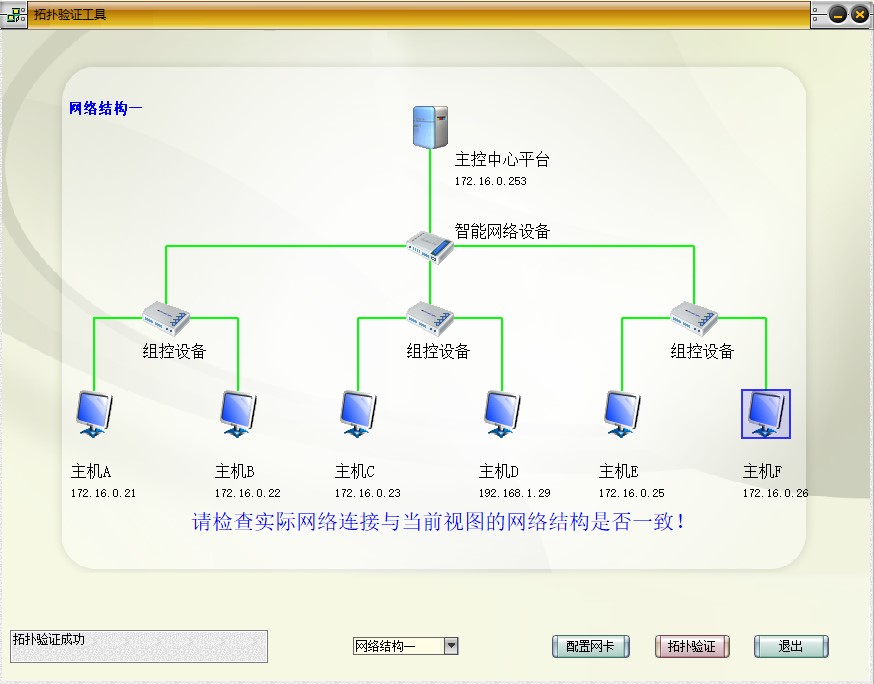
**实验五 超文本传输协议**

**报告成绩：\_\_\_\_\_\_\_\_\_ 指导教师：\_\_\_\_\_\_\_\_\_**

**姓名：黄耀增 学号：20231302231 班级：计算机科学与技术23-1**

1. **实验目的**
2. 掌握HTTP的报文格式；
3. 掌握HTTP的工作原理；
4. 掌握HTTP常用方法。
5. **实验环境（5分）**
6. 系统环境：Windows 10 专业版； TCP&IP协议实验平台； Wireshark.
7. 网络环境：网络拓扑结构（一）

本次实验主机F

【网络拓扑图及IP地址分配】

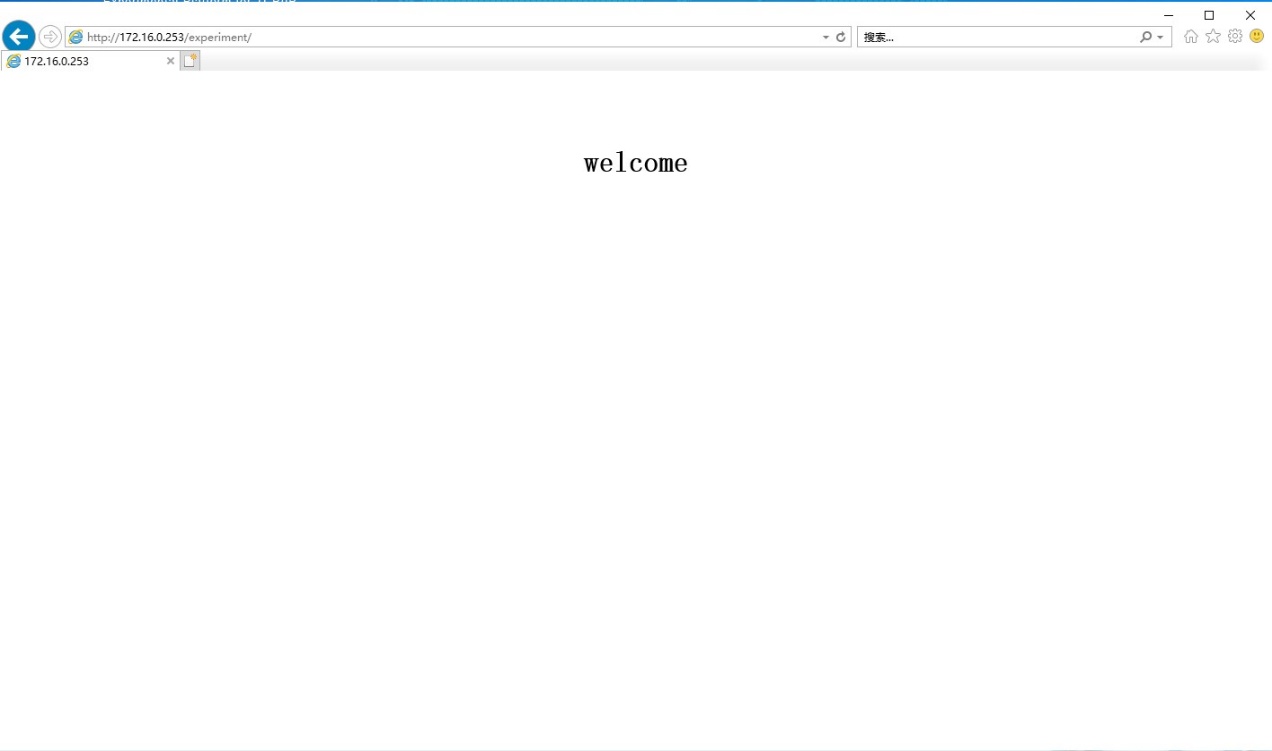
1. **实验内容及结果分析（65分）**
2. **练习一：页面访问（以下内容可以根据实际实验过程进行调整，要求完成、顺序记录，并进行结果分析）**
3. 捕获并分析HTTP协议数据包的内容
4. 本练习使用HTTP协议的哪种方法？简述这种方法的作用。
5. 根据本练习的报文内容，填写下表。

|  |  |
| --- | --- |
| 主机名 | 172.16.0.253 |
| URL | http://172.16.0.253/experiment/ |
| 服务器类型 | Apache/2.2.3 (CentOS) |
| 传输文本类型 | text/html; charset=GBK |
| 访问时间 | Fri, 30 May 2025 01:50:04 GMT |

