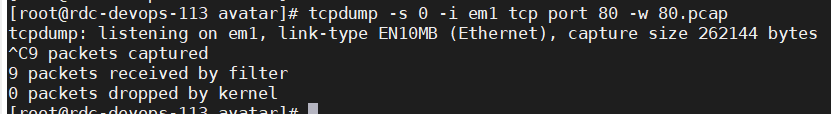
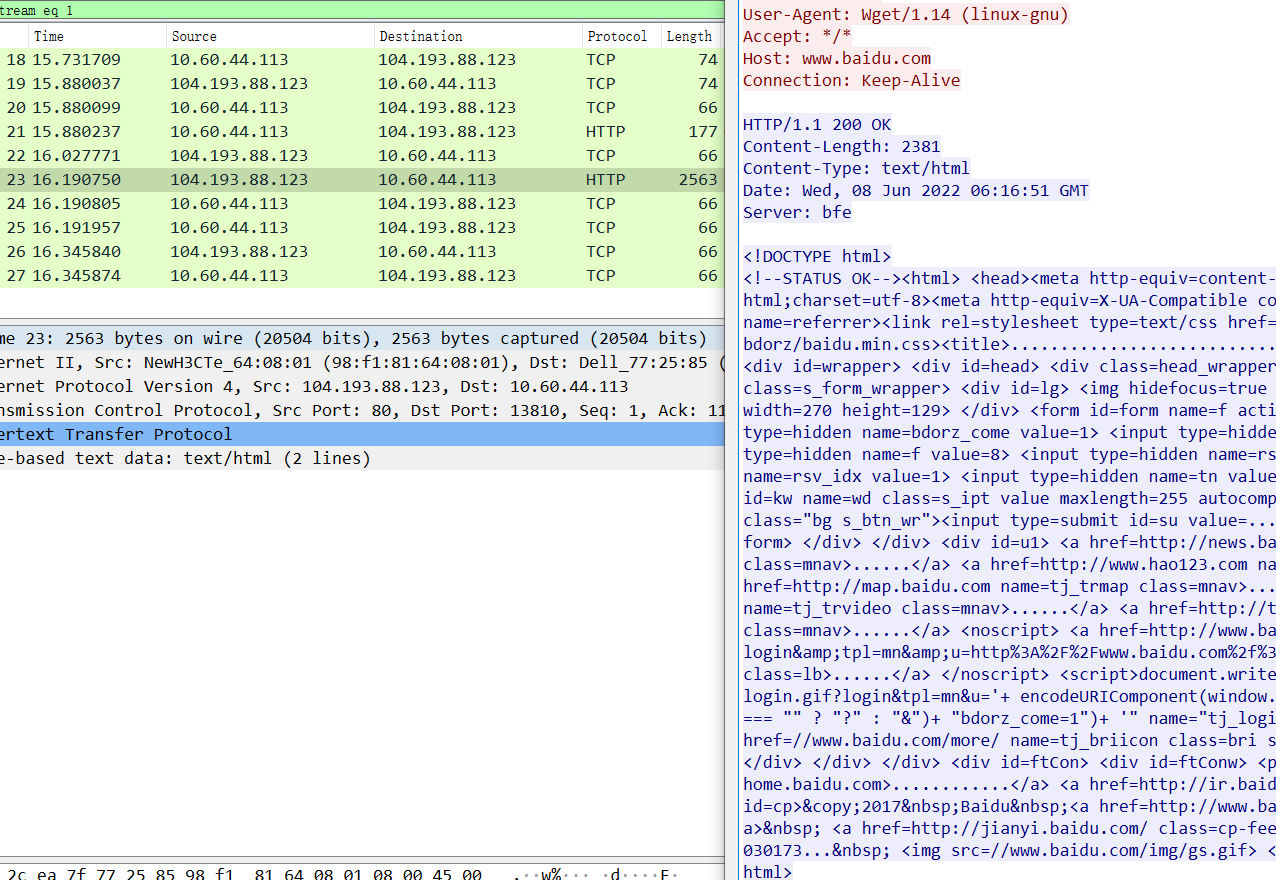
抓包：保存到pcap文件



