华东师范大学软件学院实验报告

实验课程：计算机网络实践	年级：2023 级本科	实验成绩：
实验名称：Lab1 Protocol Layer	姓名：张梓卫	
实验编号：（1）	学号：10235101526	实验日期：2024/11/15
指导老师：刘献忠	组号：	实验时间：2 课时

目录

一 实验目的

该实验是课程《计算机网络实践》的第一次实验，全名《Protocol Layer》，目标如下：

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

二 实验内容与实验步骤

- 1. 打开命令行，输入：wget http://www.baidu.com. 观察到 OK 为止。
- 2. 启动 Wireshark，在菜单栏的捕获-> 选项中进行设置，选择已连接的以太网，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. 点开命令行，重新输入:wget http://www.baidu.com
- 4. 打开 Wireshark，停止捕获。
- 5. 分析并且绘制协议包、数据包结构
- 6. 分析解复用键
- 7. 进行问题讨论

三 实验环境

使用 Wireshark v4.2.5, Windows 11 Pro, Wget Tools 进行实验。
实验报告使用 LaTeX 进行撰写，使用 Vim 编辑器进行文本编辑。

四 实验过程与分析

1 实验工具安装

安装 Wireshark 与 Wget Tools，安装完毕后，将 Wget 的 `/bin` 文件夹添加至系统环境变量 PATH，以便能够在命令行和 Powershell 中直接通过 wget 命令调用。

2 使用 Wget 通过协议获取内容

打开命令行，输入命令 `wget http://www.baidu.com`，等待回应。
显示 HTTP 200 OK，表示内容获取成功。

图 1: Wget Result

3 启动 Wireshark 捕获数据包

启动 Wireshark，按照实验要求将捕获选项设置为：已连接的以太网，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。

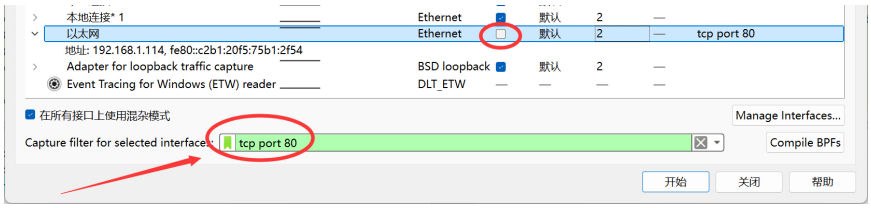


图 2: Wireshark Filter

设置 rename 选项为开启：

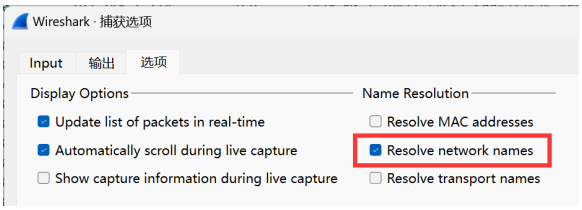


图 3: Wireshark Rename

4 通过 Wireshark 捕获数据包

在命令行中输入命令 `wget http://www.baidu.com`，等待回应。

图 4: Again Wget

在 Wireshark 捕获数据包，停止捕获。

No.	Time	Source	Destination	Protocol	Lengtl	Info
1	0.000000	192.168.1.114	180.101.50.188	TCP	66	27193 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2	0.008861	180.101.50.188	192.168.1.114	TCP	66	80 → 27193 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1440 WS=32 SACK_PERM
3	0.008958	192.168.1.114	180.101.50.188	TCP	54	27193 → 80 [ACK] Seq=1 Ack=1 Win=263424 Len=0
4	0.009565	192.168.1.114	180.101.50.188	HTTP	155	GET / HTTP/1.0
5	0.018798	180.101.50.188	192.168.1.114	TCP	60	80 → 27193 [ACK] Seq=1 Ack=102 Win=29056 Len=0
6	0.019352	180.101.50.188	192.168.1.114	TCP	682	80 → 27193 [PSH, ACK] Seq=1 Ack=102 Win=29056 Len=628 [TCP segment of a reassembled PDU]
7	0.019352	180.101.50.188	192.168.1.114	TCP	566	80 → 27193 [PSH, ACK] Seq=629 Ack=102 Win=29056 Len=512 [TCP segment of a reassembled PDU]
8	0.019352	180.101.50.188	192.168.1.114	HTTP	1411	HTTP/1.0 200 OK (text/html)
9	0.019352	180.101.50.188	192.168.1.114	TCP	60	80 → 27193 [FIN, ACK] Seq=2498 Ack=102 Win=29056 Len=0
10	0.019403	192.168.1.114	180.101.50.188	TCP	54	27193 → 80 [ACK] Seq=102 Ack=2499 Win=263424 Len=0
11	0.027867	192.168.1.114	180.101.50.188	TCP	54	27193 → 80 [FIN, ACK] Seq=102 Ack=2499 Win=263424 Len=0
12	0.037193	180.101.50.188	192.168.1.114	TCP	60	80 → 27193 [ACK] Seq=2499 Ack=103 Win=29056 Len=0
13	3.041806	180.101.50.188	192.168.1.114	TCP	60	80 → 27193 [RST] Seq=2499 Win=0 Len=0

