wireshark 抓包实验

一. 实验目的

1.利用 wireshark 软件分析 HTTP 及其下层协议（TCP 协议）

2.了解网络中数据封装的概念

3.掌握 HTTP 及 TCP 协议的工作过程

二. 实验内容

1.启动 wireshark 软件，进行报文截获

2.在浏览器访问 www.xjtu.edu.cn 页面。（打开网页，浏览并关闭页面）

3.停止 ethereal 的报文截获，将截获命名为“http—学号”，分析截获报文。

4.在思源学堂提交实验报告及存储的截获报文。

三.实验步骤

预处理

wwww.xjtu.edu.cn 为经常访问的网页，因此在本地存在缓存，为清理缓存进行以下操作

1. 在清理浏览记录功能中，清除网页缓存和 cookie 等内容

2. 进入 www.xjtu.edu.cn ，摁 F12 进入开发者工具，进入后选择网络(network)一栏，勾选禁用缓存

Ping 测试

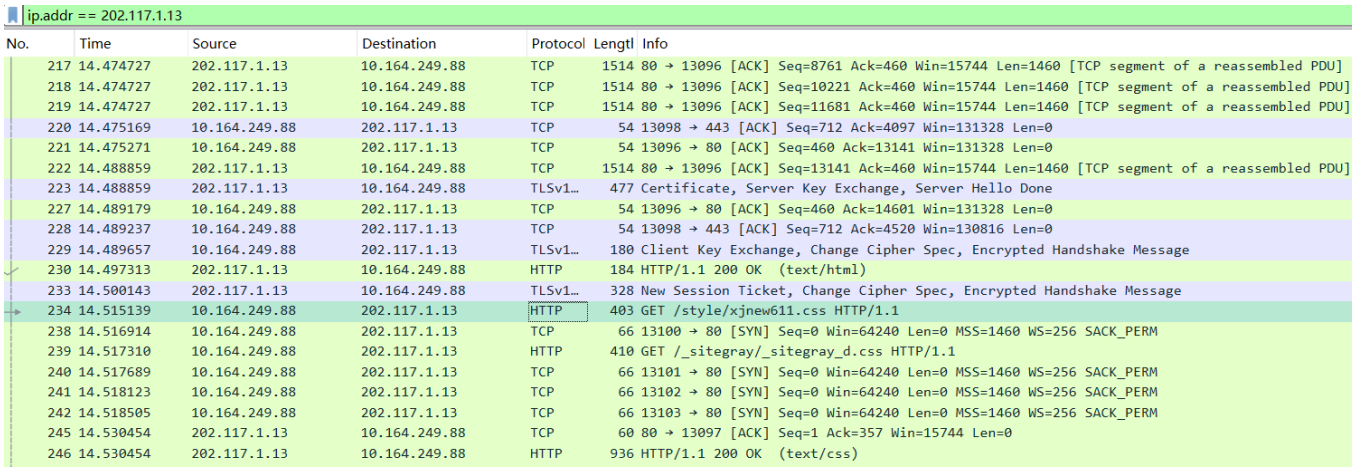
```
PS C:\Users\86178> ping www.xjtu.edu.cn

正在 Ping www.xjtu.edu.cn [202.117.1.13] 具有 32 字节的数据:
来自 202.117.1.13 的回复: 字节=32 时间=26ms TTL=60
来自 202.117.1.13 的回复: 字节=32 时间=145ms TTL=60
来自 202.117.1.13 的回复: 字节=32 时间=10ms TTL=60
来自 202.117.1.13 的回复: 字节=32 时间=647ms TTL=60

202.117.1.13 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 10ms, 最长 = 647ms, 平均 = 207ms
```