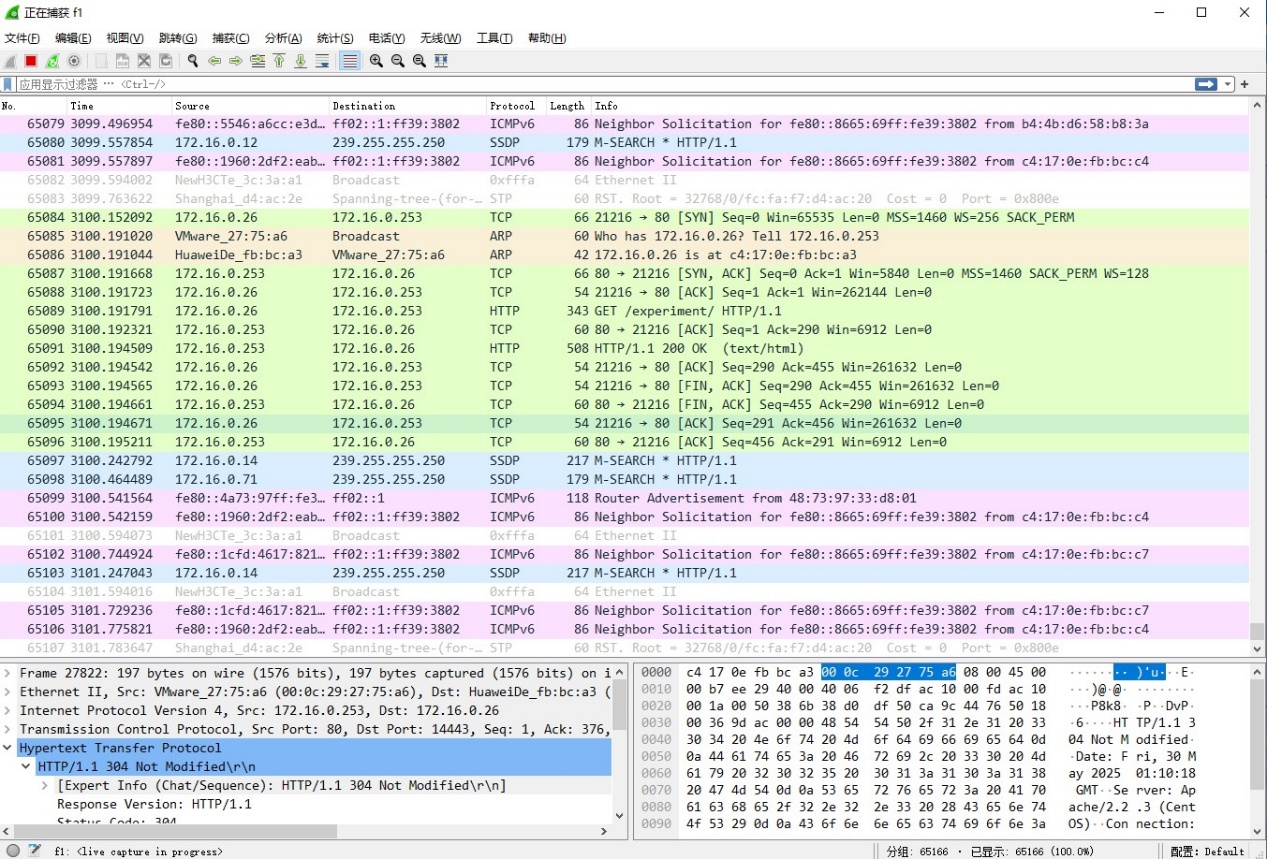
1. 参考“会话分析”视图显示结果，绘制此次访问过程的报文交互图（包括TCP协议）。
2. 简述TCP协议和HTTP协议之间的关系。
3. 主机A发送的报文的目的MAC地址和目的IP地址的含义是什么？

**实验过程：**

打开拓扑验证工具，选择网络结构一，配置网卡f1，进行拓扑验证。

首先本主机F清空IE缓存，打开WireShark抓包工具，开始抓包，然后打开IE浏览器在地址输入**<http://172.16.0.253/experiment>**，得到如下“图1.1 练习一网址网络页面”的效果：

**图1.1 练习一网址网络页面**

 在WireShark抓包工具内可以看到捕获到两个HTTP数据包，如下“图1.2 练习一数据包抓包”所示：

**图1.2 练习一数据包抓包**

图形用户界面, 应用程序

AI 生成的内容可能不正确。 分别打开这两个数据包，如下“图1.3 练习一HTTP抓包-1”、“图1.4 练习一HTTP抓包-2”所示：

图形用户界面, 文本, 应用程序, 电子邮件

AI 生成的内容可能不正确。**图1.3 练习一HTTP抓包-1**

**图1.4 练习一HTTP抓包-2**

通过观察上述两张图，我们得到如下这个“表1.1 练习一报文内容”

**表1.1 练习一报文内容**

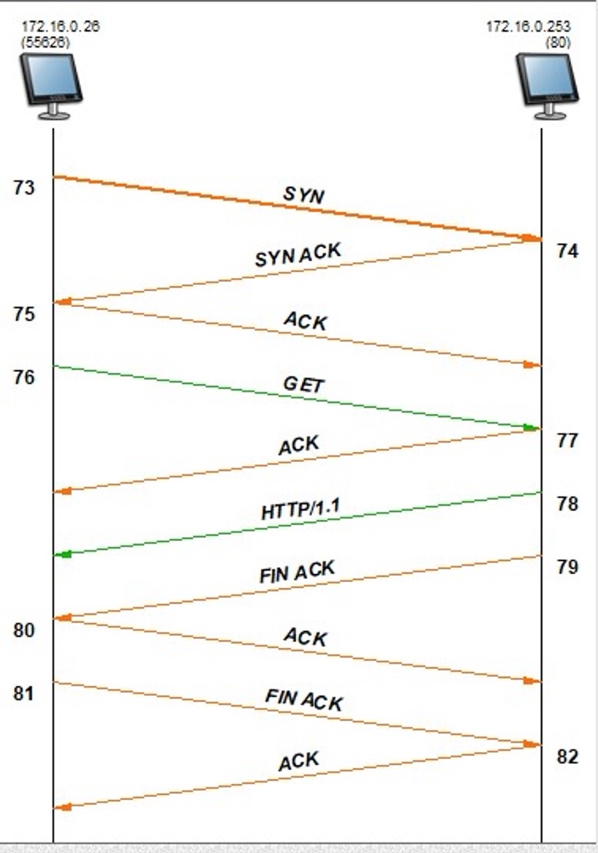
|  |  |
| --- | --- |
| 主机名 | 172.16.0.253 |
| URL | http://172.16.0.253/experiment/ |
| 服务器类型 | Apache/2.2.3 (CentOS) |
| 传输文本类型 | text/html; charset=GBK |
| 访问时间 | Fri, 30 May 2025 01:50:04 GMT |

