

实验报告：Protocol Layer 实验一

课程名称：计算机网络实践

年级：大二

上机实践成绩：

指导教师：章玥

姓名：邱吉尔

学号：10235101533

上机实践日期：

2024/11/11

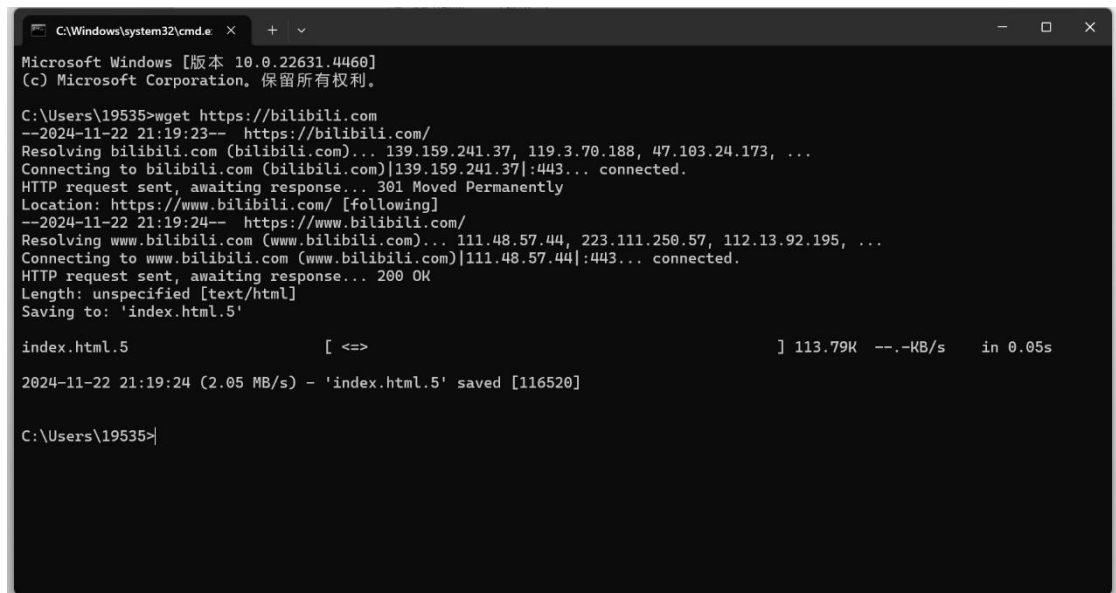
一、目的

1. 学习协议和分层如何用数据包表示；
2. 熟悉 wireshark 软件、curl、wget 等常用软件的使用，掌握网络抓包的方法，能在所用电脑上进行抓包；
3. 了解 IP 数据包格式，能应用该软件分析数据包格式，查看抓到的包的内容，并分析对应的 IP 数据包格式；
4. 抓包分析数据包，估算协议的开销；
5. 通过数据包抓取实验，将理论与实践相结合，深入理解协议层的字段与结构特征。