图 5: HTTP Get

根据 TCP 三次握手的内容，即：三次握手过程总结

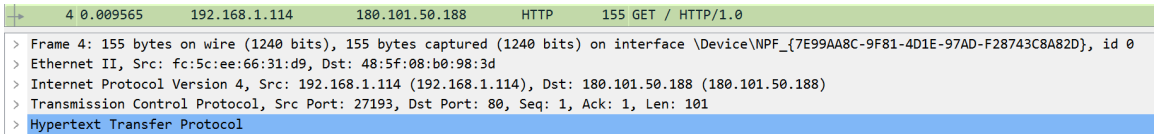
- 第一次握手（SYN）：客户端发送 SYN 报文，表示请求连接。
- 第二次握手（SYN-ACK）：服务器收到 SYN 报文后，返回 SYN-ACK 报文，表示同意连接并同步序列号。
- 第三次握手（ACK）：客户端收到 SYN-ACK 报文后，发送 ACK 报文，确认连接建立。

具体而言，抓包获取的内容如下：

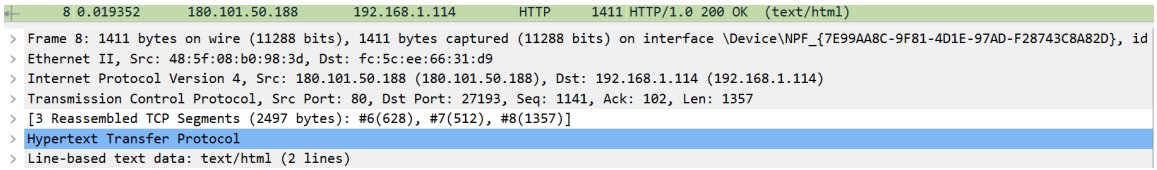
- wget 输出显示请求了 http://www.baidu.com，服务器 IP 地址为 180.101.50.188，该 IP 与抓包中相关帧中的 IP 匹配。
- 抓包的第 1 帧表示 TCP 三次握手的初始 SYN 包，第 2-3 帧完成握手的 SYN-ACK 和 ACK。
- 第 4 帧显示客户端（192.168.1.114）向服务器（180.101.50.188）发送的 HTTP GET 请求 (GET / HTTP/1.0)。
- 第 8 帧显示服务器返回 HTTP 200 OK 响应，表明内容已成功接收。此响应包含 HTTP 数据，共 1411 字节。
- 第 9-12 帧为客户端和服务器之间的连接关闭过程，其中 9 和 11 帧为 FIN 包，10 和 12 帧为 ACK 包，最后第 13 帧为服务器发出的 RST 包。

5 分析协议包

选中 HTTP 协议中含有 GET 的协议包，再查看带有 200 OK 字样的返回包，查看其内容：

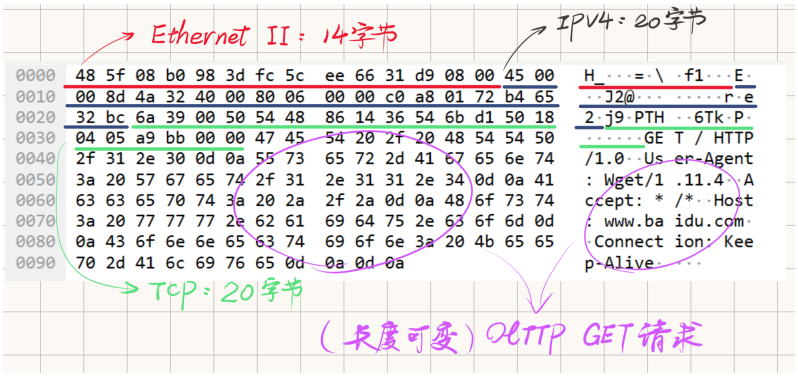


(a) HTTP GET



(b) HTTP GET 200 OK

进入下方含有十六进制码的区域，按照实验手册中所描述的协议包结构进行分析。Frame 字段是总体信息的记录，Ethernet 字段是以太网信息的记录，IP 字段是 IP 信息的记录，TCP 字段是 TCP 信息的记录，HTTP 字段是 HTTP 信息的记录。通过点击不同的字段，可以总结为下图所示的结构：



