华东师范大学软件工程学院实验报告

实验课程：计算机网络实践	**年级**：2023 级	**实验成绩**：
实验名称：Protocol Layer	**姓名**：顾翌炜	
实验编号：Lab-1	**学号**：10235101527	**实验日期**：2024/11/15
指导教师：王廷	**组号**：01	**实验时间**：2024/11/15

1 实验目的

1) 学会通过 Wireshark 获取各协议层的数据包
2) 掌握协议层数据包结构
3) 分析协议开销

2 实验内容与实验步骤

2.1 实验内容

2.1.1 获取协议层的数据包

使用 `wget` 命令发起 `HTTP` 请求，然后使用 `Wireshark` 进行抓包。

2.1.2 绘制数据包结构

分析 `HTTP GET` 协议包的内容，并绘制协议包，分别标出 `Ethernet`，`IP` 和 `TCP` 协议的头部的位置、大小以及其负载的范围。

2.1.3 分析协议开销

根据实验结构，分析 `HTTP` 应用协议额外开销。估计上面捕获的 `HTTP` 协议的额外开销。假设 `HTTP` 数据（头部和消息）是有用的，而 `TCP`，`IP` 和 `Ethernet` 头部认为是开销。对于下载的主要部分中的每一个包，我们需要分析 `Ethernet`，`IP` 和 `TCP` 的开销，和有用的 `HTTP` 数据的开销。根据以上的定义来估计下载协议的开销，来判定这种开销是否有必要。

2.1.4 分析解复用键

解复用指的是找到正确的上一层协议来处理到达的包。
观察下载的以太网和 `IP` 包的头部信息，回答下面问题：

1. Which Ethernet header field is the demultiplexing key that tells it the next higher layer is IP? What value is used in this field to indicate **"IP"**?
2. Which IP header field is the demultiplexing key that tells it the next higher layer is TCP? What value is used in this field to indicate **"TCP"**?

2.1.5 问题讨论

Explore on your own:

1. Look at a short TCP packet that carries no higher-layer data. To what entity is this packet destined? After all, if it carries no higher-layer data then it does not seem very useful to a higher layer protocol such as HTTP!
2. In a classic layered model, one message from a higher layer has a header appended by the lower layer and becomes one new message. But this is not always the case. Above, we saw a trace in which the web response (one HTTP message comprised of an HTTP header and an HTTP payload) was converted into multiple lower layer messages (being multiple TCP packets). Imagine that you have drawn the packet structure (as in step 2) for the first and last TCP packet carrying the web response. How will the drawings differ?
3. In the classic layered model described above, lower layers append headers to the messages passed down from higher layers. How will this model change if a lower layer adds encryption?
4. In the classic layered model described above, lower layers append headers to the messages passed down from higher layers. How will this model change if a lower layer adds compression?

2.2 实验步骤

1) 安装实验所需软件（为方便安装，我们使用 `Windows` 下的包管理器 `winget`）

```
1     C:\User\GHOST> winget install wget
```

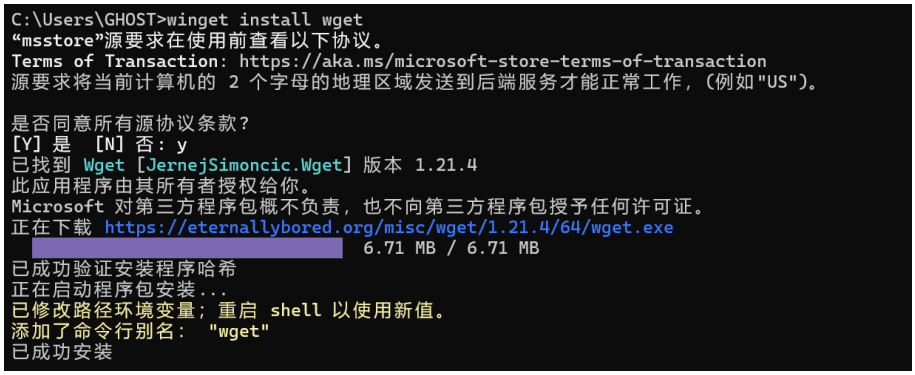

图 1: 下载 wget

并从官网按照步骤下载 wireshark 软件

2) 打开 Wireshark，在菜单栏的 **捕获 -> 选项**中进行设置，选择已连接的网络，设置捕获过滤器为 tcp port 80，将混杂模式设为关闭, 勾选 enable network name resolution，然后开始捕获。
3) 使用 wget 命令发起 HTTP 请求

```
1    C:\User\GHOST> wget http://www.baidu.com
```

