计算机网络

实验报告

（2022学年秋季学期）

教学班级	**计科二班**	**专业（方向）**	**计算机科学与技术**
学号	**20337263**	**姓名**	**俞泽斌**

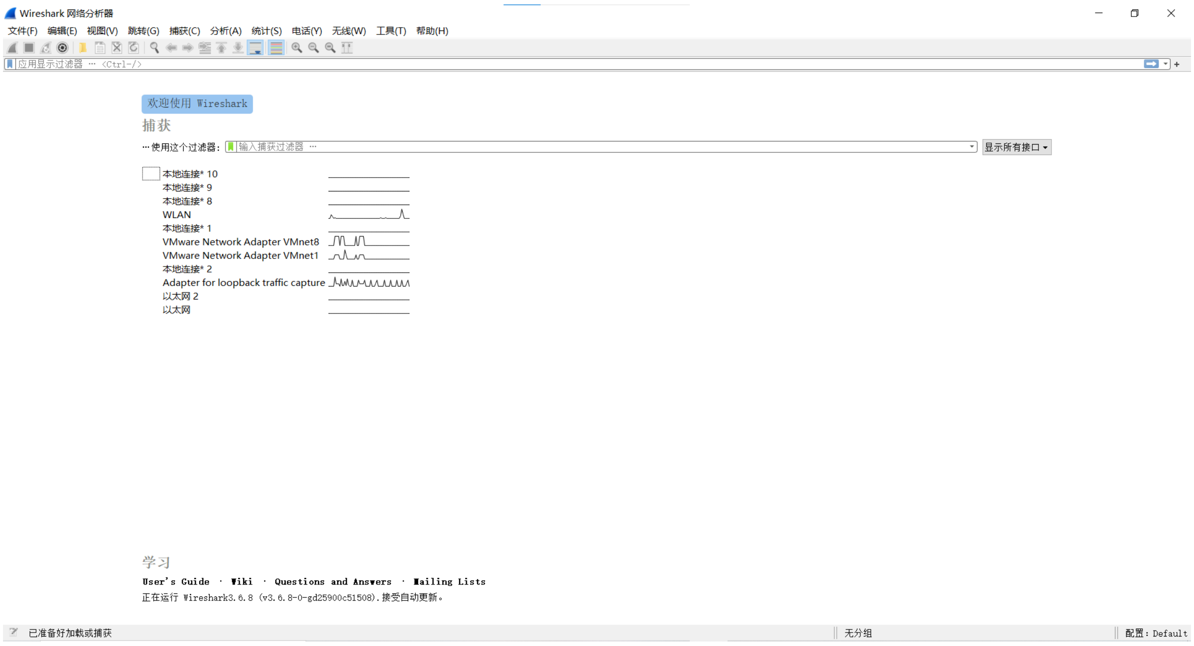
一、 实验题目

1、掌握HTTP协议获取网页的流程

2、了解HTTP请求报文和响应报文的格式，并进行报文分析

3、了解HTTP1.0和HTTP1.1的区别

二、 实验内容

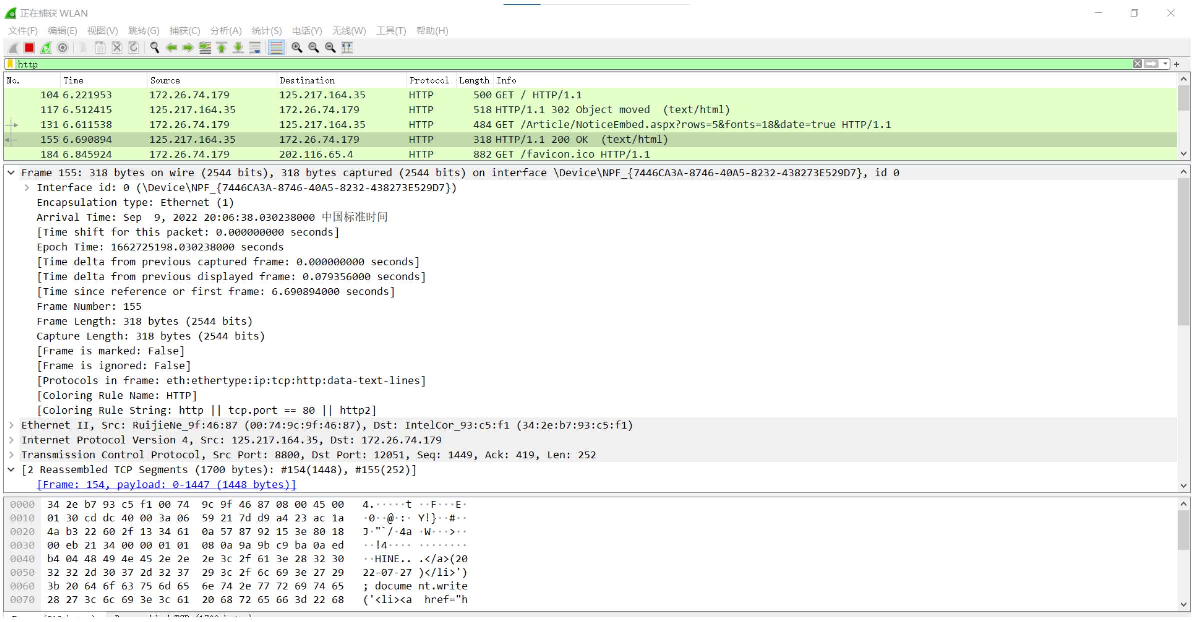

Wireshark 网络分析器
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)
应用显示过滤器 … <Ctrl-/>
欢迎使用 Wireshark
捕获
…使用这个过滤器：
输入捕获过滤器 …
显示所有接口 ▾
学习
User's Guide · Wiki · Questions and Answers · Mailing Lists