按照实验手册所要求的画为长条状的矩形，可为下图的结构：

6 分析协议开销

将 HTTP 数据（报头和消息）视为网络携带的有用数据，而将较低层报头（TCP、IP 和以太网）视为开销。

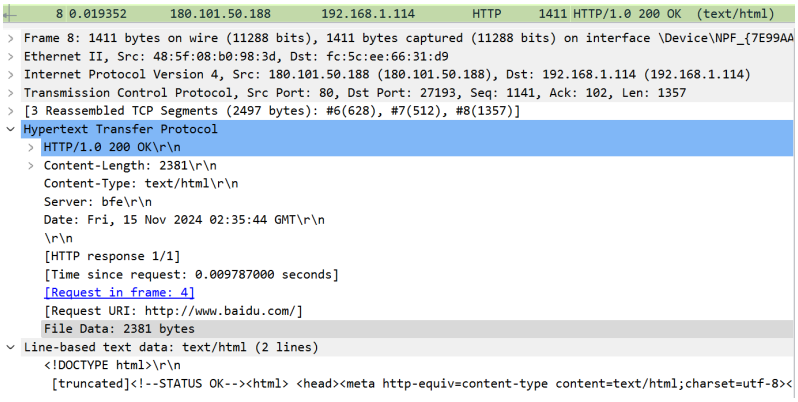


图 7: Protocol Overhead

展开 HTTP GET 200 OK 可见，真正传输的 File Data 作为有效数据实际的 HTTP 内容负载，在此应为 2381 字节。下面，我观察到 HTTP 相应数据分成了 3 个 TCP 包发送，三个重新组装的 TCP 段：

Frame 6（628 字节）、Frame 7（512 字节）和 Frame 8（1357 字节）

它们组成了一个完整的 HTTP 响应，共 2497 字节，经过重新组装形成完整的 HTTP 数据。

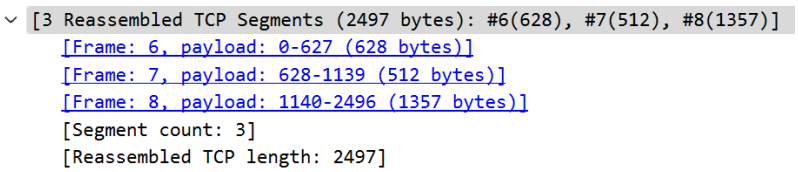


图 8: Multi TCP Packets

解答 四 .1: Cost

此时，分成 3 个 TCP 包发送会造成协议开销，按照上面所分析的过程，可以知道协议开销应该是：

$$3 \times [14(\text{Ethernet}) + 20(\text{IP}) + 20(\text{TCP})] = 162 Bytes.$$

故协议开销占总体数据传输的占比应为：

$$\frac{162}{162 + 2381} \times 100\% \approx 6.37\%.$$

7 分析解复用键

IP 必须能够确定 IP 消息的内容是 TCP 数据包，以便将其交给 TCP 协议进行处理。答案是，协议使用其报头中的信息（称为“解复用密钥”）来确定更高层。

回顾题目的问题，如下所示：

- 1. 以太网头部中哪一部分是解复用键并且告知它的下一个高层指的是 IP，在这一包内哪一个值可以表示 IP？
- 2.IP 头部中哪一部分是解复用键并且告知它的下一一个高层指的是 TCP，在这一包内哪一个值可以表示 TCP？

回顾抓包内容：

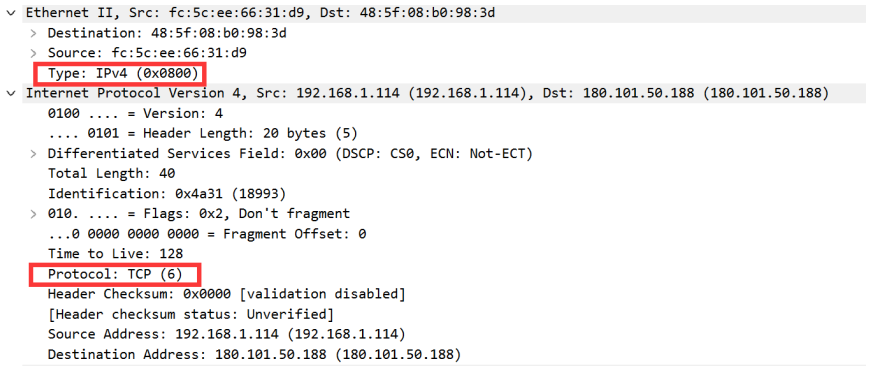
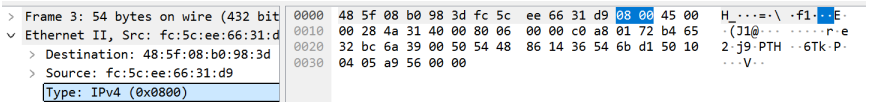
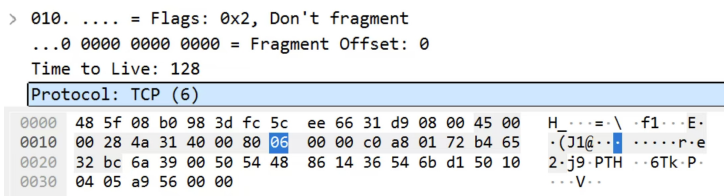