4) 在 Wireshark 中停止捕获。
5) 分析 HTTP GET 协议包的内容，并绘制协议包。
6) 分析协议开销
7) 分析解复用键
8) 问题讨论

3 实验环境

- 操作系统：Windows 11 家庭中文版 23H2 22631.4460
- 网络适配器：Killer(R)Wi-Fi 6E AX1675i 160MHz Wireless Network Adapter(211NGW)
- Wireshark：Version 4.4.1
- wget：GNU Wget 1.21.4 built on mingw32

4 实验过程与分析

4.1 获取协议层的数据包

首先，我们打开 Wireshark，在菜单栏的 **捕获 -> 选项**中进行设置，选择已连接的网络，设置捕获过滤器为 tcp port 80，将混杂模式设为关闭, 勾选 enable network name resolution，然后开始捕获。

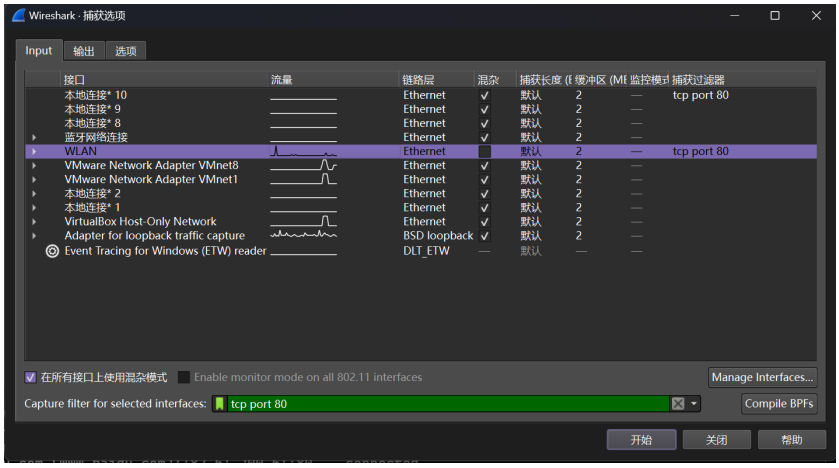

图 2: 设置捕获

然后，我们使用 `wget` 命令对 `http://www.baidu.com` 发起 HTTP 请求

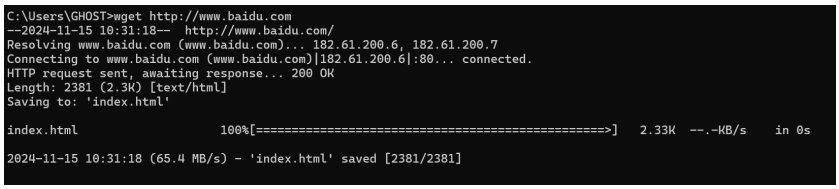

图 3: 利用 wget 抓包