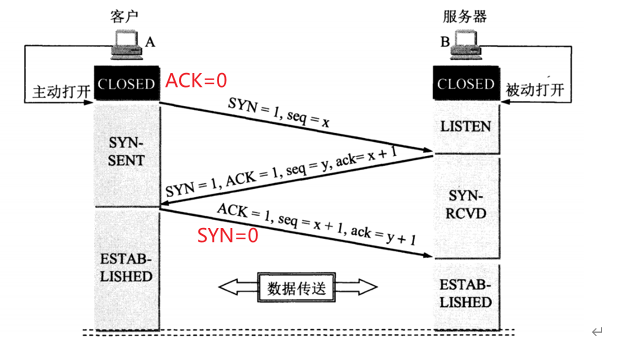
过滤



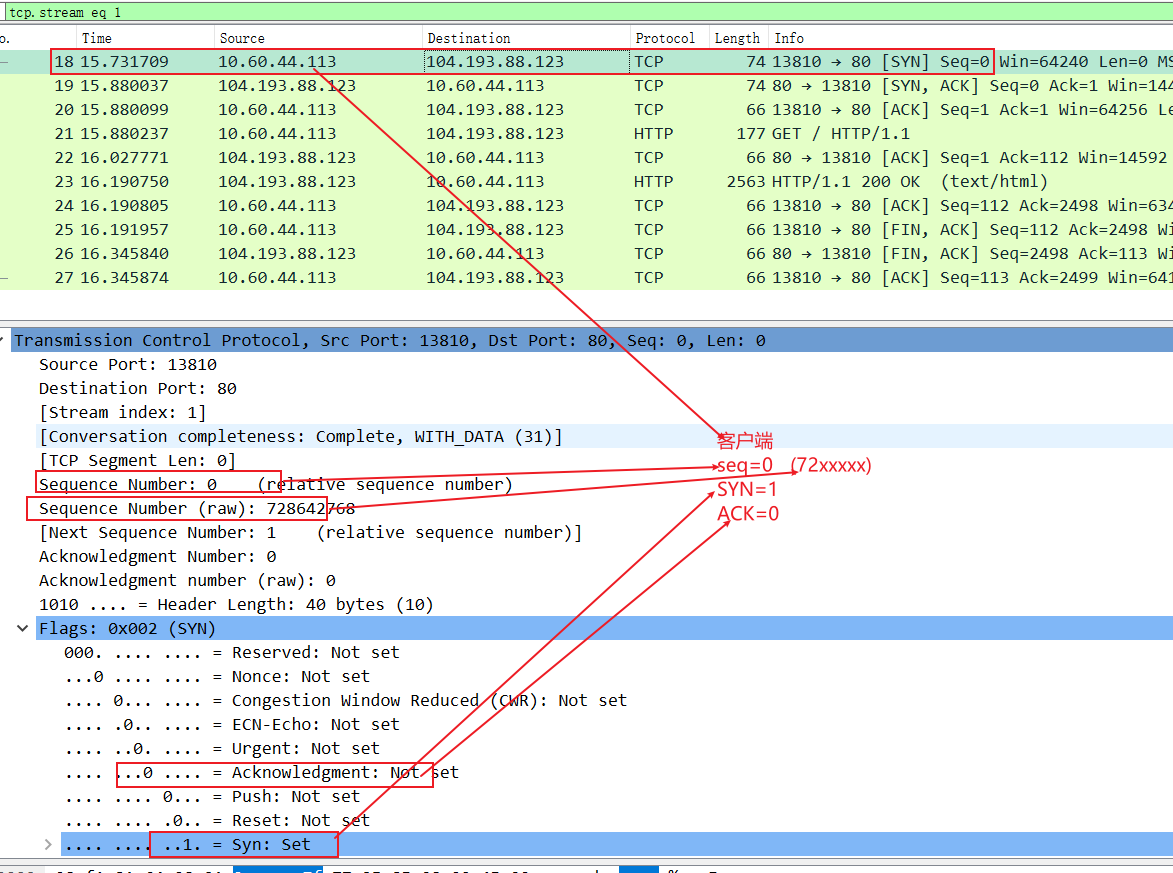
**3次握手**

先复习下

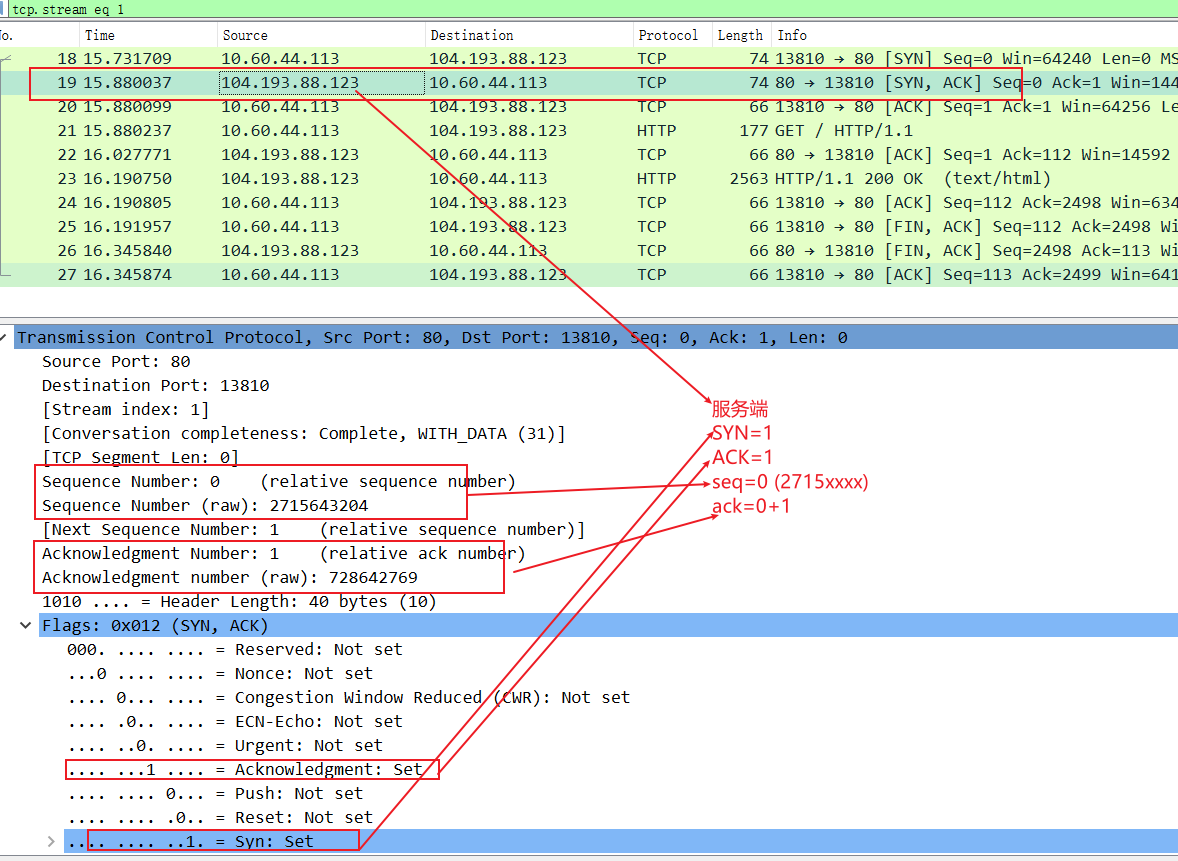
* 序号 ：用于对字节流进行编号，例如序号为 301，表示第一个字节的编号为301，如果携带的数据长度为100字节，那么下一个报文段的序号应为 401。
* 确认号 ：期望收到的下一个报文段的序号。例如 B 正确收到 A 发送来的一个报文段，序号为 501，携带的数据长度为 200 字节，因此 B 期望下一个报文段的序号为701，B 发送给A的确认报文段中确认号就为701。
* 确认 ACK ：当 ACK=1 时确认号字段有效，否则无效。TCP 规定，在连接建立后所有传送的报文段都必须把ACK 置 1。
* 同步 SYN ：在连接建立时用来同步序号。当 SYN=1，ACK=0 时表示这是一个连接请求报文段。若对方同意建立连接，则响应报文中 SYN=1，ACK=1。
* 终止 FIN ：用来释放一个连接，当 FIN=1 时，表示此报文段的发送方的数据已发送完毕，并要求释放连接。



