# 计网实验2

1）搭建Web服务器（自由选择系统），并制作简单的Web页面，包含简单文本信息（至少包含专业、学号、姓名）和自己的LOGO。（2）通过浏览器获取自己编写的Web页面，使用Wireshark捕获浏览器与Web服务器的交互过程，并进行简单的分析说明。（4）提交实验报告。

(1) 功能实现：

l Web服务器搭建、编写Web页面（提交HTML文档）（20分）

l Wireshark捕获交互过程（提交捕获pcap文件）（20分）

(2) 演示并讲解（30分）

(3) 实验报告（30分）

需要大家必须分析的是握手、http请求、挥手几个过程，其余可以自行扩展~

实验准备：

网页编写：

编写html网页如下所示，包含个人信息文本及jpeg格式图片一张。

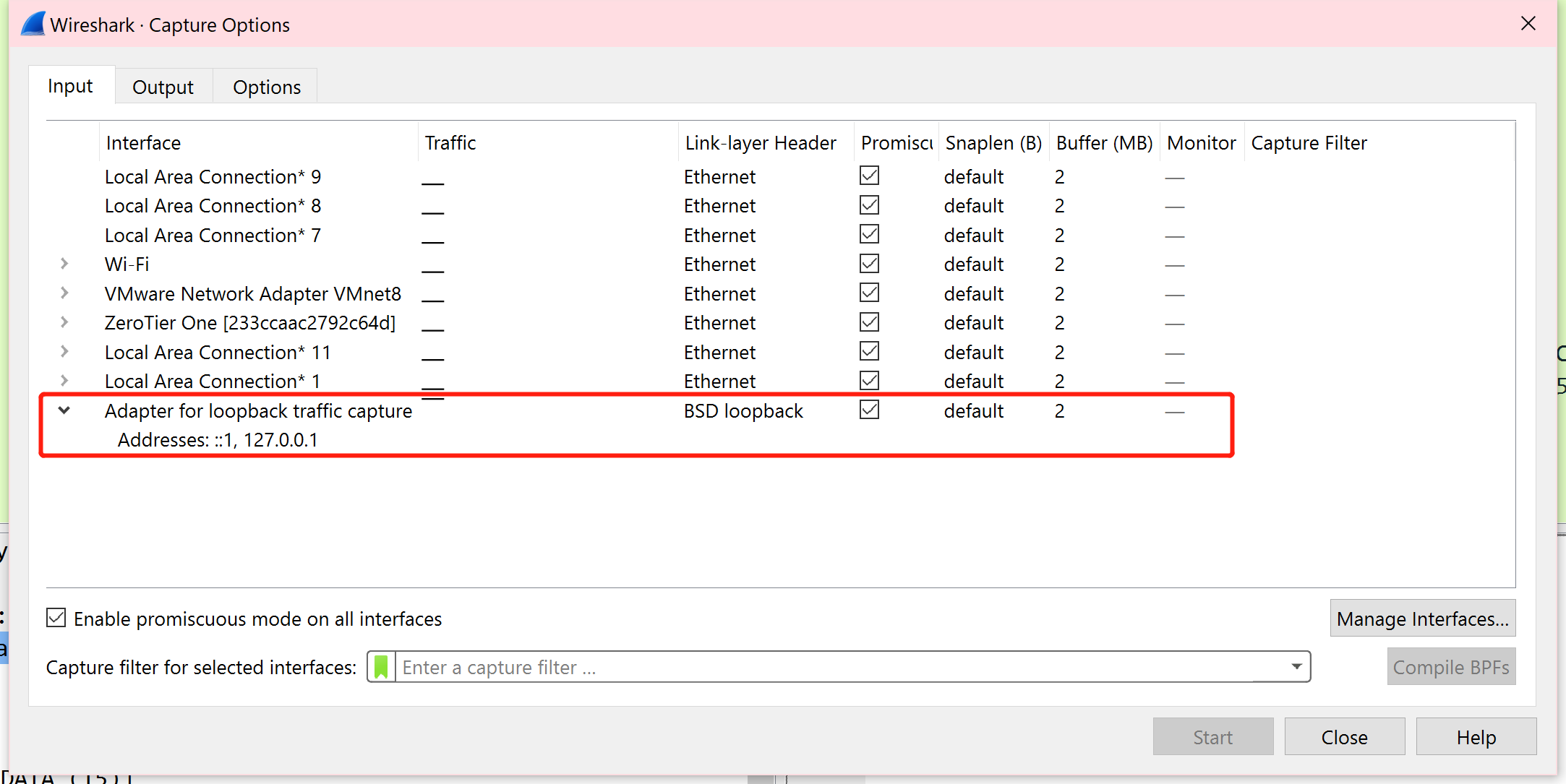
<!doctype html>
  
<html>
  
<head>
  
<meta charset='UTF-8'><meta name='viewport' content='width=device-width initial-scale=1'>
  
