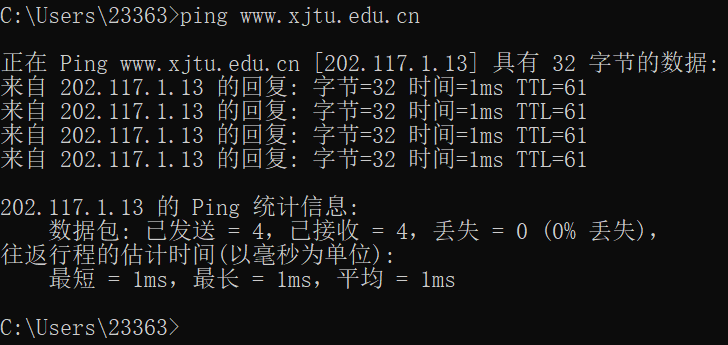
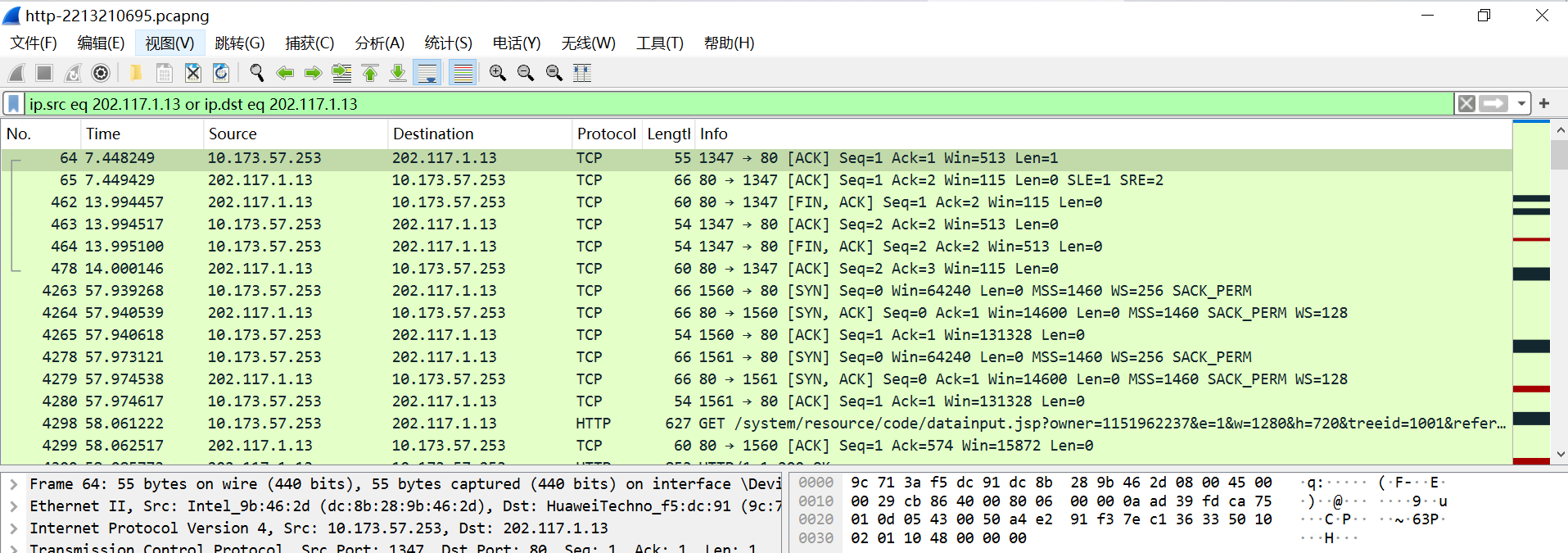
**WIRESHARK抓包**