接下来就开始了捕获的过程，如图所示：

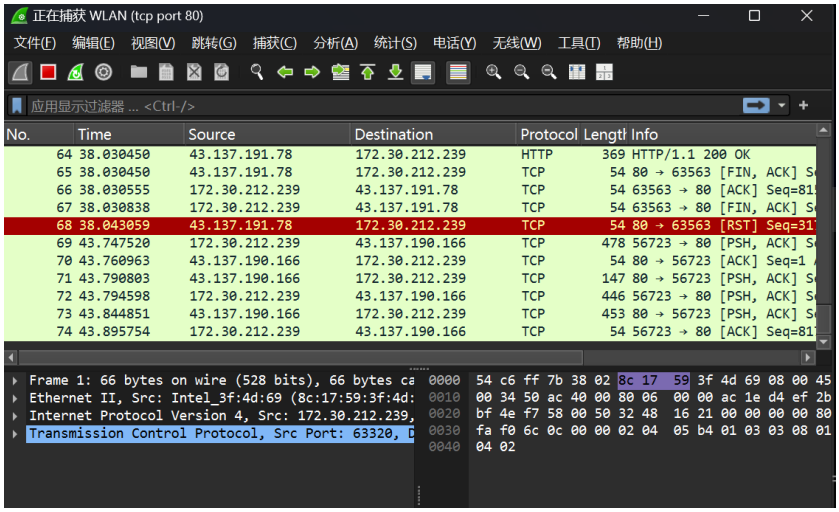

图 4: 捕获中

最后，我们在 `Wireshark` 中停止捕获，得到如下结果：

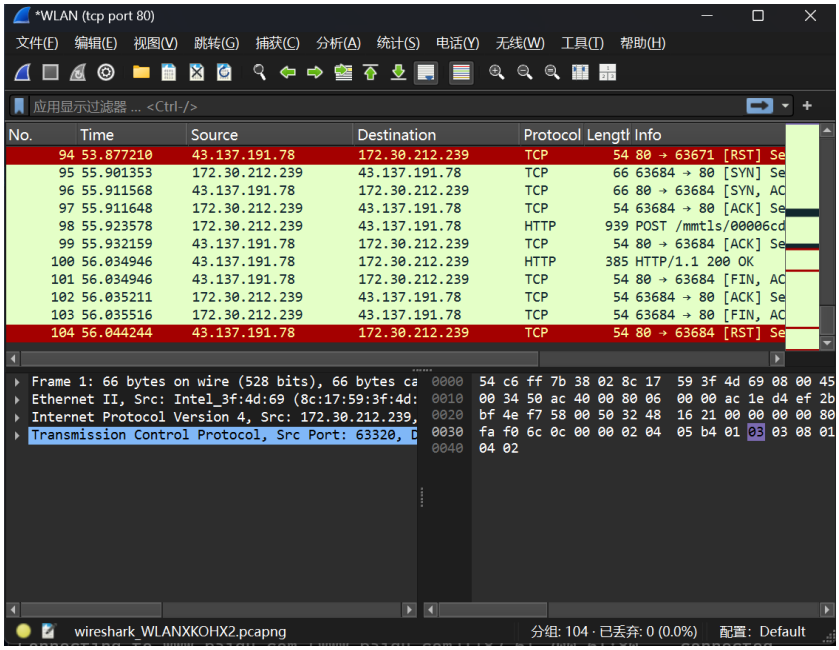

图 5: 捕获结束

4.2 绘制数据包结构

接下来，我们分析数据包的结构。在 Wireshark 中，我们找到了如图的 HTTP GET 协议包，可以看到其内容，如下图所示：

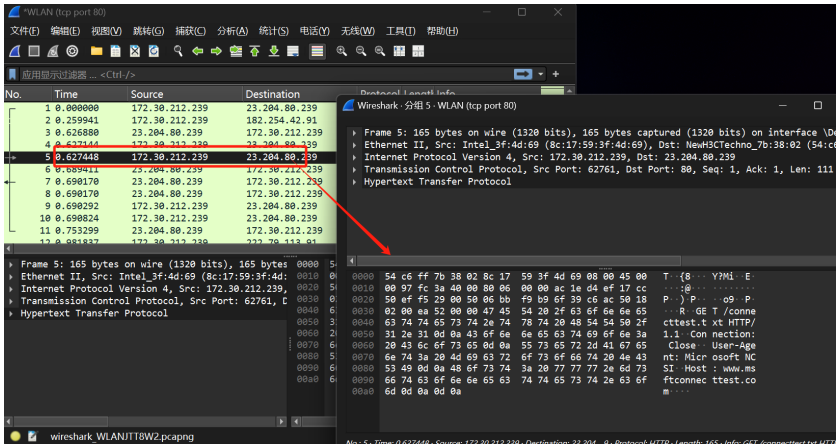

图 6: HTTP GET 协议包内容

先来大致的查阅一下数据包的结构：

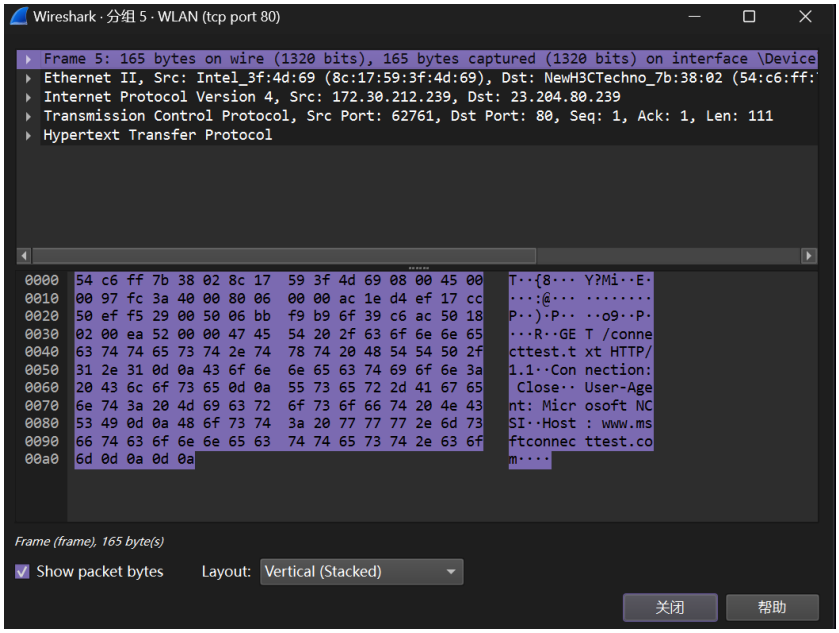

图 7: 数据包整体信息

我们可以看到：

第一个块：Frame，这不是一个协议，而是一个记录，描述有关数据包的整体信息，包括捕获时间和长度等。我们可以看到，整个数据包的长度是 165 字节。

第二个块：Ethernet II，这是以太网协议，可以看到这个标头的长度是 14 字节。

第三个块：Internet Protocol Version 4，这是 IP 协议，可以看到这个标头的长度是 20 字节。