二、实验步骤

1. 抓包

打开 cmd 终端，抓包一个网站，这里使用了 wget <https://bilibili.com>



```
C:\Windows\system32\cmd.exe X + v
Microsoft Windows [版本 10.0.22631.4460]
(c) Microsoft Corporation. 保留所有权利。

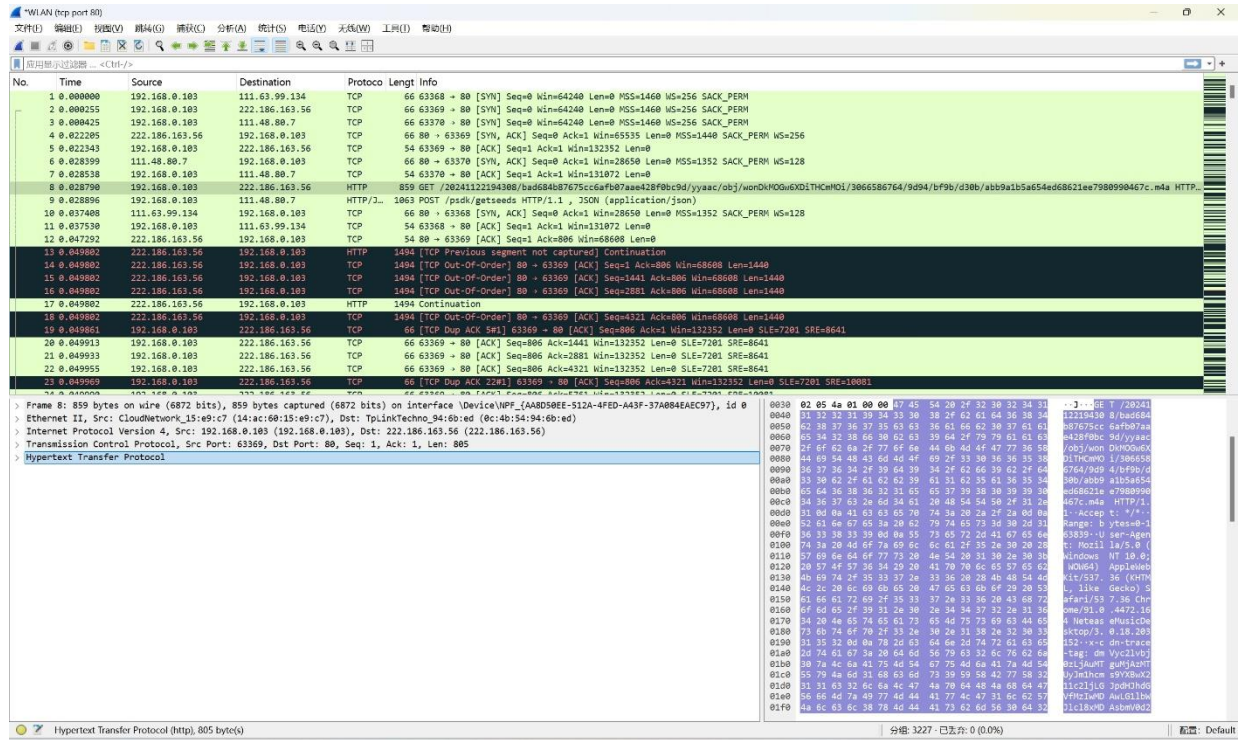
C:\Users\19535>wget https://bilibili.com
--2024-11-22 21:19:23-- https://bilibili.com/
Resolving bilibili.com (bilibili.com)... 139.159.241.37, 119.3.70.188, 47.103.24.173, ...
Connecting to bilibili.com (bilibili.com)|139.159.241.37|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://www.bilibili.com/ [following]
--2024-11-22 21:19:24-- https://www.bilibili.com/
Resolving www.bilibili.com (www.bilibili.com)... 111.48.57.44, 223.111.250.57, 112.13.92.195, ...
Connecting to www.bilibili.com (www.bilibili.com)|111.48.57.44|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html.5'

index.html.5          [ <=>                ] 113.79K --.-KB/s   in 0.05s

2024-11-22 21:19:24 (2.05 MB/s) - 'index.html.5' saved [116520]

C:\Users\19535>
```

Wireshark 抓包截图如下：



2. 分析协议包内容

找到上方 Protocol 为 HTTP，且 Info 是 GET 的，点击 Hypertext，可以看到整个窗口的最下方显示 805 byte(s)，这是 HTTP 的实际有效开销。

接下来分别查看 Ethernet、TCP、IP 占用的字节，分别如下为 14 types、20 types、20 types

*WLAN (tcp port 80)

文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(I) 帮助(H)