根据上述捕获数据包的内容回答练习一要求的问题：

(1)第一个数据包指的是报文完整地发起了一个 HTTP 1.1 的 GET 请求，目标路径为 /experiment/，没有包含请求实体（body），仅有请求头。第一个数据包指的是服务器以 HTTP/1.1 规范返回了 200 OK 的响应，正文为一段 GBK 编码的 HTML，长度 194 B。

(2)本练习使用了HTTP协议的GET方法。作用：GET 请求会将请求参数附加在 URL 之后，直接从服务器获取资源而不会修改服务器上任何数据。

(3)如上表1.1 练习一报文内容。

(4) 报文交互图如下“图1.5 练习一报文交互图”所示：

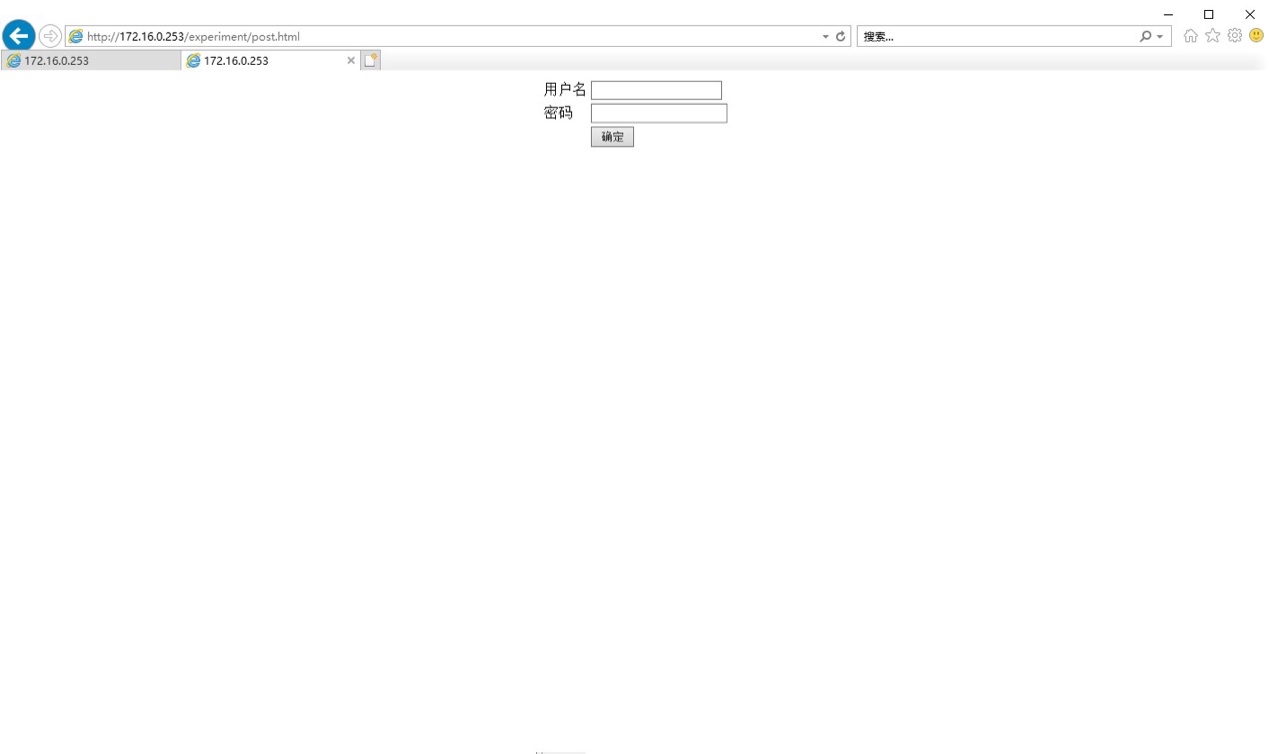
**图1.5 练习一报文交互图**

(5)TCP是“运输层”协议，为上层“应用层”的 HTTP协议提供可靠的服务。HTTP 负责告诉服务器我要哪份资源、同时告知服务器如何编码、语言、缓存策略等信息，以及服务器如何组织返回的内容。TCP 负责保证应用层发出的每一个字节都能“无差错、按序” 到达对端，若出现丢包、乱序、重复等问题，TCP 会自动重传、排序。

(6)在主机 A 第一次向目标 IP 172.16.0.253 发送数据包时，通过 ARP 查询到目标主机的 MAC，若 A 与 B 同位于 172.16.0.0/16 这样一个网段，则服务器 B 的MAC 00:0c:29:27:75:a6 就是目的 MAC。目的 IP 172.16.0.253 即服务器 B 在 IP 网络中的地址。IP 数据包内部承载的目标信息就是“我想把这个数据送给 172.16.0.253 这台主机”。

1. **练习二：页面提交（以下内容可以根据实际实验过程进行调整，要求完成、顺序记录，并进行结果分析）**
2. 截获HTTP POST请求数据中的用户名和密码等数据
3. 本练习的提交过程使用HTTP协议的哪种方法？简述这种方法的作用。
4. 此次通信分几个阶段？每个阶段完成什么工作？
5. 参考“会话分析”视图显示结果，绘制此次提交过程的报文交互图（包括TCP协议）

**实验过程：**

首先本主机F清空IE缓存，打开WireShark抓包工具，开始抓包，然后打开IE浏览器在地址输入**<http://172.16.0.253/experiment/post.html>**，得到如下“图2.1 练习二网址网络页面”的效果：