图 9: 解复用键抓包分析

Type: IPv4(0x0800) 是解复用键并且告知它的下一个高层是 IP。其中，0x0800 表示 IP。



(a) IPv4 Demultiplexing



(b) TCP Demultiplexing

Protocol: TCP (6) 是解复用键并且告知它的下一个高层指的是 TCP。其中，6 表示 TCP。

8 课后思考题

8 .1 Question 1

Look at a short TCP packet that carries no higher-layer data. To what entity is this packet destined?

观察一个携带无更高层数据的短 TCP 包。该包的目的实体是什么？

解答

一个携带无更高层数据的短 TCP 包通常用于连接管理和控制，而不是数据传输。这种包的示例包括：

SYN 和 ACK 包，用于建立连接的三次握手过程。FIN 和 RST 包，用于终止连接。空的 ACK 包，用于确认数据包的接收，而无需返回任何数据。这些包被发送到目标设备的 TCP 层，并用于管理连接状态，而不是被转发到更高层的协议（如 HTTP），因为它们不携带更高层的有效负载。

8 .2 Question 2

How would the drawings differ between the first and last TCP packet carrying the web response?
携带网页响应的第一个和最后一个 TCP 包在绘图上有何不同？

解答

在携带网页响应的第一个 TCP 包中，包含 HTTP 响应头的开始部分，包括状态码以及响应数据的初始部分。

而在携带网页响应的最后一个 TCP 包中，会包含 HTTP 负载的结尾部分。

主要区别在于每个包所承载的 HTTP 负载部分，只有第一帧具有 HTTP 协议头，而最后一个包携带消息的结尾，有效载荷。两个包都包含 TCP 头，但序列号和负载内容反映了它们在 HTTP 消息中的不同位置。

8 .3 Question 3

How would the model change if a lower layer adds encryption?
如果低层添加加密，模型将如何改变？

解答

如果低层添加加密，从更高层传递下来的数据（如 HTTP 消息）将在低层（通常是传输层如 TLS 或网络层如 IPsec）被加密。

原始的更高层消息被转变成一个加密的负载，随后由低层添加头部。只有拥有解密密钥的实体（如客户端和服务器）才能访问原始消息。这种模型通常称为“加密封装”，因为数据被封装在额外的加密层中，以保护消息在网络中的传输安全。

8 .4 Question 4

How would the model change if a lower layer adds compression?
如果低层添加压缩，模型将如何改变？

解答

如果低层添加压缩，从更高层传递下来的数据在被低层添加头部之前被压缩。这会在以下方面改变模型：

数据被低层压缩成较小的格式。随后低层会在压缩后的数据上添加自己的头部。接收端需要在将消息传递到更高层之前解压缩负载。这种压缩封装使得数据传输更高效，因为压缩减少了数据大小，但接收端必须在将消息传递给更高层之前对其进行解压缩。

五 实验结果总结

在本次实验中，我通过 Wireshark 软件学会了对 HTTP 协议进行抓包分析，验证了协议分层模型在实际传输中的应用，并深入了解了每一层协议的报文结构和协议开销。

- 使用 Wget 获取了 http://www.baidu.com 的网页内容，并通过 Wireshark 观察到 HTTP 请求和响应的传输过程。
- 使用 Wireshark 进行抓包分析，获取到了 TCP 三次握手、HTTP GET 请求、HTTP 响应以及 TCP 连接终止的完整过程。
- 计算了 HTTP 传输的协议开销，验证了在数据分片传输的情况下，各层协议头部会增加协议开销。

- 见识了了解了复用键在以太网、IP 和 TCP 层的重要性，并观察到如何通过解复用键确定上层协议。

六 附录

参考资料

- 怎么用 wireshark 看数据包的开销
- 这么详细的 Wireshark 的抓包和分析，工作中是没人告诉你的！