应用显示过滤器 ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.103	111.63.99.134	TCP	66	63368 → 80 [SYN] Seq
2	0.000255	192.168.0.103	222.186.163.56	TCP	66	63369 → 80 [SYN] Seq
3	0.000425	192.168.0.103	111.48.80.7	TCP	66	63370 → 80 [SYN] Seq
4	0.022205	222.186.163.56	192.168.0.103	TCP	66	80 → 63369 [SYN, ACK]
5	0.022343	192.168.0.103	222.186.163.56	TCP	54	63369 → 80 [ACK] Seq
6	0.028399	111.48.80.7	192.168.0.103	TCP	66	80 → 63370 [SYN, ACK]
7	0.028538	192.168.0.103	111.48.80.7	TCP	54	63370 → 80 [ACK] Seq
8	0.028790	192.168.0.103	222.186.163.56	HTTP	859	GET /20241122194308
9	0.028896	192.168.0.103	111.48.80.7	HTTP/J...	1063	POST /psdk/getseeds
10	0.037408	111.63.99.134	192.168.0.103	TCP	66	80 → 63368 [SYN, ACK]

> Frame 8: 859 bytes on wire (6872 bits), 859 bytes captured (6872 bits) on interface 0

Ethernet II, Src: CloudNetwork_15:e9:c7 (14:ac:60:15:e9:c7), Dst: 192.168.0.103 (08:00:27:00:00:00)

Internet Protocol Version 4, Src: 192.168.0.103 (192.168.0.103), Dst: 222.186.163.56 (222.186.163.56)

Transmission Control Protocol, Src Port: 63369, Dst Port: 80

Hypertext Transfer Protocol

0000 0c 4b 54 94 6b ed 14 ac 60 15

0010 03 4d b9 cd 40 00 80 06 fa da

0020 a3 38 f7 89 00 50 3c 40 33 04

0030 02 05 4a 01 00 00 47 45 54 20

0040 31 32 32 31 39 34 33 30 38 2f

0050 62 38 37 36 37 35 63 63 36 61

0060 65 34 32 38 66 30 62 63 39 64

0070 2f 6f 62 6a 2f 77 6f 6e 44 6b

0080 44 69 54 48 43 6d 4d 4f 69 2f

0090 36 37 36 34 2f 39 64 39 34 2f

00a0 33 30 62 2f 61 62 62 39 61 31

00b0 65 64 36 38 36 32 31 65 65 37

00c0 34 36 37 63 2e 6d 34 61 20 48

00d0 31 0d 0a 41 63 63 65 70 74 3a

Ethernet (eth), 14 byte(s)

分组: 3227 · 已丢弃: 0 (0.0%) 配置: Default

*WLAN (tcp port 80)

文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(I) 帮助(H)

应用显示过滤器 ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.103	111.63.99.134	TCP	66	63368 → 80 [SYN] Seq
2	0.000255	192.168.0.103	222.186.163.56	TCP	66	63369 → 80 [SYN] Seq
3	0.000425	192.168.0.103	111.48.80.7	TCP	66	63370 → 80 [SYN] Seq
4	0.022205	222.186.163.56	192.168.0.103	TCP	66	80 → 63369 [SYN, ACK]
5	0.022343	192.168.0.103	222.186.163.56	TCP	54	63369 → 80 [ACK] Seq
6	0.028399	111.48.80.7	192.168.0.103	TCP	66	80 → 63370 [SYN, ACK]
7	0.028538	192.168.0.103	111.48.80.7	TCP	54	63370 → 80 [ACK] Seq
8	0.028790	192.168.0.103	222.186.163.56	HTTP	859	GET /20241122194308
9	0.028896	192.168.0.103	111.48.80.7	HTTP/J...	1063	POST /psdk/getseeds
10	0.037408	111.63.99.134	192.168.0.103	TCP	66	80 → 63368 [SYN, ACK]

> Frame 8: 859 bytes on wire (6872 bits), 859 bytes captured (6872 bits) on interface 0

Ethernet II, Src: CloudNetwork_15:e9:c7 (14:ac:60:15:e9:c7), Dst: 192.168.0.103 (08:00:27:00:00:00)

Internet Protocol Version 4, Src: 192.168.0.103 (192.168.0.103), Dst: 222.186.163.56 (222.186.163.56)