<title>我的网页</title></head>
  
<body><p>专业：计算机科学与技术</p>
  
<p>姓名：冯朝芃</p>
  
<p>学号：2012039</p>
  
<p><img src="./xiaohui.jpeg" referrerpolicy="no-referrer" alt="xiaohui"></p>
  
</body>
  
</html>

服务器搭建：

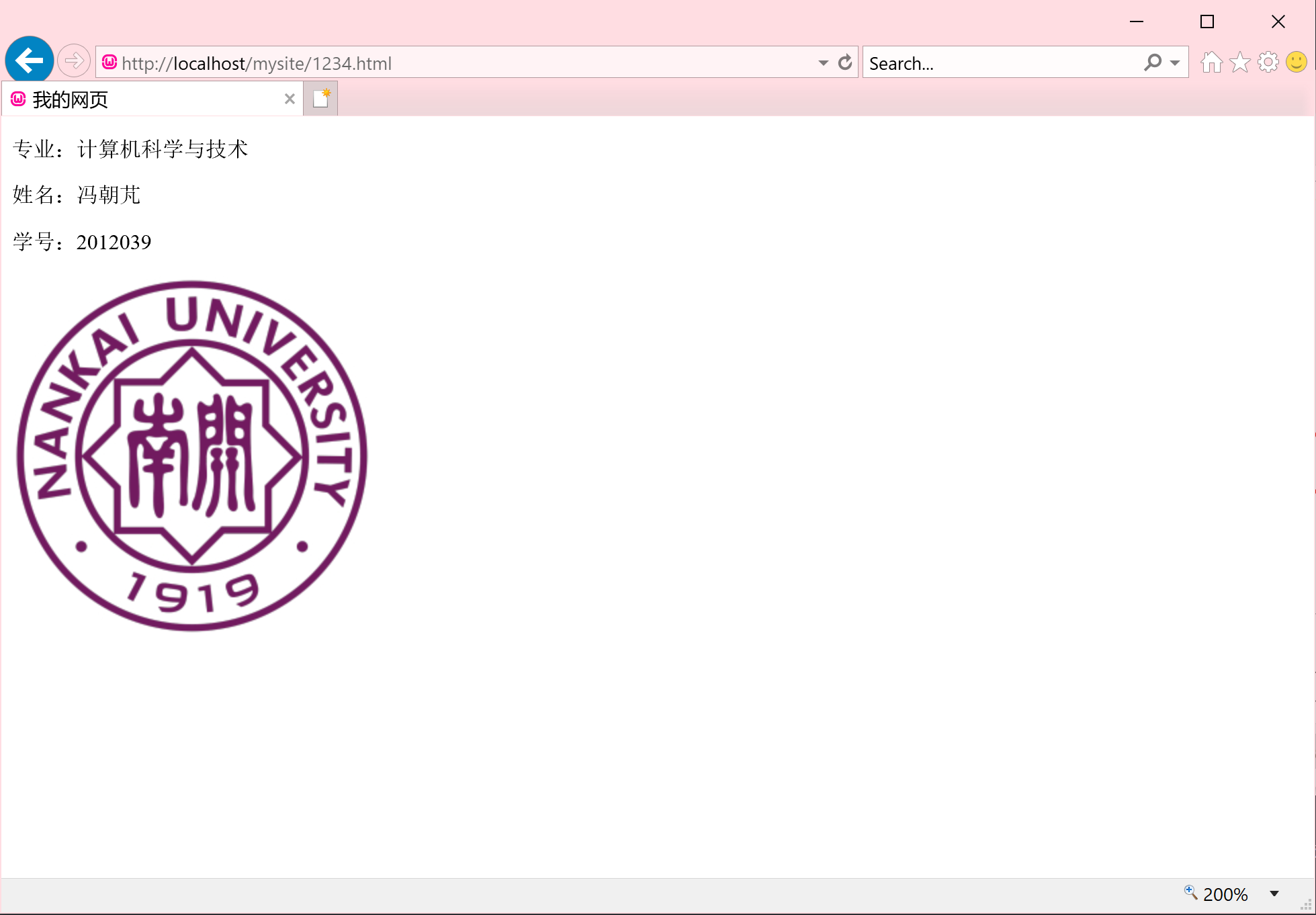
使用Wampserver软件搭建服务器环境，配置Apache2.4服务器，服务端口开放在默认的80端口上。将网页和图片放置于对应的项目文件夹中。

