

# 分析协议HTTP分组嗅探器（wireshark）分析HTTP、TCP

## 1、实验目的

利用wireshark软件分析HTTP及其下层协议（TCP协议） 了解网络中数据封装的概念 掌握HTTP及TCP协议的工作过程

## 2、实验内容

启动wireshark软件，进行报文截获 在浏览器访问www.xjtu.edu.cn页面。（打开网页，浏览并关闭页面） 停止ethereal 的报文截获，将截获命名为“http—学号” 分析截获报文。

## 3、实验要求

从截获的报文中选择HTTP请求报文（即get报文）和HTTP应答报文，并分析各字段的值；

综合分析截获的报文，概括HTTP协议的工作过程；

从截获报文中选择TCP建立连接和释放连接的报文，分析各个字段的值并概括HTTP协议的工作过程；

## 4、实验过程

### 4.1分析HTTP报文各字段值

使用ip.src==202.117.1.13||ip.dst==202.117.1.13&&http包筛选

No.	Time	Source	Destination	Protocol	Length	Info
352	19.552848	202.117.1.13	10.173.217.246	TCP	66	80 → 59259 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM WS=128
354	19.553440	10.173.217.246	202.117.1.13	HTTP	549	GET / HTTP/1.1
355	19.557054	202.117.1.13	10.173.217.246	TCP	60	80 → 59259 [ACK] Seq=1 Ack=496 Win=15744 Len=0
356	19.558280	202.117.1.13	10.173.217.246	TCP	1514	80 → 59259 [ACK] Seq=1 Ack=496 Win=15744 Len=1460 [TCP segment of a reassembled PDU]
357	19.559044	202.117.1.13	10.173.217.246	TCP	1514	80 → 59259 [ACK] Seq=1461 Ack=496 Win=15744 Len=1460 [TCP segment of a reassembled PDU]
358	19.559044	202.117.1.13	10.173.217.246	TCP	1514	80 → 59259 [ACK] Seq=2921 Ack=496 Win=15744 Len=1460 [TCP segment of a reassembled PDU]
359	19.559044	202.117.1.13	10.173.217.246	TCP	1514	80 → 59259 [ACK] Seq=4381 Ack=496 Win=15744 Len=1460 [TCP segment of a reassembled PDU]
360	19.559044	202.117.1.13	10.173.217.246	TCP	1514	80 → 59259 [ACK] Seq=5841 Ack=496 Win=15744 Len=1460 [TCP segment of a reassembled PDU]
361	19.559044	202.117.1.13	10.173.217.246	TCP	1514	80 → 59259 [ACK] Seq=7301 Ack=496 Win=15744 Len=1460 [TCP segment of a reassembled PDU]
362	19.559044	202.117.1.13	10.173.217.246	TCP	1514	80 → 59259 [ACK] Seq=8761 Ack=496 Win=15744 Len=1460 [TCP segment of a reassembled PDU]
363	19.559044	202.117.1.13	10.173.217.246	TCP	1514	80 → 59259 [ACK] Seq=10221 Ack=496 Win=15744 Len=1460 [TCP segment of a reassembled PDU]
364	19.559044	202.117.1.13	10.173.217.246	TCP	1514	80 → 59259 [ACK] Seq=11681 Ack=496 Win=15744 Len=1460 [TCP segment of a reassembled PDU]
365	19.559044	202.117.1.13	10.173.217.246	HTTP	1060	HTTP/1.1 200 OK (text/html)
385	19.663746	10.173.217.246	202.117.1.13	HTTP	465	GET /style/xjnew611.css HTTP/1.1
399	19.667469	202.117.1.13	10.173.217.246	TCP	1514	80 → 59259 [ACK] Seq=14147 Ack=907 Win=16768 Len=1460 [TCP segment of a reassembled PDU]
400	19.667469	202.117.1.13	10.173.217.246	TCP	1514	80 → 59259 [ACK] Seq=15607 Ack=907 Win=16768 Len=1460 [TCP segment of a reassembled PDU]
401	19.667469	202.117.1.13	10.173.217.246	TCP	1514	80 → 59259 [ACK] Seq=17067 Ack=907 Win=16768 Len=1460 [TCP segment of a reassembled PDU]
402	19.667469	202.117.1.13	10.173.217.246	HTTP	600	HTTP/1.1 200 OK (text/css)
430	19.670500	10.173.217.246	202.117.1.13	HTTP	461	GET /system/resource/js/counter.js HTTP/1.1

### http请求报文

### 帧信息/数据链路层

```

▼ Frame 354: 549 bytes on wire (4392 bits), 549 bytes captured (4392 bits) on interface
  Section number: 1
  > Interface id: 0 (\Device\NPF_{54C9491B-16F9-4261-8CA5-933586BA6E5C})
  Encapsulation type: Ethernet (1)
  Arrival Time: Nov 11, 2023 17:56:20.290603000 中国标准时间
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1699696580.290603000 seconds
  [Time delta from previous captured frame: 0.000477000 seconds]
  [Time delta from previous displayed frame: 0.000592000 seconds]
  [Time since reference or first frame: 19.553440000 seconds]
  Frame Number: 354
  Frame Length: 549 bytes (4392 bits)
  Capture Length: 549 bytes (4392 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]

```

- 帧号: 354
- 帧长度: 549字节
- 到达时间: 2023年11月11日 17:56:20.290603 (中国标准时间)

## 以太网 II 部分/数据链路层

```

▼ Ethernet II, Src: HuaweiTe_f5:dc:91 (9c:71:3a:f5:dc:91), Dst: IntelCor_79:ab:80 (ec:63:d7:79:ab:80)
  > Destination: IntelCor_79:ab:80 (ec:63:d7:79:ab:80)
  > Source: HuaweiTe_f5:dc:91 (9c:71:3a:f5:dc:91)
  Type: IPv4 (0x0800)

```

- 源MAC地址: IntelCor\_79:ab:80 (ec:63:d7:79:ab:80)
- 目的MAC地址: HuaweiTe\_f5:dc:91 (9c:71:3a:f5:dc:91)
- 类型: IPv4 (0x0800)