第一次握手：

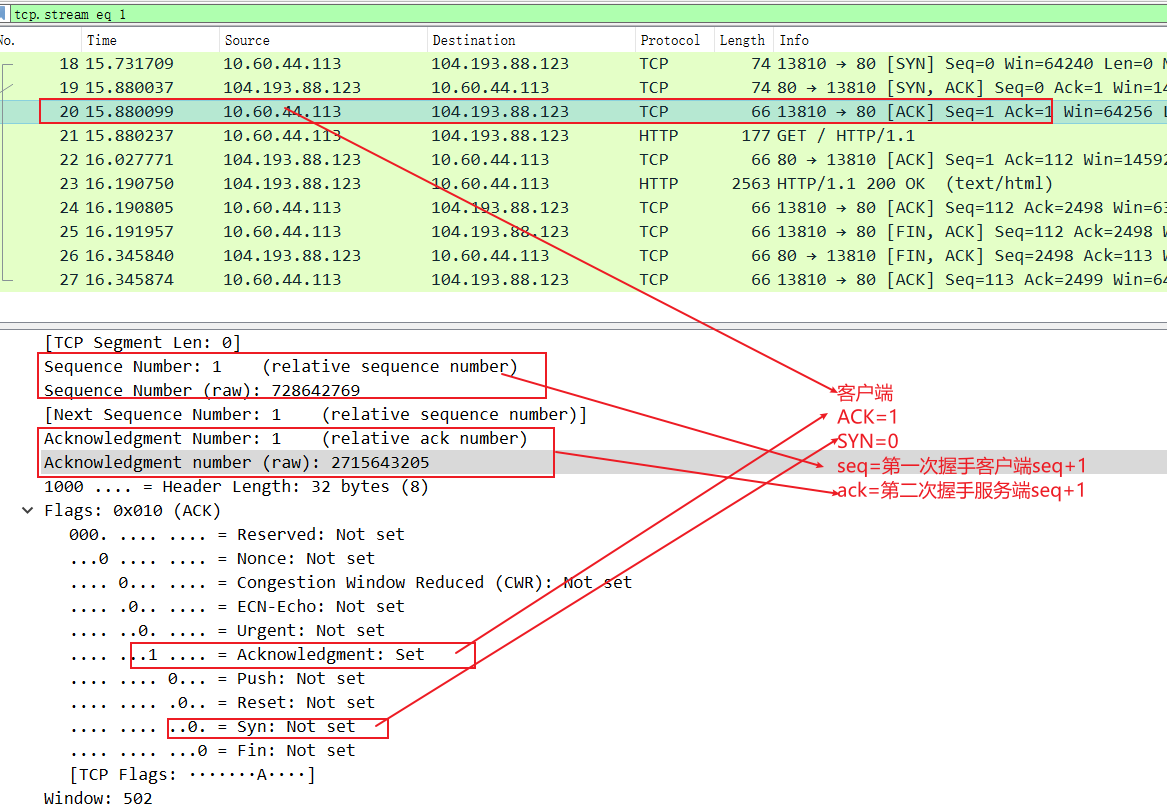


第二次握手

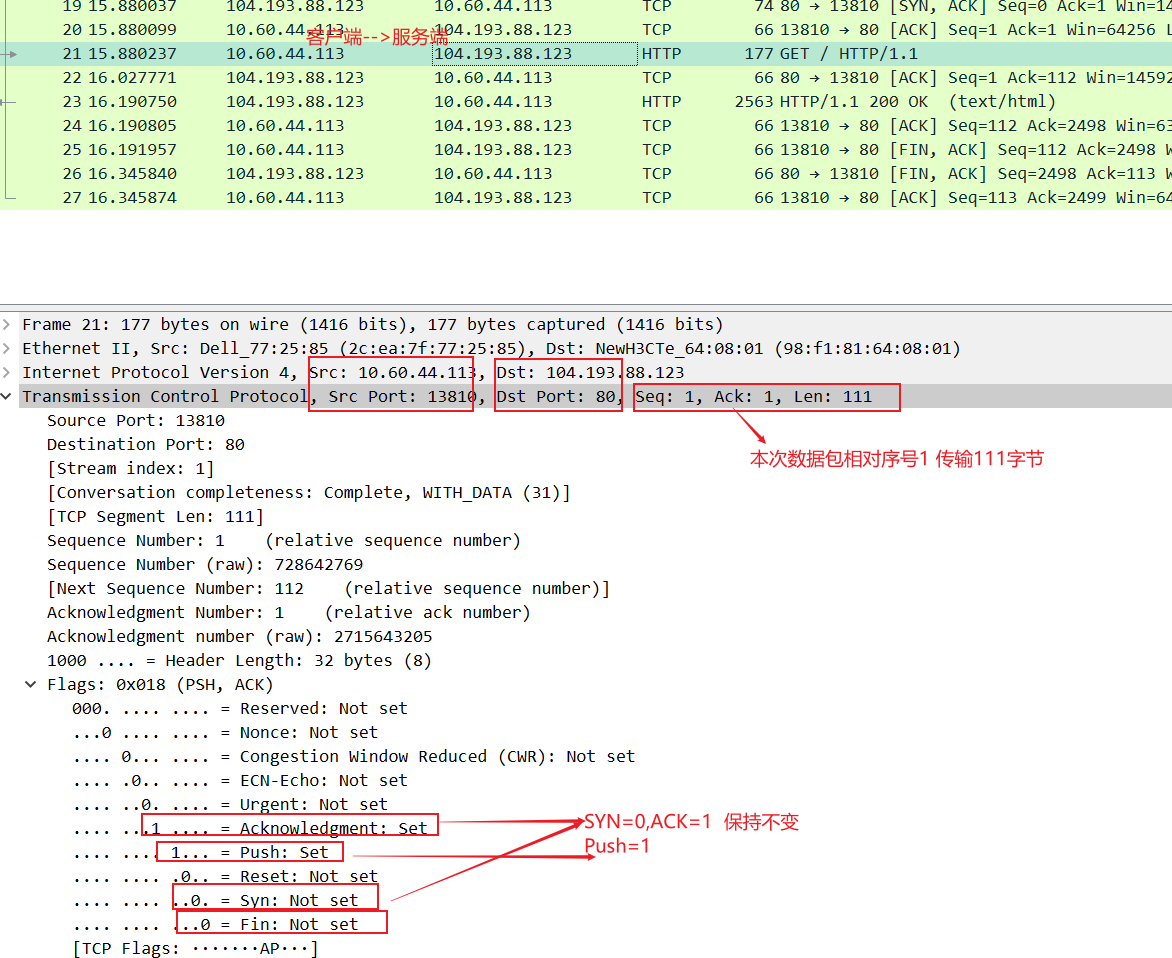


ack = 第一次握手客户端seq+1

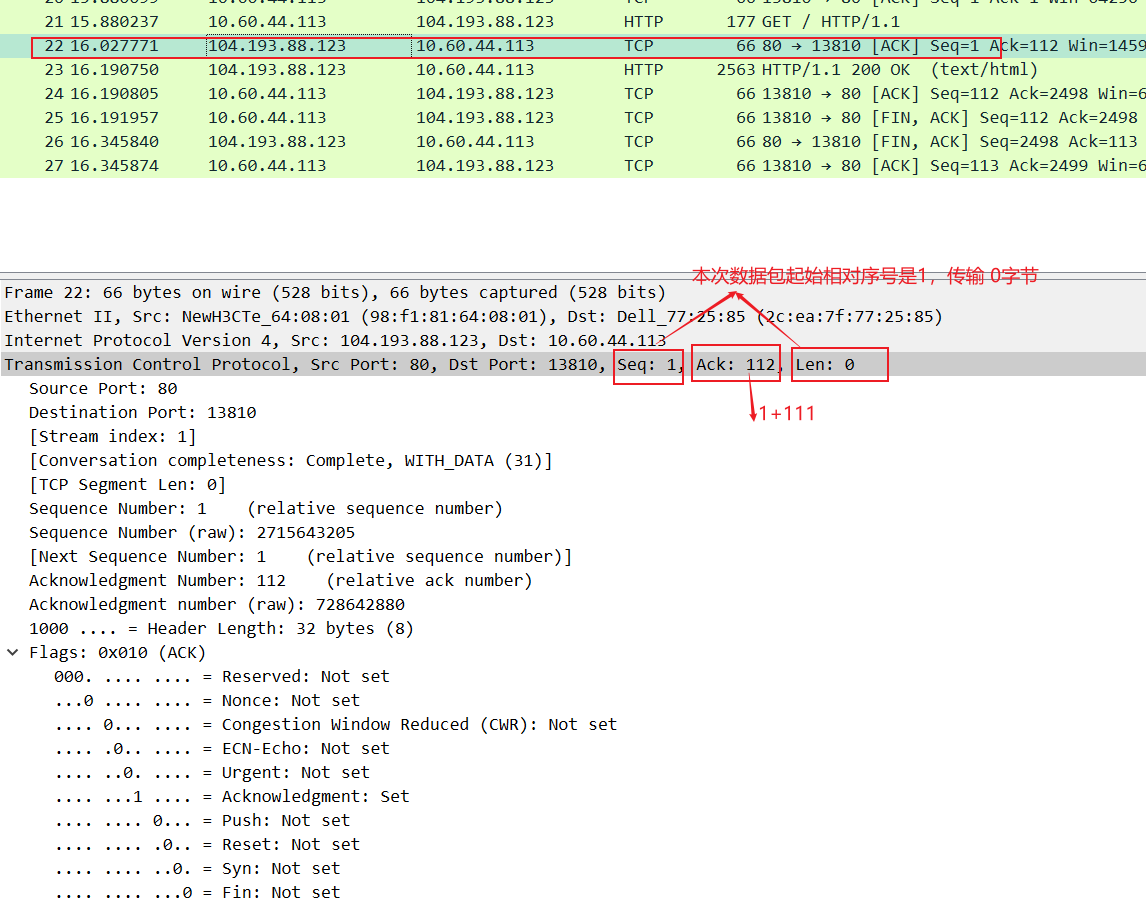
第三次握手



第1次数据交互：



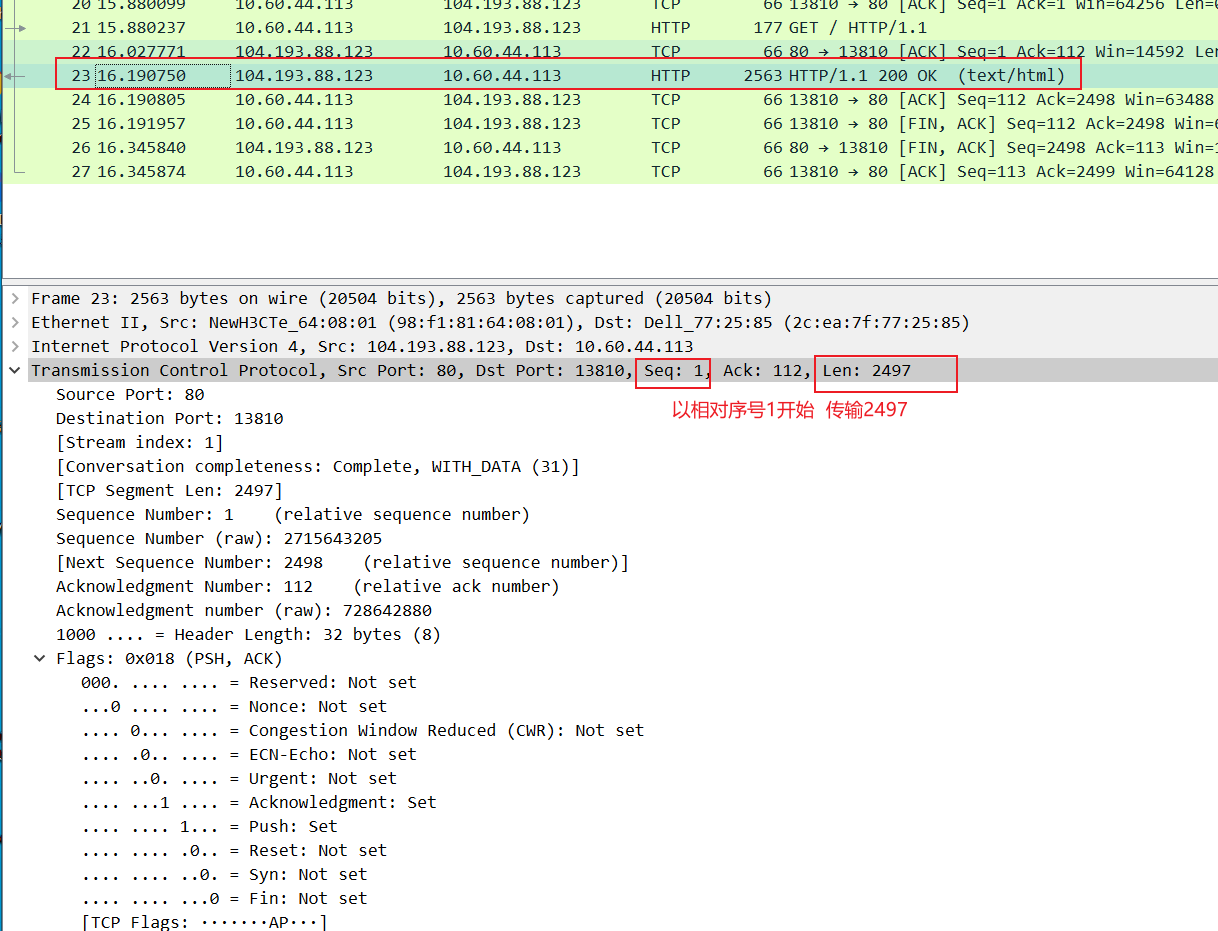
服务器开始返回数据



这里的seq=1 len=0，表示“我只是确认ack，我自己没啥好说的