通过Ping查出www.xjtu.end.cn的ip地址为

www.xjtu.edu.cn 的IP:202.117.1.13

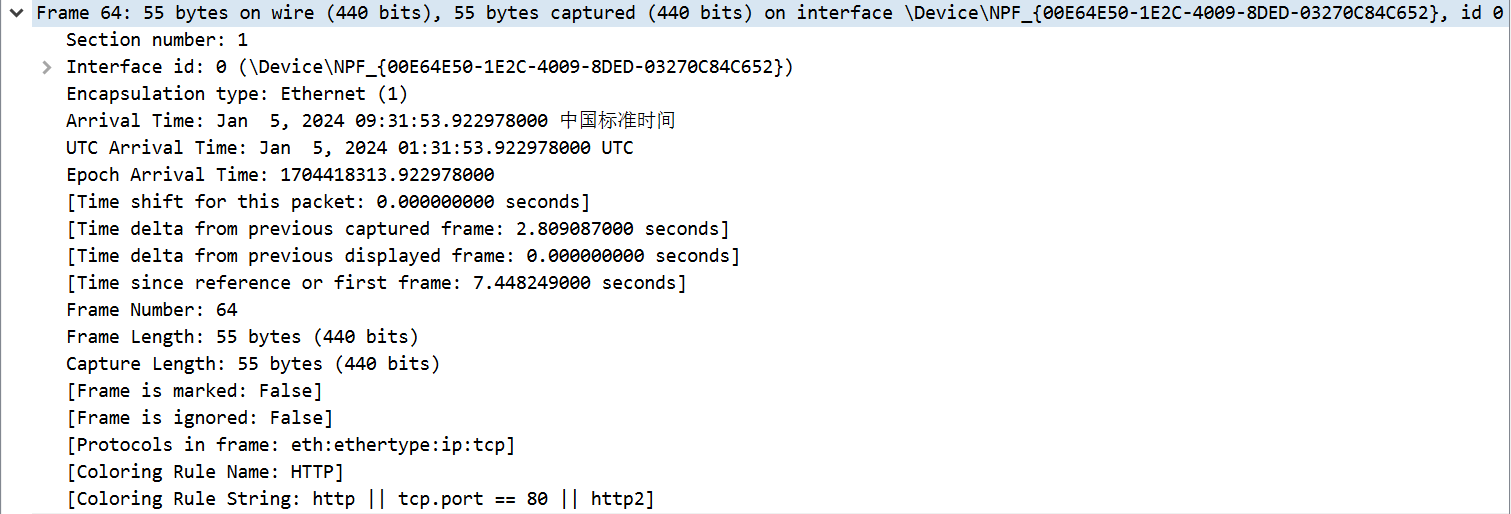
**抓到的包**



**Http分析**

**Http请求报文**

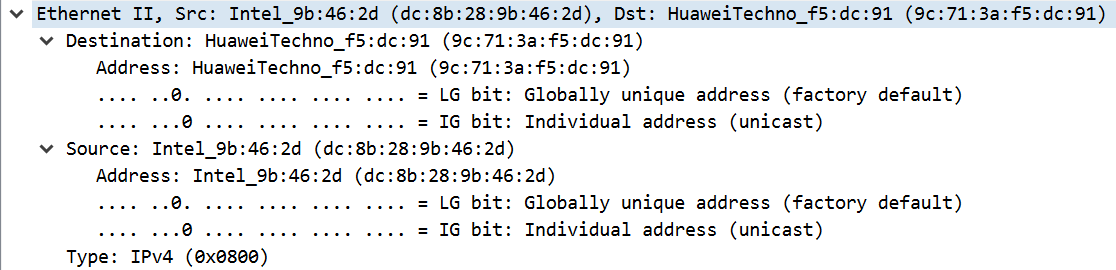
**物理层数据帧详细概要**



可以看出以下信息：

为64号帧，线路字节440bit，实际捕获55bit，端口号为0号端口，端口在计算机中明明为WLAN 2，采用的封装方式为Ethernet(1)，捕获的时间为2024年1月5号，09：31：53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。

