00:00 | 全0（未知） |
| Target IP | 192.168.3.8 | 目标IP地址 |
| Opcode | 1 (Request) | 操作码（请求） |

ARP响应字段

|  |  |  |
| --- | --- | --- |
| **字段** | **值** | **说明** |
| Sender MAC | b4:8c:9d:52:d3:79 | 响应方MAC地址 |
| Sender IP | 192.168.3.8 | 响应方IP地址 |
| Target MAC | c8:68:de:47:de:5d | 请求方MAC地址 |
| Target IP | 192.168.3.1 | 请求方IP地址 |
| Opcode | 2 (Reply) | 操作码（响应） |

## 分析免费ARP的报文结构特征

关键字段

|  |  |
| --- | --- |
| **字段** | **值** |
| Sender IP | 与Target IP相同（如192.168.3.8） |
| Target IP | 与Sender IP相同（如192.168.3.8） |
| Opcode | 1（请求，但无需响应） |

# HTTP/HTTPS实验

## 访问网站并抓包

1. Wireshark过滤器设置

host www.qq.com or host [www.sangfor.com.cn](http://www.sangfor.com.cn)

2. 清除DNS缓存

ipconfig /flushdns

3. 访问HTTP网站（腾讯网）与HTTPS网站（深信服官网）

## 分析DNS解析过程及报文结构

**1. 本地DNS缓存**

* 客户端首先检查本地缓存（浏览器、操作系统）是否有域名解析结果。
* 若缓存命中且未过期（TTL有效），直接使用缓存结果。

**2. 递归查询流程**

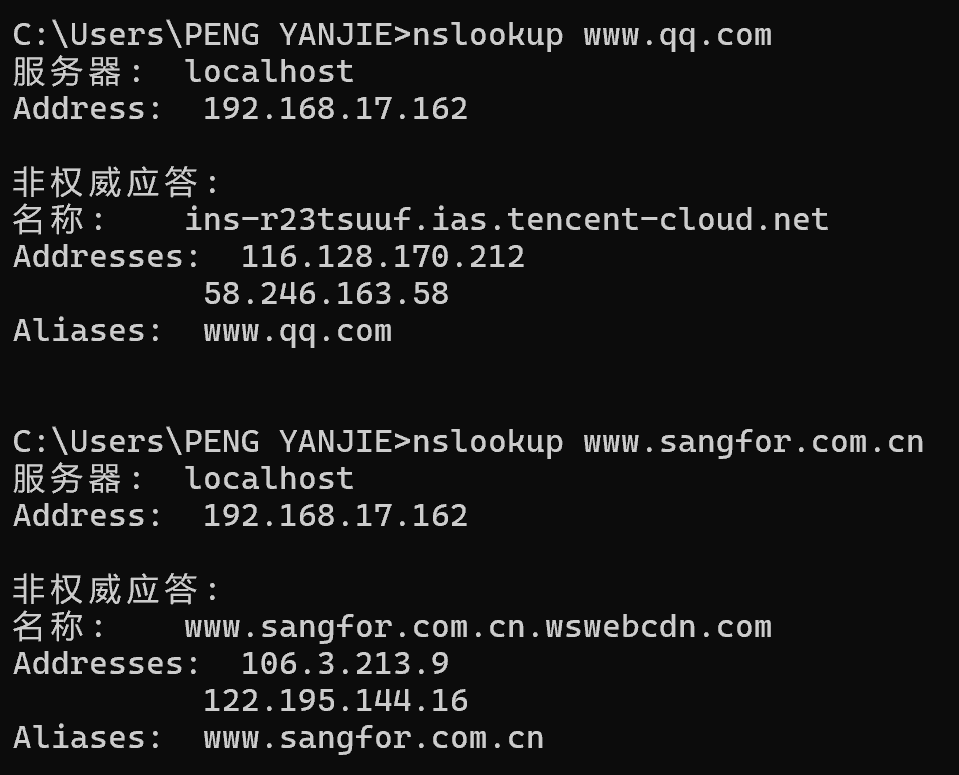
客户端向本地DNS服务器发送查询请求。

本地DNS服务器依次查询：

* 根DNS → 获取.com的顶级域DNS地址。
* 顶级域DNS → 获取qq.com的权威DNS地址。
* 权威DNS → 返回www.qq.com的A记录。