本地连接* 10
本地连接* 9
本地连接* 8
WLAN
本地连接* 1
VMware Network Adapter VMnet8
VMware Network Adapter VMnet1
本地连接* 2
Adapter for loopback traffic capture
以太网 2
以太网

正在运行 Wireshark3.6.8 (v3.6.8-0-gd25900c51508).接受自动更新。

已准备好加载或捕获 | 无分组 | 配置：Default

首先打开wireshark软件来进行抓包，然后我选择了http://home3.sysu.edu.cn/finance/作为抓包的网址，开始捕获WLAN后打开网址，抓到如下图


正在捕获 WLAN
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(T) 帮助(H)
http

```
No.   Time        Source           Destination      Protocol Length Info
  104 6.221953    172.26.74.179    125.217.164.35   HTTP      500 GET / HTTP/1.1
  117 6.512415    125.217.164.35   172.26.74.179    HTTP      518 HTTP/1.1 302 Object moved  (text/html)
  131 6.611538    172.26.74.179    125.217.164.35   HTTP      484 GET /Article/NoticeEmbed.aspx?rows=5&fonts=18&date=true HTTP/1.1
  155 6.690894    125.217.164.35   172.26.74.179    HTTP      318 HTTP/1.1 200 OK  (text/html)
  184 6.845924    172.26.74.179    202.116.65.4     HTTP      882 GET /favicon.ico HTTP/1.1
```

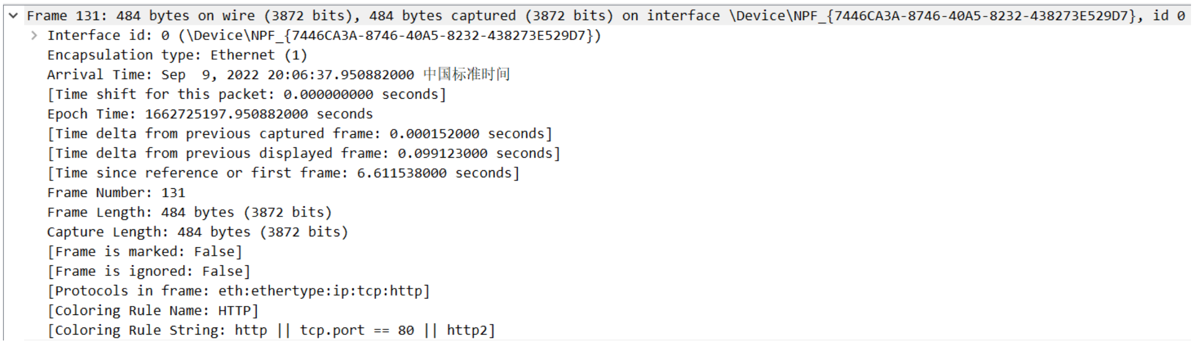
```
Frame 155: 318 bytes on wire (2544 bits), 318 bytes captured (2544 bits) on interface \Device\NPF_{7446CA3A-8746-40A5-8232-438273E529D7}, id 0
  Interface id: 0 (\Device\NPF_{7446CA3A-8746-40A5-8232-438273E529D7})
  Encapsulation type: Ethernet (1)
  Arrival Time: Sep  9, 2022 20:06:38.030238000 中国标准时间
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1662725198.030238000 seconds
  [Time delta from previous captured frame: 0.000000000 seconds]
  [Time delta from previous displayed frame: 0.079356000 seconds]
  [Time since reference or first frame: 6.690894000 seconds]
  Frame Number: 155
  Frame Length: 318 bytes (2544 bits)
  Capture Length: 318 bytes (2544 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: RuijieNe_9f:46:87 (00:74:9c:9f:46:87), Dst: IntelCor_93:c5:f1 (34:2e:b7:93:c5:f1)
Internet Protocol Version 4, Src: 125.217.164.35, Dst: 172.26.74.179
Transmission Control Protocol, Src Port: 8800, Dst Port: 12051, Seq: 1449, Ack: 419, Len: 252
[2 Reassembled TCP Segments (1700 bytes): #154(1448), #155(252)]
  [Frame: 154, payload: 0-1447 (1448 bytes)]
```

```
0000  34 2e b7 93 c5 f1 00 74  9c 9f 46 87 08 00 45 00   4.·····t ··F···E·
0010  01 30 cd dc 40 00 3a 06  59 21 7d d9 a4 23 ac 1a   ·0··@·:· Y!}··#··
0020  4a b3 22 60 2f 13 34 61  0a 57 87 92 15 3e 80 18   J·"`/·4a ·W···>··
0030  00 eb 21 34 00 00 01 01  08 0a 9a 9b c9 ba 0a ed   ··!4···· ········
0040  b4 04 48 49 4e 45 2e 2e  2e 3c 2f 61 3e 28 32 30   ··HINE.. .</a>(20
0050  32 32 2d 30 37 2d 32 37  29 3c 2f 6c 69 3e 27 29   22-07-27 )</li>')
0060  3b 20 64 6f 63 75 6d 65  6e 74 2e 77 72 69 74 65   ; docume nt.write
0070  28 27 3c 6c 69 3e 3c 61  20 68 72 65 66 3d 22 68   ('<li><a  href="h
```