通过 Ping 查出 www.xjtu.end.cn 的 ip 地址为

www.xjtu.edu.cn 的 IP:202.117.1.13

访问网页后抓包分析

HTTP 请求报文：



ip.addr == 202.117.1.13

No.	Time	Source	Destination	Protocol	Lengtl	Info
217	14.474727	202.117.1.13	10.164.249.88	TCP	1514	80 → 13096 [ACK] Seq=8761 Ack=460 Win=15744 Len=1460 [TCP segment of a reassembled PDU]
218	14.474727	202.117.1.13	10.164.249.88	TCP	1514	80 → 13096 [ACK] Seq=10221 Ack=460 Win=15744 Len=1460 [TCP segment of a reassembled PDU]
219	14.474727	202.117.1.13	10.164.249.88	TCP	1514	80 → 13096 [ACK] Seq=11681 Ack=460 Win=15744 Len=1460 [TCP segment of a reassembled PDU]
220	14.475169	10.164.249.88	202.117.1.13	TCP	54	13098 → 443 [ACK] Seq=712 Ack=4097 Win=131328 Len=0
221	14.475271	10.164.249.88	202.117.1.13	TCP	54	13096 → 80 [ACK] Seq=460 Ack=13141 Win=131328 Len=0
222	14.488859	202.117.1.13	10.164.249.88	TCP	1514	80 → 13096 [ACK] Seq=13141 Ack=460 Win=15744 Len=1460 [TCP segment of a reassembled PDU]
223	14.488859	202.117.1.13	10.164.249.88	TLSv1…	477	Certificate, Server Key Exchange, Server Hello Done
227	14.489179	10.164.249.88	202.117.1.13	TCP	54	13096 → 80 [ACK] Seq=460 Ack=14601 Win=131328 Len=0
228	14.489237	10.164.249.88	202.117.1.13	TCP	54	13098 → 443 [ACK] Seq=712 Ack=4520 Win=130816 Len=0
229	14.489657	10.164.249.88	202.117.1.13	TLSv1…	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
230	14.497313	202.117.1.13	10.164.249.88	HTTP	184	HTTP/1.1 200 OK (text/html)
233	14.500143	202.117.1.13	10.164.249.88	TLSv1…	328	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
234	14.515139	10.164.249.88	202.117.1.13	HTTP	403	GET /style/xjnew611.css HTTP/1.1
238	14.516914	10.164.249.88	202.117.1.13	TCP	66	13100 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
239	14.517310	10.164.249.88	202.117.1.13	HTTP	410	GET /_sitegray/_sitegray_d.css HTTP/1.1
240	14.517689	10.164.249.88	202.117.1.13	TCP	66	13101 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
241	14.518123	10.164.249.88	202.117.1.13	TCP	66	13102 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
242	14.518505	10.164.249.88	202.117.1.13	TCP	66	13103 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
245	14.530454	202.117.1.13	10.164.249.88	TCP	60	80 → 13097 [ACK] Seq=1 Ack=357 Win=15744 Len=0
246	14.530454	202.117.1.13	10.164.249.88	HTTP	936	HTTP/1.1 200 OK (text/css)

物理层的数据帧详细概要

```
Frame 234: 403 bytes on wire (3224 bits), 403 bytes captured (3224 bits) on interface \Device\NPF_{E17A904F-23A2-4F81-9F8B-3E8CDE61A002}, id 0
    Section number: 1
  > Interface id: 0 (\Device\NPF_{E17A904F-23A2-4F81-9F8B-3E8CDE61A002})
    Encapsulation type: Ethernet (1)
    Arrival Time: Jan  6, 2024 18:28:54.370012000 中国标准时间
    UTC Arrival Time: Jan  6, 2024 10:28:54.370012000 UTC
    Epoch Arrival Time: 1704536934.370012000
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.014996000 seconds]
    [Time delta from previous displayed frame: 0.014996000 seconds]
    [Time since reference or first frame: 14.515139000 seconds]
    Frame Number: 234
    Frame Length: 403 bytes (3224 bits)
    Capture Length: 403 bytes (3224 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp:http]
    [Coloring Rule Name: HTTP]
    [Coloring Rule String: http || tcp.port == 80 || http2]
```

该帧为 234 号帧，线路字节 3224bit，实际捕获 403bit，端口号为 0 号端口，端口在计算机中采用的封装方式为 Ethernet(1)，捕获的时间为 2024 年 1 月 6 号，18：25：54 左右。以及一些和帧有关的时间信息。该帧没有被标记，也没有被忽略。该帧的协议为 http，端口号为 80。

数据链路层以太帧的首部信息：

```
Ethernet II, Src: Intel_d2:86:3d (e4:5e:37:d2:86:3d), Dst: H3CTechnolog_b4:e0:01 (38:97:d6:b4:e0:01)
  Destination: H3CTechnolog_b4:e0:01 (38:97:d6:b4:e0:01)
      Address: H3CTechnolog_b4:e0:01 (38:97:d6:b4:e0:01)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  Source: Intel_d2:86:3d (e4:5e:37:d2:86:3d)
      Address: Intel_d2:86:3d (e4:5e:37:d2:86:3d)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
```