## IPv4 部分/网络层

```

▼ Internet Protocol Version 4, Src: 10.173.217.246, Dst: 202.117.1.13
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 535
  Identification: 0x82bf (33471)
  > 010. .... = Flags: 0x2, Don't fragment
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 128
  Protocol: TCP (6)
  Header Checksum: 0xc5fb [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 10.173.217.246
  Destination Address: 202.117.1.13
  > Transmission Control Protocol, Src Port: 59259, Dst Port: 80, Seq: 1, Ack: 1, Len: 495

```

- 版本: 4
- 头部长度的: 20字节
- 总长度: 535字节

- 协议: TCP (6)
- 源IP地址: 10.173.217.246
- 目的IP地址: 202.117.1.13

## TCP 部分/传输层

```

v Transmission Control Protocol, Src Port: 59259, Dst Port: 80, Seq: 1, Ack: 1, Len: 495
  Source Port: 59259
  Destination Port: 80
  [Stream index: 34]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 495]
  Sequence Number: 1 (relative sequence number)
  Sequence Number (raw): 4206983301
  [Next Sequence Number: 496 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 2393276913
  0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
  Window: 513
  [Calculated window size: 131328]
  [Window size scaling factor: 256]
  Checksum: 0x28e5 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  > [Timestamps]
  > [SEQ/ACK analysis]
  TCP payload (495 bytes)

```

- 1、源端口 (Source Port): 59259 表示发送端的端口号。在这个情境下，数据包是从端口59259发送的。
- 2、目标端口 (Destination Port): 80 表示接收端的端口号。在这个情境下，数据包是发送到端口80的目标。
- 3、TCP Segment Len:495:表示有效载荷的长度，这里代表这个TCP段携带495字节的数据。
- 4、Sequence Number:1 :序列号 (Sequence Number): 1 (相对序列号) 表示此数据包的序列号。相对序列号表示相对于连接建立时的初始序列号
- 5、原始序列号: 4206983301 是未经处理的序列号，这个值用于在网络中传输
- 6、下一个序列号: 496 (相对序列号) 表示期望接收的下一个数据包的序列号，这里表示客户端希望下一个收到序列号为496的数据包。
- 7、确认号 (Acknowledgment Number): 1 (相对确认号) 表示对方期望接收的下一个序列号。在这个情境下，对方期望接收序列号为1的数据包。这里的对方指的是服务器。
- 8、原始确认号: 2393276913 是未经处理的确认号，用于在网络中传输
- 9、标志 (Flags): PSH, ACK (推送数据 + 确认) 指示TCP报文的状态。PSH 标志时，表示发起端希望接收端尽快将接收到的数据交付给上层应用，而不是等待缓冲区满了再交付。ACK 标志表示确认号字段是有效的。在这个情境下，标志表示这是一个包含数据并需要确认的数据包，同时应该尽快将数据推送给应用层。
- 10、窗口大小 (Window Size): 513 表示发送端的接收窗口大小，即发送端还能接收多少字节的数据。此处发送端为服务器。
- 11、Calculated window size: 131328: 经过窗口大小缩放因子的计算后的实际窗口大小: 131328。这里使用了窗口放缩技术，TCP窗口大小放缩是为了应对网络中可能存在的不同带宽和延迟情况，以提高数据传输的效率。窗口大小放缩的主要目的是优化TCP流控制，使其适应不同网络条件，提高数据传输的吞吐量。
- 12、校验和 (Checksum): 0x28e5 [未验证]，意味着接收方尚未对校验和进行验证。可能造成未校验的原因很多，可能是信息被截断或不完整、校验和计算代价较高、网络设备问题等原因。
- 13、紧急指针 (Urgent Pointer): 0，正常的数据传输中，紧急指针值为0，表示没有紧急数据。

## HTTP 部分

```

▼ Hypertext Transfer Protocol
  > GET / HTTP/1.1\r\n
    Host: www.xjtu.edu.cn\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Safari/537.36 Edg/119.0.0.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6\r\n
  > Cookie: JSESSIONID=1DD35576BDE18C8C6E94E02B348B81A8\r\n
    Upgrade-Insecure-Requests: 1\r\n
    \r\n
    [Full request URI: http://www.xjtu.edu.cn/]
    [HTTP request 1/14]
    [Response in frame: 365]
    [Next request in frame: 385]

```

- GET: 表示使用的是GET请求方法。  
/: 表示请求的URI (Uniform Resource Identifier) , 在这里是根路径, 表示请求站点的首页。 HTTP/1.1: 表示使用的HTTP协议版本, 这里是HTTP/1.1。
- Request Method: GET  
HTTP请求方法, 这里是GET。表示客户端请求获取由URI标识的资源。
- Request URI: /  
请求的URI, 即/表示根路径。这是客户端希望访问的资源的标识。
- Request Version: HTTP/1.1  
表示使用的HTTP协议版本, 这里是HTTP/1.1。指示客户端使用的HTTP协议版本是1.1。
- User-Agent: Mozilla/5.0  
用户代理, 即客户端使用的接受包的设置, 此处使用Edge浏览器打开。
- Accept  
客户端能够接受的数据类型, 这里包括html,xhtml+xml,webp,apng等多种格式。
- Accept-Encoding  
指定编码方式为gzip和deflate格式。
- Accept-Language: zh-CN,zh 字段表明哪些语言客户端是能够理解, 并且其区域的变体是优选的, 此处请求内容为中文。
- HTTP request 1/14  
表示这是发送的第一个HTTP请求报文, 一共有14个HTTP请求报文。