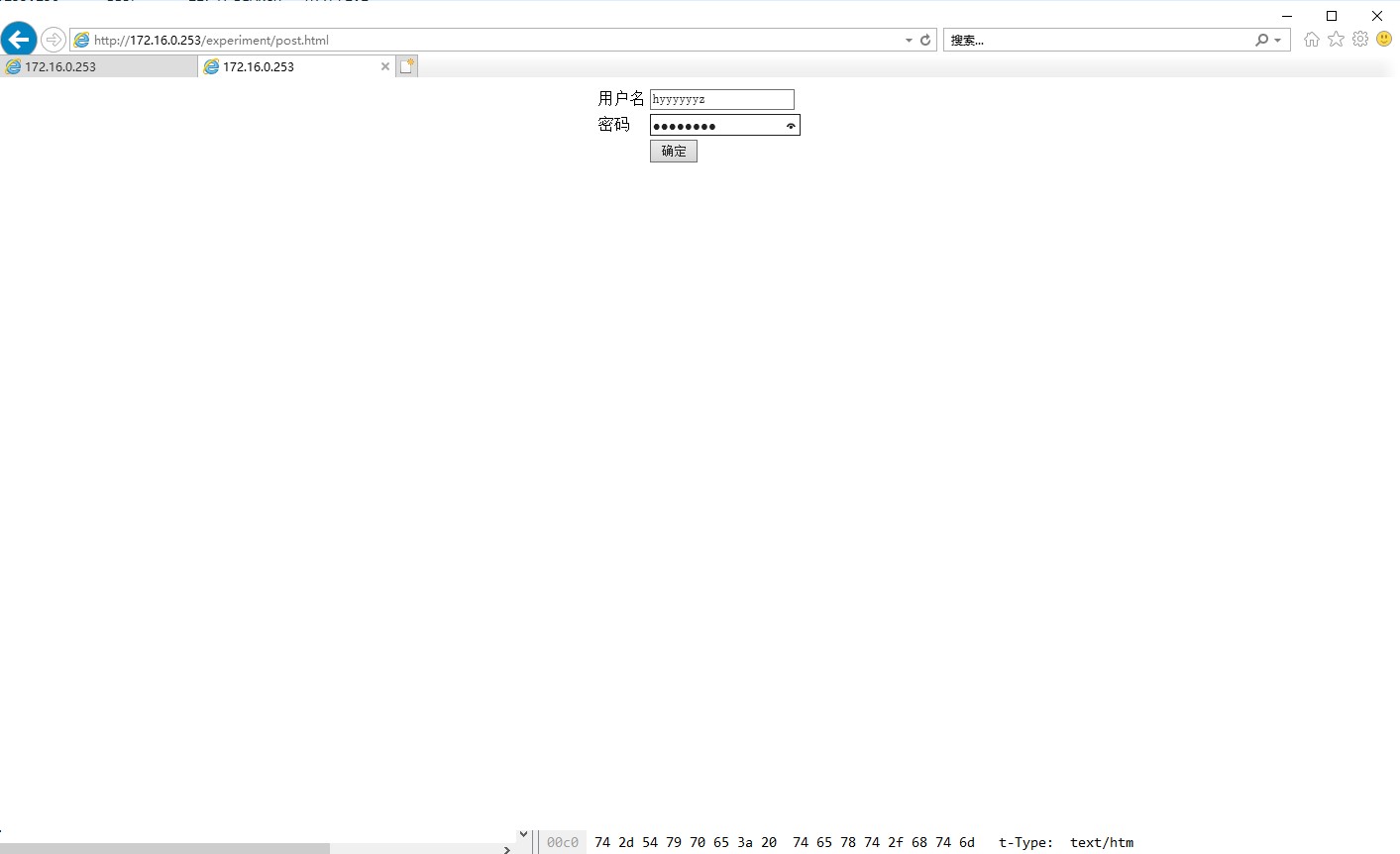
**图2.1 练习二网址网络页面**

图形用户界面, 文本, 应用程序, 电子邮件

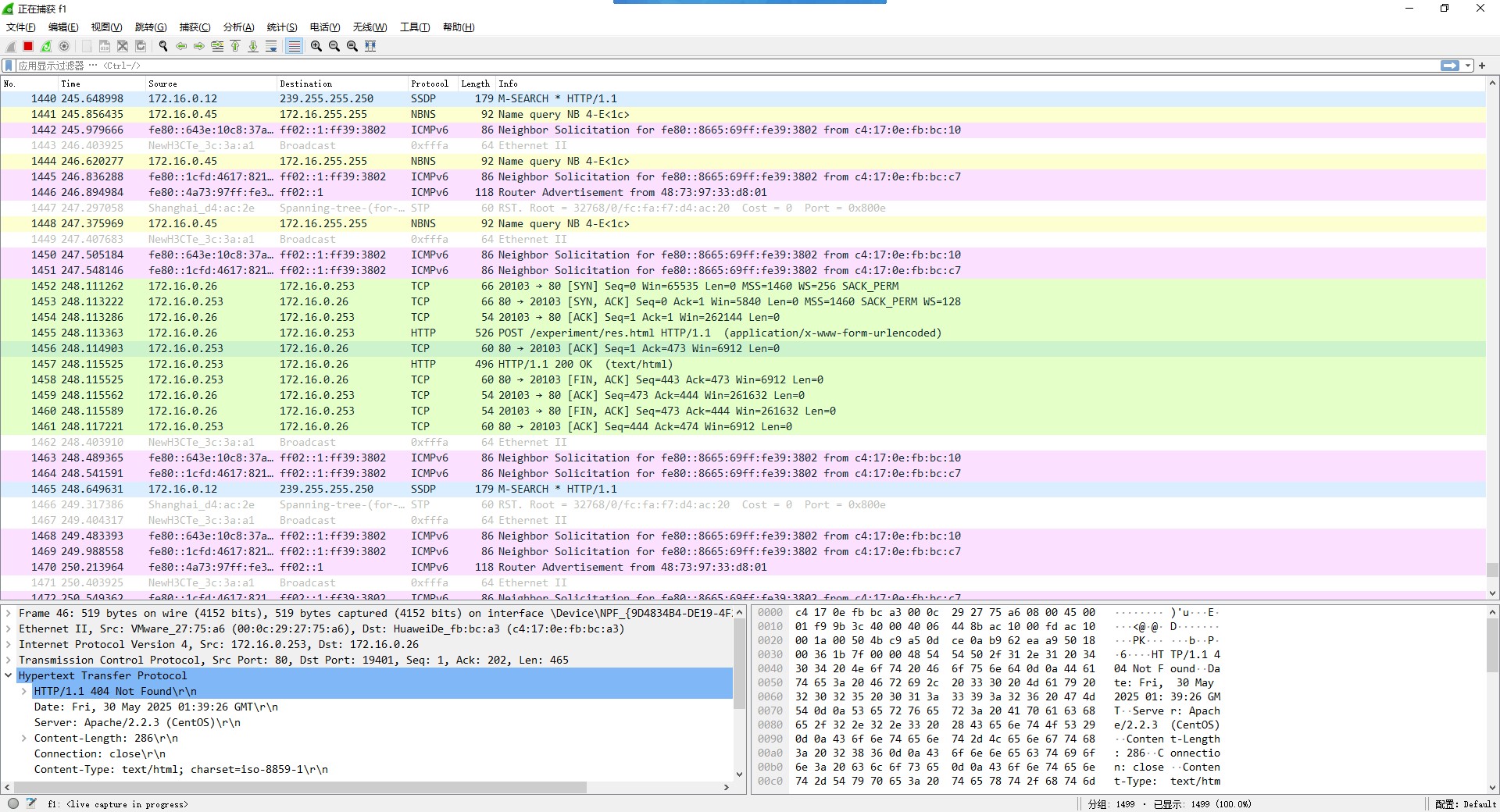
AI 生成的内容可能不正确。在WireShark抓包工具内可以看到捕获到四个HTTP数据包，如下“图2.2 练习二访问网址抓包”所示：

