华东师范大学数据学院上机实践报告

Computer Network & Coding Lab 2

课程名称： 计算机网络	年级：2023	上机实践成绩：
指导老师：张召	**姓名**：陈子谦	
上机实践名称：协议分析	**学号**：10235501454	**上机实践日期**：

一、题目要求

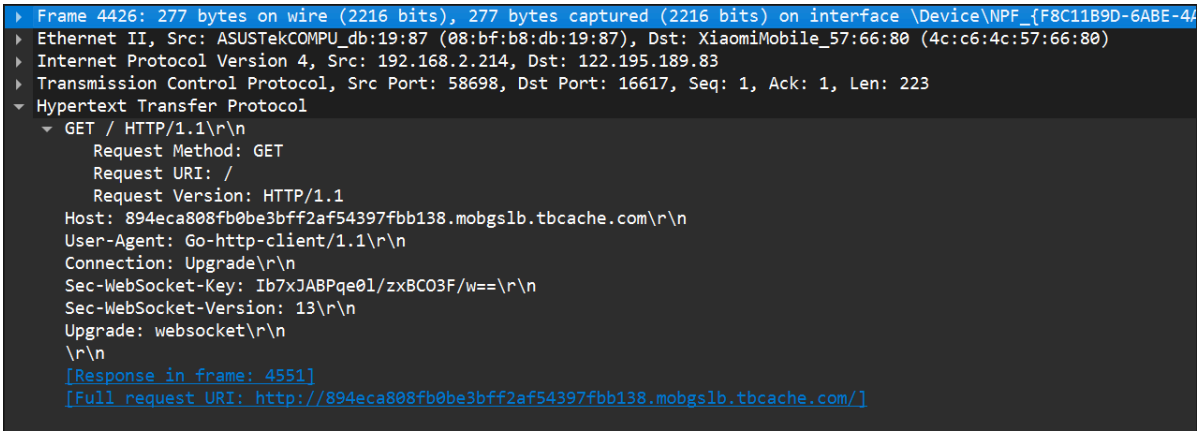
- 熟悉 HTTP、HTTPS 协议的工作原理
- 了解 HTTP、HTTPS 协议在实际网络中的运行过程
- 熟悉 SMTP 和 POP3 协议的工作原理
- 了解 SMTP 和 POP3 协议在实际网络中的运行过程

二、项目实现思路

- 通过 Wireshark 分析 HTTP 和 HTTPS 协议
- 通过 Wireshark 分析 SMTP 和 POP3 协议

三、功能实现情况

Task 1



层级	关键信息
数据链路层	源MAC: `ASUSTekCOMPU_db:19:87`，目标MAC: `XiaomiMobile_57:66:80`
网络层	源IP: `192.168.2.214`，目标IP: `122.195.189.83`（公网服务器）
传输层	源端口: `58698`，目标端口: `16617`
应用层	HTTP请求方法: `GET / HTTP/1.1`，包含WebSocket升级头

HTTP握手请求详解

（1）请求行

```
GET / HTTP/1.1\r\n
```

- **方法**: `GET`
- **URI**: `/`（根路径）
- **协议**: `HTTP/1.1`

（2）请求头部

```
Host: 894ea808fb0be3bff2af54397fbb138.mobgs1b.tbcache.com\r\n
Connection: Upgrade\r\n
Sec-WebSocket-Key: [Base64密钥]\r\n
Sec-WebSocket-Version: 13\r\n
Upgrade: websocket\r\n
\r\n
```

- `Host`: 请求的域名（CDN或缓存服务器地址）。
- `Connection: Upgrade`: 声明需要升级协议。
- `Upgrade: websocket`: 明确升级到WebSocket协议。
- `Sec-WebSocket-Key`: 客户端生成的随机Base64密钥，用于握手验证。
- `Sec-WebSocket-Version: 13`: 指定WebSocket协议版本。