抓包实验：

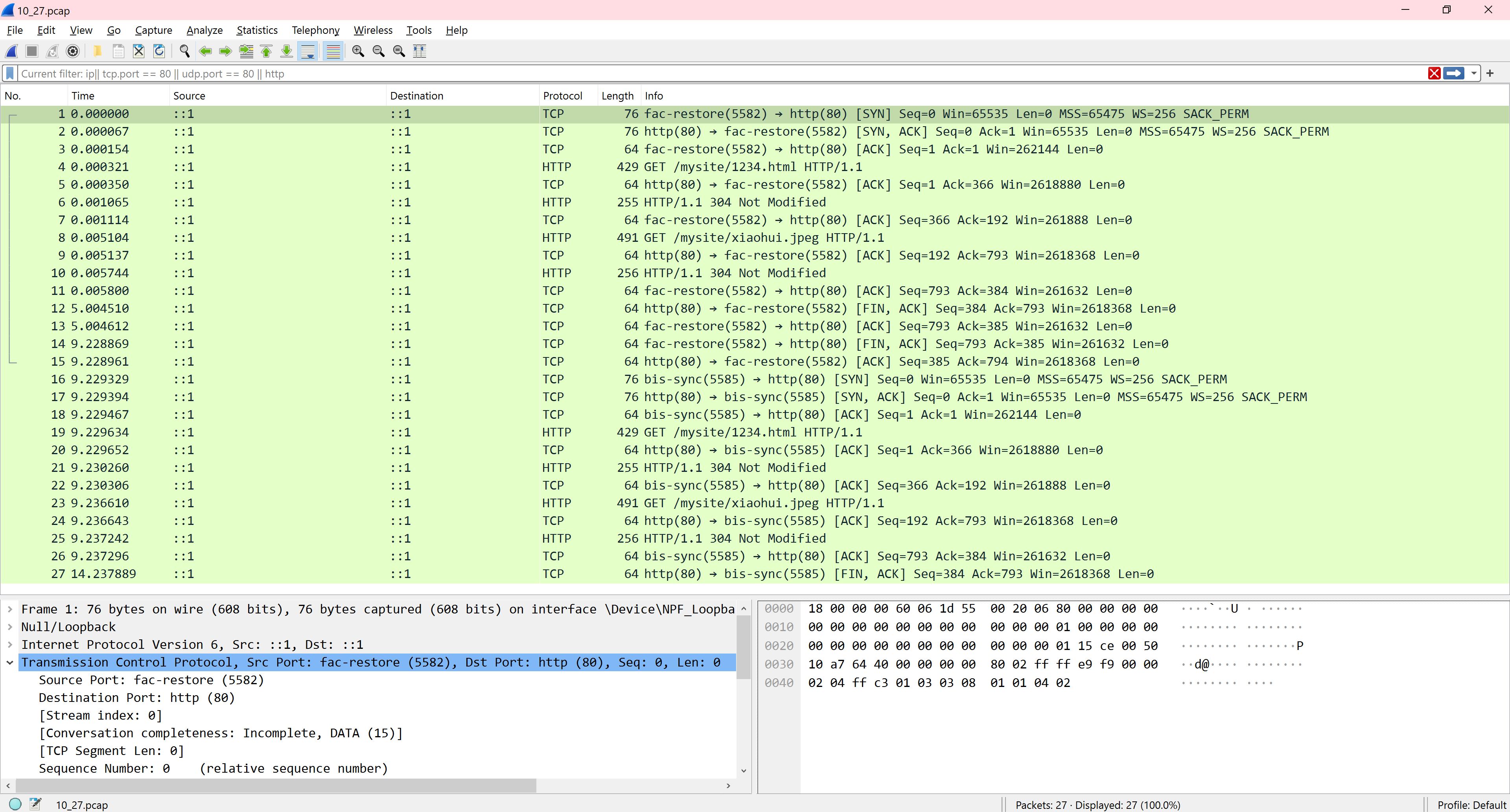
使用Wireshark4.0软件进行网络数据包抓取，选择如图所示的抓包选项，监听localhost。