本地DNS缓存结果并返回给客户端。

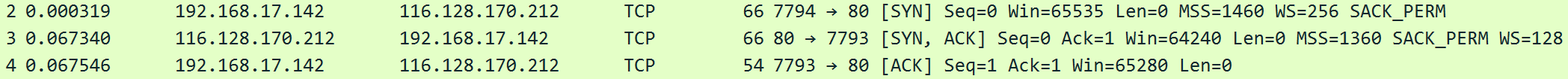
**3. 使用 nslookup 命令验证**



## 分析TCP连接与断开过程

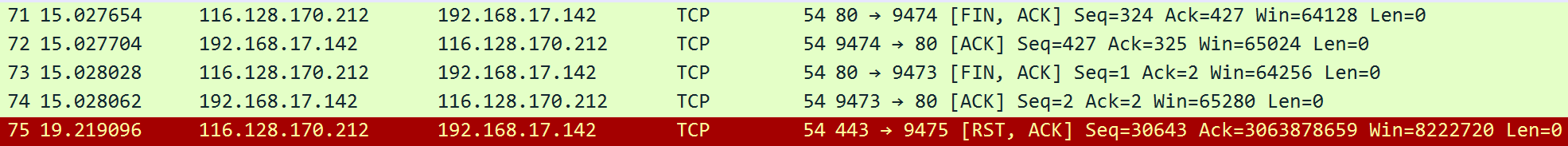
**1. TCP 三次握手**

访问腾讯网（HTTP）的三次握手：



* 第1步（SYN）：PC（源IP，如192.168.17.142）向腾讯网IP发送SYN包，序列号Seq=0，标志位SYN=1，目标端口80。
* 第2步（SYN+ACK）：腾讯网服务器返回SYN+ACK包，Seq=0（服务器序列号），Ack=1（确认PC的序列号+1），标志位 SYN=1, ACK=1。
* 第3步（ACK）：PC发送ACK包，Seq=1，Ack=1，标志位ACK=1，连接建立完成。

**2. TCP 四次挥手**



第1步（FIN）：PC发送FIN包，标志位FIN=1，请求断开连接。

第2步（ACK）：服务器返回ACK包，确认收到断开请求。

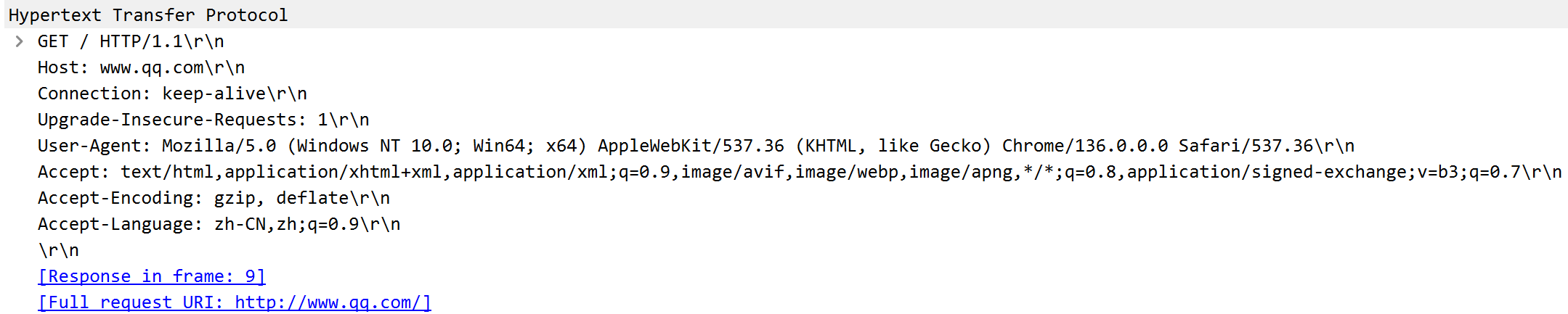
第3步（FIN）：服务器发送FIN包，标志位FIN=1，表示服务器已准备好断开。