## HTTP应答报文

### 1、帧信息/数据链路层:

```

▼ Frame 365: 1060 bytes on wire (8480 bits), 1060 bytes captured (8480 bits) on interface \Device\NPF_{54C9491B-16F9-4261-8CA5-933586BA6E5C}, id 0
  Section number: 1
  > Interface id: 0 (\Device\NPF_{54C9491B-16F9-4261-8CA5-933586BA6E5C})
    Encapsulation type: Ethernet (1)
    Arrival Time: Nov 11, 2023 17:56:20.296207000 中国标准时间
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1699696580.296207000 seconds
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 19.559044000 seconds]
    Frame Number: 365
    Frame Length: 1060 bytes (8480 bits)
    Capture Length: 1060 bytes (8480 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp:http:data-text-lines]
    [Coloring Rule Name: HTTP]
    [Coloring Rule String: http || tcp.port == 80 || http2]
  > Ethernet II, Src: HuaweiTe_f5:dc:91 (9c:71:3a:f5:dc:91), Dst: IntelCor_79:ab:80 (ec:63:d7:79:ab:80)

```

- 长度：1060字节（8480比特），从接口\Device\NPF\_{54C9491B-16F9-4261-8CA5-933586BA6E5C}捕获。
- 源MAC地址：9c:71:3a:f5:dc:91（华为设备）
- 目的MAC地址：ec:63:d7:79:ab:80（Intel设备）

## 2、网络层信息：

```
▼ Ethernet II, Src: HuaweiTe_f5:dc:91 (9c:71:3a:f5:dc:91), Dst: IntelCor_79:ab:80 (ec:63:d7:79:ab:80)
  > Destination: IntelCor_79:ab:80 (ec:63:d7:79:ab:80)
  > Source: HuaweiTe_f5:dc:91 (9c:71:3a:f5:dc:91)
  Type: IPv4 (0x0800)
```

- 源IP地址：202.117.1.13
- 目的IP地址：10.173.217.246

## 3、传输层信息：

```
▼ Internet Protocol Version 4, Src: 202.117.1.13, Dst: 10.173.217.246
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 1046
  Identification: 0x61c8 (25032)
  > 010. .... = Flags: 0x2, Don't fragment
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 61
  Protocol: TCP (6)
  Header Checksum: 0x27f4 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 202.117.1.13
  Destination Address: 10.173.217.246
```

- 源端口：80
- 目的端口：59259
- 序列号：13141，确认号：496
- 标志位：PSH, ACK
- 窗口大小：123
- 校验和：0xfbb3

## 4、TCP负载：

```
▼ [10 Reassembled TCP Segments (14146 bytes): #356(1460), #357(1460), #358(1460), #359(1460), #360(1460), #361(1460), #362(1460), #363(1460), #364(1460), #365(1006)]
  [Frame: 356, payload: 0-1459 (1460 bytes)]
  [Frame: 357, payload: 1460-2919 (1460 bytes)]
  [Frame: 358, payload: 2920-4379 (1460 bytes)]
  [Frame: 359, payload: 4380-5839 (1460 bytes)]
  [Frame: 360, payload: 5840-7299 (1460 bytes)]
  [Frame: 361, payload: 7300-8759 (1460 bytes)]
  [Frame: 362, payload: 8760-10219 (1460 bytes)]
  [Frame: 363, payload: 10220-11679 (1460 bytes)]
  [Frame: 364, payload: 11680-13139 (1460 bytes)]
  [Frame: 365, payload: 13140-14145 (1006 bytes)]
  [Segment count: 10]
  [Reassembled TCP length: 14146]
  [Reassembled TCP Data: 485454502f312e3120323030204f4b0d0a446174653a205361742c203131204e6f762032...]
```

- 由10个重新组装的TCP段组成，总长度为14146字节。
- HTTP协议：服务器返回200 OK的响应，包含了一系列的头信息和内容。
- 内容编码：gzip，内容长度13288字节，解压后变为59975字节。

## 5、HTTP/1.1协议的响应报文

```
▼ Hypertext Transfer Protocol
> HTTP/1.1 200 OK\r\n
Date: Sat, 11 Nov 2023 09:56:19 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 *.jiathis.com *.baidu.com *.bshare.cn *.eol.cn *.qq.com *.kaipuyun.cn *.bdimg.com *
Last-Modified: Fri, 10 Nov 2023 14:36:13 GMT\r\n
Accept-Ranges: bytes\r\n
Cache-Control: max-age=600\r\n
Expires: Sat, 11 Nov 2023 10:06:19 GMT\r\n
Vary: Accept-Encoding\r\n
Content-Encoding: gzip\r\n
ETag: "ea47-609cd3df9f140-gzip"\r\n
> Content-Length: 13288\r\n
Content-Type: text/html\r\n
Content-Language: zh-CN\r\n
\r\n
[HTTP response 1/14]
[Time since request: 0.005604000 seconds]
[Request in frame: 354]
[Next request in frame: 385]
[Next response in frame: 402]
[Request URI: http://www.xjtu.edu.cn/]
Content-encoded entity body (gzip): 13288 bytes -> 59975 bytes
File Data: 59975 bytes
```

- HTTP/1.1 200 OK - 表示HTTP协议版本为1.1，状态码为200，表示请求成功，/r/n表示请求的结束。
- 日期：Sat, 11 Nov 2023 09:56:19 GMT
- 服务器：\*\*\*\*\* 服务器信息，这里的内容已被截断，包含服务器的标识信息。
- X-Frame-Options: SAMEORIGIN：此字段指示浏览器是否应该加载一个iframe中的页面。值SAMEORIGIN表示页面只能被本站页面嵌入到iframe或者frame中。
- X-XSS-Protection: 1; mode=block：此字段是一种老旧的浏览器功能，用于阻止跨站脚本攻击（XSS）。当设置为1; mode=block时，如果浏览器检测到潜在的反射型XSS攻击，将不会渲染页面，而是阻止页面加载123。
- X-Content-Type-Options: nosniff：此字段用于阻止浏览器对资源进行MIME类型嗅探。当设置为nosniff时，浏览器将严格遵循从服务器发送的Content-Type头的MIME类型，而不会尝试嗅探并更改资源的MIME类型45。
- Referer-Policy: no-referer-when-downgrade：此字段用于控制Referer头的内容。当设置为no-referer-when-downgrade时，只有在协议安全级别保持不变或提高（例如，从HTTP到HTTP，或从HTTP到HTTPS）的情况下，才会在Referer头中发送来源、路径和查询字符串。如果目标的安全级别降低（例如，从HTTPS到HTTP），则不会发送Referer头678910。
- X-Download-Options: noopen：此字段是Internet Explorer特有的安全功能，用于防止基于"Open"命令的攻击。当设置为noopen时，用户无法直接打开下载的文件，而必须先保存文件，然后再打开。这样可以防止恶意代码在用户的网站上下文中运行11121314。
- Last-Modified: Fri, 10 Nov 2023 14:36:13 GMT：此字段指明服务器对象的最后修订时间，即2023年11月10日14:36:13 GMT。
- Accept-Ranges: bytes：此字段用于告知客户端服务器是否能够处理范围请求，以指定获取服务器端某个部分的资源。当服务器可以处理范围请求时，指定为bytes。
- Cache-Control: max-age=600：此字段用以实现缓存机制，其值max-age=600表示如果缓存资源的缓存时间值小于600秒则使用缓存。
- Expires: Sat, 11 Nov 2023 10:06:19 GMT\r\n，表示资源过期的日期
- Vary: Accept-Encoding:这里是指要选择一种Encoding编码方法。



