实验2 配置Web服务器，编写简单页面，分析交互过程

一、Web服务器搭建

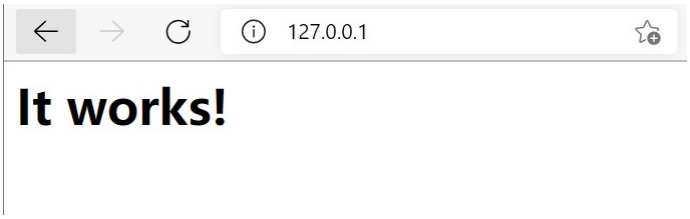
系统：Windows11　**软件**：Apache　**端口**：8000(修改了默认端口)

（一）安装配置

官网下载Apache，进入bin文件夹，输入命令`httpd -k install`安装。修改文档`Apache\conf\httpd.conf`，更改路径`Define SRVROOT "C:\web\Apache"`，配置`ServerName`，输入命令`httpd -k start`启动服务。

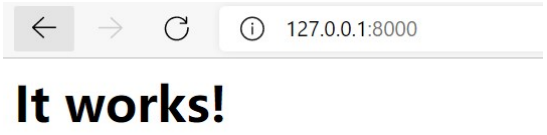
```
36  #
37  Define SRVROOT "C:\web\Apache"
38
39  ServerRoot "${SRVROOT}"
40
41  ServerName localhost
42  #
```

用浏览器进入127.0.0.1，成功



修改端口为8000，重启Apache services

用浏览器进入127.0.0.1:8000，成功

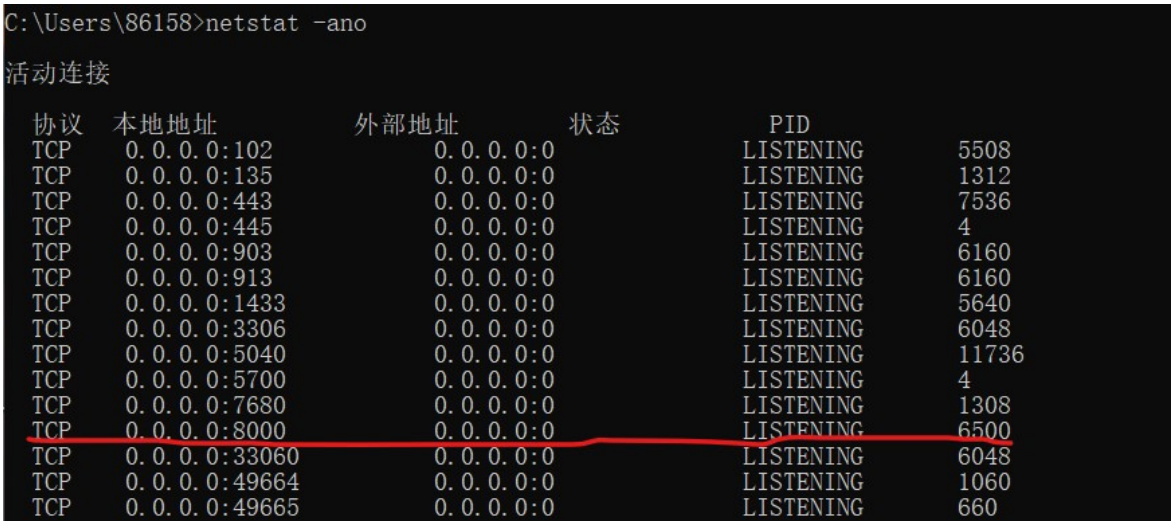


使用命令`netstat -ano`查看本机所有端口的使用情况，8000端口已启用

```
C:\Users\86158>netstat -ano

活动连接

  协议  本地地址          外部地址        状态           PID
  TCP    0.0.0.0:80             0.0.0.0:0              LISTENING       4648
  TCP    0.0.0.0:102            0.0.0.0:0              LISTENING       5508
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING       1312
  TCP    0.0.0.0:443            0.0.0.0:0              LISTENING       7536
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:903            0.0.0.0:0              LISTENING       6160
  TCP    0.0.0.0:913            0.0.0.0:0              LISTENING       6160
  TCP    0.0.0.0:1433           0.0.0.0:0              LISTENING       5640
  TCP    0.0.0.0:3306           0.0.0.0:0              LISTENING       6048
  TCP    0.0.0.0:5040           0.0.0.0:0              LISTENING       11736
  TCP    0.0.0.0:5700           0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:7680           0.0.0.0:0              LISTENING       1308
  TCP    0.0.0.0:33060          0.0.0.0:0              LISTENING       6048
  TCP    0.0.0.0:49664          0.0.0.0:0              LISTENING       1060
  TCP    0.0.0.0:49665          0.0.0.0:0              LISTENING       660
```



二、简单的web页面

设置标签页标题和icon图标，内容设置为姓名学号专业，还有一个logo图标。

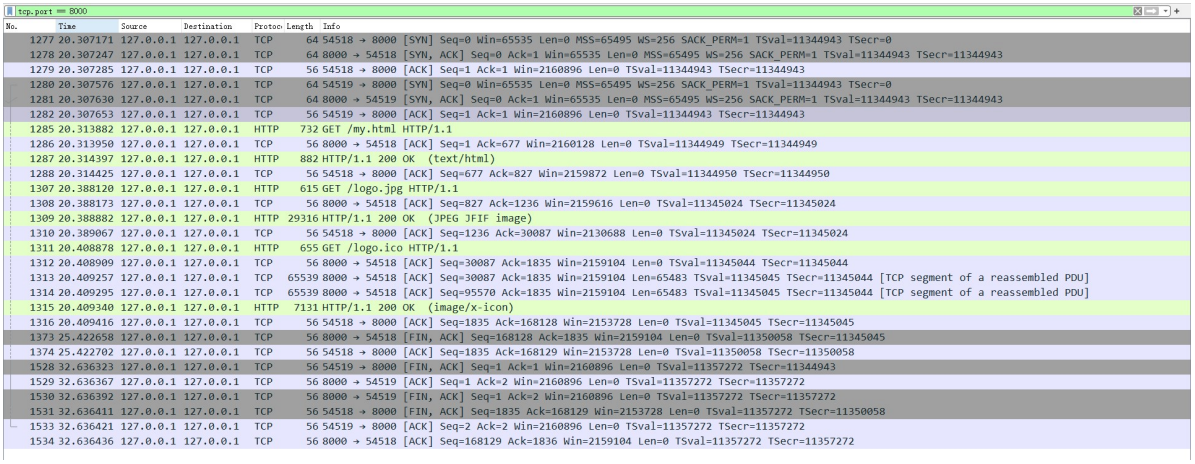
```
<html>
<meta charset="utf-8">
<head>
    <title>my web</title>
    <link rel="shortcut icon" href="logo.ico" type="logo" />
</head>
<body>
    <center>
        <p><br><br><br><br><br><br>
            姓名:
            <br>
            学号:
            <br>
            专业:
            <br>
        </p>
        <p>
            下面是我的logo<br>
            <img src="logo.jpg" referrerpolicy="no-referrer"
style="zoom:50%;">
        </p>
    </center>
</body>
</html>
```