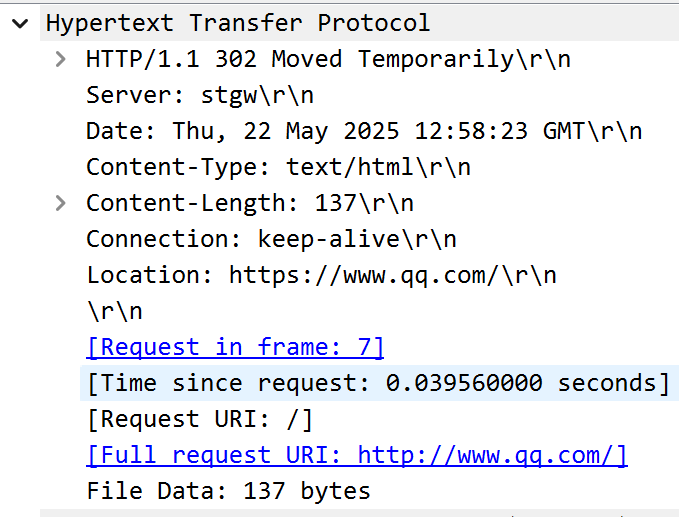
第4步（ACK）：PC返回ACK包，确认服务器的断开请求，连接彻底关闭。

强制关闭浏览器，捕获到RST包（标志位RST=1），表示连接被重置（非正常断开）。

## HTTP/HTTPS请求与响应报文

**1. 分析HTTP报文（腾讯网）**





**请求行：**

GET / HTTP/1.1 # 请求方法为GET，URI为根路径，HTTP版本1.1

**请求头字段：**

Host: www.qq.com：目标主机名。

User-Agent：浏览器类型（Chrome）

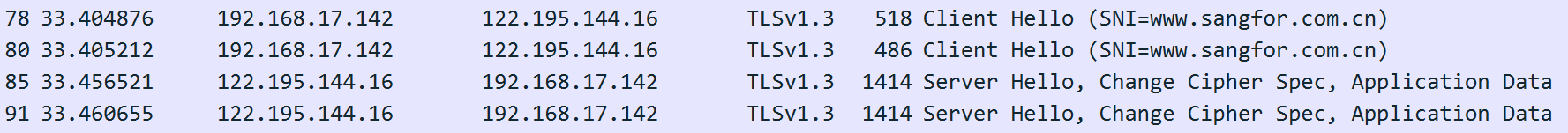
Accept：支持的响应内容类型（text/html）

**响应报文：**

状态行：HTTP/1.1 302 Moved Temporarily（状态码 302 表示临时重定向）

响应头字段：Content-Type: text/html（内容类型为 HTML），Content-Length（响应体长度137）

**2. 分析HTTPS报文（深信服官网）**



**握手阶段：**

* 客户端发送Client Hello，包含支持的TLS版本、加密算法列表
* 服务器返回Server Hello，选择TLS版本和加密算法，发送证书证书（含公钥）
* 客户端验证证书后生成密钥，用服务器公钥加密后发送

**3. HTTP报文关键字段**

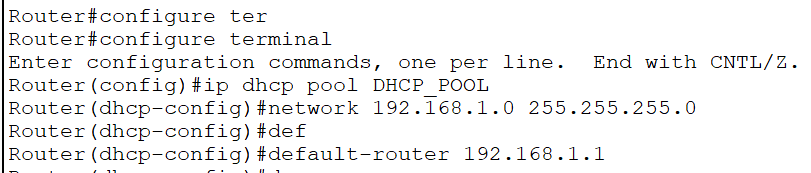
HTTP报文关键字段

|  |  |  |
| --- | --- | --- |
| **请求头** | **响应头** | **说明** |
| Host | Server | 目标主机/服务器软件信息 |
| User-Agent | Content-Type | 客户端标识/响应数据类型 |
| Cookie | Set-Cookie | 客户端/服务器管理会话状态 |
| Accept | Content-Length | 客户端支持的数据类型/数据长度 |