- Content-Encoding: gzip: 此字段表示服务器使用gzip压缩了内容。
- ETag (实体标签) 是服务器生成的资源的唯一标识符, 通常是内容的哈希或版本号。当客户端再次请求该资源时, 它会发送一个If-None-Match头, 其中包含先前接收到的ETag值。如果资源没有更改, 服务器将返回一个304 Not Modified状态, 告诉客户端可以使用其缓存的版本。如果资源已更改, 服务器将返回新的资源和新的ETag值。在这个例子中, "ea47-609cd3df9f140-gzip"就是资源的ETag值。
- Content-Length: 13288: 此字段表示响应正文的长度, 即13288字节。
- Content-Type: text/html: 此字段表示响应正文的媒体类型, 即HTML文本。
- Content-Language: zh-CN: 此字段表示响应正文的语言, 即简体中文。
- Content-encoded entity body (gzip): 13288 bytes -> 59975 bytes---以gzip压缩格式提供的HTML文档, 原始大小13288字节, 压缩后大小59975字节

## 6、文件信息:

```
Line-based text data: text/html (1023 lines)
\u0000<!DOCTYPE HTML>\r\n
<HTML><HEAD><TITLE>西安交通大学</TITLE>\r\n
\r\n
\r\n
\r\n
\r\n
<META name="360-site-verification" content="845cb73defc117caad1186ca8fac8532"><script type="text/javascript">\r\n
\r\n
if (/AppleWebKit.*Mobile/i.test(navigator.userAgent) || (/MIDP|SymbianOS|NOKIA|SAMSUNG|LG|NEC|TCL|Alcatel|BIRD|DBTEL|Dopod|PHILIPS|HAIER|LENOVO|MOT-|Nokia|SonyEricsson|SIE-|Amoi|ZTE/.test(
if(window.location.href.indexOf("?mobile")<0){\r\n
    try{\r\n
        if (/Android|Windows Phone|webOS|iPhone|iPod|BlackBerry/i.test(navigator.userAgent)){\r\n
            window.location.href="http://mob.xjtu.edu.cn/";\r\n
        }\r\n
    }catch(e){}\r\n
    }\r\n
}\r\n
\r\n
\r\n
</script>\r\n
<META content="text/html; charset=UTF-8" http-equiv="Content-Type">\r\n
[truncated]<META content="IE=edge,chrome=1" http-equiv="X-UA-Compatible"><LINK rel="stylesheet" type="text/css" href="style/xjnew611.css"><script type="text/javascript" src="js/jquery.
<META name="baidu-site-verification" content="r23jsHKdp1">\r\n
<STYLE>\r\n
.menuLink ul li: hover ul{width: 230px !important;}\r\n
</STYLE>\r\n
\r\n
\r\n
<!--Announced by Visual SiteBuilder 9-->\r\n
<link rel="stylesheet" type="text/css" href="_sitegray/_sitegray_d.css" />\r\n
<script language="javascript" src="_sitegray/_sitegray.js"></script>\r\n
<!-- CustomerNO:7765626265723230697547545352544203050000 -->\r\n
<link rel="stylesheet" type="text/css" href="index.vsb.css" />\r\n
<META Name="keywords" Content="西安交通大学" />\r\n
<script type="text/javascript" src="system/resource/js/counter.js"></script>\r\n
<script type="text/javascript">_jsq_(1001, '/index.jsp', -1, 1151962237)</script>\r\n
</HEAD>\r\n
<BODY>\r\n
<div class="feedback"></div><div class="feedback"></div><div class="feedback"></div></div>
```

- 内容经过gzip压缩, 压缩前13288字节, 压缩后59975字节。
- 文件数据包含1023行文本。
- 请求URI为http://www.xjtu.edu.cn/

## 4.2概括HTTP协议的工作

通过分析上述截获的HTTP请求和响应报文, 可以概括HTTP协议的工作过程如下:

**建立连接:** 客户端发起与服务器的连接, 这通常是通过TCP协议的三次握手建立的连接。在这里, 客户端的IP地址为10.173.217.246, 服务器的IP地址为202.117.1.13。客户端使用端口59259, 服务器使用端口80。

**发送HTTP请求:** 客户端向服务器发送HTTP请求, 请求的内容包括请求方法 (GET)、请求的资源 (http://www.xjtu.edu.cn/), 以及其他相关的头部信息。

以本实验抓获的第一个HTTP请求报文示例。报文内容包括请求方法(GET)和HTTP版本, 请求URL, 和客户端的信息。

```
Host: www.xjtu.edu.cn\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/119.0.0.0 Safari/537.36 Edg/119.0.0.0\r\n
```