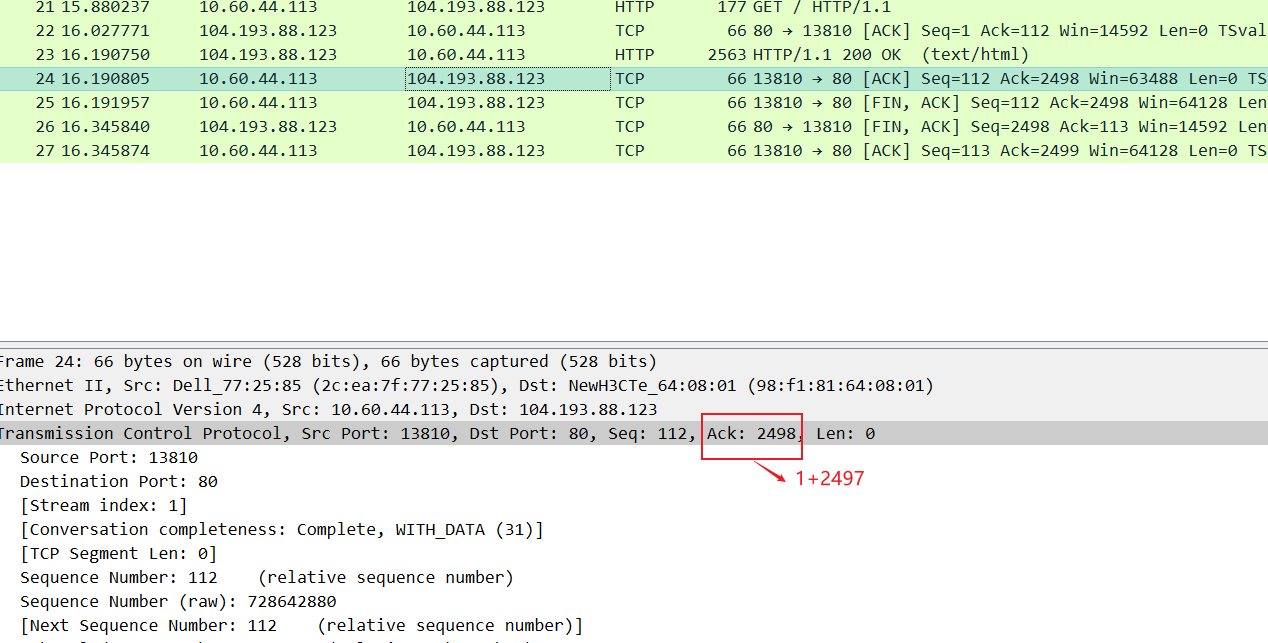
客户端没有的ack响应

服务器继续发送



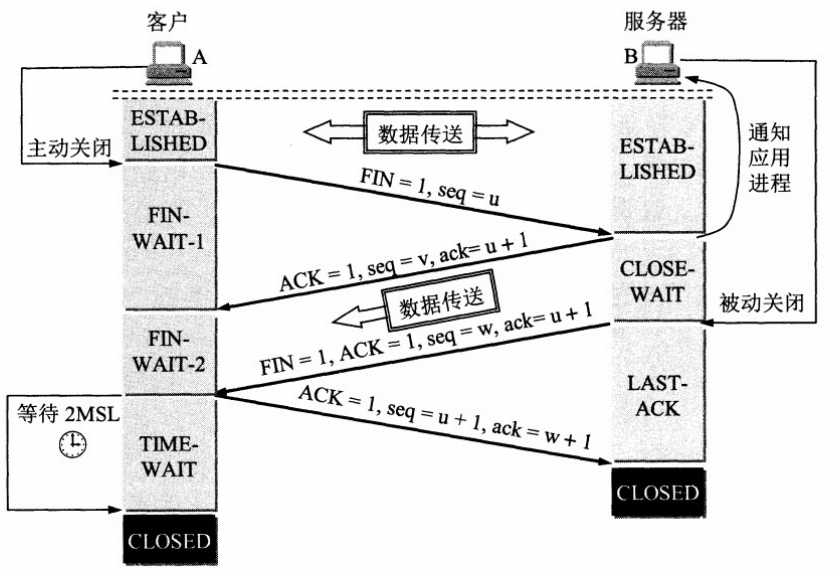
服务端第一次seq=1 len=0 Fin=0 所以第二次返回还是seq=1

客户端的ack

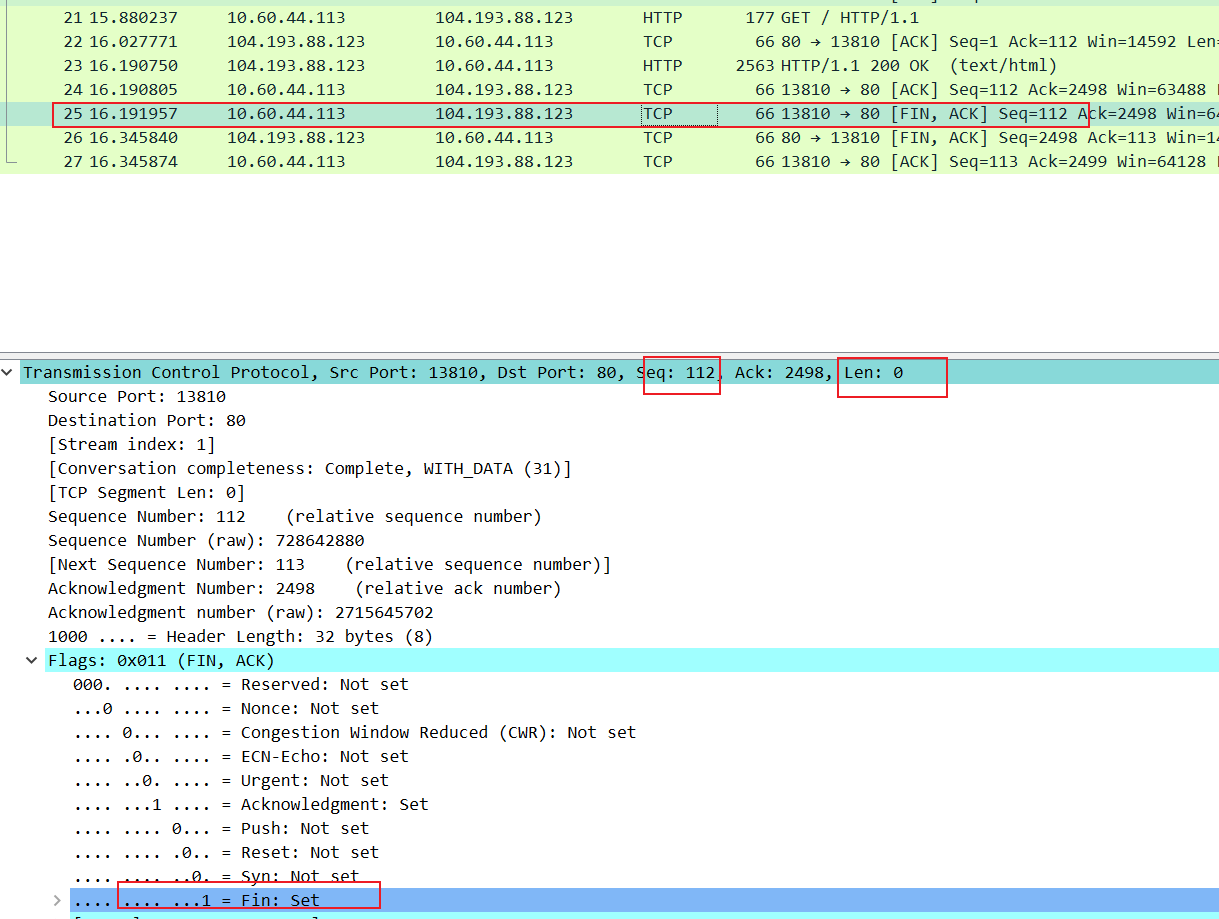


ack是对方的消息的确认，seq+len是本次自己发送的数据

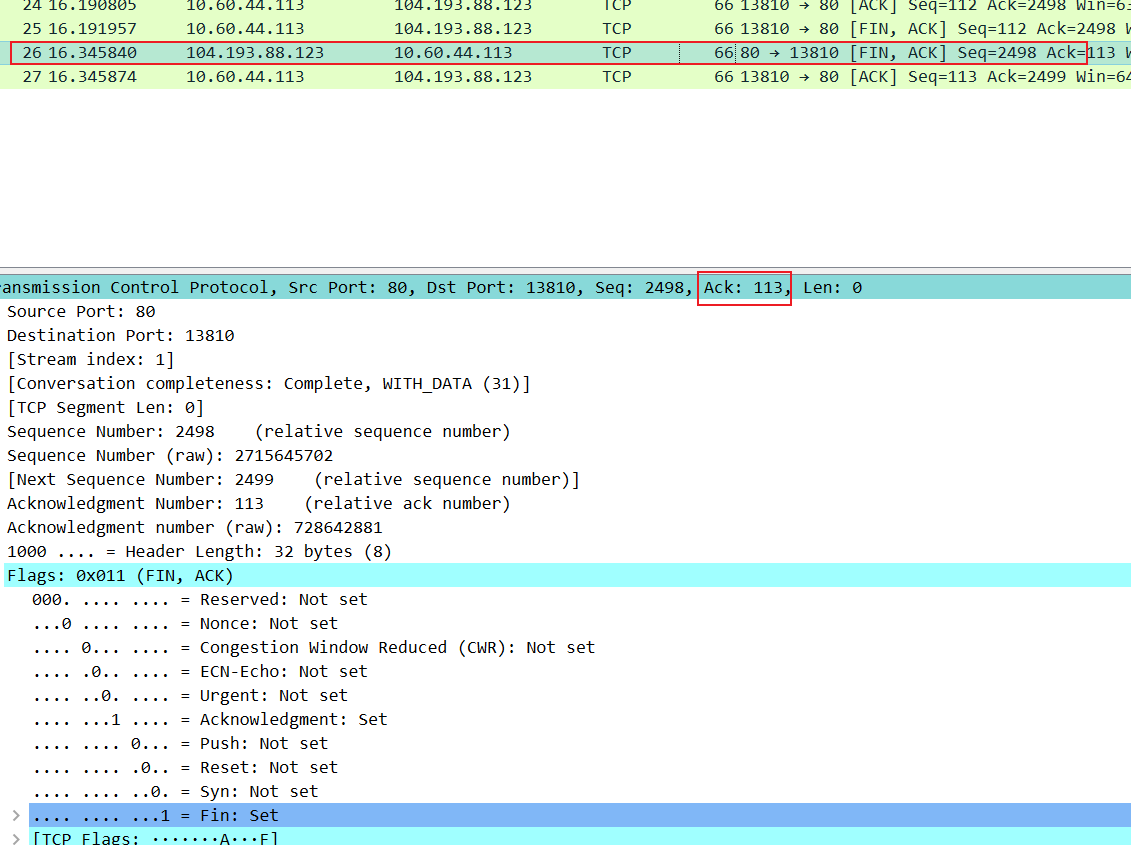