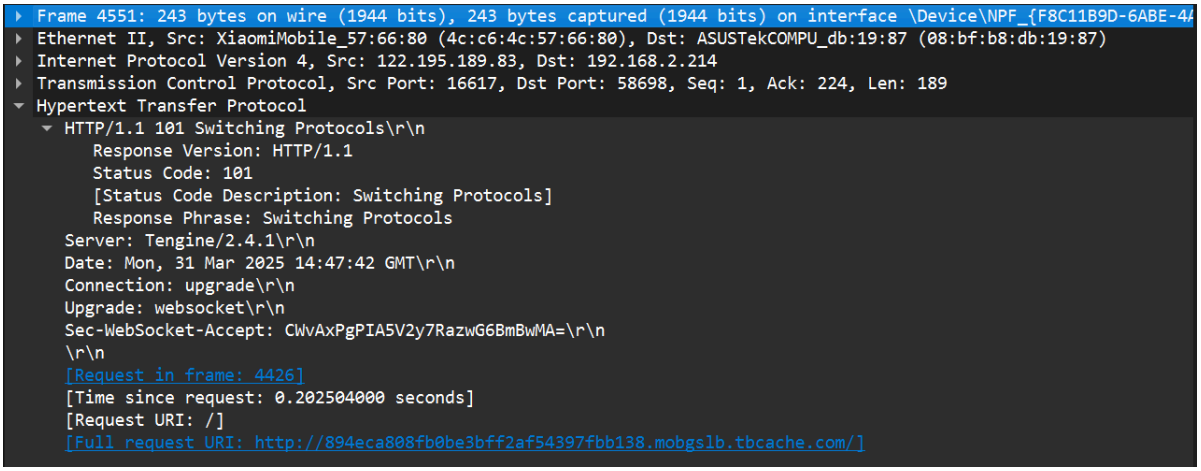
WebSocket握手流程解析

1. **握手请求**（当前报文）
   - 客户端通过HTTP `GET` 请求发起WebSocket连接，目标端口为非常规端口（`16617`），表明是自定义服务。
   - 服务器需返回 `HTTP 101 Switching Protocols` 响应以完成握手。
2. **握手响应**（根据提示，响应在Frame 4551中）
   - **预期响应头**:

     ```
     HTTP/1.1 101 Switching Protocols\r\n
     Upgrade: websocket\r\n
     Connection: Upgrade\r\n
     Sec-WebSocket-Accept: [服务端计算的密钥]\r\n
     ```

   - **密钥验证**：服务端会用客户端的 `Sec-WebSocket-Key` 计算 `Sec-WebSocket-Accept`，确保握手合法性。

Task 2



HTTP响应报文结构

组成部分	规范要求（图片内容）	实际响应（Frame 4551）
状态行	`协议版本 状态码 状态描述\r\n`	`HTTP/1.1 101 Switching Protocols\r\n`
响应头部	`字段名: 值\r\n`（每行一个字段）	`Upgrade: websocket\r\n` + `Sec-WebSocket-Accept: [密钥]\r\n`
空行	`\r\n`（分隔头部与正文）	头部结束后空一行
响应正文	可选（如HTML/JSON）	无正文（WebSocket握手无需传输数据）

请求包与响应包逐层对比

1. 数据链路层 & 网络层

对比项	HTTP请求（Frame 4426）	HTTP响应（Frame 4551）
源MAC地址	`ASUSTekCOMPU_db:19:87`	`XiaomiMobile_57:66:80`
目标MAC地址	`XiaomiMobile_57:66:80`	`ASUSTekCOMPU_db:19:87`
源IP地址	`192.168.2.214`（客户端）	`122.195.189.83`（服务器）
目标IP地址	`122.195.189.83`（服务器）	`192.168.2.214`（客户端）

2. 传输层

对比项	HTTP请求	HTTP响应
源端口	`58698`（客户端随机端口）	`16617`（服务器监听端口）
目标端口	`16617`（非标准HTTP端口）	`58698`（客户端端口）
TCP标志位	`PSH, ACK`（推送数据并确认）	`ACK`（确认接收）