服务器处理请求：服务器接收到HTTP请求后，根据请求的内容进行相应的处理。在这个例子中，服务器返回了一个HTML文档作为响应。

发送HTTP响应：服务器将处理结果封装成HTTP响应，包括状态码、响应头以及响应体。这个响应报文在帧365中可以看到。以本实验抓获的第一个HTTP请求报文的响应的HTML报文为例,包括相应行(HTTP/1.1 200 OK\r\n)，响应头(余下内容)，响应体(具体html的内容，过长此处不显示)。

```
HTTP/1.1 200 OK\r\n
Date: Sat, 11 Nov 2023 09:56:19 GMT\r\n
Server: *****\r\n
Content-Encoding: gzip\r\n
Content-Length: 13288\r\n
Content-Type: text/html\r\n
Content-Language: zh-CN\r\n
```

关闭连接：一旦响应被传输给客户端，连接可以根据需要立即关闭，或者保持打开状态以进行后续的请求和响应。给出的信息中，未包含 "Connection" 头部字段。在HTTP/1.1中，如果未指定 "Connection" 头部字段，通常默认为 "Connection: keep-alive"，这表示服务器愿意保持TCP连接打开以进行后续请求和响应。与前面的一共有14个报文还要发送的事实相符（例如其他JavaScript，CSS包），后续也有捕捉到，如下图。

363	19.559044	202.117.1.13	10.173.217.246	TCP	1514 80 → 59259 [ACK] Seq=10221 Ack=496 Win=15744 Len=1460 [TCP segment of a reassembled PDU]
364	19.559044	202.117.1.13	10.173.217.246	TCP	1514 80 → 59259 [ACK] Seq=11681 Ack=496 Win=15744 Len=1460 [TCP segment of a reassembled PDU]
365	19.559044	202.117.1.13	10.173.217.246	HTTP	1060 HTTP/1.1 200 OK (text/html)
385	19.663746	10.173.217.246	202.117.1.13	HTTP	465 GET /style/xjnew611.css HTTP/1.1
399	19.667469	202.117.1.13	10.173.217.246	TCP	1514 80 → 59259 [ACK] Seq=14147 Ack=907 Win=16768 Len=1460 [TCP segment of a reassembled PDU]
400	19.667469	202.117.1.13	10.173.217.246	TCP	1514 80 → 59259 [ACK] Seq=15607 Ack=907 Win=16768 Len=1460 [TCP segment of a reassembled PDU]
401	19.667469	202.117.1.13	10.173.217.246	TCP	1514 80 → 59259 [ACK] Seq=17067 Ack=907 Win=16768 Len=1460 [TCP segment of a reassembled PDU]
402	19.667469	202.117.1.13	10.173.217.246	HTTP	600 HTTP/1.1 200 OK (text/css)
430	19.670500	10.173.217.246	202.117.1.13	HTTP	461 GET /system/resource/js/counter.js HTTP/1.1
433	19.672766	202.117.1.13	10.173.217.246	TCP	1514 80 → 59259 [ACK] Seq=19073 Ack=1314 Win=17920 Len=1460 [TCP segment of a reassembled PDU]
434	19.672766	202.117.1.13	10.173.217.246	HTTP	209 HTTP/1.1 200 OK (application/javascript)

显示内容：客户端接收到服务器的响应后，根据响应的内容进行相应的处理，通常是渲染HTML页面，显示在用户的浏览器中。

## 4.3分析TCP协议

### 4.3.1建立TCP连接的3次握手：

把过滤条件改为ip.src==202.117.1.13||ip.dst==202.117.1.13&&tcp，抓取到以下符合条件的包。

No.	Time	Source	Destination	Protocol	Length	Info
345	19.550038	10.173.217.246	202.117.1.13	TCP	66	59259 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
352	19.552848	202.117.1.13	10.173.217.246	TCP	66	80 → 59259 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM WS=128
353	19.552963	10.173.217.246	202.117.1.13	TCP	54	59259 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
354	19.553440	10.173.217.246	202.117.1.13	HTTP	549	GET / HTTP/1.1
355	19.557054	202.117.1.13	10.173.217.246	TCP	60	80 → 59259 [ACK] Seq=1 Ack=496 Win=15744 Len=0
356	19.558280	202.117.1.13	10.173.217.246	TCP	1514	80 → 59259 [ACK] Seq=1 Ack=496 Win=15744 Len=1460 [TCP segment of a reassembled PDU]
357	19.559044	202.117.1.13	10.173.217.246	TCP	1514	80 → 59259 [ACK] Seq=1461 Ack=496 Win=15744 Len=1460 [TCP segment of a reassembled PDU]
358	19.559044	202.117.1.13	10.173.217.246	TCP	1514	80 → 59259 [ACK] Seq=2921 Ack=496 Win=15744 Len=1460 [TCP segment of a reassembled PDU]
359	19.559044	202.117.1.13	10.173.217.246	TCP	1514	80 → 59259 [ACK] Seq=4381 Ack=496 Win=15744 Len=1460 [TCP segment of a reassembled PDU]
360	19.559044	202.117.1.13	10.173.217.246	TCP	1514	80 → 59259 [ACK] Seq=5841 Ack=496 Win=15744 Len=1460 [TCP segment of a reassembled PDU]
361	19.559044	202.117.1.13	10.173.217.246	TCP	1514	80 → 59259 [ACK] Seq=7301 Ack=496 Win=15744 Len=1460 [TCP segment of a reassembled PDU]
362	19.559044	202.117.1.13	10.173.217.246	TCP	1514	80 → 59259 [ACK] Seq=8761 Ack=496 Win=15744 Len=1460 [TCP segment of a reassembled PDU]
363	19.559044	202.117.1.13	10.173.217.246	TCP	1514	80 → 59259 [ACK] Seq=10221 Ack=496 Win=15744 Len=1460 [TCP segment of a reassembled PDU]
364	19.559044	202.117.1.13	10.173.217.246	TCP	1514	80 → 59259 [ACK] Seq=11681 Ack=496 Win=15744 Len=1460 [TCP segment of a reassembled PDU]
365	19.559044	202.117.1.13	10.173.217.246	HTTP	1060	HTTP/1.1 200 OK (text/html)
366	19.559321	10.173.217.246	202.117.1.13	TCP	54	59259 → 80 [ACK] Seq=496 Ack=14147 Win=131328 Len=0
385	19.663746	10.173.217.246	202.117.1.13	HTTP	465	GET /style/xjnew611.css HTTP/1.1
399	19.667469	202.117.1.13	10.173.217.246	TCP	1514	80 → 59259 [ACK] Seq=14147 Ack=907 Win=16768 Len=1460 [TCP segment of a reassembled PDU]
400	19.667469	202.117.1.13	10.173.217.246	TCP	1514	80 → 59259 [ACK] Seq=15607 Ack=907 Win=16768 Len=1460 [TCP segment of a reassembled PDU]