抓包开始后，使用IE11浏览器访问自建网站，等待一会使服务器与浏览器断开连接，之后再刷新一次，使浏览器与服务器主动断开连接。浏览器打开页面后效果如图所示。



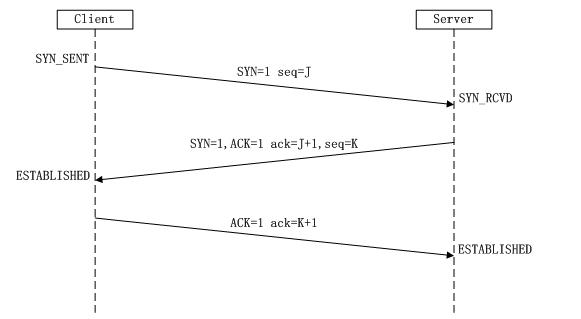
抓包完成后界面如下图所示，从时间上可以很清晰的看到：以0s开头的是三次握手和http请求的数据包，以5s开头的是服务器端超时与浏览器挥手断开连接的数据包，以9s开头的是浏览器刷新网页，与服务器挥手断连的数据包。Wireshark软件也在左侧以灰色线框标注出了一次完整的TCP连接过程。



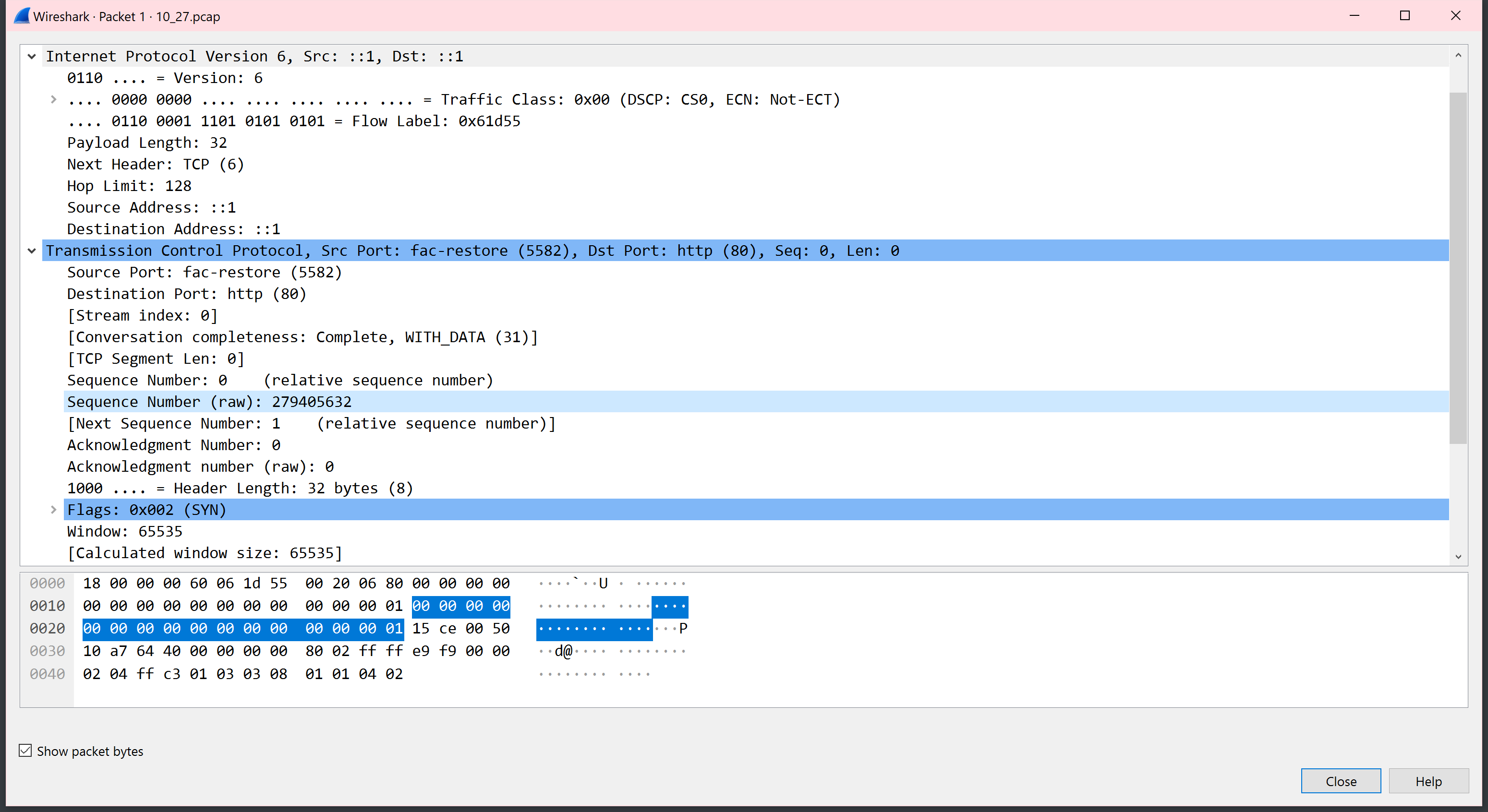
以下按时间顺序分析TCP握手、HTTP请求和响应、TCP挥手的数据包内容。

TCP握手：