访问网址127.0.0.1:8000/my.html，下图为网页内容。

三、Wireshark捕获交互过程

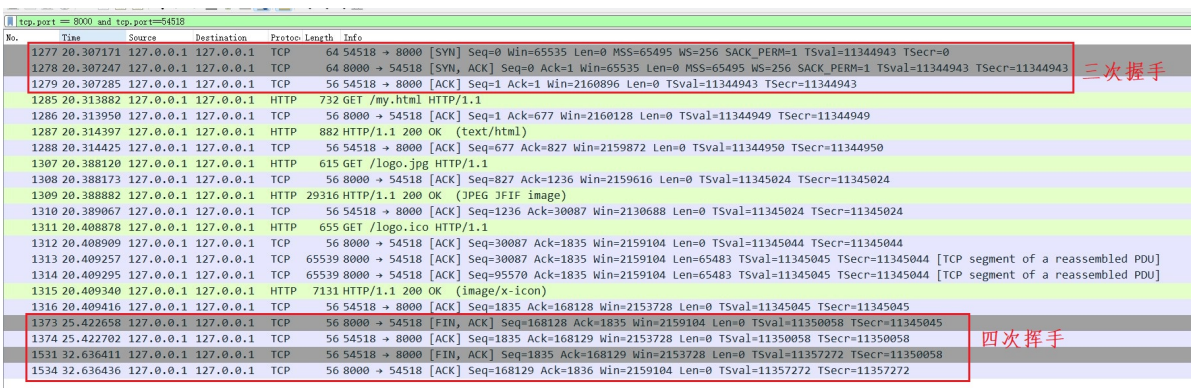
（一）抓包

由于是抓取通过 127.0.0.1 本地环回地址的包，所以需要选择Adapter for loopback traffic capture。查看端口为8000的所有包（设置tcp.port == 8000）。下图为访问时，抓取8000端口所有的包。



No.	Time	Source	Destination	Protocol	Length	Info
1277	20.307171	127.0.0.1	127.0.0.1	TCP	64	54518 → 8000 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1 TSval=11344943 TSecr=0
1278	20.307247	127.0.0.1	127.0.0.1	TCP	64	8000 → 54518 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1 TSval=11344943 TSecr=11344943
1279	20.307285	127.0.0.1	127.0.0.1	TCP	56	54518 → 8000 [ACK] Seq=1 Ack=1 Win=2160896 Len=0 TSval=11344943 TSecr=11344943
1280	20.307576	127.0.0.1	127.0.0.1	TCP	64	54519 → 8000 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1 TSval=11344943 TSecr=0
1281	20.307630	127.0.0.1	127.0.0.1	TCP	64	8000 → 54519 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1 TSval=11344943 TSecr=11344943
1282	20.307653	127.0.0.1	127.0.0.1	TCP	56	54519 → 8000 [ACK] Seq=1 Ack=1 Win=2160896 Len=0 TSval=11344943 TSecr=11344943
1285	20.313882	127.0.0.1	127.0.0.1	HTTP	732	GET /my.html HTTP/1.1
1286	20.313950	127.0.0.1	127.0.0.1	TCP	56	8000 → 54518 [ACK] Seq=1 Ack=677 Win=2160128 Len=0 TSval=11344949 TSecr=11344949
1287	20.314397	127.0.0.1	127.0.0.1	HTTP	882	HTTP/1.1 200 OK (text/html)
1288	20.314425	127.0.0.1	127.0.0.1	TCP	56	54518 → 8000 [ACK] Seq=677 Ack=827 Win=2159872 Len=0 TSval=11344950 TSecr=11344950
1307	20.388120	127.0.0.1	127.0.0.1	HTTP	615	GET /logo.jpg HTTP/1.1
1308	20.388173	127.0.0.1	127.0.0.1	TCP	56	8000 → 54518 [ACK] Seq=827 Ack=1236 Win=2159616 Len=0 TSval=11345024 TSecr=11345024
1309	20.388882	127.0.0.1	127.0.0.1	HTTP	29316	HTTP/1.1 200 OK (JPEG JFIF image)
1310	20.389067	127.0.0.1	127.0.0.1	TCP	56	54518 → 8000 [ACK] Seq=1236 Ack=30087 Win=2130688 Len=0 TSval=11345024 TSecr=11345024
1311	20.408878	127.0.0.1	127.0.0.1	HTTP	655	GET /logo.ico HTTP/1.1
1312	20.408909	127.0.0.1	127.0.0.1	TCP	56	8000 → 54518 [ACK] Seq=30087 Ack=1835 Win=2159104 Len=0 TSval=11345044 TSecr=11345044
1313	20.409257	127.0.0.1	127.0.0.1	TCP	65539	8000 → 54518 [ACK] Seq=30087 Ack=1835 Win=2159104 Len=65483 TSval=11345045 TSecr=11345044 [TCP segment of a reassembled PDU]
1314	20.409295	127.0.0.1	127.0.0.1	TCP	65539	8000 → 54518 [ACK] Seq=95570 Ack=1835 Win=2159104 Len=65483 TSval=11345045 TSecr=11345044 [TCP segment of a reassembled PDU]
1315	20.409340	127.0.0.1	127.0.0.1	HTTP	7131	HTTP/1.1 200 OK (image/x-icon)
1316	20.409416	127.0.0.1	127.0.0.1	TCP	56	54518 → 8000 [ACK] Seq=1835 Ack=168128 Win=2153728 Len=0 TSval=11345045 TSecr=11345045
1373	25.422658	127.0.0.1	127.0.0.1	TCP	56	8000 → 54518 [FIN, ACK] Seq=168128 Ack=1835 Win=2159104 Len=0 TSval=11350058 TSecr=11345045
1374	25.422702	127.0.0.1	127.0.0.1	TCP	56	54518 → 8000 [ACK] Seq=1835 Ack=168129 Win=2153728 Len=0 TSval=11350058 TSecr=11350058
1528	32.636323	127.0.0.1	127.0.0.1	TCP	56	54519 → 8000 [FIN, ACK] Seq=1 Ack=1 Win=2160896 Len=0 TSval=11357272 TSecr=11344943
1529	32.636367	127.0.0.1	127.0.0.1	TCP	56	8000 → 54519 [ACK] Seq=1 Ack=2 Win=2160896 Len=0 TSval=11357272 TSecr=11357272
1530	32.636392	127.0.0.1	127.0.0.1	TCP	56	8000 → 54519 [FIN, ACK] Seq=1 Ack=2 Win=2160896 Len=0 TSval=11357272 TSecr=11357272
1531	32.636411	127.0.0.1	127.0.0.1	TCP	56	54518 → 8000 [FIN, ACK] Seq=1835 Ack=168129 Win=2153728 Len=0 TSval=11357272 TSecr=11350058
1533	32.636421	127.0.0.1	127.0.0.1	TCP	56	54519 → 8000 [ACK] Seq=2 Ack=2 Win=2160896 Len=0 TSval=11357272 TSecr=11357272
1534	32.636436	127.0.0.1	127.0.0.1	TCP	56	8000 → 54518 [ACK] Seq=168129 Ack=1836 Win=2159104 Len=0 TSval=11357272 TSecr=11357272