红框框起的三个包即为建立连接的“三次握手”。

**注：由于之前在对应用层的HTTP协议分析时，已经对TCP报文进行了较为完整的分析，此处着重分析对建立TCP连接三次握手机报文中与其他TCP包不同之处，着重分析信息用加黑字体表示。**

**建立TCP连接第一次握手：**

```
Transmission Control Protocol, Src Port: 59259, Dst Port: 80, Seq: 0, Len: 0
  Source Port: 59259
  Destination Port: 80
  [Stream index: 34]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 0]
  Sequence Number: 0 (relative sequence number)
  Sequence Number (raw): 4206983300
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 0
  Acknowledgment number (raw): 0
  1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x002 (SYN)
  Window: 64240
  [Calculated window size: 64240]
  Checksum: 0x68e7 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  > Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted
  > [Timestamps]
```

这是一个TCP三次握手之中的第一个报文，用于建立一个TCP连接。以下是对报文各部分的解释：

1. 源端口 (Source Port): 59259

- 表示发送端的端口号。

2. 目标端口 (Destination Port): 80

- 表示接收端的端口号，通常是HTTP服务的默认端口。

3. 序列号 (Sequence Number): SEQ=0 (相对序列号)

- 表示此数据包的序列号。在第一个握手阶段，初始序列号通常为0。

4. 确认号 (Acknowledgment Number): 0

- 表示对方期望接收的下一个序列号。在第一个握手阶段，客户端通常将确认号设置为0。这里指希望对方下一个发送回来的数据包序列号为0。

5. Flags: 0x002 (SYN)

- 表示SYN标志被设置为1，表明这是一个发起连接的同步报文。

6. Calculated window size: 64240

- 经过计算后的实际窗口大小。

7. Checksum Status: Unverified

- 校验和状态显示为未验证，表示校验和字段的值尚未被验证。

8. Timestamps

- 包含了时间戳信息，用于支持更精准的计时和延迟计算。

这个报文是一个TCP连接建立的初始阶段，由客户端发起。通过SYN标志、序列号、窗口大小等信息，客户端表明了建立连接的愿望和自身的初始参数。

## 建立连接第二次握手

这是一个TCP三次握手中的第二个报文，用于建立一个TCP连接。以下是对报文各部分的解释：

1. Source Port: 80源端口号

- 表示发送端的端口号，通常是HTTP服务的默认端口。

2. Destination Port: 59259目的地址端口号

- 表示接收端的端口号。

3. **Sequence Number: 0 (relative sequence number)相对序列号**

- 表示此数据包的序列号为0。对应第一次握手中的ACK=0，即回应一个序列号为0的数据包。

4. Sequence Number (raw): 2393276912

- 未经处理的序列号，用于在网络中传输。

5. **Acknowledgment Number: 1 (relative ack number)相对序列号**

- 表示确认号为1，即接收方期望接收的下一个序列号。

6. Header Length: 32 bytes (8)

- 表示TCP头部的长度，以32位字为单位。在这里，头部长度为32字节。

7. **Flags: 0x012 (SYN, ACK)**

- 表示SYN和ACK标志均被设置，表明这是一个确认连接的同步报文。

8. Checksum: 0x1a29 [unverified]

- 校验和字段，用于检测数据在传输过程中是否发生了错误。

9. Checksum Status: Unverified

- 校验和状态显示为未验证，表示校验和字段的值尚未被验证。

10. Timestamps

- 包含了时间戳信息，用于支持更精准的计时和延迟计算。

11. SEQ/ACK analysis

- 该部分可能包含有关序列号和确认号的分析，有关具体分析的信息未提供。

## 建立连接第三次握手

这是TCP三次握手中的第三个报文，用于完成TCP连接的建立。以下是对报文各部分的解释：

1. Source Port: 59259

- 表示发送端的端口号。

## 2. Destination Port: 80

- 表示接收端的端口号，通常是HTTP服务的默认端口。

## 3. Sequence Number: 1 (relative sequence number)

- 表示此数据包的序列号为1。在TCP的确认机制中，这是确认号为1的数据包的起始序列号。对应第二次握手中对方发来的希望收到序列号为1的包(ACK=1)的请求

## 4. Acknowledgment Number: 1 (relative ack number)ACK=1

- 确认号为1，表示希望收到的下一个数据包的序列号为1。

## 5. SEQ/ACK analysis

- 该部分可能包含有关序列号和确认号的分析，有关具体分析的信息未提供。

综上，TCP连接经过三次握手，建立成功。可总结为

- 第一次握手:** 请求建立连接方发送一个**SYN=1,SEQ=0**的数据包，SYN=1表示期望建立连接，SEQ为数据包的相对序列号，需要强调的是，虽然第一次握手的数据包中的Flags表示中没有ACK，但是对应的报文中仍然包含ACK=0的内容。
- 第二次握手:** 接收到另一方发来的**SYN=1,SEQ=0**的数据包后，回复一个**SYN=1,SEQ=0,ACK=1**的数据包，表示**同意建立连接、发送序号为0的数据包、期望收到序列号为1的数据包**，以便于实现第三次握手。
- 第三次握手:** 请求建立连接方收到发来的**SYN=1,SEQ=0,ACK=1**的数据包，回复**SEQ=0,ACK=1**的数据包，三次握手完成，连接建立。