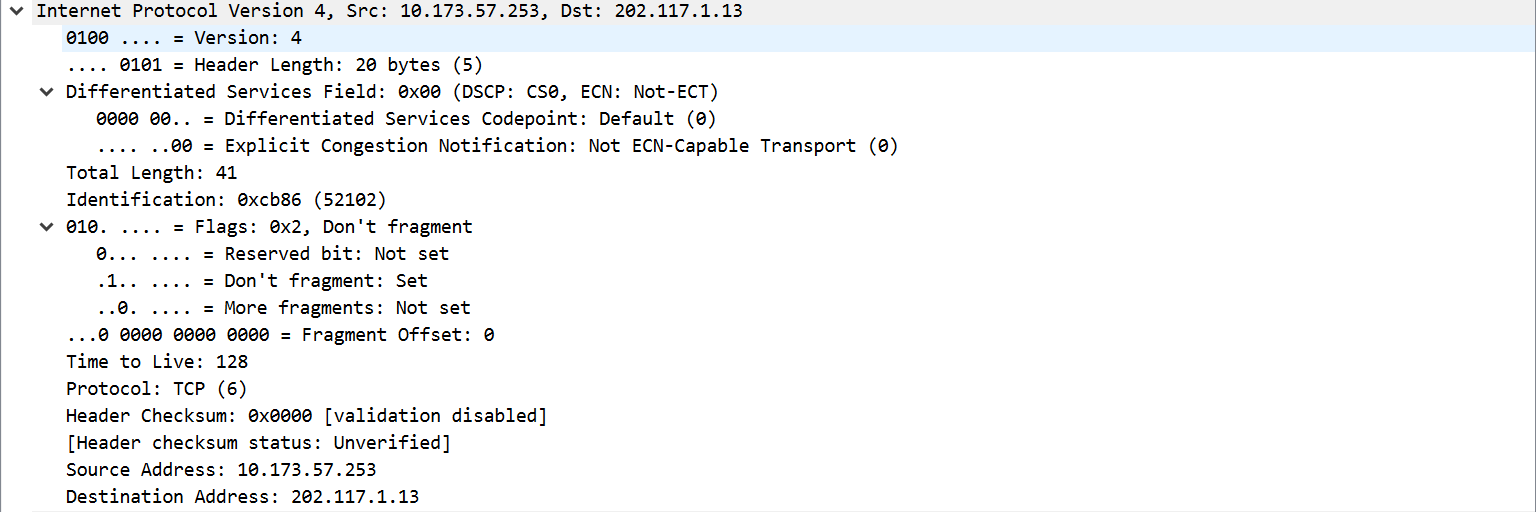
**数据链路层以太帧的首部信息：**



目的地的MAC地址: 9c:71:3a:f5:dc:91

源MAC地址:dc:8b:28:9b:46:2d

**网络层ip包首部信息：**



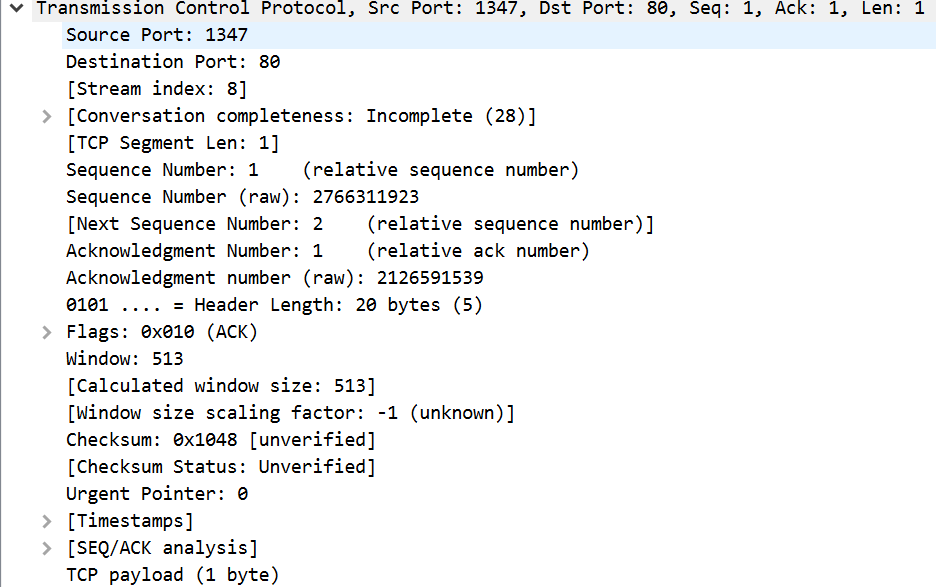
第1行看出使用了IPv4协议

首部长20字节。IP包共长41字节，标记字段为52102，生存周期为128。

源地址为10.173.57.253

目的地址为202.117.1.13

**传输层数据包首部信息**



可以看出以下信息：

源端口号：1347

目的端口号：80

流量控制端口号：513

**应用层分析：**



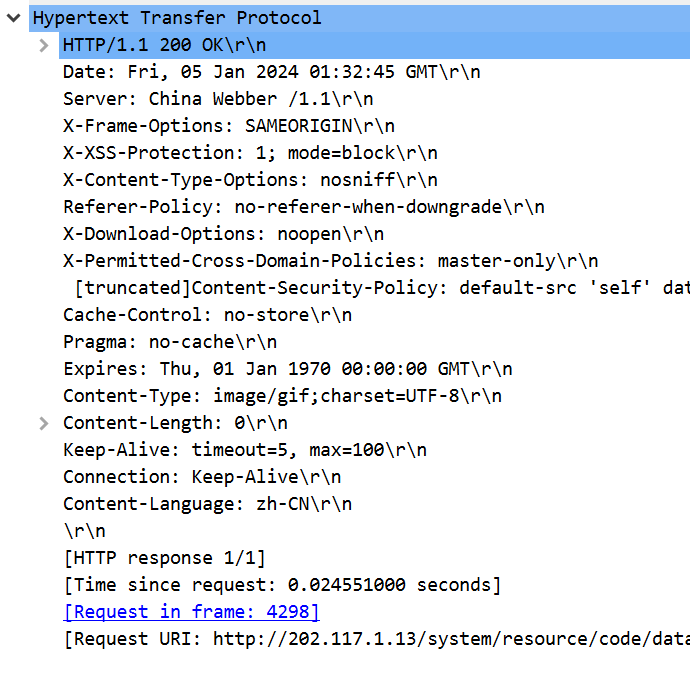
即具体的http报文，

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：**

前面的物理层，数据链路层，网络层，传输层，http请求报文大同小异，直接看应用层http报文。