**图2.2 练习二访问网址抓包**

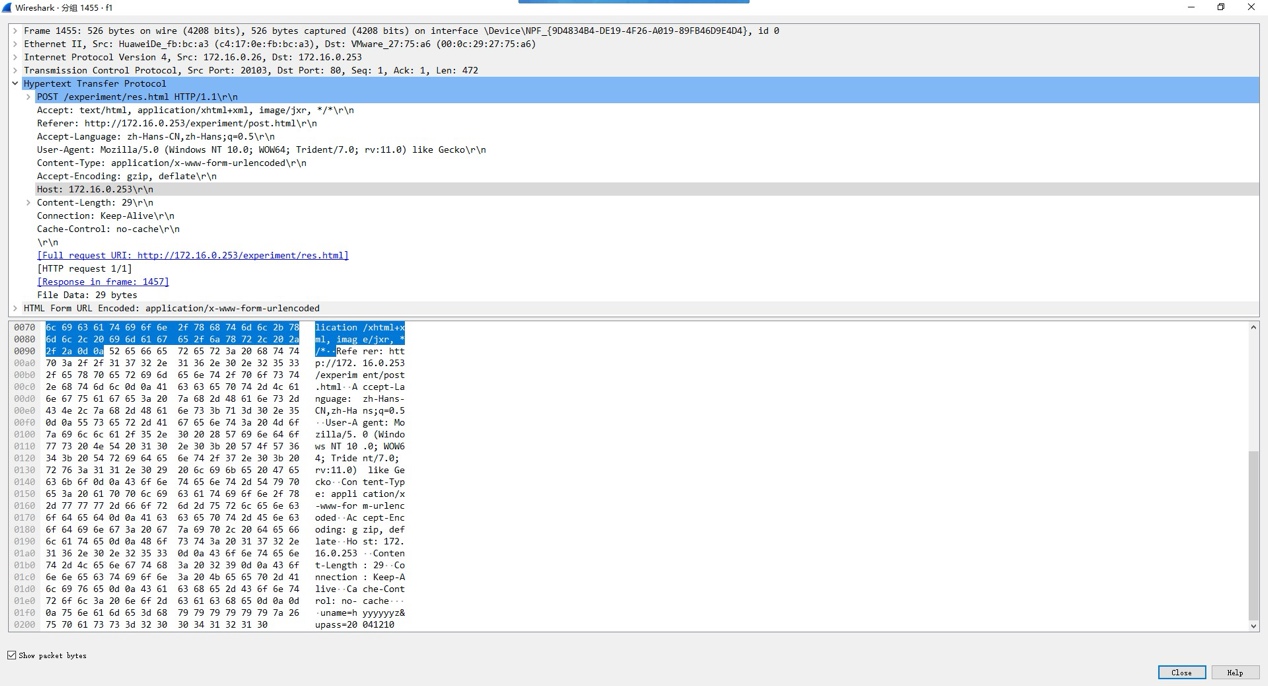
由于这部分抓包数据没有什么特殊性，固不进行详细展示。

 接下来在网址的输入框里面填写用户名“hyyyyyyz”和密码“20041210”，如下“图2.3 练习二网址填写数据”所示：

**图2.3 练习二网址填写数据**

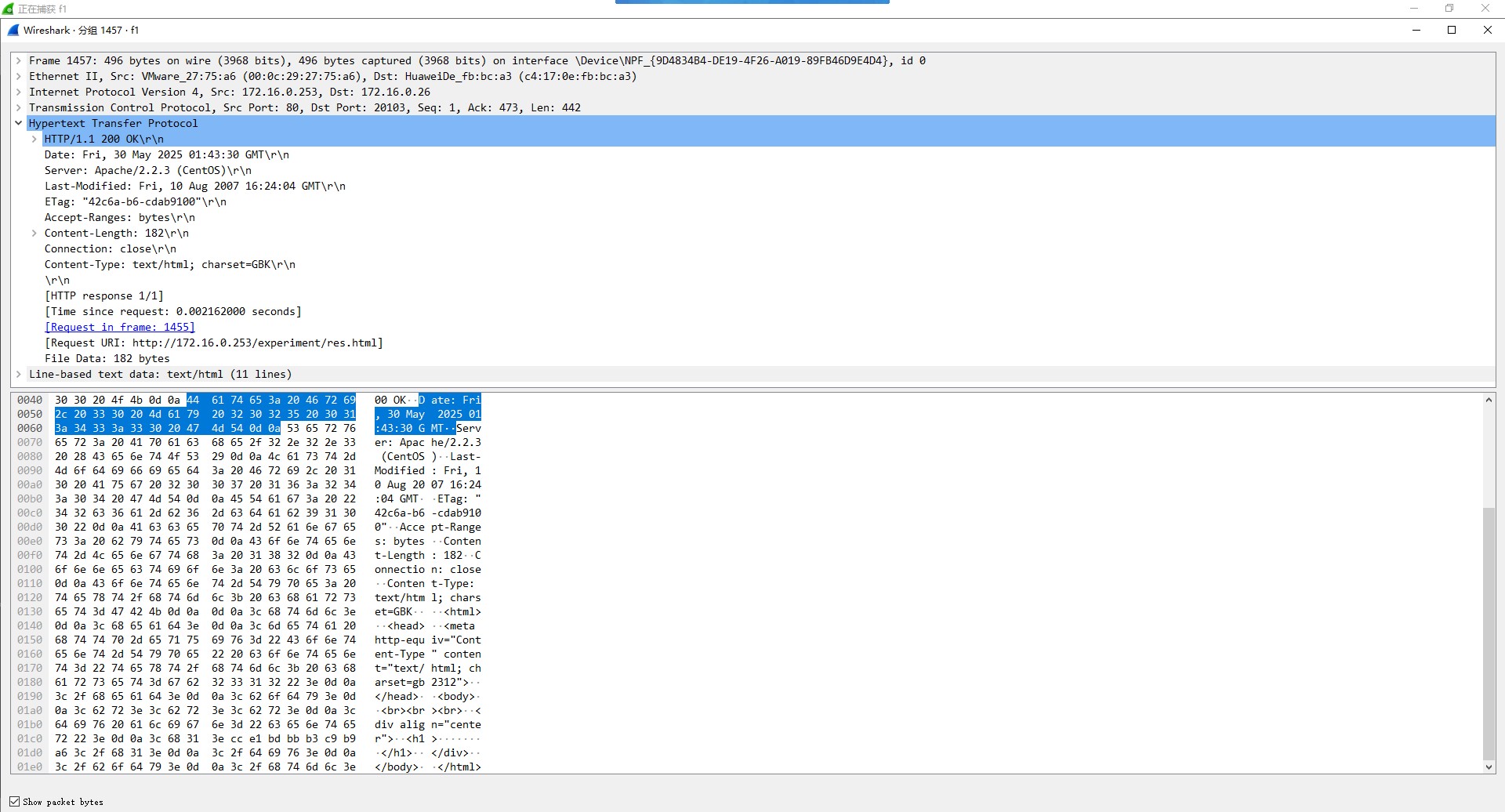
 点击“确定”按键，可以在WireShark抓包到两个HTTP数据包，如下“图2.4 练习二提交数据抓包”所示：

**图2.4 练习二提交数据抓包**

 分别打开这两个HTTP数据包，如下“图2.5 练习二提交数据HTTP抓包-1”、“图2.6 练习二提交数据HTTP抓包-2”所示：

提交的账号密码

**图2.5 练习二提交数据HTTP抓包-1**

**图2.6 练习二提交数据HTTP抓包-2**

观察第一个数据包，可以看到数据的最后包含最初当时设置的账号以及密码；第二个数据包的数据内可以看到上面所有的内容，“/r/n”把所有的数据进行的换行处理。

根据上述捕获数据包的内容回答练习二要求的问题：

(1)见“图2.5 练习二提交数据HTTP抓包-1”中可以看到截获到的HTTP POST请求数据中的用户名和密码等数据。

(2)使用 HTTP POST 方法。作用：向服务器提交数据，数据封装在请求正文中，而非附加在URL后。适用于敏感数据或大量数据的传输，安全性优于GET方法。相比于GET方法，POST方法可能修改服务器状态。