3. 应用层

对比项	HTTP请求	HTTP响应
起始行	`GET / HTTP/1.1\r\n`	`HTTP/1.1 101 Switching Protocols\r\n`
关键字段	`Sec-WebSocket-Key: [Base64密钥]`	`Sec-WebSocket-Accept: [服务端密钥]`
协议升级	请求升级（`Upgrade: websocket`）	确认升级（`Upgrade: websocket`）
空行与正文	无正文	无正文

对比

特性	HTTP请求	HTTP响应
交互方向	客户端 → 服务器	服务器 → 客户端
协议角色	主动发起连接	确认协议升级
后续通信	升级为WebSocket二进制帧	同上
端口方向	目标端口为服务端监听端口	目标端口为客户端临时端口

HTTP请求用于协议升级协商，而响应用于确认升级，两者共同完成WebSocket握手

Task 3 - GET 和 POST 方法对比分析

• **GET**：用于请求资源，通常用于获取数据（查询操作），参数通过 URL 传递。
• **POST**：用于提交数据，通常用于修改服务器资源（提交操作），参数通过请求体传递。

请求报文差异

特性	GET	POST
请求行	`GET /path?param=value HTTP/1.1`	`POST /path HTTP/1.1`
参数位置	URL 查询字符串	请求体中
请求头	无 `Content-Type` 或 `Content-Length`	需包含 `Content-Type` 和 `Content-Length`
长度限制	受 URL 长度限制	无限制，依赖服务器配置
安全性	参数暴露在 URL 和日志中，不安全	参数在请求体中，安全性略高
缓存	可被缓存、书签保存	不可缓存，浏览器通常不保留

请求报文示例

GET 请求

```
GET /search?q=hello&category=books HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0
```

POST 请求

```
POST /submit-form HTTP/1.1
Host: example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 21

username=alice&age=25
```

响应报文差异

特性	GET	POST
状态码	通常返回 `200 OK`（直接返回资源）	可能返回 `201 Created` 或 `303 See Other`（重定向）
响应头	可能包含缓存头（如 `Cache-Control`）	通常包含 `Location` 头（重定向时）
响应体	请求的资源（如 HTML、JSON）	处理结果（如确认消息、新资源 ID）

Task 4 - SMTP 和 POP3 数据包分析

SMTP 数据流分析

1. 报文结构概览

```
220 ************************************************************
EHLO LAPTOP-GIHO6R83
250-mail...（服务器支持的功能列表）
AUTH LOGIN
334 dXNlcm5hbWU6
MTg2MjUxNzEzODZAMTYzLmNvbQ==
334 UGFzc3dvcmQ6
QlFxVmdwZU15VlRoTm5tUw==
235 Authentication successful
MAIL FROM: <18625171386@163.com>
250 Mail OK
RCPT TO: <3380595029@qq.com>
250 Mail OK
DATA
354 End data with <CR><LF>.<CR><LF>
（邮件内容及附件）
.
250 Mail OK queued as...
```

```
QUIT
221 Bye
```

2. 报文组成解析

命令与响应

命令/响应	说明
`220`	服务器就绪响应，标识 SMTP 服务启动。
`EHLO LAPTOP-GIH06R83`	客户端发起握手，声明自身域名。
`250-mail...`	服务器支持的功能列表（如 `PIPELINING`、`AUTH`、`STARTTLS`）。
`AUTH LOGIN`	客户端请求使用明文认证（Base64 编码）。
`334 dXNlcm5hbWU6`	服务器要求客户端发送 Base64 编码的用户名。
`MTg2...`	客户端发送用户名（解码后为 `18625171386@163.com`）。
`334 UGFzc3dvcmQ6`	服务器要求客户端发送 Base64 编码的密码。
`QlFx...`	客户端发送密码（解码后为 `BQqVgpeMyVThNnmS`）。
`235 Authentication`	服务器认证成功。
`MAIL FROM`	指定发件人地址。
`RCPT TO`	指定收件人地址。
`DATA`	开始传输邮件内容。
`354`	服务器同意接收邮件数据，提示以 `.` 结束。