采用的是HTTP/1.1，使用双端口进行连接，防止头阻塞，所以有两个端口（54518、54519）与8000端口同时交互，这也是浏览器http请求并发性的体现。所以在这个过程中出现了两组TCP三次握手和两组TCP四次挥手。但从图中可以看到端口54518完成了所有的交互过程，并未出现阻塞，54519只进行了三次握手和四次挥手，所以下面只分析54518。

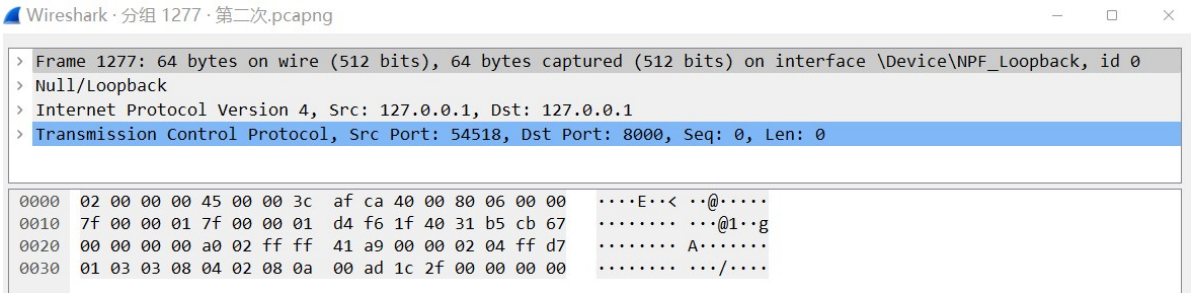


No.	Time	Source	Destination	Protocol	Length	Info
1277	20.307171	127.0.0.1	127.0.0.1	TCP	64	54518 → 8000 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1 TSval=11344943 TSecr=0
1278	20.307247	127.0.0.1	127.0.0.1	TCP	64	8000 → 54518 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1 TSval=11344943 TSecr=11344943
1279	20.307285	127.0.0.1	127.0.0.1	TCP	56	54518 → 8000 [ACK] Seq=1 Ack=1 Win=2160896 Len=0 TSval=11344943 TSecr=11344943
1285	20.313882	127.0.0.1	127.0.0.1	HTTP	732	GET /my.html HTTP/1.1
1286	20.313950	127.0.0.1	127.0.0.1	TCP	56	8000 → 54518 [ACK] Seq=1 Ack=677 Win=2160128 Len=0 TSval=11344949 TSecr=11344949
1287	20.314397	127.0.0.1	127.0.0.1	HTTP	882	HTTP/1.1 200 OK (text/html)
1288	20.314425	127.0.0.1	127.0.0.1	TCP	56	54518 → 8000 [ACK] Seq=677 Ack=827 Win=2159872 Len=0 TSval=11344950 TSecr=11344950
1307	20.388120	127.0.0.1	127.0.0.1	HTTP	615	GET /logo.jpg HTTP/1.1
1308	20.388173	127.0.0.1	127.0.0.1	TCP	56	8000 → 54518 [ACK] Seq=827 Ack=1236 Win=2159616 Len=0 TSval=11345024 TSecr=11345024
1309	20.388882	127.0.0.1	127.0.0.1	HTTP	29316	HTTP/1.1 200 OK (JPEG JFIF image)
1310	20.389067	127.0.0.1	127.0.0.1	TCP	56	54518 → 8000 [ACK] Seq=1236 Ack=30087 Win=2130688 Len=0 TSval=11345024 TSecr=11345024
1311	20.408878	127.0.0.1	127.0.0.1	HTTP	655	GET /logo.ico HTTP/1.1
1312	20.408909	127.0.0.1	127.0.0.1	TCP	56	8000 → 54518 [ACK] Seq=30087 Ack=1835 Win=2159104 Len=0 TSval=11345044 TSecr=11345044
1313	20.409257	127.0.0.1	127.0.0.1	TCP	65539	8000 → 54518 [ACK] Seq=30087 Ack=1835 Win=2159104 Len=65483 TSval=11345045 TSecr=11345044 [TCP segment of a reassembled PDU]
1314	20.409295	127.0.0.1	127.0.0.1	TCP	65539	8000 → 54518 [ACK] Seq=95570 Ack=1835 Win=2159104 Len=65483 TSval=11345045 TSecr=11345044 [TCP segment of a reassembled PDU]
1315	20.409340	127.0.0.1	127.0.0.1	HTTP	7131	HTTP/1.1 200 OK (image/x-icon)
1316	20.409416	127.0.0.1	127.0.0.1	TCP	56	54518 → 8000 [ACK] Seq=1835 Ack=168128 Win=2153728 Len=0 TSval=11345045 TSecr=11345045
1373	25.422658	127.0.0.1	127.0.0.1	TCP	56	8000 → 54518 [FIN, ACK] Seq=168128 Ack=1835 Win=2159104 Len=0 TSval=11350058 TSecr=11345045
1374	25.422702	127.0.0.1	127.0.0.1	TCP	56	54518 → 8000 [ACK] Seq=1835 Ack=168129 Win=2153728 Len=0 TSval=11350058 TSecr=11350058
1531	32.636411	127.0.0.1	127.0.0.1	TCP	56	54518 → 8000 [FIN, ACK] Seq=1835 Ack=168129 Win=2153728 Len=0 TSval=11357272 TSecr=11350058
1534	32.636436	127.0.0.1	127.0.0.1	TCP	56	8000 → 54518 [ACK] Seq=168129 Ack=1836 Win=2159104 Len=0 TSval=11357272 TSecr=11357272