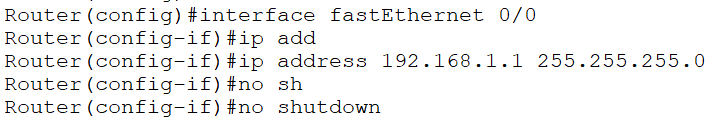
# DHCP实验

## 配置DHCP服务器（路由器）

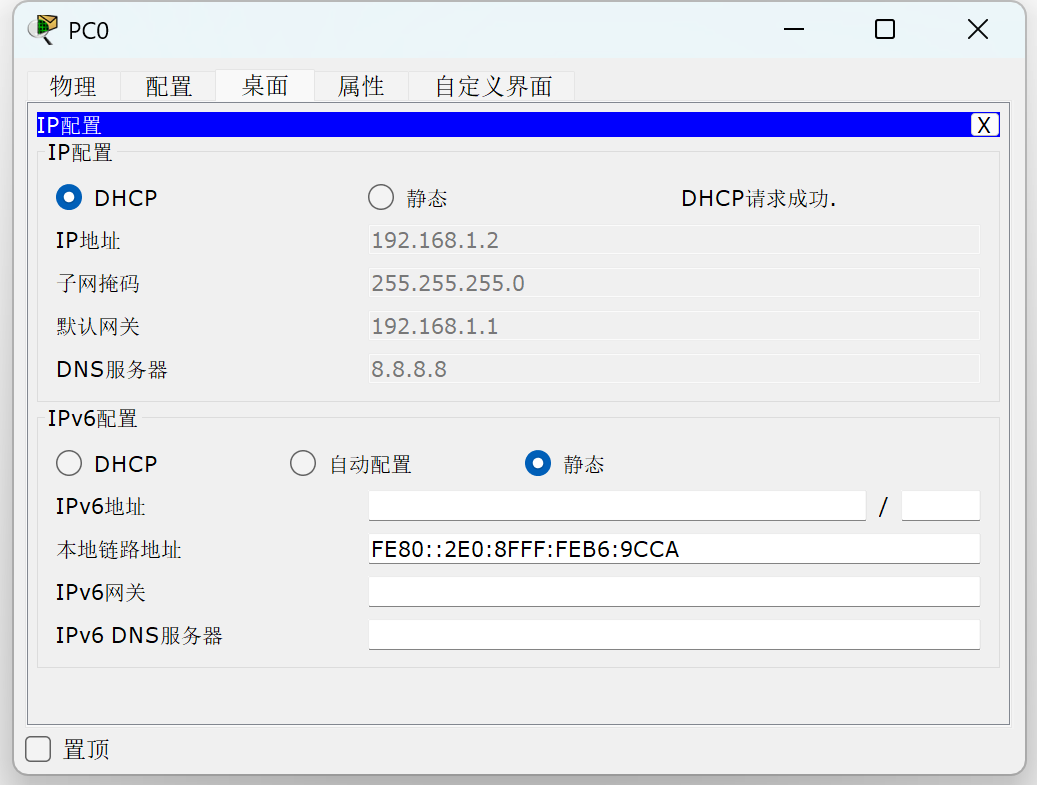
1. 创建DHCP地址池



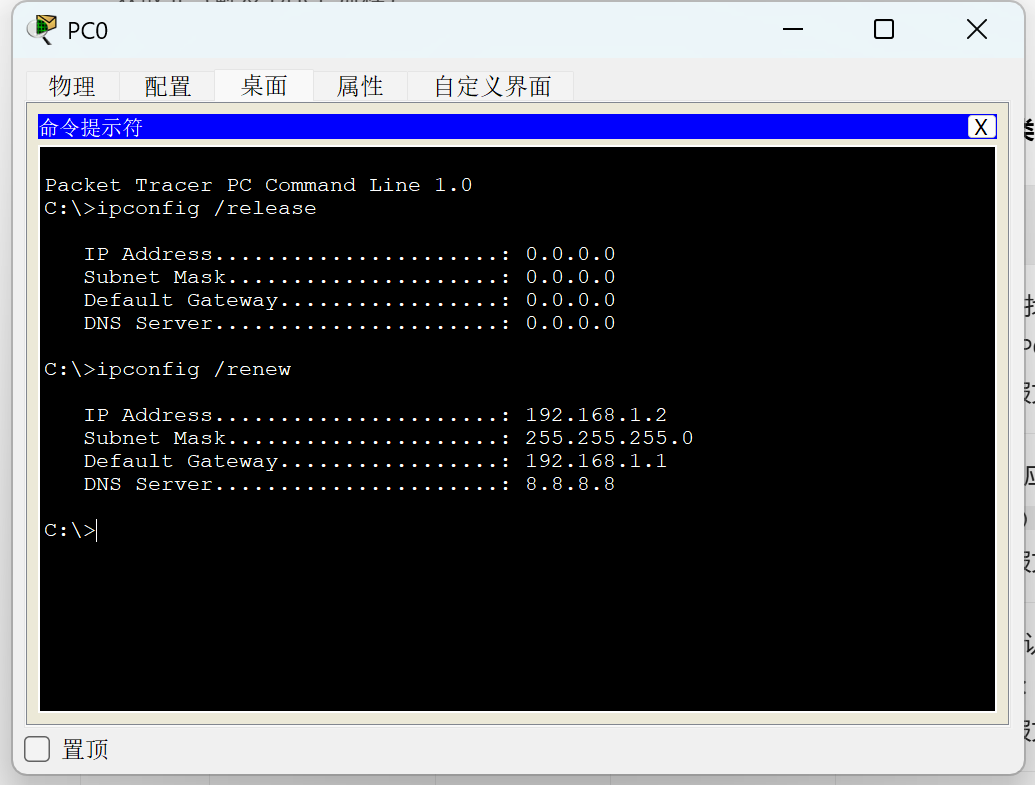
2. 配置路由器接口IP（作为DHCP网关）

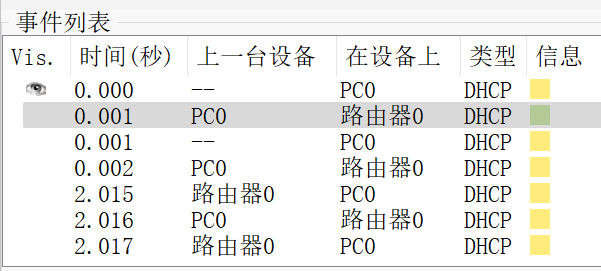


3. 配置客户端PC（自动获取IP）



## 抓包



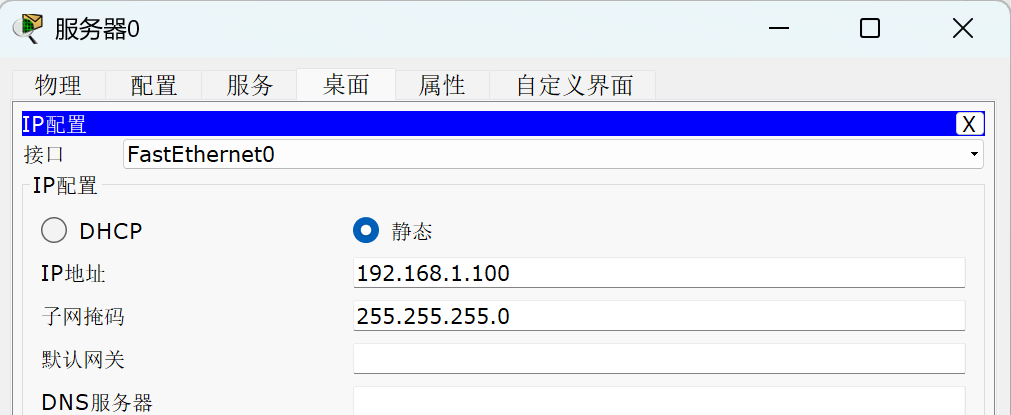


分析DHCP报文

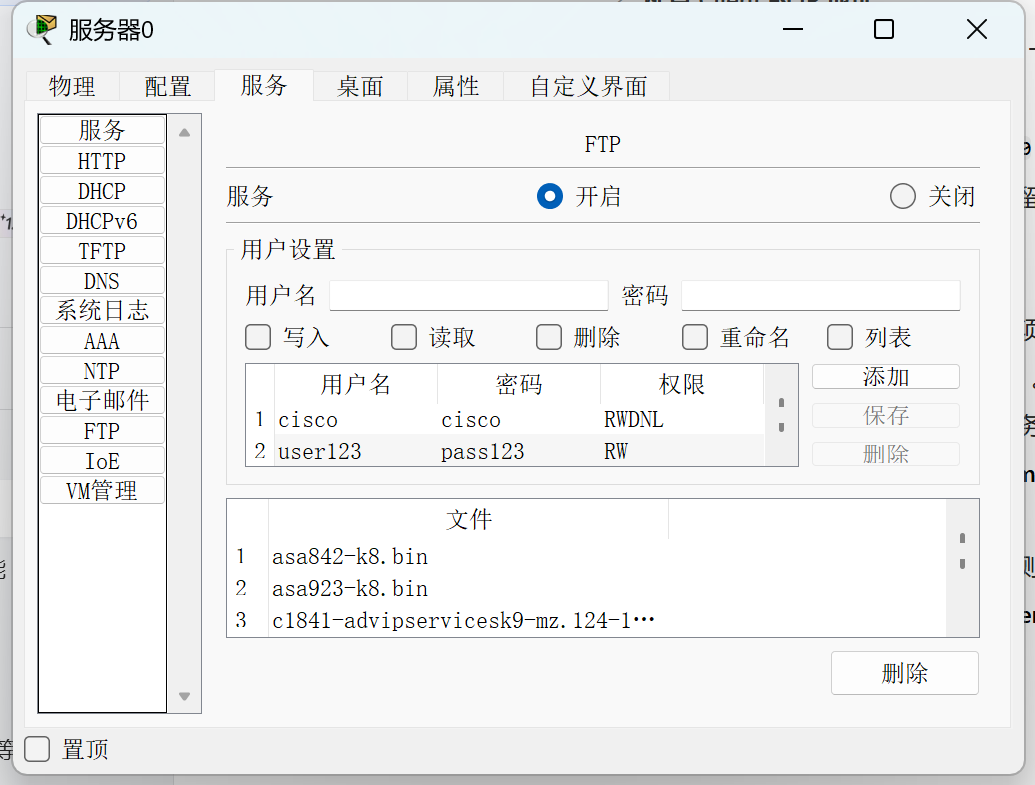
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **阶段** | **报文类型** | **源IP** | **目的IP** | **关键字段分析** |
| 1. 发现 | DHCPDISCOVER | 0.0.0.0 | 255.255.255.255 | - 客户端广播寻找DHCP服务器  - Client MAC：PC 的MAC地址  - Option 53：报文类型 = 1（Discover） |
| 2. 提供 | DHCPOFFER | 192.168.1.1 | 255.255.255.255 | - 服务器回应  - Your (client) IP：分配的IP（192.168.1.2）  - Option 53：报文类型 = 2（Offer） |
| 3. 请求 | DHCPREQUEST | 0.0.0.0 | 255.255.255.255 | - 客户端广播确认接受 IP  - Requested IP：服务器提供的 IP  - Option 53：报文类型 = 3（Request） |
| 4. 确认 | DHCPACK | 192.168.1.1 | 255.255.255.255 | - 服务器广播确认分配  - Option 53：报文类型 = 5（Ack） |