3. 邮件内容结构

- **邮件头**：

```
Date: Sun, 23 Mar 2025 21:30:24 +0800
From: "18625171386@163.com" <18625171386@163.com>
To: 3380595029 <3380595029@qq.com>
Subject: Hello World!
Mime-Version: 1.0
Content-Type: multipart/alternative; boundary="----=_001_NextPart316532538204_=----"
```

- **邮件体**：
  - **文本部分**（`text/plain`）：Base64 编码的明文内容（解码后为 `Hello World!`）。
  - **HTML 部分**（`text/html`）：使用 Quoted-Printable 编码的富文本内容。
  - **附件**：通过 `multipart/alternative` 和 `boundary` 分隔多部分内容。

POP3 数据流分析

1. 报文结构概览

```
+OK POP3 ready
USER 18625171386@163.com
+OK
PASS BQqVgpeMyVThNnmS
+OK 11 message(s) [450543 byte(s)]
STAT
+OK 11 450543
LIST
+OK 11 450543（邮件列表）
UIDL
+OK 11 450543（唯一邮件标识）
RETR 10
+OK 2700 octets（邮件内容）
.
RETR 11
+OK 3534 octets（邮件内容）
.
QUIT
+OK core mail
```

2. 报文组成解析

命令与响应

命令/响应	说明
`USER`	客户端发送用户名。
`PASS`	客户端发送密码（明文传输）。
`STAT`	查询邮箱状态（邮件总数和总大小）。
`LIST`	列出每封邮件的编号和大小。
`UIDL`	获取每封邮件的唯一标识符（防止重复下载）。
`RETR <n>`	下载第 `n` 封邮件的内容。
`QUIT`	结束会话。

3. 邮件内容结构

- **邮件头**：

```
From: "3380595029" <3380595029@qq.com>
To: "18625171386@163.com" <18625171386@163.com>
Subject: =?utf-8?B?6Ieq5Yqo5Zue5aSNOiBIZWxsbyBXb3JMZCE=?=
Date: Sun, 23 Mar 2025 21:30:26 +0800
DKIM-Signature: ...（邮件签名验证）
Content-Type: text/html; charset="utf-8"
```

- **邮件体**：
  - **Base64 编码内容**：解码后为 `您好，这是您发送的邮件。`。
  - **多部分结构**：通过 `multipart/alternative` 和 `boundary` 分隔文本和 HTML 内容。

Task 5

以下为抓取的 SMTP 会话文本：

```
S: 220 ************************************************************
C: EHLO LAPTOP-GIHO6R83
S: 250-mail.example.com
S: 250-PIPELINING
S: 250-AUTH LOGIN PLAIN
S: 250-STARTTLS
S: 250 8BITMIME
C: AUTH LOGIN
S: 334 dXNlcm5hbWU6
C: MTg2MjUxNzEzODZAMTYzLmNvbQ==
S: 334 UGFzc3dvcmQ6
C: QlFxVmdwZU15VlRoTm5tUw==
S: 235 Authentication successful
C: MAIL FROM: <18625171386@163.com>
S: 250 Mail OK
C: RCPT TO: <3380595029@qq.com>
S: 250 Mail OK
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: Date: Sun, 23 Mar 2025 21:30:24 +0800
C: From: "18625171386@163.com" <18625171386@163.com>
C: To: 3380595029 <3380595029@qq.com>
C: Subject: Hello World!
C: Content-Type: multipart/alternative; boundary="----
=_001_NextPart316532538204_=----"
C:
C: ------=_001_NextPart316532538204_=----
C: Content-Type: text/plain; charset="us-ascii"
C: Content-Transfer-Encoding: base64
C:
C: SGVsbG8gV29ybGQhDQo=
C: ------=_001_NextPart316532538204_=----
C: .
S: 250 Mail OK queued as gzga-smtp-mtada-g0-1
C: QUIT
S: 221 Bye
```

报文交互分析

1. 连接建立阶段

- **服务器响应（S→C）**：

  ```
  220 ************************************************************
  ```