## 4.3.2释放连接的四次握手

ip.src==202.117.1.13 ip.dst==202.117.1.13&&tcp&&tcp.port == 59259						
No.	Time	Source	Destination	Protocol	Length	Info
4380	20.082979	202.117.1.13	10.173.217.246	HTTP	1382	HTTP/1.1 200 OK (PNG)
4382	20.083292	202.117.1.13	10.173.217.246	TCP	1514	80 → 59276 [ACK] Seq=834887 Ack=5845 Win=28544 Len=1460 [TCP segment of a reassembled PDU]
4383	20.083292	202.117.1.13	10.173.217.246	HTTP	410	HTTP/1.1 200 OK (PNG)
4385	20.121232	10.173.217.246	202.117.1.13	TCP	54	59259 → 80 [ACK] Seq=6369 Ack=768892 Win=130048 Len=0
4421	25.067679	202.117.1.13	10.173.217.246	TCP	60	80 → 59273 [FIN, ACK] Seq=716006 Ack=5919 Win=28544 Len=0
4424	25.069649	202.117.1.13	10.173.217.246	TCP	60	80 → 59273 [ACK] Seq=716007 Ack=5920 Win=28544 Len=0
4425	25.072357	202.117.1.13	10.173.217.246	TCP	60	80 → 59259 [FIN, ACK] Seq=768892 Ack=6369 Win=29696 Len=0
4426	25.072390	10.173.217.246	202.117.1.13	TCP	54	59259 → 80 [ACK] Seq=6369 Ack=768893 Win=130048 Len=0
4427	25.072424	10.173.217.246	202.117.1.13	TCP	54	59259 → 80 [FIN, ACK] Seq=6369 Ack=768893 Win=130048 Len=0
4428	25.073195	202.117.1.13	10.173.217.246	TCP	60	80 → 59272 [FIN, ACK] Seq=719663 Ack=5494 Win=27520 Len=0
4431	25.075789	202.117.1.13	10.173.217.246	TCP	60	80 → 59259 [ACK] Seq=768893 Ack=6370 Win=29696 Len=0
4432	25.075789	202.117.1.13	10.173.217.246	TCP	60	80 → 59272 [ACK] Seq=719664 Ack=5495 Win=27520 Len=0
4433	25.082560	202.117.1.13	10.173.217.246	TCP	60	80 → 59274 [FIN, ACK] Seq=908563 Ack=6331 Win=29696 Len=0
4436	25.082903	202.117.1.13	10.173.217.246	TCP	60	80 → 59275 [FIN, ACK] Seq=1021646 Ack=5531 Win=27648 Len=0
4439	25.085183	202.117.1.13	10.173.217.246	TCP	60	80 → 59276 [FIN, ACK] Seq=836703 Ack=5845 Win=28544 Len=0
4442	25.085487	202.117.1.13	10.173.217.246	TCP	60	80 → 59274 [ACK] Seq=908564 Ack=6332 Win=29696 Len=0
4443	25.085487	202.117.1.13	10.173.217.246	TCP	60	80 → 59275 [ACK] Seq=1021647 Ack=5532 Win=27648 Len=0
4444	25.086764	202.117.1.13	10.173.217.246	TCP	60	80 → 59276 [ACK] Seq=836704 Ack=5846 Win=28544 Len=0

筛选条件为：**ip.src==202.117.1.13|ip.dst==202.117.1.13&&tcp&&tcp.port == 59259**，选择刚刚建立TCP连接的端口进行TCP释放的分析。

## 释放连接第一次握手，服务器端向客户端发起

```

Transmission Control Protocol, Src Port: 80, Dst Port: 59259, Seq: 768892, Ack: 6369, Len: 0
  Source Port: 80
  Destination Port: 59259
  [Stream index: 34]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 0]
  Sequence Number: 768892      (relative sequence number)
  Sequence Number (raw): 2394045804
  [Next Sequence Number: 768893      (relative sequence number)]
  Acknowledgment Number: 6369      (relative ack number)
  Acknowledgment number (raw): 4206989669
  0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x011 (FIN, ACK)
    Window: 232
    [Calculated window size: 29696]
    [Window size scaling factor: 128]
    Checksum: 0xb3eb3 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > [Timestamps]
    [Time since first frame in this TCP stream: 5.522319000 seconds]
    [Time since previous frame in this TCP stream: 4.951125000 seconds]

```

### 1. Source Port: 80

- 表示发送端的端口号，通常是HTTP服务的默认端口。

### 2. Destination Port: 59259

- 表示接收端的端口号

### 3. Acknowledgment Number: 6369 (relative ack number)

- 表示确认号为6369，即接收方期望接收的下一个序列号。

### 4. Flags: 0x011 (FIN, ACK)

- 表示FIN和ACK标志均被设置，表明这是一个带有释放请求和确认的报文。

## 释放连接第二次握手，客户端向服务器端回应

```

Transmission Control Protocol, Src Port: 59259, Dst Port: 80, Seq: 6369, Ack: 768893, Len: 0
  Source Port: 59259
  Destination Port: 80
  [Stream index: 34]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 0]
  Sequence Number: 6369      (relative sequence number)
  Sequence Number (raw): 4206989669
  [Next Sequence Number: 6369      (relative sequence number)]
  Acknowledgment Number: 768893      (relative ack number)
  Acknowledgment number (raw): 2394045805
  0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
  Window: 508
  [Calculated window size: 130048]
  [Window size scaling factor: 256]
  Checksum: 0xbd9f [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  > [Timestamps]
    [Time since first frame in this TCP stream: 5.522352000 seconds]
    [Time since previous frame in this TCP stream: 0.000033000 seconds]
  > [SEQ/ACK analysis]
    [This is an ACK to the segment in frame: 4425]
    [The RTT to ACK the segment was: 0.000033000 seconds]
    [iRTT: 0.002925000 seconds]

```

### 1. Source Port: 59259

- 表示发送端的端口号。

### 2. Destination Port: 80

- 表示接收端的端口号，通常是HTTP服务的默认端口。

### 3. Sequence Number: 6369 (relative sequence number)

- 表示此数据包的相对序列号为6369。在TCP的释放阶段，这是确认方发送的确认报文，表示接收到了释放请求的数据包。对应释放连接第一次握手时的ACK=6369,此处也能看出此分组是回应FIN的分组。