目的地的 MAC 地址: 38:97:d6:b4:e0:01

源 MAC 地址:e4:5e:37:d2:86:3d

网络层 ip 包首部信息：

```
Internet Protocol Version 4, Src: 10.164.249.88, Dst: 202.117.1.13
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      0000 00.. = Differentiated Services Codepoint: Default (0)
      .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
    Total Length: 389
    Identification: 0xd23e (53822)
  010. .... = Flags: 0x2, Don't fragment
      0... .... = Reserved bit: Not set
      .1.. .... = Don't fragment: Set
      ..0. .... = More fragments: Not set
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: TCP (6)
    Header Checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.164.249.88
    Destination Address: 202.117.1.13
```

使用了 IPv4 协议,首部长 20 字节。

IP 包共长 389 字节，标记字段为 53822，生存周期为 128。

源地址为 10.164.249.88

目的地址为 202.117.1.13

传输层数据包首部信息

```
Transmission Control Protocol, Src Port: 13096, Dst Port: 80, Seq: 460, Ack: 14731, Len: 349
    Source Port: 13096
    Destination Port: 80
    [Stream index: 35]
  [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 349]
    Sequence Number: 460    (relative sequence number)
    Sequence Number (raw): 2585473147
    [Next Sequence Number: 809    (relative sequence number)]
    Acknowledgment Number: 14731    (relative ack number)
    Acknowledgment number (raw): 536011713
    0101 .... = Header Length: 20 bytes (5)
  Flags: 0x018 (PSH, ACK)
    Window: 512
    [Calculated window size: 131072]
    [Window size scaling factor: 256]
    Checksum: 0xd0f6 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  [Timestamps]
  [SEQ/ACK analysis]
    TCP payload (349 bytes)
```

源端口号：13096

目的端口号：80

流量控制的窗口大小：131072

应用层分析：

```
Hypertext Transfer Protocol
  GET /style/xjnew611.css HTTP/1.1\r\n
  Host: www.xjtu.edu.cn\r\n
  Connection: keep-alive\r\n
  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 Edg/120.0.0.0\r\n
  Accept: text/css,*/*;q=0.1\r\n
  Referer: http://www.xjtu.edu.cn/\r\n
  Accept-Encoding: gzip, deflate\r\n
  Accept-Language: zh-CN,zh;q=0.9\r\n
  \r\n
  [Full request URI: http://www.xjtu.edu.cn/style/xjnew611.css]
  [HTTP request 2/10]
  [Prev request in frame: 201]
  [Response in frame: 254]
  [Next request in frame: 267]
```

即具体的 http 报文，

给出了请求主机的地址：202.117.1.13

给出了一些自身信息：这是一个在 Windows 10 上运行的 64 位系统的浏览器，基于 WebKit 引擎，其浏览器核心为 Chrome（版本号 120.0.0.0）和 Microsoft Edge（版本号也是 120.0.0.0）。客户端支持两种压缩算法：gzip 和 deflate。如果可能的话，首选返回中文内容。如果中文不可用，可以考虑返回英语内容，依次考虑英国英语和美国英语。以及 cookie。

HTTP 应答报文

```
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
  Date: Sat, 06 Jan 2024 10:28:54 GMT\r\n
  Server: *********\r\n
  X-Frame-Options: SAMEORIGIN\r\n
  X-XSS-Protection: 1; mode=block\r\n
  X-Content-Type-Options: nosniff\r\n
  Referer-Policy: no-referer-when-downgrade\r\n
  X-Download-Options: noopen\r\n
  X-Permitted-Cross-Domain-Policies: master-only\r\n
   [truncated]Content-Security-Policy: default-src 'self' data: blob: *.conac.cn  *.xjtu.edu.cn *.gov.cn *.
  Last-Modified: Wed, 19 Dec 2012 05:54:24 GMT\r\n
  Accept-Ranges: bytes\r\n
  Content-Length: 20\r\n
  Cache-Control: max-age=3600\r\n
  Expires: Sat, 06 Jan 2024 11:28:54 GMT\r\n
  ETag: "14-4d12e3f16c400-gzip"\r\n
  Keep-Alive: timeout=5, max=100\r\n
  Connection: Keep-Alive\r\n
  Content-Type: text/css\r\n
  Content-Language: zh-CN\r\n
  \r\n
  [HTTP response 1/9]
  [Time since request: 0.013144000 seconds]
  [Request in frame: 239]
  [Next request in frame: 266]
  [Next response in frame: 280]
  [Request URI: http://www.xjtu.edu.cn/_sitegray/_sitegray_d.css]
  File Data: 20 bytes
```