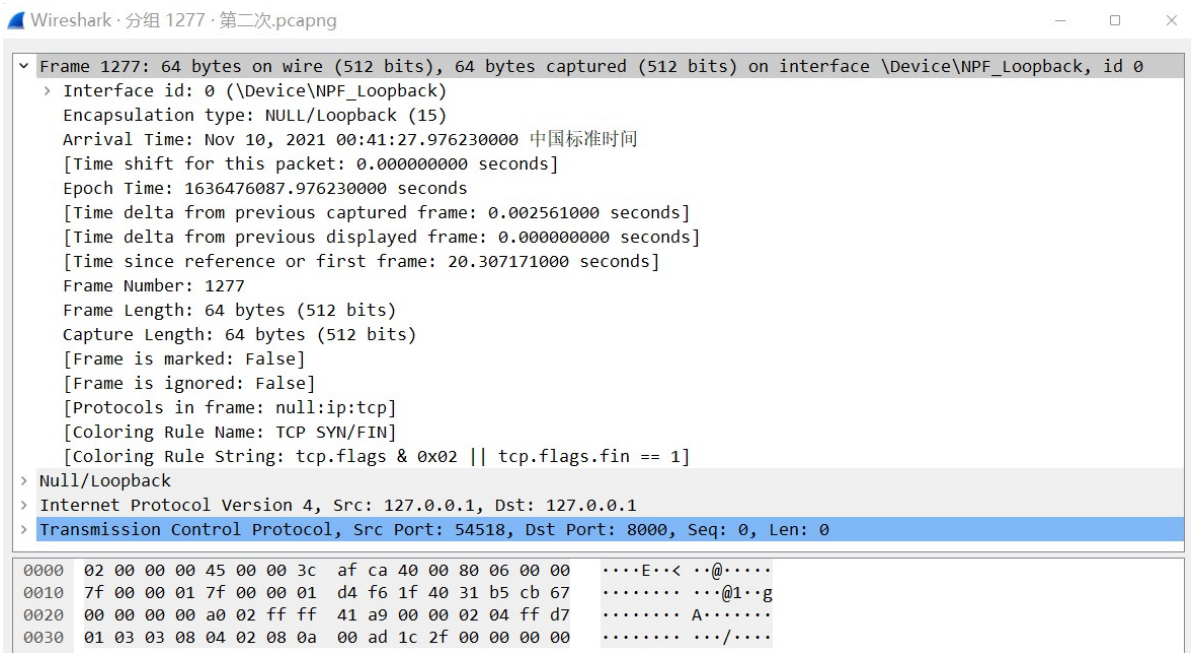
（二）报文

1. TCP报文

报文分为四部分

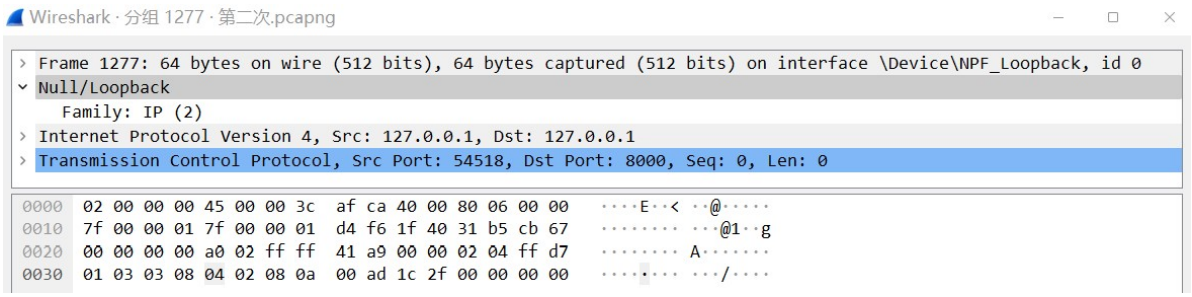


（一）物理层数据帧概况



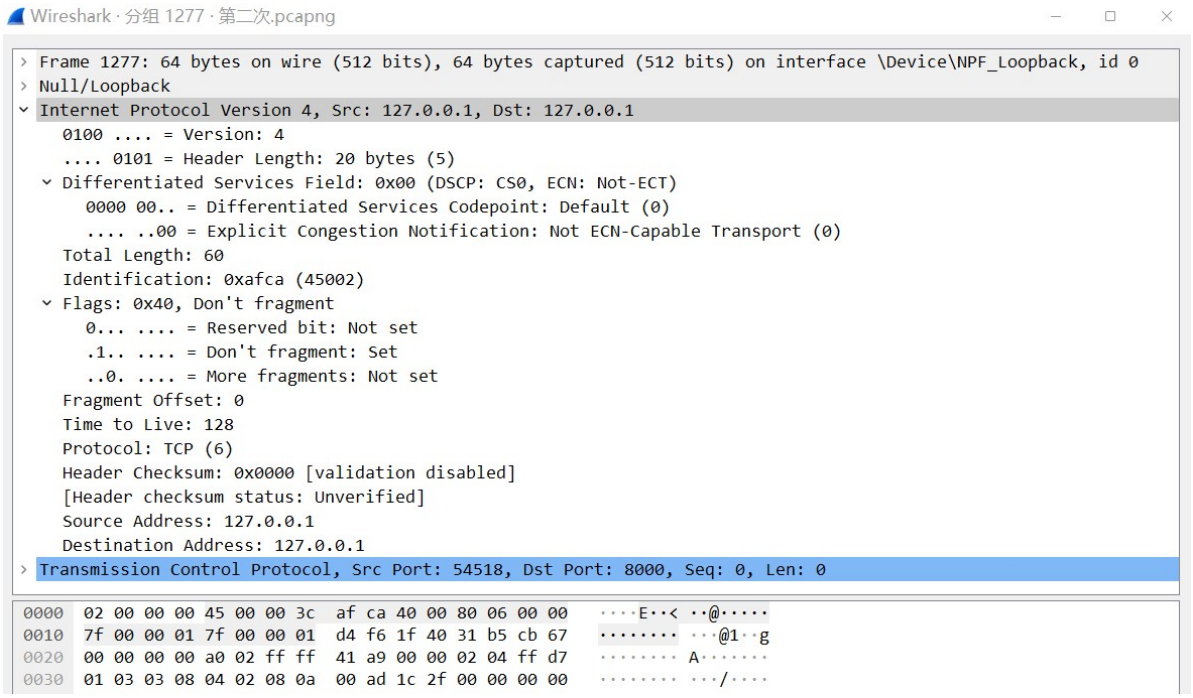
（二）数据层头部信息

由于是本地回环，所以头部并没有什么信息。



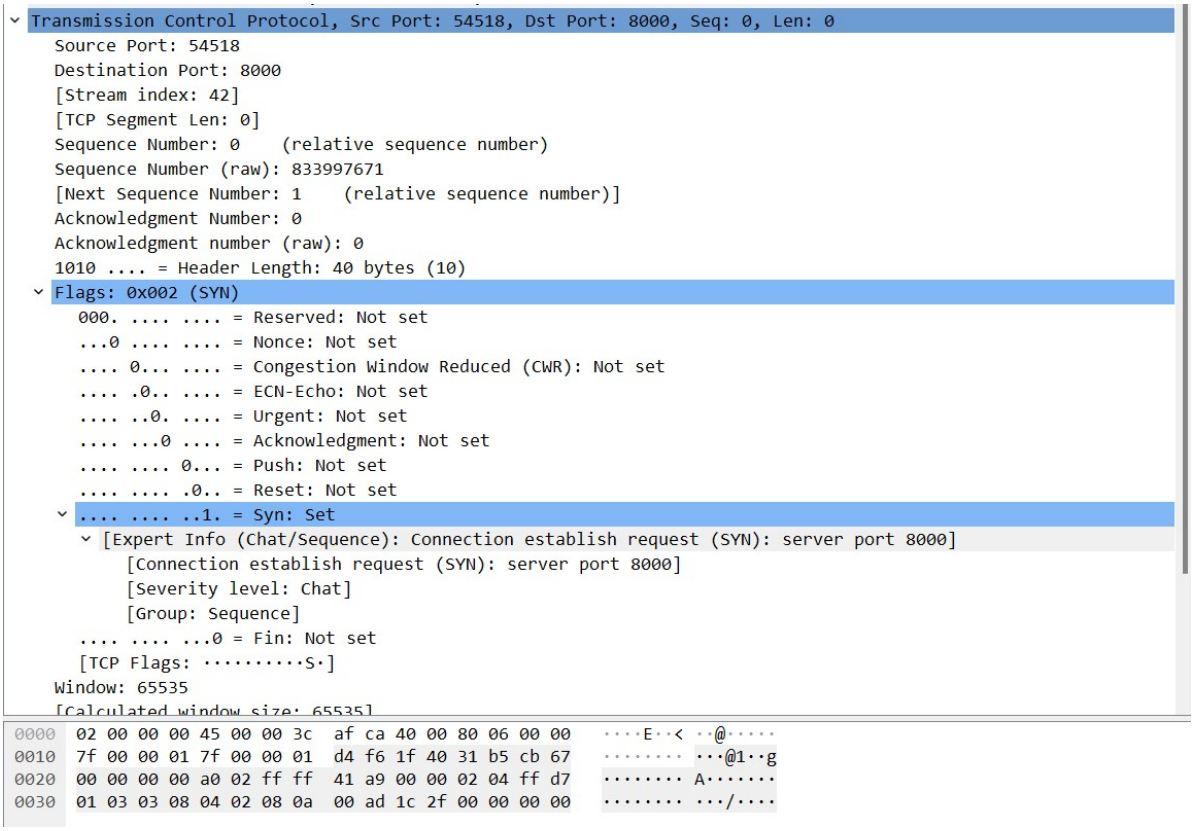
（三）互联网层IP包头部信息

> 这部分信息包含：
>
> 指明为IPV4协议，包头长度为20bytes，IP包总长度为60，标识字段45002，标记字段为0x40，生存期为128，包内封装的上层协议为TCP，头部校验和，源IP地址（127.0.0.1），目的IP地址（127.0.0.1）。