Transmission Control Protocol, Src Port: 63369, Dst Port: 80

Hypertext Transfer Protocol

0020 a3 38 f7 89 00 50 3c 40 33 04

0030 02 05 4a 01 00 00 47 45 54 20

0040 31 32 32 31 39 34 33 30 38 2f

0050 62 38 37 36 37 35 63 63 36 61

0060 65 34 32 38 66 30 62 63 39 64

0070 2f 6f 62 6a 2f 77 6f 6e 44 6b

0080 44 69 54 48 43 6d 4d 4f 69 2f

0090 36 37 36 34 2f 39 64 39 34 2f

00a0 33 30 62 2f 61 62 62 39 61 31

00b0 65 64 36 38 36 32 31 65 65 37

00c0 34 36 37 63 2e 6d 34 61 20 48

00d0 31 0d 0a 41 63 63 65 70 74 3a

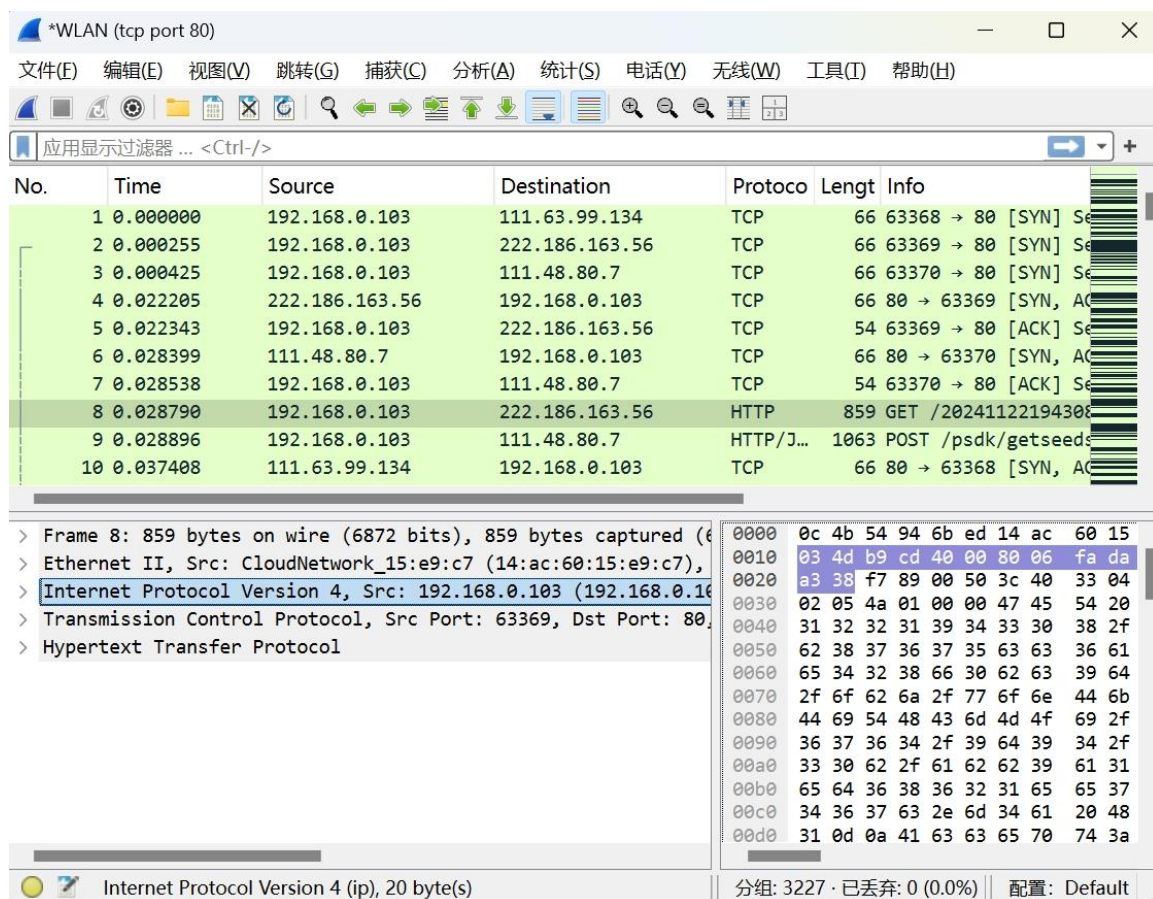
00e0 52 61 6e 67 65 3a 20 62 79 74

00f0 36 33 38 33 39 0d 0a 55 73 65