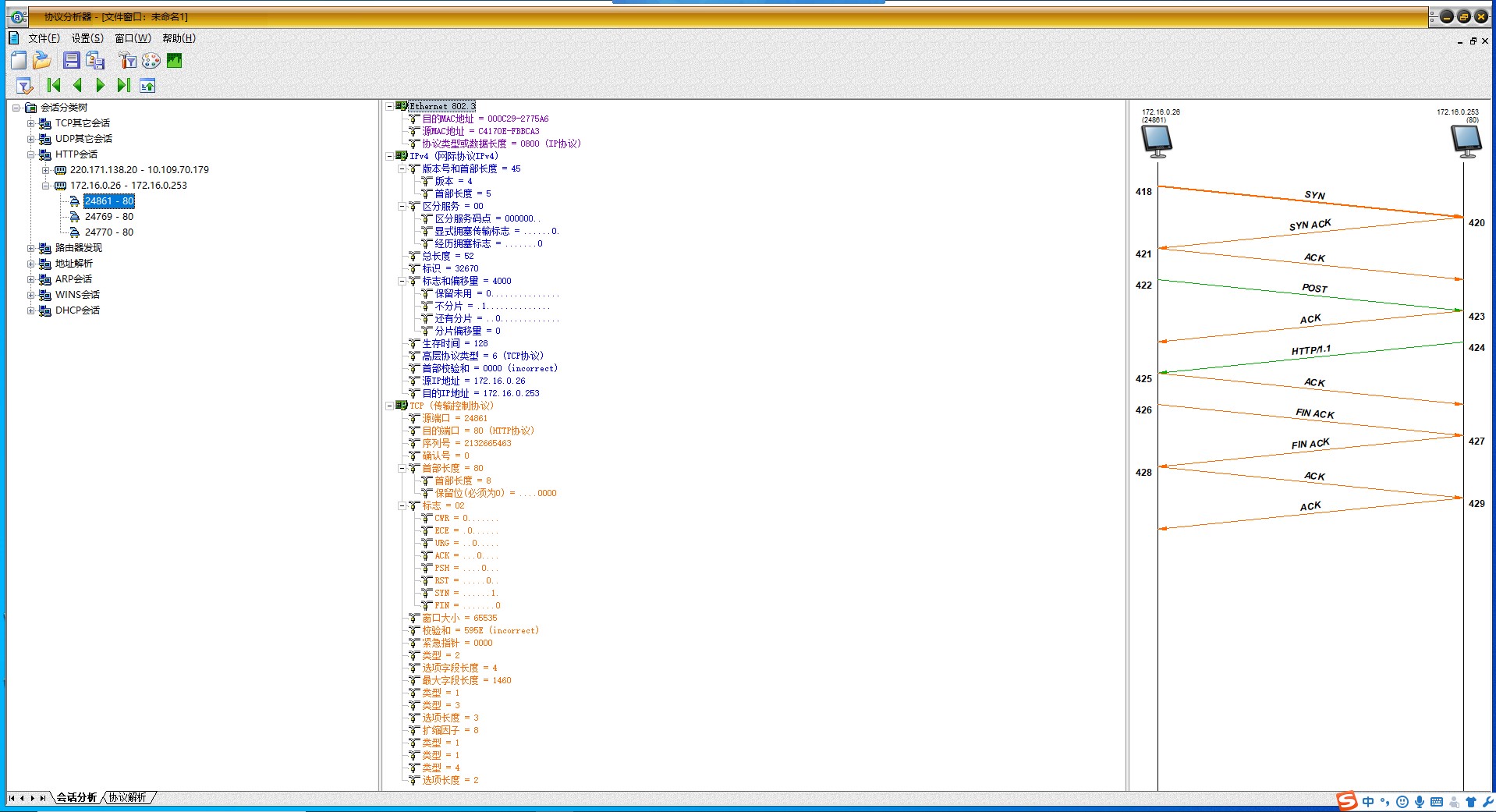
(3)通信分为 4个阶段：

首先是TCP连接建立（三次握手）：客户端发送 SYN → 服务器回复 SYN-ACK → 客户端确认 ACK，从而建立可靠传输通道。

然后是HTTP请求与响应的阶段1（加载页面）：客户端发送 GET /experiment/post.html → 服务器返回 200 OK（HTML页面）。

接下来是HTTP请求与响应的阶段2（提交表单）：客户端发送 POST /experiment/post.html（含用户名密码）→ 服务器返回 200 OK（提交结果）。

最后是TCP连接释放（四次挥手）：客户端发送 FIN → 服务器确认 ACK → 服务器发送 FIN → 客户端确认 ACK，从而安全关闭连接。

(4)报文交互图如下“图2.7 练习二报文交互图”所示：

**图2.7 练习二报文交互图**

1. **练习三：获取页面信息（以下内容可以根据实际实验过程进行调整，要求完成、顺序记录，并进行结果分析）**