（四）传输层数据段头部信息

> 包含信息：源端口，目的端口，标志为SYN（发送SYN报文到服务器），seq=0（此处分析的是第一次握手）。

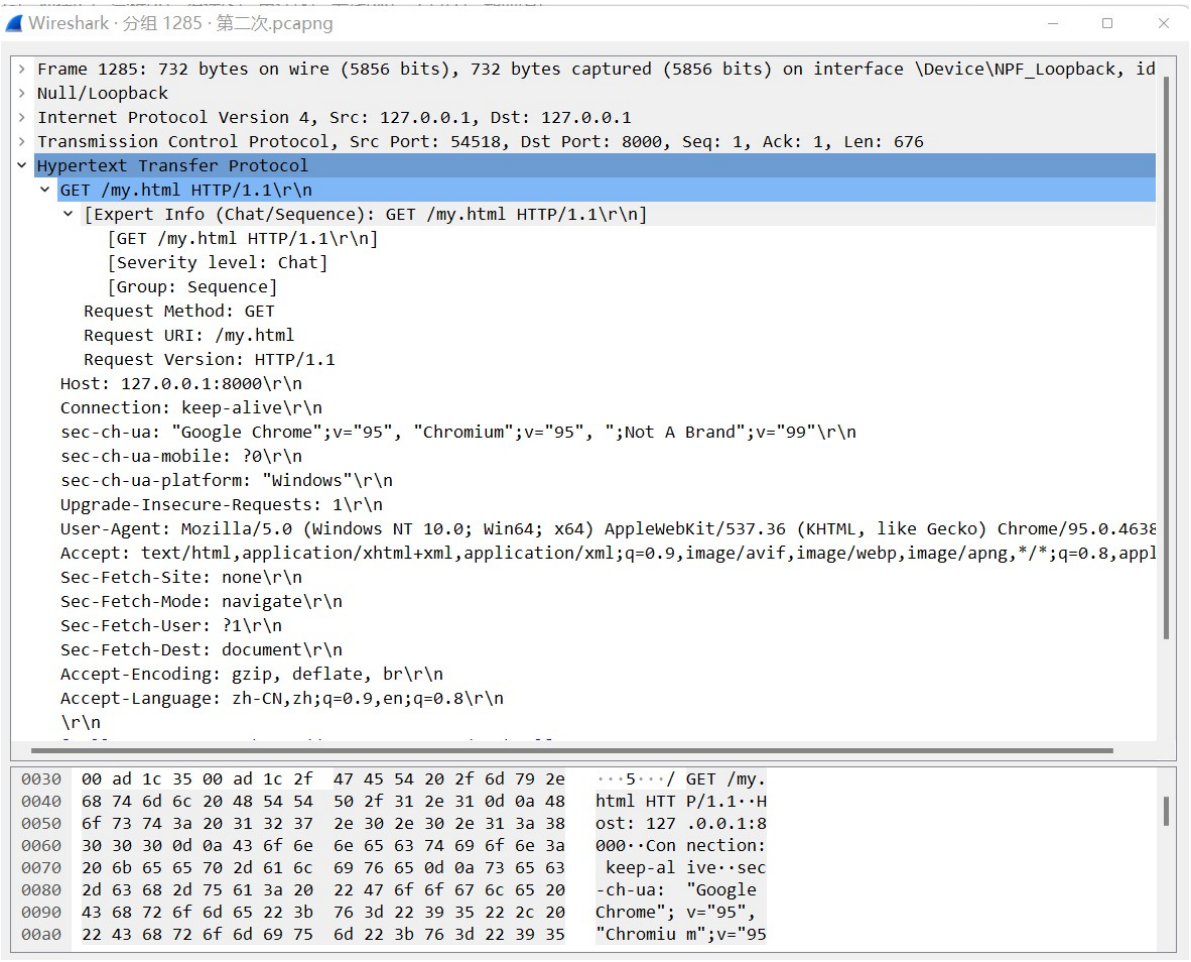


2. HTTP报文

2.1 HTTP请求报文

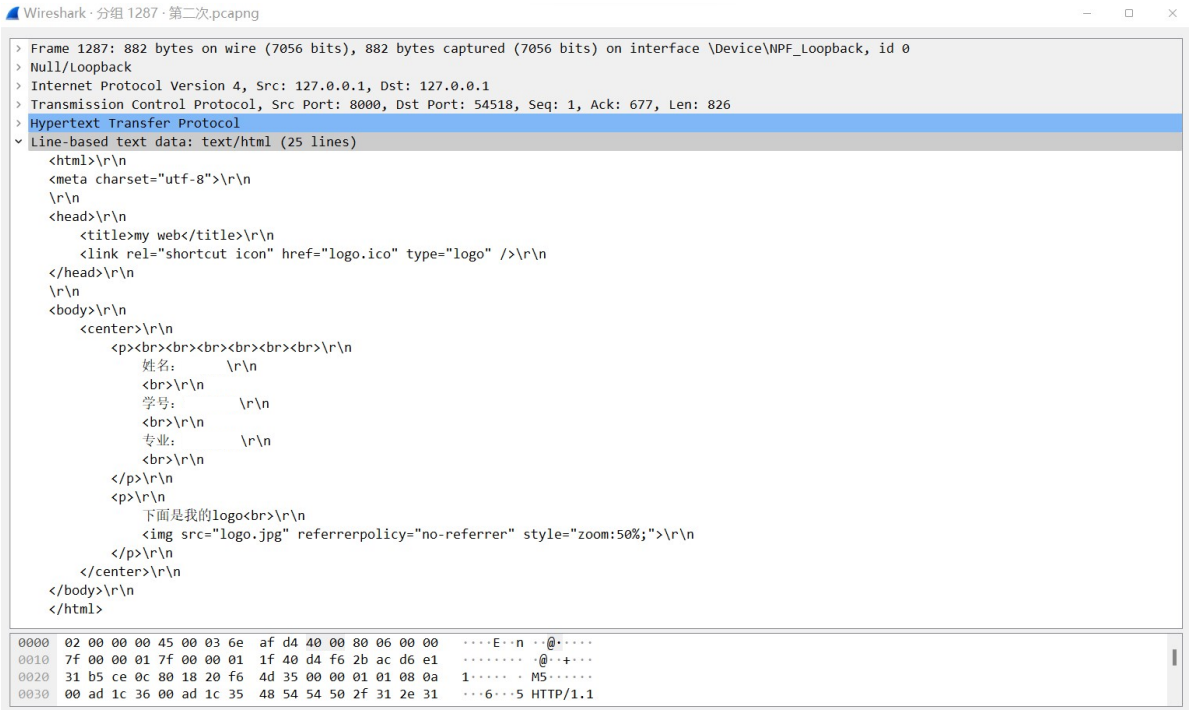
其消息格式在原有的TCP基础上增加了差文本传输协议部分。

> 请求行指明方法为GET，URI为my.html，HTTP版本为HTTP1.1
>
> 指明请求源host为127.0.0.1:8000，连接类型，一些Cookie信息