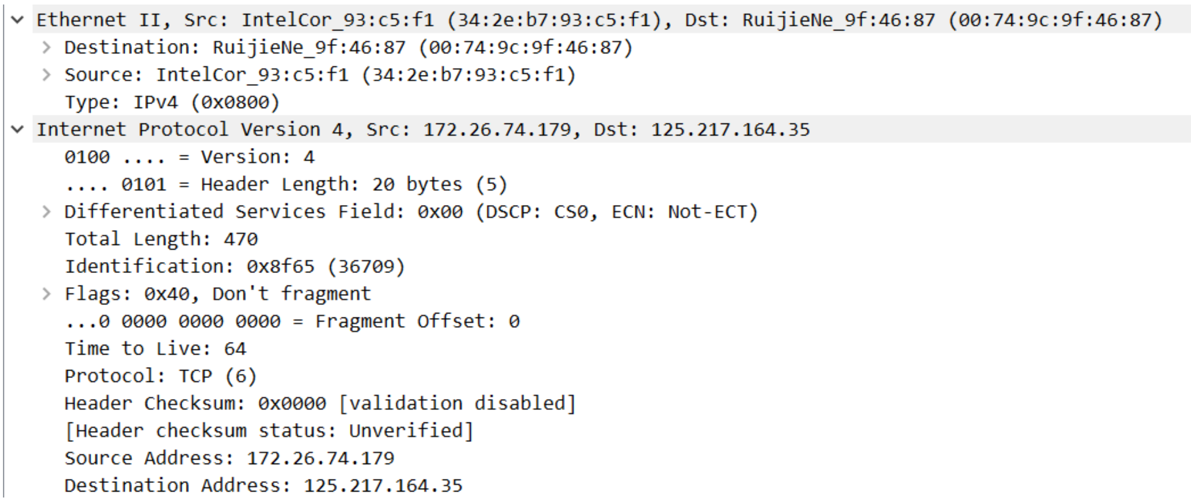
其实这里开始的时候发现有两次的来回，第一次网站返回的是302的代码，也就是要跳转的网站，后面才得到200 ok的代码，说明网站访问成功，具体这里的理解主要是第一次可能是类似于握手的操作，就是确认一下有这个网址，然后第二次才是正式的访问，访问的来回两个包，一个是浏览器发送到网站对应位置的，另一个获取的网站返回的信息。

我们这里主要取的是返回代码为200 的两个报文，然后下面的报文我们做具体的分析，首先是请求报文的部分，首先把所有的报文都展开来

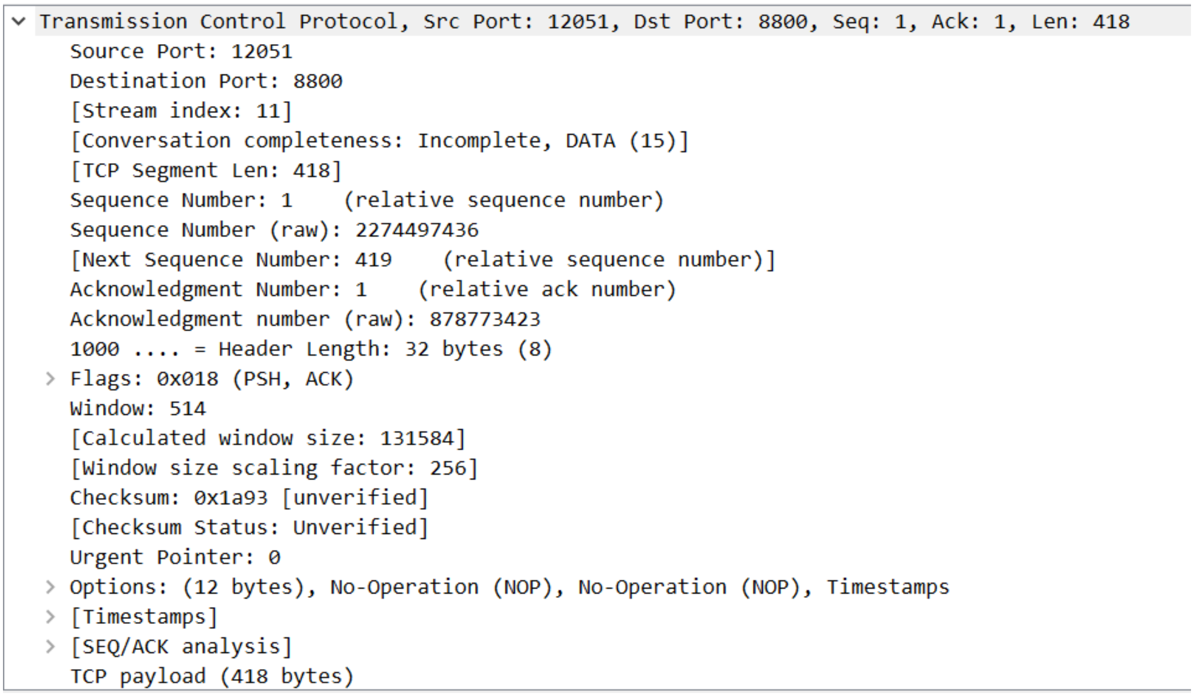
(1)



```
Frame 131: 484 bytes on wire (3872 bits), 484 bytes captured (3872 bits) on interface \Device\NPF_{7446CA3A-8746-40A5-8232-438273E529D7}, id 0
  Interface id: 0 (\Device\NPF_{7446CA3A-8746-40A5-8232-438273E529D7})
  Encapsulation type: Ethernet (1)
  Arrival Time: Sep  9, 2022 20:06:37.950882000 中国标准时间
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1662725197.950882000 seconds
  [Time delta from previous captured frame: 0.000152000 seconds]
  [Time delta from previous displayed frame: 0.099123000 seconds]
  [Time since reference or first frame: 6.611538000 seconds]
  Frame Number: 131
  Frame Length: 484 bytes (3872 bits)
  Capture Length: 484 bytes (3872 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]
```