第四个块：Transmission Control Protocol，这是 TCP 协议，可以看到这个标头的长度是 20 字节。

第五个块：Hypertext Transfer Protocol，这是 HTTP 协议，可以看到这个标头的长度是 111 字节。

接下来就是每一块的具体内容：

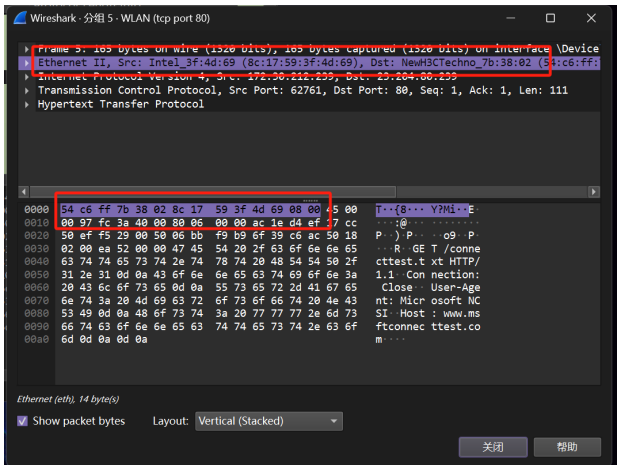

图 8: 以太网数据

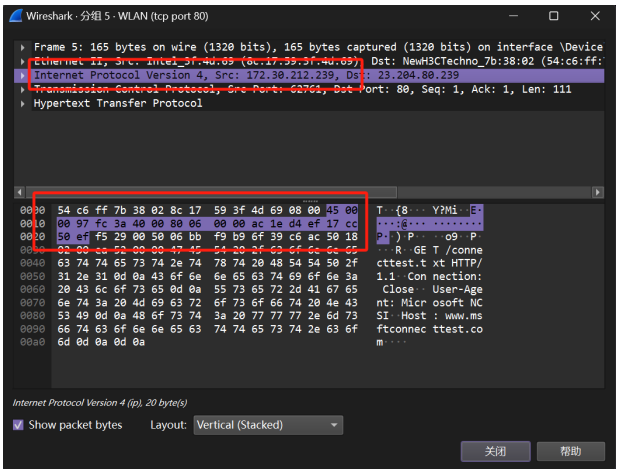

图 9: IP 数据

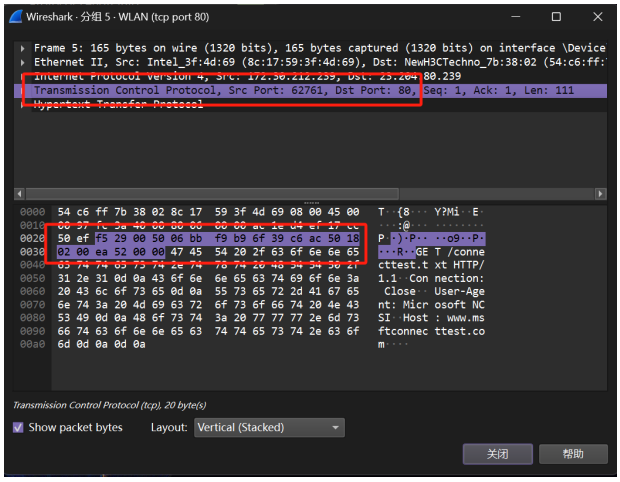

图 10: TCP 数据

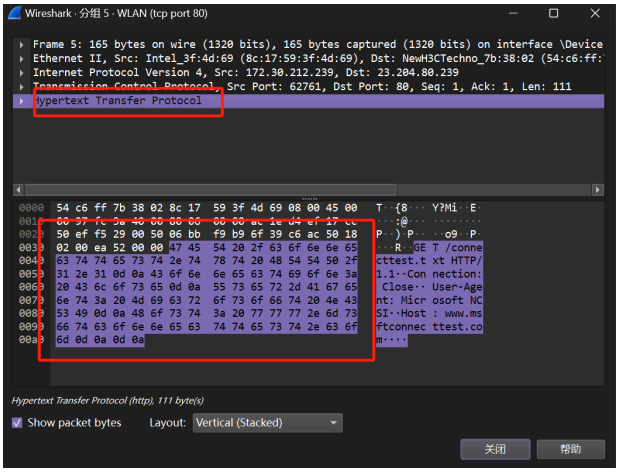

图 11: HTTP 数据

所以由此，绘制的协议包的结构如下图所示：

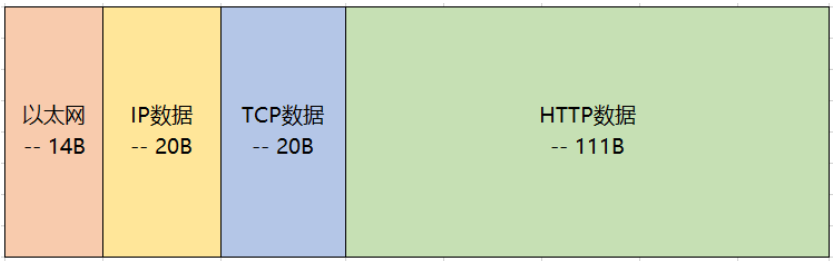

图 12: 绘制数据协议包

4.3 分析协议开销

以太网的开销为

$$\frac{14}{165} = 8.48\%$$

IP 的开销为

$$\frac{20}{165} = 12.12\%$$

TCP 的开销为

$$\frac{20}{165} = 12.12\%$$

整个捕获过程（从 SYN ACK 开始，到 HTTP 后的第一个 TCP 数据包结束）的开销为