### 4. Acknowledgment Number: 768893 (relative ack number)

- 表示确认号为768893，即接收方期望接收的下一个序列号。

### 5. Flags: 0x010 ACK(ack of FIN)

- 表示ACK标志被设置，表明这是一个确认连接释放的报文。

### 6. Time since first frame in this TCP stream: 5.522352000 seconds

- 表示自TCP流的第一帧以来的时间。

### 7. Time since previous frame in this TCP stream: 0.000033000 seconds

- 表示自前一帧以来的时间。

### 8. SEQ/ACK analysis

- 包含了对序列号和确认号的分析信息，包括此报文是对哪个数据包的确认，以及确认的往返时间（RTT）等信息。

### 释放连接第三次握手，客户端向服务器端发起（与第一次握手类似）

```

Transmission Control Protocol, Src Port: 59259, Dst Port: 80, Seq: 6369, Ack: 768893, Len: 0
  Source Port: 59259
  Destination Port: 80
  [Stream index: 34]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 0]
  Sequence Number: 6369      (relative sequence number)
  Sequence Number (raw): 4206989669
  [Next Sequence Number: 6370      (relative sequence number)]
  Acknowledgment Number: 768893      (relative ack number)
  Acknowledgment number (raw): 2394045805
  0101 .... = Header Length: 20 bytes (5)
  Flags: 0x011 (FIN, ACK)
  Window: 508
  [Calculated window size: 130048]
  [Window size scaling factor: 256]
  Checksum: 0xbd9e [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  [Timestamps]
    [Time since first frame in this TCP stream: 5.522386000 seconds]
    [Time since previous frame in this TCP stream: 0.000034000 seconds]

```

#### 1. Source Port: 59259

- 表示发送端的端口号。

#### 2. Destination Port: 80

- 表示接收端的端口号，通常是HTTP服务的默认端口。

#### 3. Sequence Number: 6369 (relative sequence number)

- SEQ=6369**表示此数据包的相对序列号为6369。在TCP的释放阶段，这是确认方发送的确认报文，表示接收到了释放请求的数据包。

#### 4. Acknowledgment Number: 768893 (relative ack number)

- ACK=768893,表示确认号为768893，即接收方期望接收的下一个序列号。

#### 5. Flags: 0x011 (FIN, ACK)

- 表示FIN和ACK标志均被设置，表明这是一个带有释放请求和确认的报文。

#### 6. Timestamps

- 包含了时间戳信息，用于支持更精准的计时和延迟计算。

#### 7. Time since first frame in this TCP stream: 5.522352000 seconds

- 表示自TCP流的第一帧以来的时间。为5.522352000s。

#### 8. Time since previous frame in this TCP stream: 0.000033000 seconds

- 表示自前一帧以来的时间。为0.000033000s。

#### 9. SEQ/ACK analysis



- 包含了序列号和确认号的分析信息，包括此报文是对哪个数据包的确认，以及确认的往返时间（RTT）等信息。

### 释放连接第四次握手，服务器端向客户端回应（与第二次握手类似）

```

Transmission Control Protocol, Src Port: 80, Dst Port: 59259, Seq: 768893, Ack: 6370, Len: 0
  Source Port: 80
  Destination Port: 59259
  [Stream index: 34]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 0]
  Sequence Number: 768893 (relative sequence number)
  Sequence Number (raw): 2394045805
  [Next Sequence Number: 768893 (relative sequence number)]
  Acknowledgment Number: 6370 (relative ack number)
  Acknowledgment number (raw): 4206989670
  0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
  Window: 232
  [Calculated window size: 29696]
  [Window size scaling factor: 128]
  Checksum: 0xb2eb2 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  > [Timestamps]
    [Time since first frame in this TCP stream: 5.525751000 seconds]
    [Time since previous frame in this TCP stream: 0.003365000 seconds]
  > [SEQ/ACK analysis]
    [This is an ACK to the segment in frame: 4427]
    [The RTT to ACK the segment was: 0.003365000 seconds]
    [iRTT: 0.002925000 seconds]

```

#### 1. Source Port: 80

- 表示发送端的端口号，是HTTP服务的默认端口。

#### 2. Destination Port: 59259

- 表示接收端的端口号。

#### 3. Sequence Number: 768893 (relative sequence number)

- 表示此数据包的相对序列号为768893。在TCP的释放阶段，这是服务器端在确认回复客户端连接释放请求的报文，对应释放连接第三次握手中客户端发出的 ACK=768893。

#### 4. Acknowledgment Number: 6370 (relative ack number)

- 表示确认号为6370，即期望接收的下一个序列号。

#### 5. Flags: 0x010 ACK(ack of FIN)

- 表示ACK标志被设置，表明这是一个确认连接释放的报文。

#### 6. Timestamps

- 包含了时间戳信息，用于支持更精准的计时和延迟计算。

### 总结TCP连接释放四次握手

- 第一次握手**，服务器端向客户端发送完了所有内容，不再有消息发送，申请释放连接，发送一个有FIN标志的。

- **第二次握手**，客户端向服务器端发起的释放连接报文发起回应，不带有FIN标志，但是回应的数据包的SEQ号为第一次握手请求释放连接数据包所要求的ACK号，服务器端能够根据SEQ号知道客户端收到了连接释放请求。**此外，第二次握手完成后，客户端仍可以接受服务器端已经发送但仍未到达的数据包。只是服务器端不再保留向客户端发送数据包的窗口。**
- **第三次握手**，客户端收到服务器端发来的所有包以后，向服务器发送有FIN标志的请求释放连接的数据包。
- **第四次握手**，服务器端收到客户端发来的请求断开连接的包后，回应收到的确认包，类似于第二次握手。  
**此外，观察到这四次握手数据包的长度都为0，这是因为作为释放连接的包已经没有数据需要传输，所以长度len都为0**

## 5、实验收获

经过本次抓包分析实验，让我对HTTP协议和TCP协议的工作方式有了更加深入的认识，尤其是亲眼所见网络的分层情况后，让我对学到的知识有了更形象的认识，加深了我对课堂上学到的知识的理解。