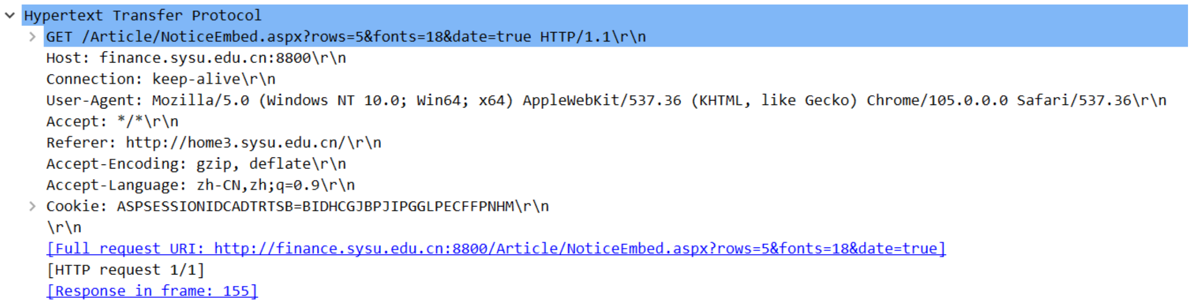


```
Ethernet II, Src: IntelCor_93:c5:f1 (34:2e:b7:93:c5:f1), Dst: RuijieNe_9f:46:87 (00:74:9c:9f:46:87)
  Destination: RuijieNe_9f:46:87 (00:74:9c:9f:46:87)
  Source: IntelCor_93:c5:f1 (34:2e:b7:93:c5:f1)
  Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 172.26.74.179, Dst: 125.217.164.35
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 470
  Identification: 0x8f65 (36709)
  Flags: 0x40, Don't fragment
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 64
  Protocol: TCP (6)
  Header Checksum: 0x0000 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 172.26.74.179
  Destination Address: 125.217.164.35
```



```
Transmission Control Protocol, Src Port: 12051, Dst Port: 8800, Seq: 1, Ack: 1, Len: 418
    Source Port: 12051
    Destination Port: 8800
    [Stream index: 11]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 418]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 2274497436
    [Next Sequence Number: 419    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 878773423
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x018 (PSH, ACK)
    Window: 514
    [Calculated window size: 131584]
    [Window size scaling factor: 256]
    Checksum: 0x1a93 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  > [Timestamps]
  > [SEQ/ACK analysis]
    TCP payload (418 bytes)
```

其实主要看的是下面的这段报文



```
Hypertext Transfer Protocol
  > GET /Article/NoticeEmbed.aspx?rows=5&fonts=18&date=true HTTP/1.1\r\n
    Host: finance.sysu.edu.cn:8800\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/105.0.0.0 Safari/537.36\r\n
    Accept: */*\r\n
    Referer: http://home3.sysu.edu.cn/\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: zh-CN,zh;q=0.9\r\n
  > Cookie: ASPSESSIONIDCADTRTSB=BIDHCGJBPJIPGGLPECFFPNHM\r\n
    \r\n
    [Full request URI: http://finance.sysu.edu.cn:8800/Article/NoticeEmbed.aspx?rows=5&fonts=18&date=true]
    [HTTP request 1/1]
    [Response in frame: 155]
```

http请求报文

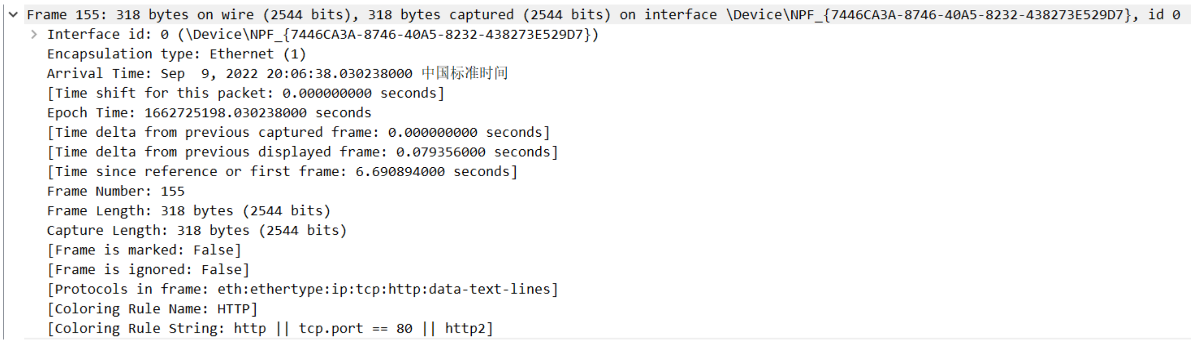
方法	get	版本	1.1	URL	

首部字段名	字段值	字段所表达的信息
HOST	finance.sysu.edu.cn:8800	请求的主机名，资源所在的服务器
Connection	keep-alive	客户端和服务端指定的请求，响应有关选项
User-Agent	Mozilla,AppleWebKit/537.36,Chrome/105.0.0.0 Safari/537.36	发送请求的操作系统和浏览器信息
Accept	* / *	客户端可识别的内容类型列表，这里表示都可以