TCP三次握手（Three-Way Handshake）即建立TCP连接，指建立一个TCP连接时，需要客户端和服务端总共发送3个包以确认连接的建立。图示如下。



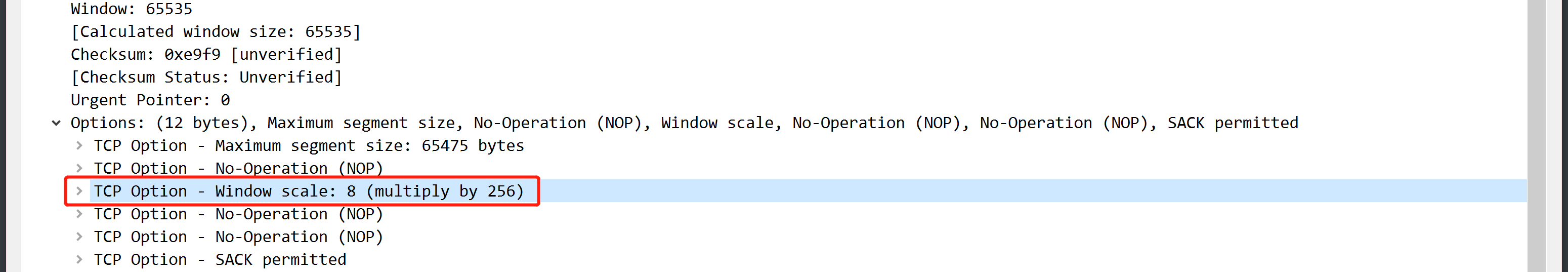
第一次握手：Client将标志位SYN置为1，随机产生一个序号seq=J，并将该数据包发送给Server，Client进入SYN\_SENT状态，等待Server确认。展示数据包如下。



TCP是一种流式协议，而TCP流（TCP STREAM）可以用来形容建连-传输-挥手这个过程，Wireshark也采用了这种说法，故在之后的分析中也采用这种说法。观察图片内容可以看到，这个数据包的随机序列号是279405632，相对于整个流中客户端所发送的数据包是第0个数据包。Flags标志位只有SYN被置1，这说明它是TCP握手中的第一个数据包（第一次握手）——客户端程序向服务器发送建连请求。