$$\frac{14+20+20+66+54+60}{165+66+54+60} = 67.82\%$$

有用的 HTTP 数据的开销为

$$\frac{111}{165+66+54+60}=32.17\%$$

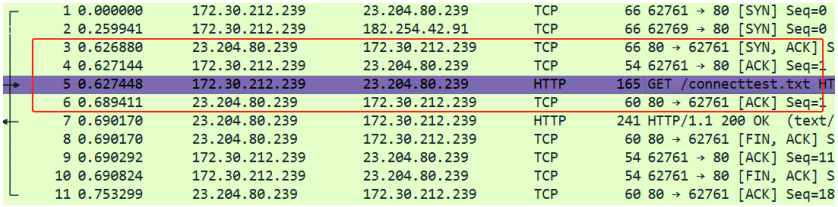

图 13: 整个捕获过程

4.4 分析解复用键

1. Which Ethernet header field is the demultiplexing key that tells it the next higher layer is IP? What value is used in this field to indicate **“IP”**？
   以太网头部中的 type 字段是解复用键，它的值为 0x0800，表示 IP 协议。
2. Which IP header field is the demultiplexing key that tells it the next higher layer is TCP? What value is used in this field to indicate **“TCP”**？
   IP 头部中的 protocol 字段是解复用键，它的值为 0x06，表示 TCP 协议。