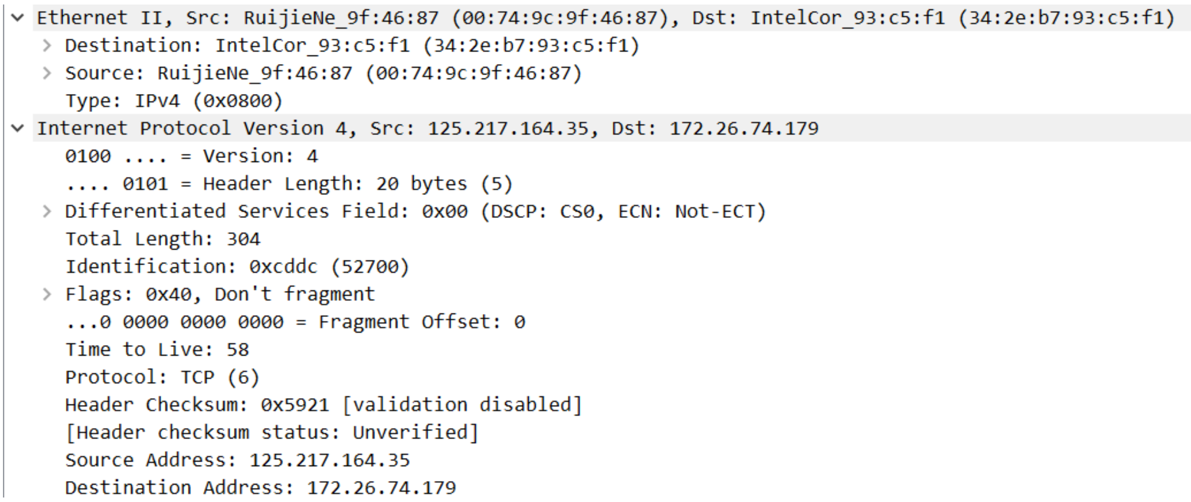
Referer	http://home3.sysu.edu.cn/	判断来源页面
Accept-Encoding	gzip, deflate	客户端可识别的数据编码
Accept-Language	zh-CN,zh;q=0.9	浏览器所支持的语言类型
Cookie	ASPSESSIONIDCADTRTSB=BIDHCGJBPJIPGGLPECFFPNHM	发出http请求时会默认携带的一段文本片段

(2)

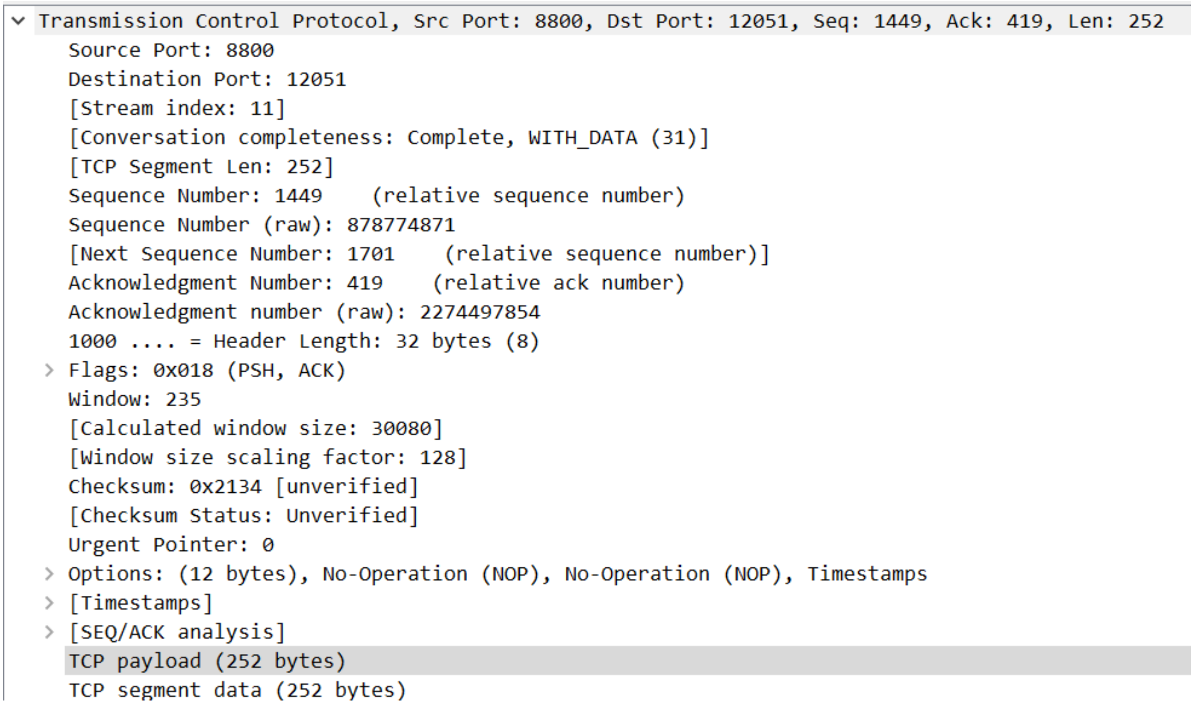
接下来是返回的报文



```
v Frame 155: 318 bytes on wire (2544 bits), 318 bytes captured (2544 bits) on interface \Device\NPF_{7446CA3A-8746-40A5-8232-438273E529D7}, id 0
  > Interface id: 0 (\Device\NPF_{7446CA3A-8746-40A5-8232-438273E529D7})
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep  9, 2022 20:06:38.030238000 中国标准时间
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1662725198.030238000 seconds
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.079356000 seconds]
    [Time since reference or first frame: 6.690894000 seconds]
    Frame Number: 155
    Frame Length: 318 bytes (2544 bits)
    Capture Length: 318 bytes (2544 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]
```