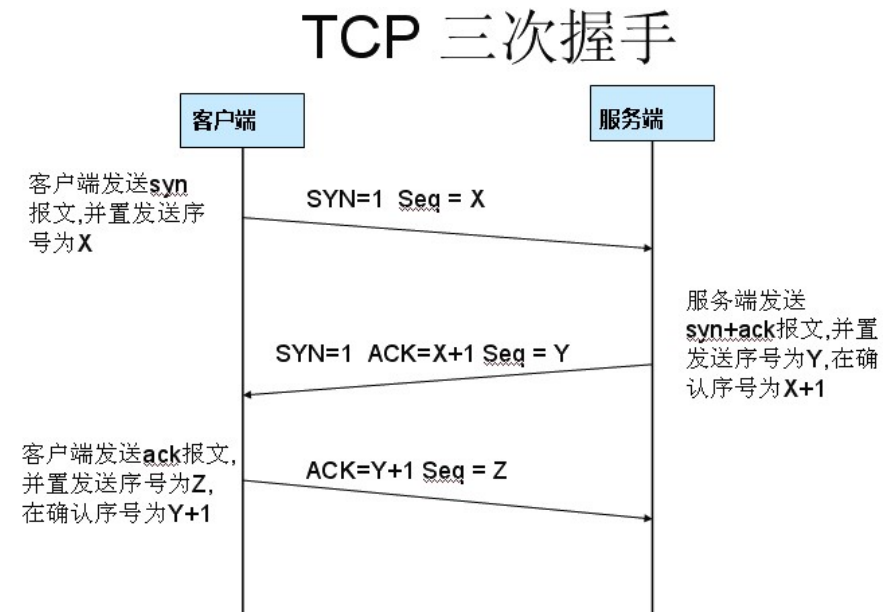
2.1 HTTP响应报文

响应报文的消息格式又在HTTP请求报文中增加了响应体，包含请求需要得到的数据，这里是HTML文件内容。



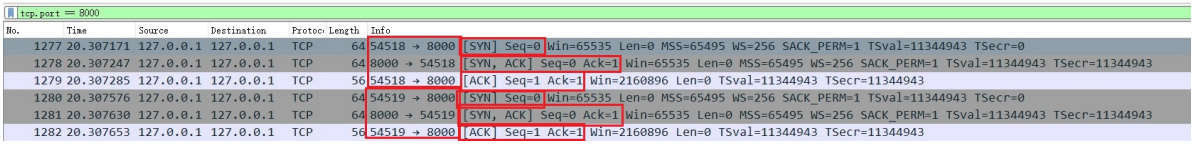
（三）三次握手和四次挥手

1. 三次握手

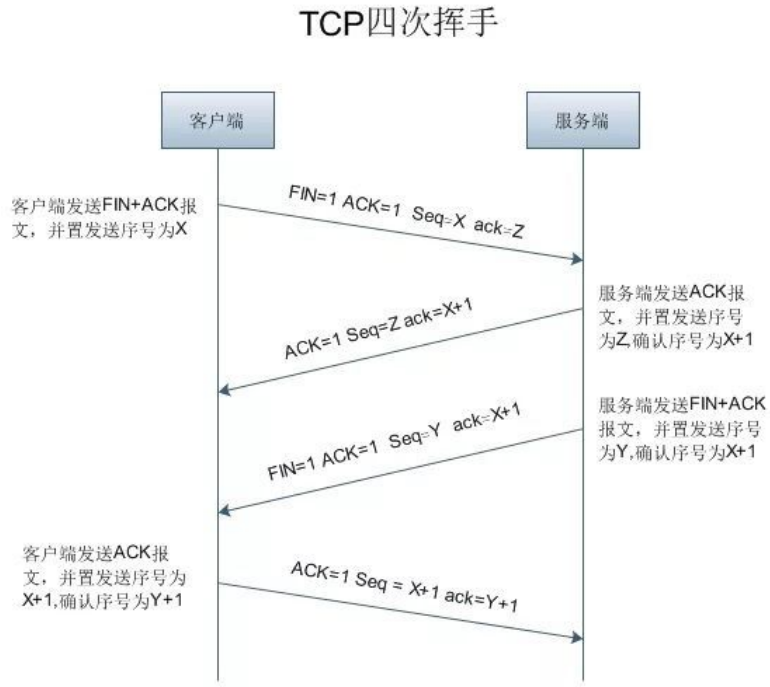


从下图中的几条记录可以看到

对于每一个端口，1. 由客户端向服务器发送连接建立请求SYN，2. 服务器向客户端回复SYN消息并携带确认消息ACK，3. 客户端收到服务器的回复并再次向服务器发送ACK确认。