# FTP实验

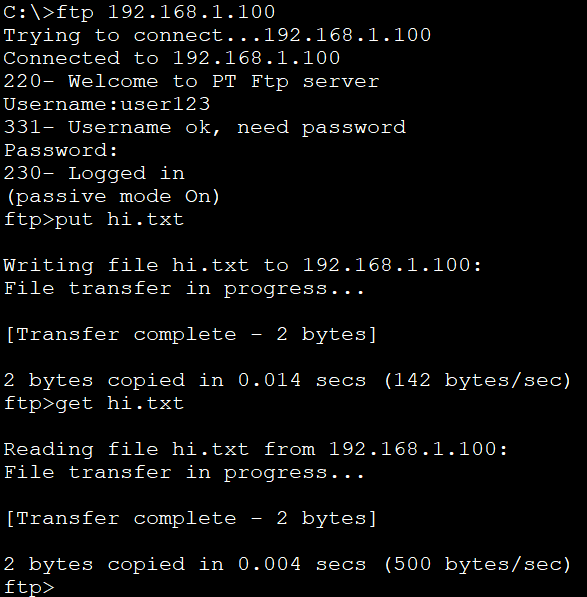
## 搭建FTP环境

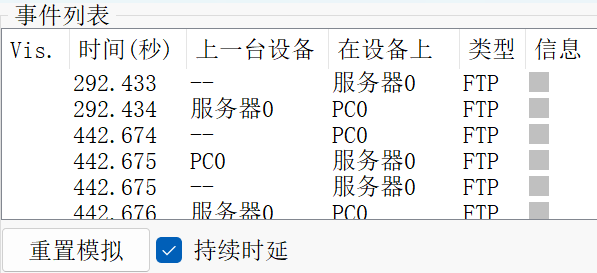






## FTP文件传输抓包





## 分析FTP协议

**1. 控制连接建立阶段（TCP三次握手）**

报文1：Client → Server（SYN）

* 源端口：随机临时端口（如 1030）
* 目的端口：21（FTP 控制端口）
* 标志位：SYN=1

报文2：Server → Client（SYN+ACK）

* 源端口：21
* 目的端口：1031
* 标志位：SYN=1，ACK=1

报文3：Client → Server（ACK）

* 标志位：ACK=1

**2. 登录认证阶段（控制连接）**

报文4：Client → Server（USER 命令）

* FTP 命令：USER user1（发送用户名）
* 服务器响应：331 User name okay, need password（状态码331表示需要密码）