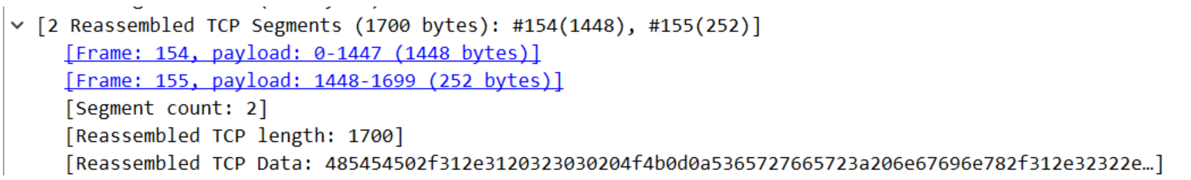


```
v Ethernet II, Src: RuijieNe_9f:46:87 (00:74:9c:9f:46:87), Dst: IntelCor_93:c5:f1 (34:2e:b7:93:c5:f1)
  > Destination: IntelCor_93:c5:f1 (34:2e:b7:93:c5:f1)
  > Source: RuijieNe_9f:46:87 (00:74:9c:9f:46:87)
    Type: IPv4 (0x0800)
v Internet Protocol Version 4, Src: 125.217.164.35, Dst: 172.26.74.179
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 304
    Identification: 0xcddc (52700)
  > Flags: 0x40, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 58
    Protocol: TCP (6)
    Header Checksum: 0x5921 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 125.217.164.35
    Destination Address: 172.26.74.179
```

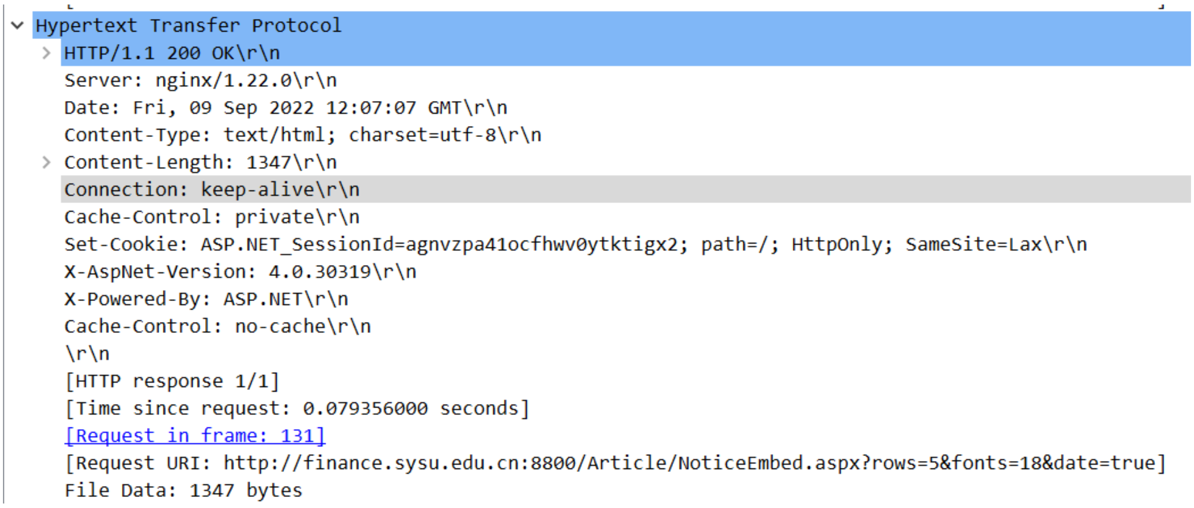


```
v Transmission Control Protocol, Src Port: 8800, Dst Port: 12051, Seq: 1449, Ack: 419, Len: 252
    Source Port: 8800
    Destination Port: 12051
    [Stream index: 11]
    [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 252]
    Sequence Number: 1449    (relative sequence number)
    Sequence Number (raw): 878774871
    [Next Sequence Number: 1701    (relative sequence number)]
    Acknowledgment Number: 419    (relative ack number)
    Acknowledgment number (raw): 2274497854
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x018 (PSH, ACK)
    Window: 235
    [Calculated window size: 30080]
    [Window size scaling factor: 128]
    Checksum: 0x2134 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  > [Timestamps]
  > [SEQ/ACK analysis]
    TCP payload (252 bytes)
    TCP segment data (252 bytes)
```

其实主要看的也是下面的这段报文



```
v [2 Reassembled TCP Segments (1700 bytes): #154(1448), #155(252)]
    [Frame: 154, payload: 0-1447 (1448 bytes)]
    [Frame: 155, payload: 1448-1699 (252 bytes)]
    [Segment count: 2]
    [Reassembled TCP length: 1700]
    [Reassembled TCP Data: 485454502f312e3120323030204f4b0d0a5365727665723a206e67696e782f312e32322e…]
```



```
v Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
    Server: nginx/1.22.0\r\n
    Date: Fri, 09 Sep 2022 12:07:07 GMT\r\n
    Content-Type: text/html; charset=utf-8\r\n
  > Content-Length: 1347\r\n
    Connection: keep-alive\r\n
    Cache-Control: private\r\n
    Set-Cookie: ASP.NET_SessionId=agnvzpa41ocfhwv0ytktigx2; path=/; HttpOnly; SameSite=Lax\r\n
    X-AspNet-Version: 4.0.30319\r\n
    X-Powered-By: ASP.NET\r\n
    Cache-Control: no-cache\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.079356000 seconds]
    [Request in frame: 131]
    [Request URI: http://finance.sysu.edu.cn:8800/Article/NoticeEmbed.aspx?rows=5&fonts=18&date=true]
    File Data: 1347 bytes
```