2. 四次挥手



从下图中几条记录中可以看到

对于每个端口，1. 客户端向服务器发送连接断开请求，2. 服务器回复确认，3. 服务器关闭与客户端的连接，并发送FIN和ACK，4. 客户端收到消息，回复确认

```
1373 25.422658 127.0.0.1 127.0.0.1   TCP      56 8000 → 54518 [FIN, ACK] Seq=168128 Ack=1835 Win=2159104 Len=0 TSval=11350058 TSecr=11345045
1374 25.422702 127.0.0.1 127.0.0.1   TCP      56 54518 → 8000 [ACK] Seq=1835 Ack=168129 Win=2153728 Len=0 TSval=11350058 TSecr=11350058
1528 32.636323 127.0.0.1 127.0.0.1   TCP      56 54519 → 8000 [FIN, ACK] Seq=1 Ack=1 Win=2160896 Len=0 TSval=11357272 TSecr=11344943
1529 32.636367 127.0.0.1 127.0.0.1   TCP      56 8000 → 54519 [ACK] Seq=1 Ack=2 Win=2160896 Len=0 TSval=11357272 TSecr=11357272
1530 32.636392 127.0.0.1 127.0.0.1   TCP      56 8000 → 54519 [FIN, ACK] Seq=1 Ack=2 Win=2160896 Len=0 TSval=11357272 TSecr=11357272
1531 32.636411 127.0.0.1 127.0.0.1   TCP      56 54518 → 8000 [FIN, ACK] Seq=1835 Ack=168129 Win=2153728 Len=0 TSval=11357272 TSecr=11350058
1533 32.636421 127.0.0.1 127.0.0.1   TCP      56 54519 → 8000 [ACK] Seq=2 Ack=2 Win=2160896 Len=0 TSval=11357272 TSecr=11357272
1534 32.636436 127.0.0.1 127.0.0.1   TCP      56 8000 → 54518 [ACK] Seq=168129 Ack=1836 Win=2159104 Len=0 TSval=11357272 TSecr=11357272
```

（四）网页信息

由于HTTP协议版本为HTTP1.1，所以在一次连接中可以传送多个请求和响应，多个请求可以重叠和同时进行。

请求和响应

1. 客户端向服务器发送HTTP请求报文
2. 服务器收到客户端的请求并向客户端发送ACK确认TCP报文
3. 服务器向客户端发送HTTP响应报文
4. 客户端向服务器发送ACK确认TCP报文

图中可以看到对于ico的请求，服务器发送了三个ACK确认，其中[TCP segment of a reassembled PDU]表明这三个确认都是针对客户端对于ico请求报文的确认。

```
1285 20.313882 127.0.0.1 127.0.0.1   HTTP    732 GET /my.html HTTP/1.1
1286 20.313950 127.0.0.1 127.0.0.1   TCP      56 8000 → 54518 [ACK] Seq=1 Ack=677 Win=2160128 Len=0 TSval=11344949 TSecr=11344949
1287 20.314397 127.0.0.1 127.0.0.1   HTTP    882 HTTP/1.1 200 OK  (text/html)
1288 20.314425 127.0.0.1 127.0.0.1   TCP      56 54518 → 8000 [ACK] Seq=677 Ack=827 Win=2159872 Len=0 TSval=11344950 TSecr=11344950
1307 20.388120 127.0.0.1 127.0.0.1   HTTP    615 GET /logo.jpg HTTP/1.1
1308 20.388173 127.0.0.1 127.0.0.1   TCP      56 8000 → 54518 [ACK] Seq=827 Ack=1236 Win=2159616 Len=0 TSval=11345024 TSecr=11345024
1309 20.388882 127.0.0.1 127.0.0.1   HTTP  29316 HTTP/1.1 200 OK  (JPEG JFIF image)
1310 20.389067 127.0.0.1 127.0.0.1   TCP      56 54518 → 8000 [ACK] Seq=1236 Ack=30087 Win=2130688 Len=0 TSval=11345024 TSecr=11345024
1311 20.408878 127.0.0.1 127.0.0.1   HTTP    655 GET /logo.ico HTTP/1.1
1312 20.408909 127.0.0.1 127.0.0.1   TCP      56 8000 → 54518 [ACK] Seq=30087 Ack=1835 Win=2159104 Len=0 TSval=11345044 TSecr=11345044
1313 20.409257 127.0.0.1 127.0.0.1   TCP   65539 8000 → 54518 [ACK] Seq=30087 Ack=1835 Win=2159104 Len=65483 TSval=11345045 TSecr=11345044 [TCP segment of a reassembled PDU]
1314 20.409295 127.0.0.1 127.0.0.1   TCP   65539 8000 → 54518 [ACK] Seq=95570 Ack=1835 Win=2159104 Len=65483 TSval=11345045 TSecr=11345044 [TCP segment of a reassembled PDU]
1315 20.409340 127.0.0.1 127.0.0.1   HTTP   7131 HTTP/1.1 200 OK  (image/x-icon)
1316 20.409416 127.0.0.1 127.0.0.1   TCP      56 54518 → 8000 [ACK] Seq=1835 Ack=168128 Win=2153728 Len=0 TSval=11345045 TSecr=11345045
```