- **说明**：服务器返回状态码 `220`，表示服务就绪，准备接收客户端请求。
- **字段**：`220` 是 SMTP 标准响应码，后接服务器标识（部分隐藏）。

- **客户端握手（C→S）**：

```
EHLO LAPTOP-GIH06R83
```

  - **说明**：客户端发送 `EHLO` 命令，声明自身域名（`LAPTOP-GIH06R83`），启动扩展 SMTP（ESMTP）会话。
  - **字段**：`EHLO` 是改进的 `HELO` 命令，支持扩展功能协商（如认证、加密）。

2. 功能协商阶段

- 服务器响应（S→C）：

```
250-mail.example.com
250-PIPELINING
250-AUTH LOGIN PLAIN
250-STARTTLS
250 8BITMIME
```

  服务器返回支持的功能列表：
  - `PIPELINING`：允许客户端连续发送多个命令。
  - `AUTH LOGIN PLAIN`：支持明文认证（Base64 编码）。
  - `STARTTLS`：支持升级到 TLS 加密连接。
  - `8BITMIME`：支持 8 位二进制数据传输。
  - **字段**：每行以 `250` 开头，表示请求成功。

3. 认证阶段

- **客户端请求认证（C→S）**：

```
AUTH LOGIN
```

  - **说明**：客户端选择 `LOGIN` 认证方式（需用户名和密码）。

- **服务器挑战（S→C）**：

```
334 dXNlcm5hbWU6
```

  - **说明**：服务器返回 `334`，要求客户端发送 Base64 编码的用户名。
  - **字段**：`dXNlcm5hbWU6` 解码后为 `username:`。

- **客户端发送用户名（C→S）**：

```
MTg2MjUxNzEzODZAMTYzLmNvbQ==
```

  - **说明**：Base64 编码的用户名，解码后为 `18625171386@163.com`。

- **服务器再次挑战（S→C）**：

```
334 UGFzc3dvcmQ6
```

- **说明**：服务器要求客户端发送 Base64 编码的密码。
- **字段**：`UGFzc3dvcmQ6` 解码后为 `Password:`。

- **客户端发送密码（C→S）**：

```
QlFxVmdwZU15VlRoTm5tUw==
```

- **说明**：Base64 编码的密码，解码后为 `BQqVgpeMyVThNnmS`。

- **服务器认证成功（S→C）**：

```
235 Authentication successful
```

- **说明**：状态码 `235` 表示认证成功。

4. 邮件传输阶段

- **发件人指定（C→S）**：

```
MAIL FROM: <18625171386@163.com>
```

- **说明**：客户端指定发件人地址，服务器返回 `250 Mail OK`。

- **收件人指定（C→S）**：

```
RCPT TO: <3380595029@qq.com>
```

- **说明**：客户端指定收件人地址，服务器返回 `250 Mail OK`。

- **邮件内容传输（C→S）**：

```
DATA
354 End data with <CR><LF>.<CR><LF>
（邮件头及正文）
.
```

- `DATA` 命令启动邮件内容传输。
- 服务器返回 `354`，提示客户端以单行 `.` 结束数据传输。
- 邮件内容包含 `Date`、`From`、`To`、`Subject` 等头字段，以及 MIME 编码的正文。

5. 会话终止

- **客户端退出（C→S）**：

```
QUIT
```

- **说明**：客户端请求结束会话。

- **服务器确认（S→C）**：

```
221 Bye
```

- **说明**：服务器返回 `221`，确认关闭连接。

四、总结

本次实验通过抓包分析HTTP、HTTPS、SMTP和POP3协议，加深了对应用层协议的理解：

1. 通过分析HTTP请求和响应报文，掌握了其报文结构和工作流程，明确了GET和POST方法在参数传递、安全性和缓存方面的差异。
2. 观察了WebSocket基于HTTP建立连接的过程，理解了协议升级机制和密钥验证方法。
3. 通过分析SMTP和POP3数据包，掌握了邮件发送和接收的完整流程，了解了认证、报文格式和邮件内容编码方式。
4. 对比HTTP与HTTPS、明文传输与加密传输的差异，认识到网络安全的重要性。