第一行 200 ok 表示准备好了，后面是一些设定，与请求报文相呼应。

http 协议工作过程：

1. 建立连接：

当用户在浏览器中输入 URL 或点击链接时，浏览器会尝试与目标服务器建立 TCP 连接。这通常涉及到 TCP 的三次握手过程。

2. 发起请求：

浏览器通过建立的连接向服务器发送 HTTP 请求。请求中包含了要访问的资源的信息，如请求方法（GET、POST 等）、目标 URL、请求头部（包含浏览器信息、Cookie 等）以及请求体（对于 POST 请求）。

3. 服务器处理请求：

服务器接收到 HTTP 请求后，根据请求的信息执行相应的操作。这可能涉及查询数据库、处理业务逻辑等。

4. 服务器返回响应：

服务器生成 HTTP 响应，包含状态码、响应头部（包括服务器信息、内容类型等）以及响应体（实际的数据，如 HTML 文档、图片等）。

5. 传输数据：

服务器将生成的 HTTP 响应通过 TCP 连接传输回客户端。这涉及将响应分割为数据包，并通过网络传输。

6. 浏览器接收响应：

浏览器接收到响应后，根据响应头中的信息判断响应的类型（文本、图像等），然后将其显示给用户。

7. 渲染页面：

如果响应是 HTML 文档，浏览器会解析 HTML、CSS 和 JavaScript，并将页面呈现给用户。这可能会触发对其他资源（如图像、样式表、脚本等）的额外 HTTP 请求。

8. 保持连接（可选）：

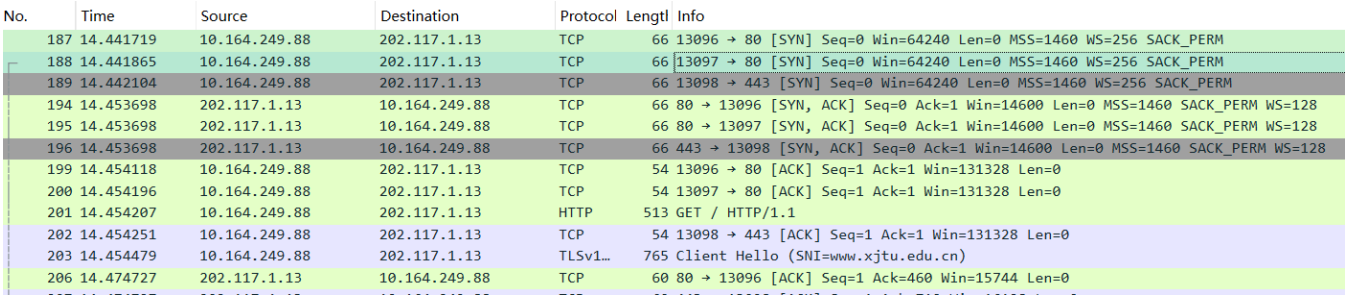
HTTP/1.1 引入了持久连接（Keep-Alive）机制，允许在单个连接上发送多个 HTTP 请求和响应，以减少连接的建立和关闭开销。

9. 连接关闭：

一旦浏览器获取到所需的资源，或者服务器决定关闭连接，TCP 连接将被关闭。

TCP 分析：

TCP 建立连接



No.	Time	Source	Destination	Protocol	Lengtl	Info
187	14.441719	10.164.249.88	202.117.1.13	TCP	66	13096 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
188	14.441865	10.164.249.88	202.117.1.13	TCP	66	13097 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
189	14.442104	10.164.249.88	202.117.1.13	TCP	66	13098 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
194	14.453698	202.117.1.13	10.164.249.88	TCP	66	80 → 13096 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM WS=128
195	14.453698	202.117.1.13	10.164.249.88	TCP	66	80 → 13097 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM WS=128
196	14.453698	202.117.1.13	10.164.249.88	TCP	66	443 → 13098 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM WS=128
199	14.454118	10.164.249.88	202.117.1.13	TCP	54	13096 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
200	14.454196	10.164.249.88	202.117.1.13	TCP	54	13097 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
201	14.454207	10.164.249.88	202.117.1.13	HTTP	513	GET / HTTP/1.1
202	14.454251	10.164.249.88	202.117.1.13	TCP	54	13098 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
203	14.454479	10.164.249.88	202.117.1.13	TLSv1…	765	Client Hello (SNI=www.xjtu.edu.cn)
206	14.474727	202.117.1.13	10.164.249.88	TCP	60	80 → 13096 [ACK] Seq=1 Ack=460 Win=15744 Len=0