2. 采用HTTP HEAD方法获取服务器资源的元数据
3. 主机A在“TCP工具”上，设置“发送数据（文本）”为以下内容：

HEAD /experiment/ HTTP/1.1<CRLF>

Host: 172.16.0.253<CRLF>

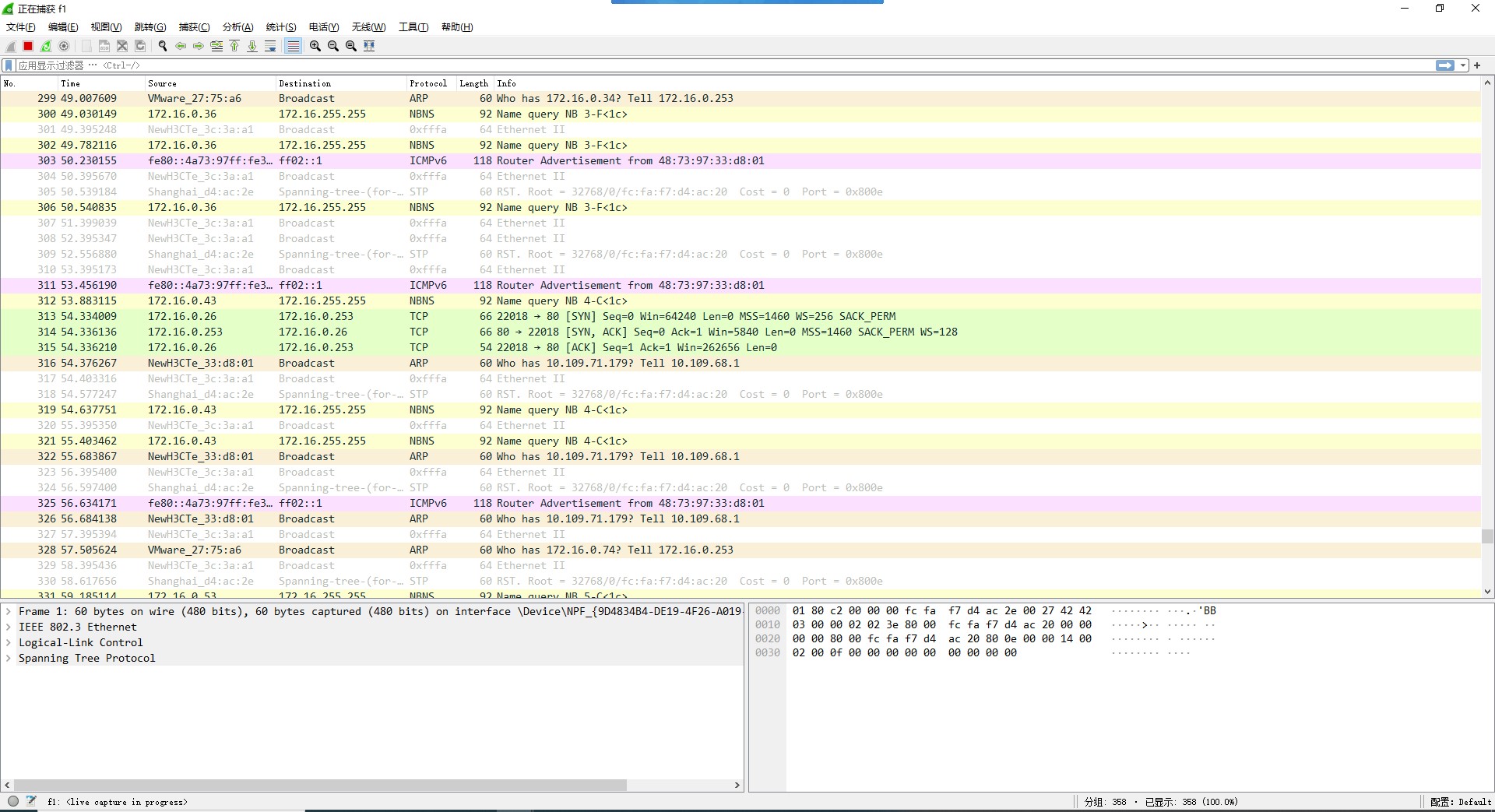
<CRLF>

点击[发送]按钮。 （注：<CRLF>是回车换行）

点击[断开]按钮，断开TCP连接（由于不同http版本所遵循的规范不同，有些HTTP服务器不需要断开操作）。

1. 主机A在“TCP工具”上的“显示数据（文本）”中察看服务器返回信息。

**实验过程：**

打开TCP工具，选择客户端，在地址栏以及端口栏分别输入**172.16.0.253**和**80**，然后打开WireShark抓包工具开始抓包，在TCP工具中点击连接，成功连接后，可以观察到抓包了几个TCP数据包，如下“图3.1 建立连接抓包”所示：