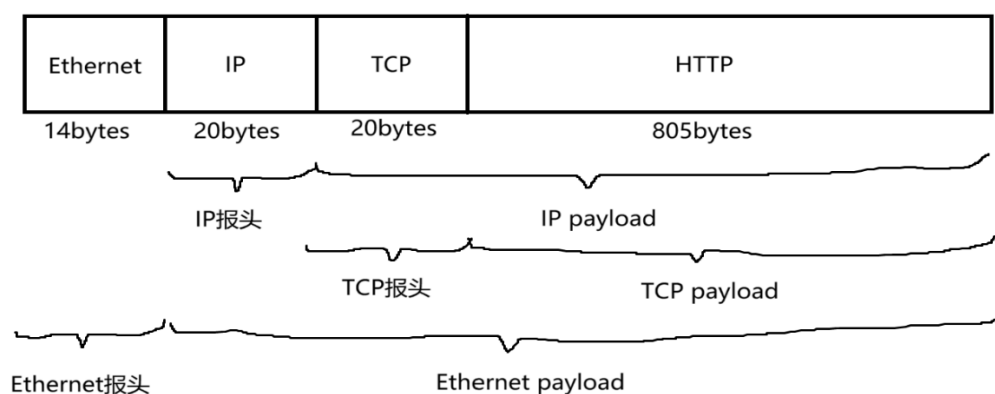
Transmission Control Protocol (tcp), 20 byte(s)

分组: 3227 · 已丢弃: 0 (0.0%) 配置: Default

第 3 页 共 7 页



3. 画一个关于使用 GET 方法的 HTTP 请求的图，如下：



4. 根据数据包的抓取结果，分析协议开销：

协议总开销为 $66+54+859+66=1045$

5. 估计协议的开销或者是协议开销占用下载字节的百分比。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，你认为这种开销是必要的吗？（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。)：

百分比为 $805/1045=77.03\%$

有必要

①协议开销会直接影响成本。分析后可以评估实际的有用数据比例，从而优化传输。

②开销可能需要更多处理资源（如 CPU 或内存），分析后可评估现有硬件是否匹配需求。

③通过对开销的细致检查，可以发现可能导致数据丢失或性能下降的问题

观察下载的以太网和 IP 头包回答下面问题：

- 1、以太网头部中哪一部分是解复用（解复用：找到正确的上一层协议来处理到达的包的行为叫做 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？