187 号报文为浏览器向服务器发起 TCP 连接的第一条报文，具体内容未申请连接信号 SYN，初始序号 seq=0,发送窗口大小为 64240，MSS=1460 等初始信息。由于超过一定时间之后浏览器仍然没有收到服务器的应答报文，为了避免报文丢失或者损坏，浏览器又向服务器发送了同样内容的 188 号报文，类似的浏览器还没有在规定时间内收到应答报文，所以又发送了 189 号报文；

服务器收到浏览器的请求报文后发送了 194 号报文，具体内容为确认信号 ACK=1，表示服务器已经接收到 1 号报文之前的信息，现在请求对方发送给服务器 1 号报文的信息,申请连接信号 SYN,初始序号 seq=0，接收窗口大小为 14600，MSS 为 1460 等初始信息，因为同样超时未收到报文，所以服务器又发送了同样内容的 195，196 号报文。

199 号报文为浏览器收到服务器的确认报文之后做出的确认信息，具体内容为确认信号 ACK=1，表示浏览器已经接收到 1 号报文之前的信息，现在请求对方发送给浏览器 1 号报文的信息,同时告诉对方自己当前的窗口大小。

TCP 释放连接

ip.addr == 202.117.1.13

分组列表 | 宽窄 | 区分大小写 | 字符串 | FIN

No.	Time	Source	Destination	Protocol	Lengtl	Info
4258	15.667080	10.164.249.88	202.117.1.13	TCP	54	13100 → 80 [ACK] Seq=5151 Ack=471773 Win=130048 Len=0
4344	20.608944	202.117.1.13	10.164.249.88	TCP	60	80 → 13096 [FIN, ACK] Seq=793983 Ack=4015 Win=25344 Len=0
4345	20.609039	10.164.249.88	202.117.1.13	TCP	54	13096 → 80 [ACK] Seq=4015 Ack=793984 Win=131328 Len=0
4348	20.609888	202.117.1.13	10.164.249.88	TCP	60	80 → 13097 [FIN, ACK] Seq=734670 Ack=3589 Win=24320 Len=0
4349	20.609949	10.164.249.88	202.117.1.13	TCP	54	13097 → 80 [ACK] Seq=3589 Ack=734671 Win=130048 Len=0
4350	20.613127	202.117.1.13	10.164.249.88	TCP	60	80 → 13102 [FIN, ACK] Seq=505486 Ack=4763 Win=27520 Len=0
4351	20.613257	10.164.249.88	202.117.1.13	TCP	54	13102 → 80 [ACK] Seq=4763 Ack=505487 Win=129792 Len=0
4352	20.622810	202.117.1.13	10.164.249.88	TCP	60	80 → 13103 [FIN, ACK] Seq=573144 Ack=6334 Win=31872 Len=0
4353	20.622903	10.164.249.88	202.117.1.13	TCP	54	13103 → 80 [ACK] Seq=6334 Ack=573145 Win=131328 Len=0
4356	20.623827	202.117.1.13	10.164.249.88	TCP	60	80 → 13100 [FIN, ACK] Seq=471773 Ack=5151 Win=28544 Len=0
4357	20.623898	10.164.249.88	202.117.1.13	TCP	54	13100 → 80 [ACK] Seq=5151 Ack=471774 Win=130048 Len=0
4370	20.664218	202.117.1.13	10.164.249.88	TCP	60	80 → 13101 [FIN, ACK] Seq=637092 Ack=6748 Win=32896 Len=0
4371	20.664323	10.164.249.88	202.117.1.13	TCP	54	13101 → 80 [ACK] Seq=6748 Ack=637093 Win=131328 Len=0
4527	26.752748	10.164.249.88	202.117.1.13	TCP	54	13096 → 80 [FIN, ACK] Seq=4015 Ack=793984 Win=131328 Len=0
4528	26.752824	10.164.249.88	202.117.1.13	TCP	54	13102 → 80 [FIN, ACK] Seq=4763 Ack=505487 Win=129792 Len=0
4529	26.752867	10.164.249.88	202.117.1.13	TCP	54	13097 → 80 [FIN, ACK] Seq=3589 Ack=734671 Win=130048 Len=0
4530	26.752909	10.164.249.88	202.117.1.13	TCP	54	13100 → 80 [FIN, ACK] Seq=5151 Ack=471774 Win=130048 Len=0
4531	26.752949	10.164.249.88	202.117.1.13	TCP	54	13103 → 80 [FIN, ACK] Seq=6334 Ack=573145 Win=131328 Len=0
4532	26.752988	10.164.249.88	202.117.1.13	TCP	54	13101 → 80 [FIN, ACK] Seq=6748 Ack=637093 Win=131328 Len=0
4534	26.759172	202.117.1.13	10.164.249.88	TCP	60	80 → 13102 [ACK] Seq=505487 Ack=4764 Win=27520 Len=0
4535	26.762399	202.117.1.13	10.164.249.88	TCP	60	80 → 13096 [ACK] Seq=793984 Ack=4016 Win=25344 Len=0
4536	26.762399	202.117.1.13	10.164.249.88	TCP	60	80 → 13097 [ACK] Seq=734671 Ack=3590 Win=24320 Len=0
4537	26.766401	202.117.1.13	10.164.249.88	TCP	60	80 → 13100 [ACK] Seq=471774 Ack=5152 Win=28544 Len=0
4538	26.766401	202.117.1.13	10.164.249.88	TCP	60	80 → 13103 [ACK] Seq=573145 Ack=6335 Win=31872 Len=0
4539	26.766401	202.117.1.13	10.164.249.88	TCP	60	80 → 13101 [ACK] Seq=637093 Ack=6749 Win=32896 Len=0
7941	31.563816	10.164.249.88	202.117.1.13	TCP	66	13165 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
7942	31.564011	10.164.249.88	202.117.1.13	TCP	66	13166 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
7943	31.583458	202.117.1.13	10.164.249.88	TCP	66	80 → 13165 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM WS=128
7944	31.583458	202.117.1.13	10.164.249.88	TCP	66	80 → 13166 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM WS=128
7946	31.583796	10.164.249.88	202.117.1.13	TCP	54	13165 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
7948	31.583895	10.164.249.88	202.117.1.13	TCP	54	13166 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
8002	34.521458	202.117.1.13	10.164.249.88	TLSv1…	85	Encrypted Alert