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：客户端与服务器之间通过TCP（通常是TCP）建立连接。默认的HTTP端口是80，或者使用安全的HTTPS协议时为443。

2. 发送请求：客户端发送一个HTTP请求给服务器。这个请求通常包含以下部分：

请求行：包含请求的方法（GET、POST等）、请求的资源路径和使用的协议版本。

请求头部：包含一系列的键值对，描述客户端的信息、所需的文档类型、支持的压缩算法等信息。

请求主体：仅在使用POST、PUT等方法时存在，包含发送给服务器的数据。

3.处理请求：服务器接收并解析请求，然后根据请求执行相应的操作。这可能涉及到从文件系统中获取文件、查询数据库，或执行其他服务器端的逻辑。

4. 发送响应：服务器生成一个HTTP响应，包含以下部分：

状态行（Status Line）：包含协议版本、状态码和状态消息。

响应头部（Response Headers）：与请求头类似，包含服务器信息、响应的文档类型、日期等信息。

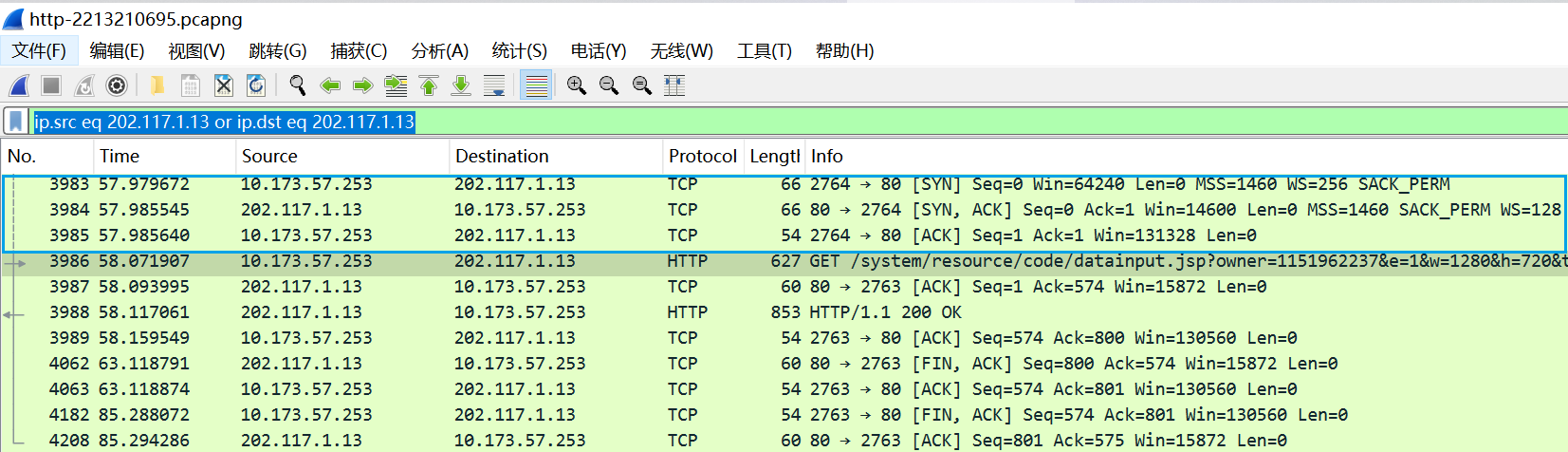
响应主体（Response Body）：包含实际的数据，比如HTML页面、图像、文本等。