请求信息

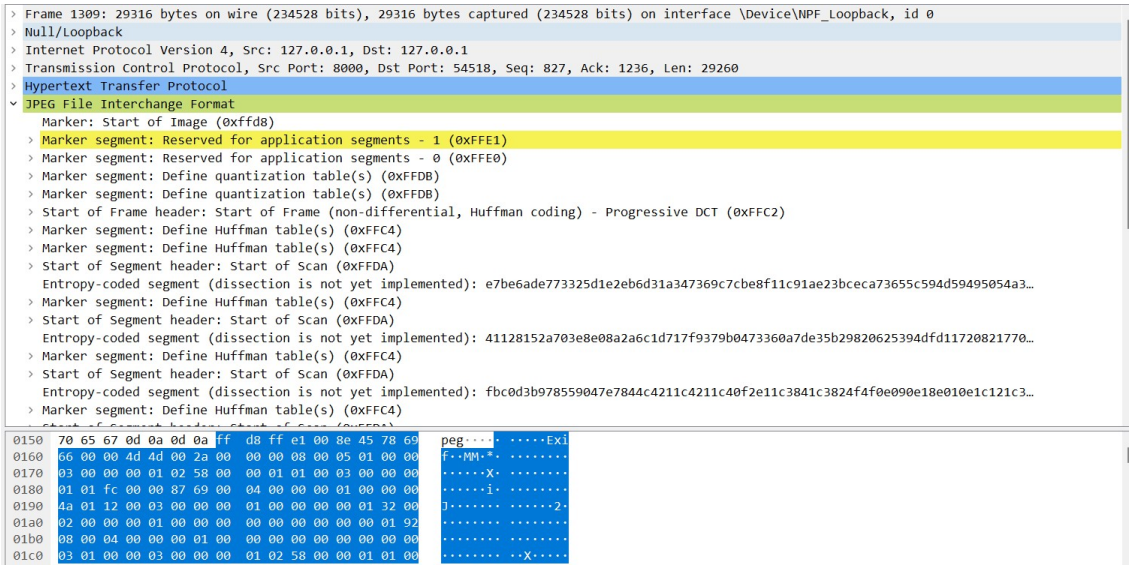
其中HTTP响应报文中包含请求信息

1. **文本信息**

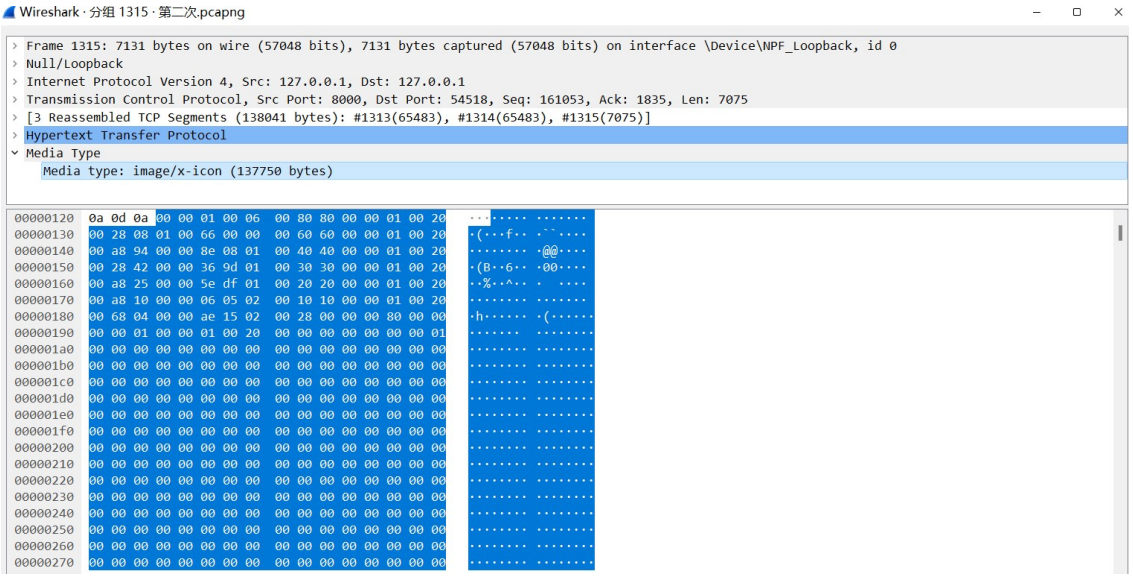
```
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 8000, Dst Port: 54518, Seq: 1, Ack: 677, Len: 826
> Hypertext Transfer Protocol
v Line-based text data: text/html (25 lines)
    <html>\r\n
    <meta charset="utf-8">\r\n
    \r\n
    <head>\r\n
        <title>my web</title>\r\n
        <link rel="shortcut icon" href="logo.ico" type="logo" />\r\n
    </head>\r\n
    \r\n
    <body>\r\n
        <center>\r\n
            <p><br><br><br><br><br><br>\r\n
                姓名:       \r\n
                <br>\r\n
                学号:       \r\n
                <br>\r\n
                专业:       \r\n
                <br>\r\n
            </p>\r\n
            <p>\r\n
                下面是我的logo<br>\r\n
                <img src="logo.jpg" referrerpolicy="no-referrer" style="zoom:50%;">\r\n
            </p>\r\n
        </center>\r\n
    </body>\r\n
```

2. **图片信息**

   下图中十六进制编码即为图片编码


Frame 1309: 29316 bytes on wire (234528 bits), 29316 bytes captured (234528 bits) on interface \Device\NPF_Loopback, id 0
Null/Loopback
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 8000, Dst Port: 54518, Seq: 827, Ack: 1236, Len: 29260
Hypertext Transfer Protocol
JPEG File Interchange Format
Marker: Start of Image (0xffd8)
Marker segment: Reserved for application segments - 1 (0xFFE1)
Marker segment: Reserved for application segments - 0 (0xFFE0)
Marker segment: Define quantization table(s) (0xFFDB)
Marker segment: Define quantization table(s) (0xFFDB)
Start of Frame header: Start of Frame (non-differential, Huffman coding) - Progressive DCT (0xFFC2)
Marker segment: Define Huffman table(s) (0xFFC4)
Marker segment: Define Huffman table(s) (0xFFC4)
Start of Segment header: Start of Scan (0xFFDA)
Entropy-coded segment (dissection is not yet implemented): e7be6ade773325d1e2eb6d31a347369c7cbe8f11c91ae23bceca73655c594d59495054a3…
Marker segment: Define Huffman table(s) (0xFFC4)
Start of Segment header: Start of Scan (0xFFDA)
Entropy-coded segment (dissection is not yet implemented): 41128152a703e8e08a2a6c1d717f9379b0473360a7de35b29820625394dfd11720821770…
Marker segment: Define Huffman table(s) (0xFFC4)
Start of Segment header: Start of Scan (0xFFDA)
Entropy-coded segment (dissection is not yet implemented): fbc0d3b978559047e7844c4211c4211c40f2e11c3841c3824f4f0e090e18e010e1c121c3…
Marker segment: Define Huffman table(s) (0xFFC4)
0150 70 65 67 0d 0a 0d 0a ff d8 ff e1 00 8e 45 78 69 peg· ·····Exi
0160 66 00 00 4d 4d 00 2a 00 00 00 08 00 05 01 00 00 f··MM·*· ····
0170 03 00 00 00 01 02 58 00 00 01 01 00 03 00 00 00 ····· X· ····
0180 01 01 fc 00 00 87 69 00 04 00 00 00 01 00 00 00 ····· i· ····
0190 4a 01 12 00 03 00 00 00 01 00 00 00 00 01 32 00 J··· ····· ····· 2·
01a0 02 00 00 00 01 00 00 00 00 00 00 00 00 00 01 92 ····· ····· ····
01b0 08 00 04 00 00 00 01 00 00 00 00 00 00 00 00 00 ····· ····· ····
01c0 03 01 00 00 03 00 00 00 01 02 58 00 00 01 01 00 ····· ··X· ····

3. icon信息


Wireshark · 分组 1315 · 第二次.pcapng
Frame 1315: 7131 bytes on wire (57048 bits), 7131 bytes captured (57048 bits) on interface \Device\NPF_Loopback, id 0
Null/Loopback
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 8000, Dst Port: 54518, Seq: 161053, Ack: 1835, Len: 7075
[3 Reassembled TCP Segments (138041 bytes): #1313(65483), #1314(65483), #1315(7075)]
Hypertext Transfer Protocol
Media Type
Media type: image/x-icon (137750 bytes)
00000120 0a 0d 0a 00 00 01 00 06 00 80 80 00 00 01 00 20 ··· ····· ·····
00000130 00 28 08 01 00 66 00 00 00 60 60 00 00 01 00 20 ·(···f·· ·`` ····
00000140 00 a8 94 00 00 8e 08 01 00 40 40 00 00 01 00 20 ····· ·@@ ····
00000150 00 28 42 00 00 36 9d 01 00 30 30 00 00 01 00 20 ·(B··6·· ·00 ····
00000160 00 a8 25 00 00 5e df 01 00 20 20 00 00 01 00 20 ··%··^·· · ····
00000170 00 a8 10 00 00 06 05 02 00 10 10 00 00 01 00 20 ····· ····· ·····
00000180 00 68 04 00 00 ae 15 02 00 28 00 00 00 80 00 00 ·h··· ··· ·(·····
00000190 00 00 01 00 00 01 00 20 00 00 00 00 00 00 00 01 ····· ···· ·····
000001a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ····· ····· ·····
000001b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ····· ····· ·····
000001c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ····· ····· ·····
000001d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ····· ····· ·····
000001e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ····· ····· ·····
000001f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ····· ····· ·····
00000200 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ····· ····· ·····
00000210 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ····· ····· ·····
00000220 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ····· ····· ·····
00000230 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ····· ····· ·····
00000240 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ····· ····· ·····
00000250 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ····· ····· ·····
00000260 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ····· ····· ·····
00000270 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ····· ····· ·····