主机发起请求 FIN,ACK（第一次挥手），

本机接收到并返回两个包，一个为 ACK（第二次挥手），

另一个为 FIN,ACK（第三次挥手），

最后主机发一个 ACK 返回给本机（第四次挥手）然后结束。

TCP 过程

1. 建立连接：

客户端向服务器发送 SYN（同步）报文：客户端选择一个随机的初始序列号（ISN）并发送一个带有 SYN 标志的 TCP 报文给服务器。

服务器收到 SYN 报文并回应:服务器接收到客户端的 SYN 报文后,选择自己的随机 ISN,并发送一个带有 SYN 和 ACK（确认）标志的报文给客户端。

客户端发送 ACK 报文：客户端收到服务器的响应后，发送一个带有 ACK 标志的报文给服务器，完成连接的建立。

这个过程称为三次握手，建立了双方的连接，确保双方都能够接收和发送数据。

2. 数据传输：

客户端和服务器通过已建立的连接传输数据。数据被分割成 TCP 段，并分别被封装成 TCP

报文，然后通过网络传输。

3. 关闭连接：

客户端发送 FIN 报文：当一方（通常是客户端）希望关闭连接时，它发送一个带有 FIN（结
束）标志的 TCP 报文。

服务器收到 FIN 并回应：服务器收到 FIN 后，发送一个带有 ACK 标志的报文给客户端，表示确认收到关闭请求。

服务器发送 FIN 报文：服务器也 7 希望关闭连接，因此发送一个带有 FIN 标志的报文给
客户端。

客户端收到 FIN 并回应：客户端收到服务器的 FIN 后，发送一个带有 ACK 标志的报文，确认收到服务器的关闭请求。

这个过程称为四次挥手，确保双方都完成了数据的传输并准备好关闭连接。

TCP 通过这样的握手和挥手过程，提供了可靠的连接，保证了数据的完整性和有序性。这种可靠性建立在序列号、确认和重传机制等基础上，确保数据在传输过程中不会丢失、损坏或乱序