**四次挥手**



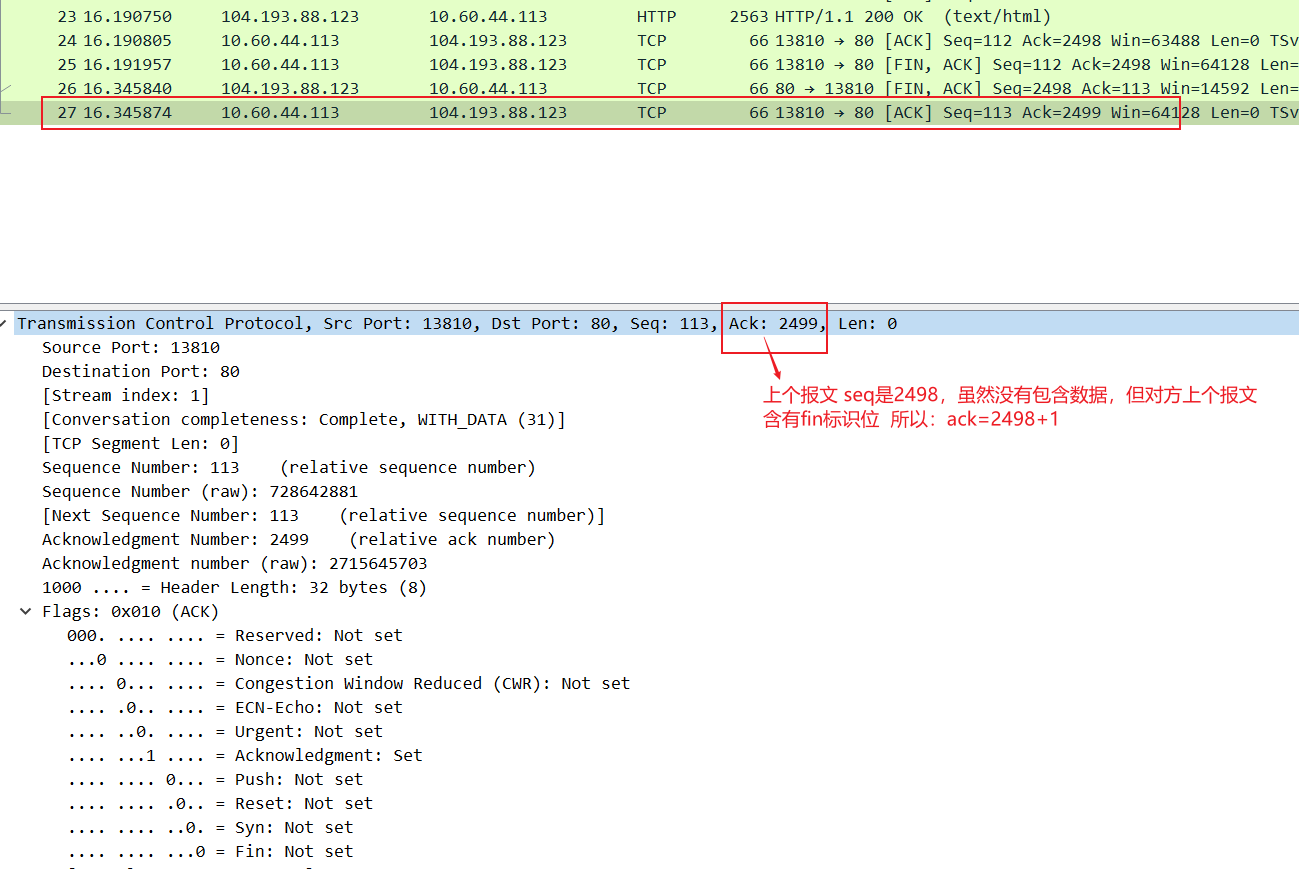
第一次挥手



第二次挥手与第三次挥手 因为没有数据传输



第四次挥手



什么内核发送缓冲区？什么是内核接收缓冲区？

内核发送缓冲区：将数据拷贝进入socket的内核发送缓冲区，发给应用程序（服务端）

内核接收缓冲区：，对端发来的数据都会经由内核接收并且缓存到socket的内核接收缓冲区之中，调用**recv()**，就是把内核缓冲区中的数据拷贝到应用层用户的buffer里面