报文5：Client → Server（PASS命令）

* FTP 命令：PASS pass123（发送密码，明文传输）
* 服务器响应：230 Login successful（状态码230表示登录成功）

**3. 文件上传阶段（被动模式数据连接）**

步骤1：Client发送PASV命令请求被动模式

步骤2：Client建立数据连接（TCP三次握手）

步骤3：传输文件数据

报文6：Client → Server（put hi.txt 命令）

* FTP命令：put hi.txt（请求上传文件）

报文7：Client → Server（文件数据传输）

* 通过数据连接上传hi.txt内容。
* 服务器响应：226 Transfer complete

4. 文件下载阶段（被动模式数据连接）

步骤1：Client 发送PASV命令（若数据连接已断开）

步骤2：Client 建立数据连接

步骤3：传输文件数据

报文 8：Client → Server（get hi.txt 命令）

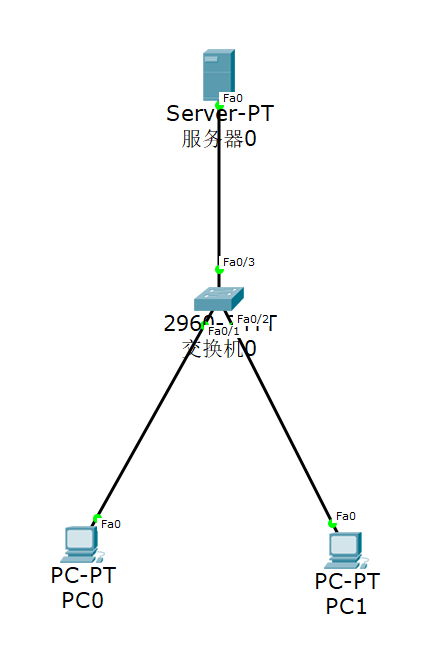
* FTP 命令：get hi.txt（请求下载文件）

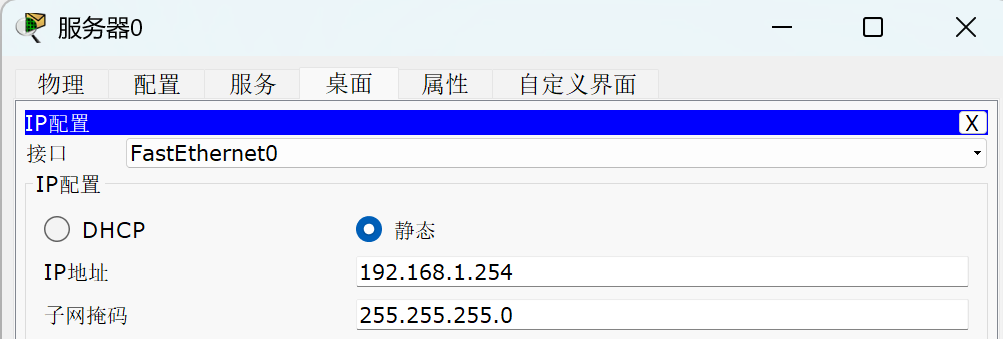
报文 9：Server → Client（文件数据传输）

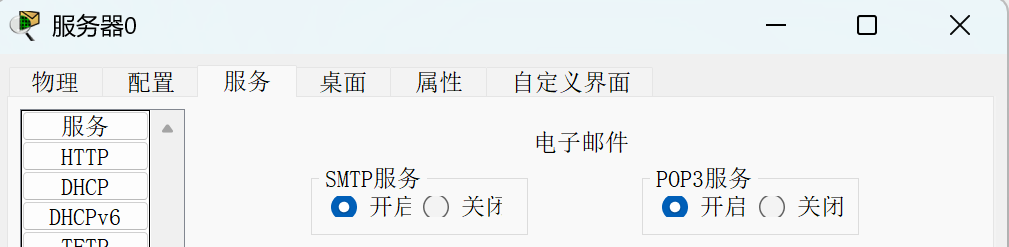
* 传输hi.txt的内容，以TCP分段形式发送。
* 服务器响应：226 Transfer complete（状态码 226 表示传输完成）