**图3.1 建立连接抓包**

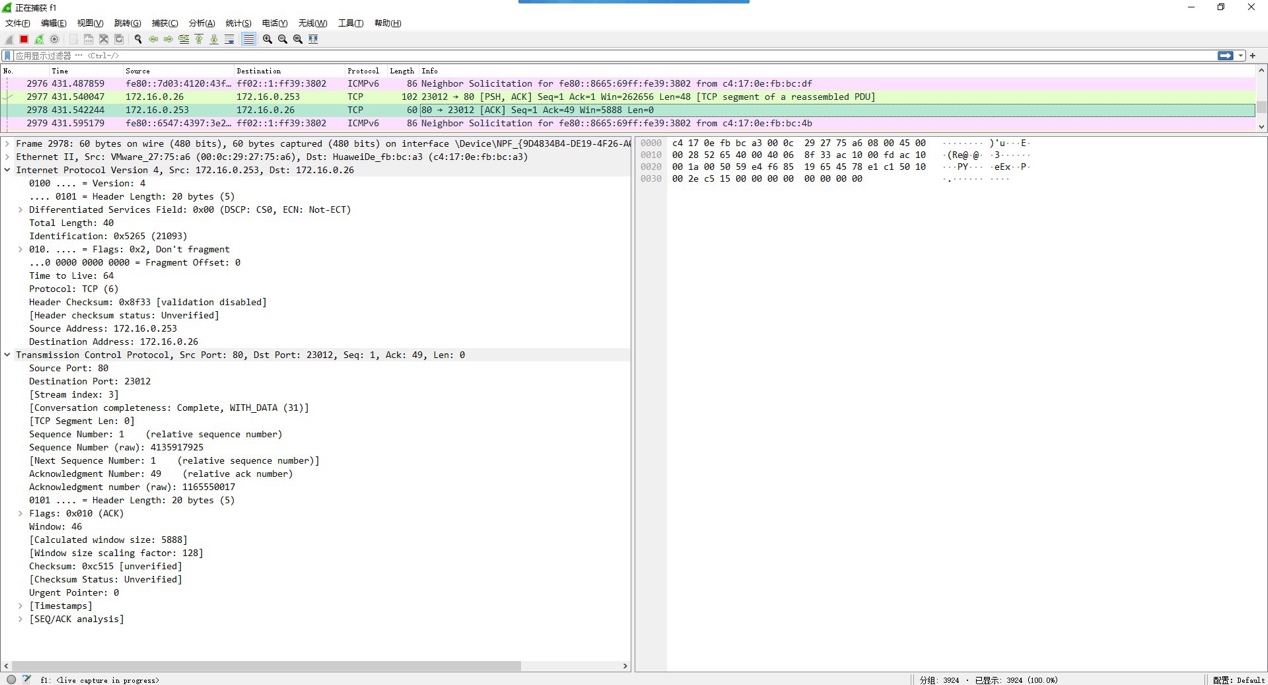
图形用户界面

AI 生成的内容可能不正确。在TCP工具中输入实验要求的文本内容，然后点击发送，如下“图3.2 发送文本数据”所示：



**图3.2 发送文本数据**

可以在WireShark抓包到两个TCP数据包，分别打开这两个数据包如下“图3.3 发送数据抓包-1”、“图3.4 发送数据抓包-2”所示：

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。**图3.3 发送数据抓包-1**

**图3.4 发送数据抓包-2**

观察上述两个数据包，由第一个数据包箭头所示出，可以观察到即TCP工具中发送的内容，手动构造了HEAD请求HEAD /experiment/ HTTP/1.1和主机号Host: 172.16.0.253，才使得数据能够正常发送出去。

HEAD方法相对于GET方法只请求资源的头部信息，不返回实际内容体，可以做到节省带宽，用于检查资源是否存在、是否被修改。

1. **思考题（20分）**
2. **一个主页是否只有一个连接？**

不一定。现代网页通常包含多个资源，由上述实验抓包数据可知，浏览器会建立多个并行TCP连接同时下载资源。对于HTTP/1.1，可能复用同一TCP连接。HTTP/2支持允许通过单一连接并行传输多个资源。

1. **HTTP请求方法有哪些？分别表示什么含义？**

|  |  |
| --- | --- |
| **方法** | **含义** |
| **GET** | 获取资源，参数在URL中，不应修改服务器数据 |
| **POST** | 提交数据，参数在请求体中，可能修改服务器状态（如登录/下单） |
| **HEAD** | 获取资源的元数据（响应头），不返回内容体 |
| **PUT** | 上传资源到指定位置（完整替换） |
| **DELETE** | 删除指定资源 |
| **PATCH** | 对资源进行部分修改 |
| **OPTIONS** | 获取服务器支持的HTTP方法 |
| **TRACE** | 回显请求消息（用于诊断） |

1. **什么是URL？URL的完整格式是说明？用一个示例说明URL中的各个部分。**

**定义**：统一资源定位符（Uniform Resource Locator），互联网资源的地址

**完整格式**：<协议>://<用户名>:<密码>@<主机>:<端口>/<路径>?<查询参数>#<片段>

**示例**：<https://www.example.com:443/search?q=network#results>

协议：https

主机：www.example.com

端口：443（HTTPS默认端口，可省略）

路径：/search

查询参数：q=network

片段：results（页面内锚点）

1. **实验总结（10分）**

通过本次实验，我深入理解了HTTP协议的核心机制：

验证了HTTP依赖TCP提供可靠传输（三次握手建立连接，四次挥手断开），观察到MAC地址用于局域网寻址，IP地址用于网络层路由。

成功捕获并解析GET/POST/HEAD方法的报文结构，对比了GET（参数在URL）与POST（参数在body）的安全差异，通过HEAD方法验证了元数据获取机制。

抓包暴露了POST请求的明文密码传输风险，认识到实际开发中必须使用HTTPS加密敏感数据。

完整跟踪了从TCP连接到HTTP请求的端到端过程，通过手动构造HEAD请求深入理解协议规范

本次实验让我认识到协议设计中的安全考量（如GET/POST的选择），以及在复杂网络环境中调试协议的能力重要性。