http应答报文

版本	1.1	状态码	200	短语	ok

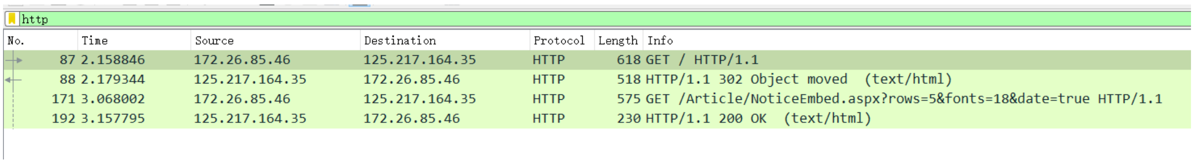
首部字段名	字段值	字段所表达的信息
Server	nginx/1.22.0	服务器信息及版本
Date	Fri ,09 Sep 2022 12:07:01 GMT	响应时间
Content-Type	text/html;charset=utf-8	告诉客户端实际返回的类型
Content-Length	1347	http消息主体的大小

首部字段名	字段值	字段所表达的信息
Connection	keep-alive	连接状态，长连接
cache-control	private	缓存机制，仅开放给部分用户做缓存
set-cookie	ASP.NET_SessionId=p4yohj4wlrf1sthdqd3hwnx0; path=/; HttpOnly; SameSite=Lax	默认携带的一段文本片段

（3）

现在综合分析捕获的报文，理解HTTP协议的工作过程

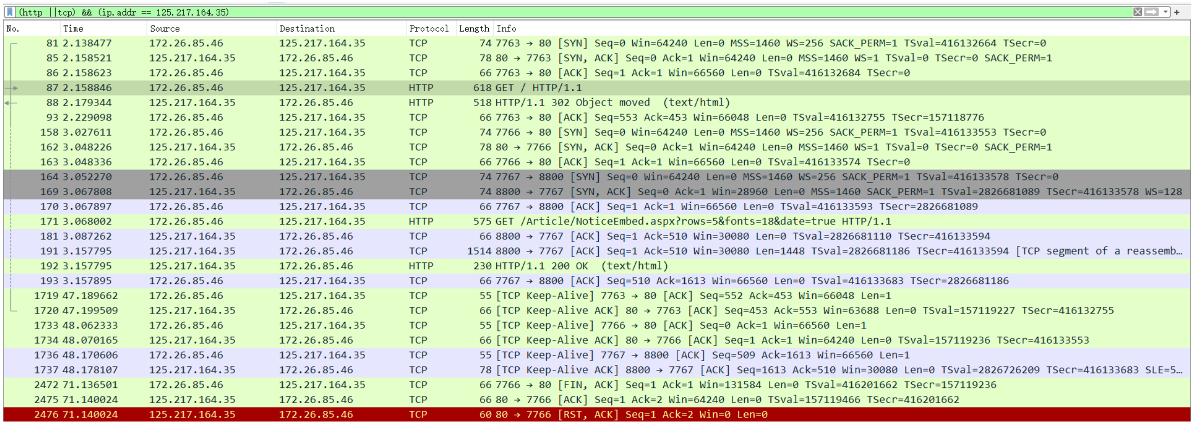
首先我们来看具体的捕获到的包



```
http
No.      Time      Source          Destination     Protocol  Length Info
    87 2.158846    172.26.85.46    125.217.164.35  HTTP      618 GET / HTTP/1.1
    88 2.179344    125.217.164.35  172.26.85.46    HTTP      518 HTTP/1.1 302 Object moved  (text/html)
   171 3.068002    172.26.85.46    125.217.164.35  HTTP      575 GET /Article/NoticeEmbed.aspx?rows=5&fonts=18&date=true HTTP/1.1
   192 3.157795    125.217.164.35  172.26.85.46    HTTP      230 HTTP/1.1 200 OK  (text/html)
```

可以看到首先finance.sysu.edu.cn的请求，服务器将原先的 http 链接重定向至 https 链接，所以第一段返回的是304 not modified