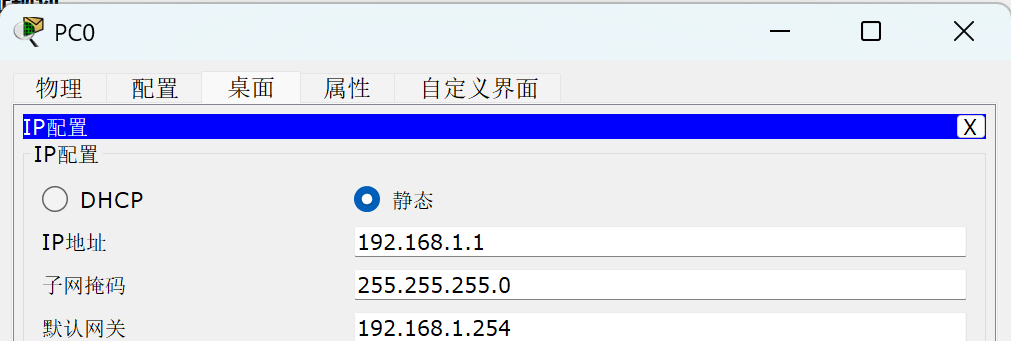
# SMTP和POP3实验

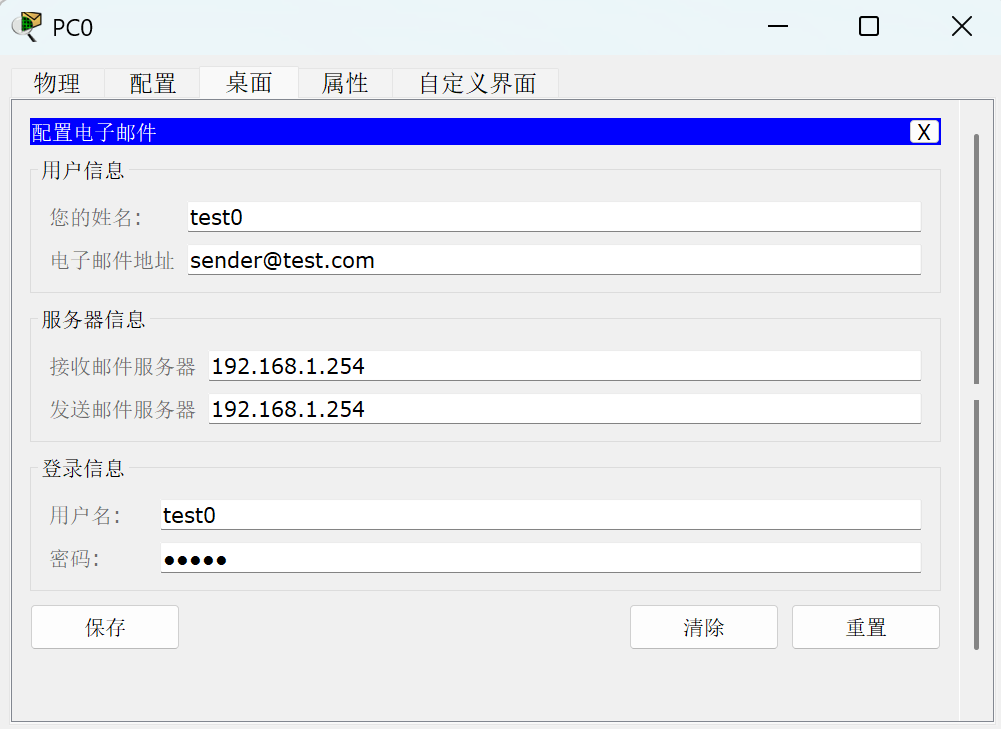
## 搭建环境

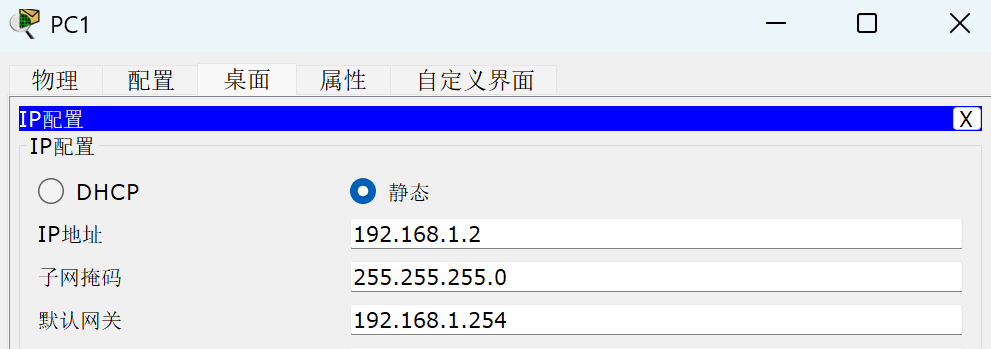


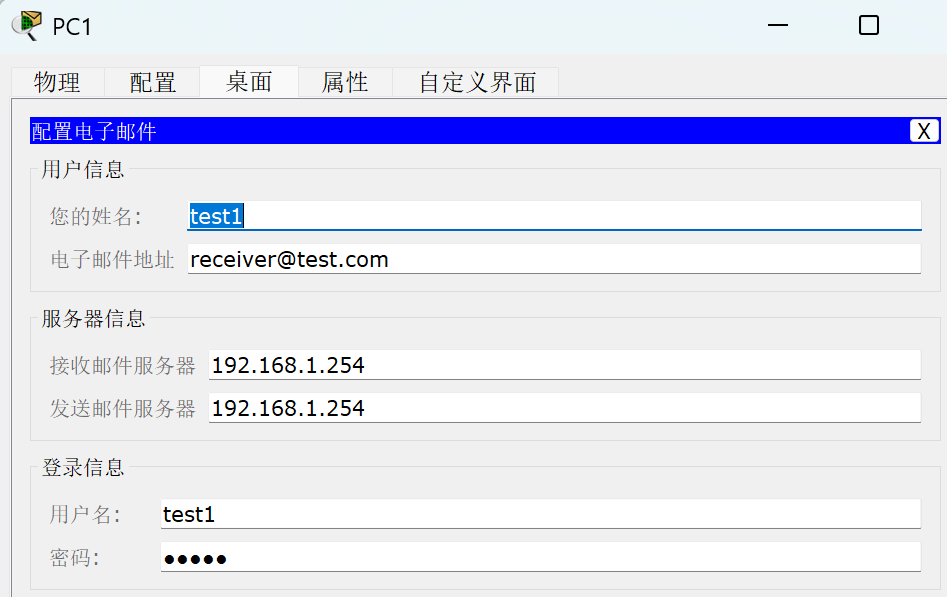












## 发送和接收邮件

**1. SMTP 协议基于TCP 25端口**

**阶段1：客户端与服务器建立TCP连接**

* 数据包类型：TCP SYN、SYN-ACK、ACK（三次握手）。

**阶段2：SMTP 命令交互**

数据包内容：客户端通过明文发送SMTP命令，服务器返回状态码。

客户端命令：

* HELO client.com：客户端标识自己（client.com 为任意标识）。
* MAIL FROM: <sender@test.com>：指定发件人。
* RCPT TO: <receiver@test.com >：指定收件人。
* DATA：开始传输邮件内容。
* 邮件内容（主题、正文），以 .\r\n 结束。
* QUIT：断开连接。

服务器响应：

* 250 OK：命令成功。
* 354 Start mail input：准备接收邮件内容。
* 221 Service closing transmission channel：连接关闭。

**阶段 3：TCP 连接释放**

四次挥手：客户端和服务器交换 FIN、ACK 包，断开连接。

**2. POP3协议基于TCP 110端口**

**阶段1：客户端与服务器建立TCP连接**

* 数据包类型：TCP SYN、SYN-ACK、ACK（三次握手）。

**阶段 2：POP3 命令交互**

客户端命令：

* USER test1：发送用户名。
* PASS test1：发送密码。

LIST：列出邮件列表。

RETR 1：检索第 1 封邮件内容。

QUIT：退出会话。

服务器响应：

+OK：命令成功（如 +OK User name accepted）。

-ERR：错误（如密码错误）。

邮件内容以明文返回，包含发件人、主题、正文等。