https://blog.csdn.net/Swallow\_he/article/details/84392285?spm=1001.2101.3001.6650.6&utm\_medium=distribute.pc\_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7Edefault-6-84392285-blog-78871093.pc\_relevant\_downloadblacklistv1&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7Edefault-6-84392285-blog-78871093.pc\_relevant\_downloadblacklistv1&utm\_relevant\_index=12

可读可写的本质什么？

可读：内核接收缓存区里有数据，后调用recv()方法，把内核缓冲区中的数据拷贝到应用层用户的buffer里面

可写:   内核发送缓存区有空间可以放入待发送的数据,后通过网线传给服务端，服务端在通过内核接受并缓冲到服务端的内核缓冲区里

阐述一下TCPIP协议5层的每层分别解决了什么问题？

* 应用层 ：为特定应用程序提供数据传输服务。
* 传输层 ：为进程提供通用数据传输服务。
* 网络层 ：为主机提供数据传输服务。网络层把传输层传递下来的报文段或者用户数据报封装成分组。
* 数据链路层 ：主机之间可以有很多链路，链路层协议就是为同一链路的主机提供数据传输服务。数据链路层把网络层传下来的分组封装成帧。
* 物理层 ：考虑的是怎样在传输媒体上传输数据比特流，而不是指具体的传输媒体。物理层的作用是尽可能屏蔽传输媒体和通信手段的差异，使数据链路层感觉不到这些差异。

阐述一下每一层如何跟应用程序/socket/内核缓冲区/网卡 这些映射起来，每层协议在哪个地方发挥了什么作用，谈谈你的体会。

对应关系：