后面定位到了home3.sysu.edu.cn/finance 后才返回200 ok，现在我们将过滤放开来看下具体的工作，因为此时重定向后的这个网站的ip地址为125.217.164.35，所以将这个来作为过滤条件，得到过滤条件为(http ||tcp) && (ip.addr == 125.217.164.35)



```
(http ||tcp) && (ip.addr == 125.217.164.35)
No.      Time       Source          Destination     Protocol Length Info
    81 2.138477     172.26.85.46    125.217.164.35  TCP        74 7763 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=416132664 TSecr=0
    85 2.158521     125.217.164.35  172.26.85.46    TCP        78 80 → 7763 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 WS=1 TSval=0 TSecr=0 SACK_PERM=1
    86 2.158623     172.26.85.46    125.217.164.35  TCP        66 7763 → 80 [ACK] Seq=1 Ack=1 Win=66560 Len=0 TSval=416132684 TSecr=0
    87 2.158846     172.26.85.46    125.217.164.35  HTTP      618 GET / HTTP/1.1
    88 2.179344     125.217.164.35  172.26.85.46    HTTP      518 HTTP/1.1 302 Object moved  (text/html)
    93 2.229098     172.26.85.46    125.217.164.35  TCP        66 7763 → 80 [ACK] Seq=553 Ack=453 Win=66048 Len=0 TSval=416132755 TSecr=157118776
   158 3.027611     172.26.85.46    125.217.164.35  TCP        74 7766 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=416133553 TSecr=0
   162 3.048226     125.217.164.35  172.26.85.46    TCP        78 80 → 7766 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 WS=1 TSval=0 TSecr=0 SACK_PERM=1
   163 3.048336     172.26.85.46    125.217.164.35  TCP        66 7766 → 80 [ACK] Seq=1 Ack=1 Win=66560 Len=0 TSval=416133574 TSecr=0
   164 3.052270     172.26.85.46    125.217.164.35  TCP        74 7767 → 8800 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=416133578 TSecr=0
   169 3.067808     125.217.164.35  172.26.85.46    TCP        74 8800 → 7767 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=2826681089 TSecr=416133578 WS=128
   170 3.067897     172.26.85.46    125.217.164.35  TCP        66 7767 → 8800 [ACK] Seq=1 Ack=1 Win=66560 Len=0 TSval=416133593 TSecr=2826681089
   171 3.068002     172.26.85.46    125.217.164.35  HTTP      575 GET /Article/NoticeEmbed.aspx?rows=5&fonts=18&date=true HTTP/1.1
   181 3.087262     125.217.164.35  172.26.85.46    TCP        66 8800 → 7767 [ACK] Seq=1 Ack=510 Win=30080 Len=0 TSval=2826681110 TSecr=416133594
   191 3.157795     125.217.164.35  172.26.85.46    TCP      1514 8800 → 7767 [ACK] Seq=1 Ack=510 Win=30080 Len=1448 TSval=2826681186 TSecr=416133594 [TCP segment of a reassemb…
   192 3.157795     125.217.164.35  172.26.85.46    HTTP      230 HTTP/1.1 200 OK  (text/html)
   193 3.157895     172.26.85.46    125.217.164.35  TCP        66 7767 → 8800 [ACK] Seq=510 Ack=1613 Win=66560 Len=0 TSval=416133683 TSecr=2826681186
  1719 47.189662    172.26.85.46    125.217.164.35  TCP        55 [TCP Keep-Alive] 7763 → 80 [ACK] Seq=552 Ack=453 Win=66048 Len=1
  1720 47.199509    125.217.164.35  172.26.85.46    TCP        66 [TCP Keep-Alive ACK] 80 → 7763 [ACK] Seq=453 Ack=553 Win=63688 Len=0 TSval=157119227 TSecr=416132755
  1733 48.062333    172.26.85.46    125.217.164.35  TCP        55 [TCP Keep-Alive] 7766 → 80 [ACK] Seq=0 Ack=1 Win=66560 Len=1
  1734 48.070165    125.217.164.35  172.26.85.46    TCP        66 [TCP Keep-Alive ACK] 80 → 7766 [ACK] Seq=1 Ack=1 Win=64240 Len=0 TSval=157119236 TSecr=416133553
  1736 48.170606    172.26.85.46    125.217.164.35  TCP        55 [TCP Keep-Alive] 7767 → 8800 [ACK] Seq=509 Ack=1613 Win=66560 Len=1
  1737 48.178107    125.217.164.35  172.26.85.46    TCP        78 [TCP Keep-Alive ACK] 8800 → 7767 [ACK] Seq=1613 Ack=510 Win=30080 Len=0 TSval=2826726209 TSecr=416133683 SLE=5…
  2472 71.136501    172.26.85.46    125.217.164.35  TCP        66 7766 → 80 [FIN, ACK] Seq=1 Ack=1 Win=131584 Len=0 TSval=416201662 TSecr=157119236
  2475 71.140024    125.217.164.35  172.26.85.46    TCP        66 80 → 7766 [ACK] Seq=1 Ack=2 Win=64240 Len=0 TSval=157119466 TSecr=416201662
  2476 71.140024    125.217.164.35  172.26.85.46    TCP        60 80 → 7766 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0
```

http协议工作过程

客户机端口号	服务器端口号	所包括的报文号	工作过程
7763	80	81	客户端发送给服务器 SYN 数据包，并请求连接
80	7763	85	服务端返回 SYN 和 ACK数据包，返回连接请求和数据
7763	80	86	客户端返回 ACK 报文，说明客户端可以开始接受数据，也是tcp的三次握手成功
7763	80	87	第一次开始发送GET请求
80	7763	88	服务端重定向至其他地址，返回 302 object moved
7767	80	164	客户端发送给服务器 SYN 数据包，再次请求连接

客户机端口号	服务器端口号	所包括的报文号	工作过程
80	7767	169	服务端返回 SYN 和 ACK数据包，返回连接请求和数据
7767	80	170	客户端返回 ACK 报文，说明客户端可以开始接受数据，tcp的三次握手成功
7767	80	171	第二次开始发送GET请求
80	7767	181,191	客户端返回 ACK 报文
80	7767	192	客户端返回200 ok， 开始传输数据

（4）第一个HTTP会话中，Web服务器对Web客户端GET请求的回应是302 object moved， 相当于一次握手过程

第二个HTTP会话中，Web服务器对Web客户端GET请求的回应是200 ok，表明请求成功

实验思考

（1）连接是由客户端发起建立的，通过先向服务端发送SYN数据包并请求连接的方式，然后等服务端返回SYN和ACK数据包，在发送ACK报文，握手成功后发送请求，请求被服务端接收并返回200 ok时，开始传输数据，HTTP会话建立成功

（2）关闭连接是由服务端发起的，断开连接时，发送 close_notify 报文。之后再发送 TCP FIN 报文来关闭与 TCP 的通信

（3）GET 用于信息获取，而且应该是安全的和幂等的POST 表示可能修改变服务器上的资源的请求，POST提交表单，因为这样就会涉及对服务器上的资源的修改，所以用POST，如果直接访问的话，用GET就可以了。