5. 传输数据：服务器将响应数据通过建立的TCP连接传输到客户端。

6. 关闭连接：客户端接收到响应后，如果没有指定连接保持活动（HTTP/1.1默认保持活动，除非指定关闭），则客户端或服务器将关闭连接。

**TCP分析:**

**TCP建立连接**



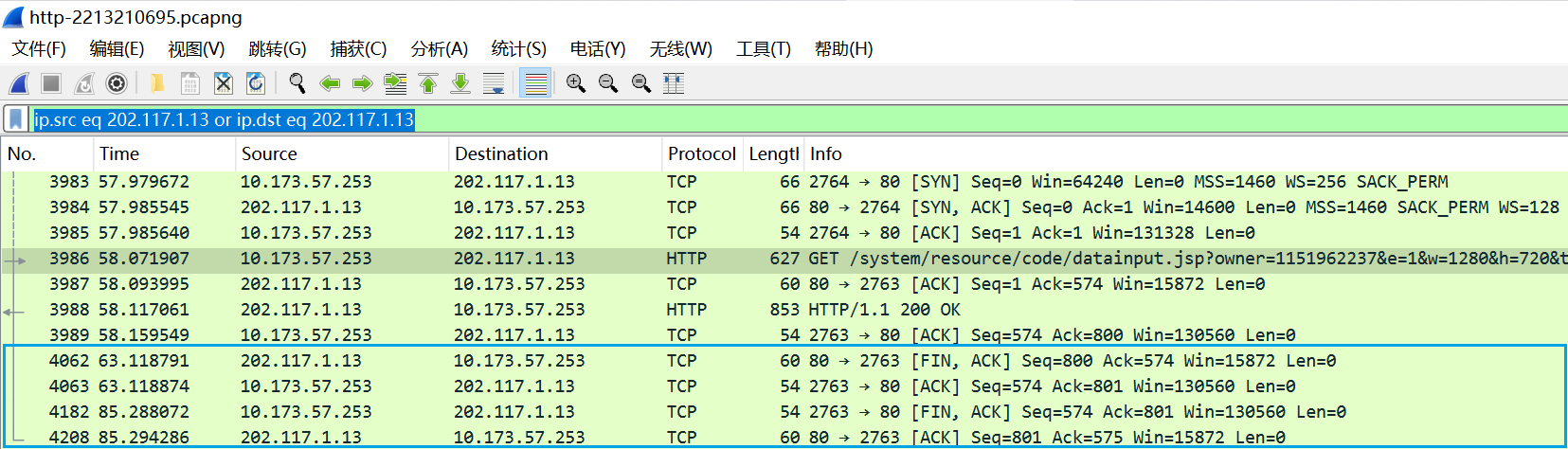
tcp建立时由客户端端口2764向服务器80端口发送第一次握手SYN=1 seq=0

主机端返回第二次握手SYN=1 ACK=1 seq=0 ack=1

客户端收到后第三次握手ACK=1，seq=1，ack=1

至此，tcp成功建立连接

**TCP释放链接**



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。