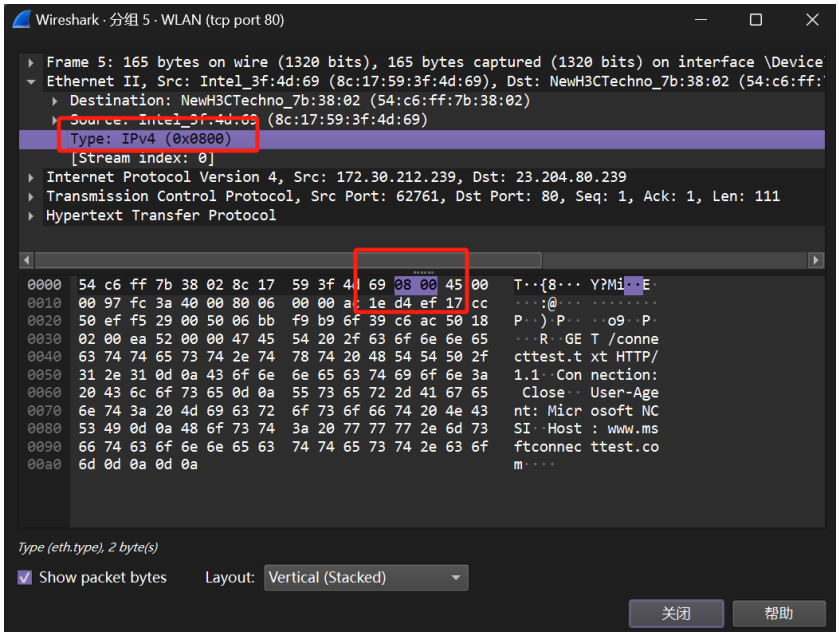

图 14: **以太网**头部

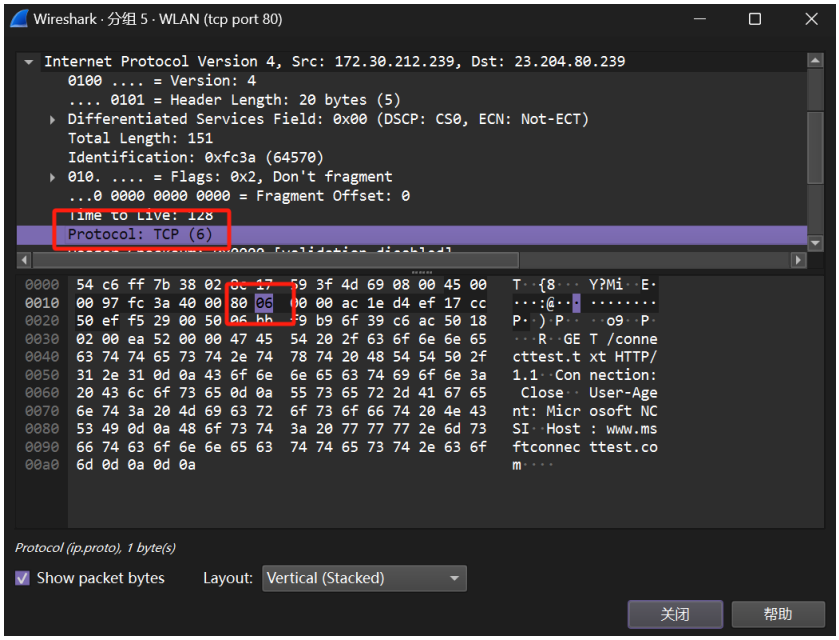

图 15: IP 头部

4.5 问题讨论

1. Look at a short TCP packet that carries no higher-layer data. To what entity is this packet destined? After all, if it carries no higher-layer data then it does not seem very useful to a higher layer protocol such as HTTP!
   该数据包的目的地是我们试图访问的网站或我们自己的设备。它主要包含确认和控制信息，对于建立 TCP 连接至关重要。这类报文是 TCP 三次握手过程的一部分，对于 HTTP 协议的运作是必不可少的。
2. In a classic layered model, one message from a higher layer has a header appended by the lower layer and becomes one new message. But this is not always the case. Above, we saw a trace in which the web response (one HTTP message comprised of an HTTP header and an HTTP payload) was converted into multiple lower layer messages (being multiple TCP packets). Imagine that you have drawn the packet structure (as in step 2) for the first and last TCP packet carrying the web response. How will the drawings differ?
   首个 TCP 数据包携带的是 HTTP 协议的头部信息，而随后的 TCP 数据包则承载 HTTP 协议的有效载荷内容。
3. In the classic layered model described above, lower layers append headers to the messages passed down from higher layers. How will this model change if a lower layer adds encryption?
   双方必须就加密算法达成一致，这样在数据传输过程中，接收方才能够正确解密数据。否则，接收方将无法解析数据包中的内容。
4. In the classic layered model described above, lower layers append headers to the messages passed down from higher layers. How will this model change if a lower layer adds compression?

发送方可以在数据包的头部附加解压缩方法的说明，这样接收方在接收到数据包后，便能够先进行解压缩，随后顺利解析数据包中的内容。

5 实验结果总结

在本次实验中，我不仅学会了如何使用 Wireshark 工具捕获不同协议层的数据包，还深入掌握了这些数据包的结构。通过对协议开销的细致分析，我进一步理解了复用键的概念。这些经历极大地加深了我对计算机网络中协议层的理解和认识。

6 附录

无