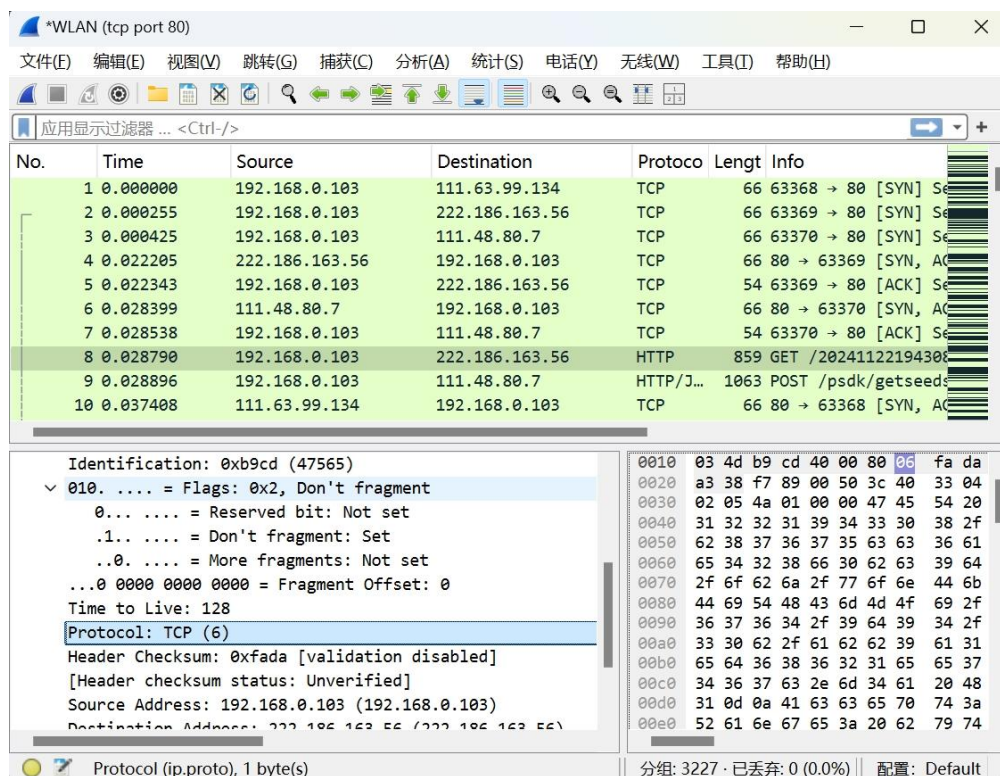
The screenshot shows a Wireshark packet capture of a network interface *WLAN (tcp port 80). The packet list shows a series of TCP and HTTP packets. The selected packet (No. 8) is an HTTP GET request. The packet details pane shows the Ethernet II header with the following fields:

- Destination: TpLinkTechno_94:6b:ed (0c:4b:54:94:6b:ed)
- Source: CloudNetwork_15:e9:c7 (14:ac:60:15:e9:c7)
- Type: IPv4 (0x0800)

The packet bytes pane shows the raw data of the packet, including the Ethernet II header and the IP header.

以太网解复用键是 Type，对应的值是 0x0800

- 2、IP 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 TCP，在这一包内哪一个值可以表示 TCP？



IP 解复用键是 Protocol，对应的值是 6

三、问题与思考

1. 查看不包含高层数据的短 TCP 数据包，查看它发往哪？不携带高层数据的数据包有用吗？

有用，此数据报可能是用于建立链接，也有可能是 ACK，NAK，带有确认信息，查看这些数据包的目的地，可以帮助确认它们是否有用，以及是否需要优化查找异常

2. 在经典的分层模型中，低层字段包装到高层数据包外面，成为一条新消息。但这并非总是如此，Web 响应（一个包含 HTTP 标头和 HTTP 有效负载的 HTTP 消息）可能被转换为多个较低层的消息（即多个 TCP 数据包）。假设你为 Web 响应的第一个和最后一个 TCP 数据包绘制了数据包结构，那么该结构与经典分层模型有什么不同？

第一个 TCP 数据包：

Ethernet 头 | IP 头 | TCP 头 | HTTP 部分（头部数据）

最后一个 TCP 数据包：

Ethernet 头 | IP 头 | TCP 头 | HTTP 部分（尾部数据）

不同点：

- ① 经典模型中假定每条消息从高层到低层逐步封装，形成单一完整的数据包
实际情况：HTTP 消息通常会被分割成多个 TCP 数据包，每个 TCP 包只包含部分高层数据，接收端需要重组这些分段
- ② 经典模型中每条完整的消息仅对应一个低层头部

实际情况：每段高层数据都需要单独添加低层头部，造成更多的协议开销

3. 在上述经典分层模型中，低层字段包装到高层数据包外面，如果较低层添加加密，此模型将如何更改？

在较低层的报头处添加密文，解密后才能继续向后读取

4. 在上述经典分层模型中，低层字段包装到高层数据包外面,如果较低的层添加压缩，此模型将如何更改？

添加额外的头部或字段，用于描述压缩方式和解压缩所需的信息，传输时传输已被压缩后的信息