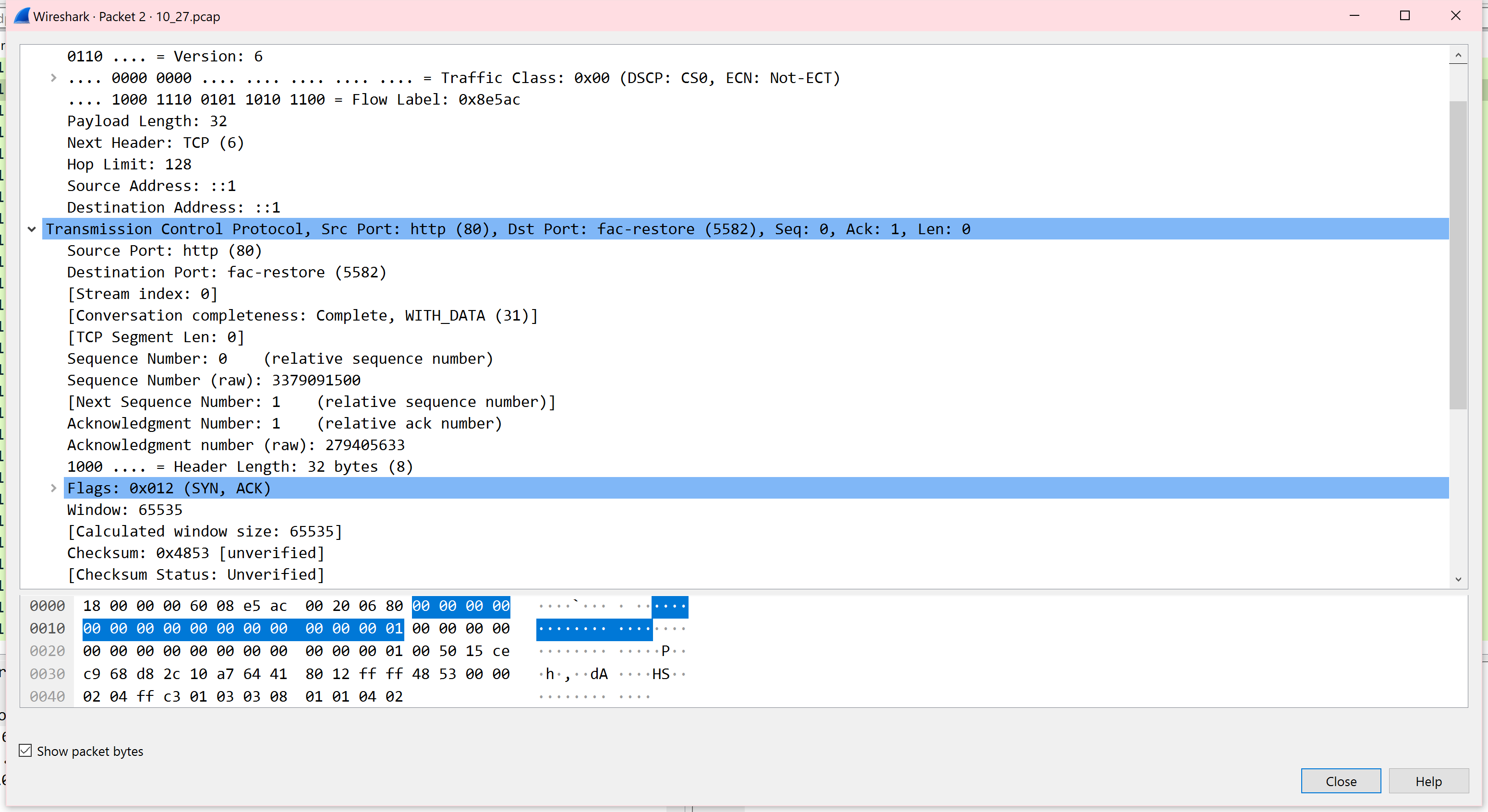
数据包中还有一些其他信息。如发送端口是5582，目标端口是80，由于本机启用了ipv6，故目标和来源IP都是::1，也就是localhost。此时的确认号是0，因为这个TCP流中没有上一个数据包。该TCP首部的长度是32bytes。

可以看到TCP首部中有一个Window Size字段，它其实是接收窗口（滑动窗口）的大小。用来告知发送端自己所能接收的数据量，从而达到一部分流控的目的，与拥塞控制有关。该数据包Window大小为65535（16bits能表示最大值），在下图中我们可以看到option字段有一个缩放系数，这里是8，要左移八位，也就是说接收窗口为；缩放系数如果以后不调整就固定了。发送方的缩放系数和接收方的乘积因子可以不同，由各自决定，观察后续数据包发现这里两方恰巧是一样的。Window大小是动态调整的，后续我们可以看到很多动态调整的例子。



此时ACK为0，事实上TCP规定除了在最初建立连接时候的SYN包之外该位必须设置为1，该位为1时，确认应答的字段为有效。

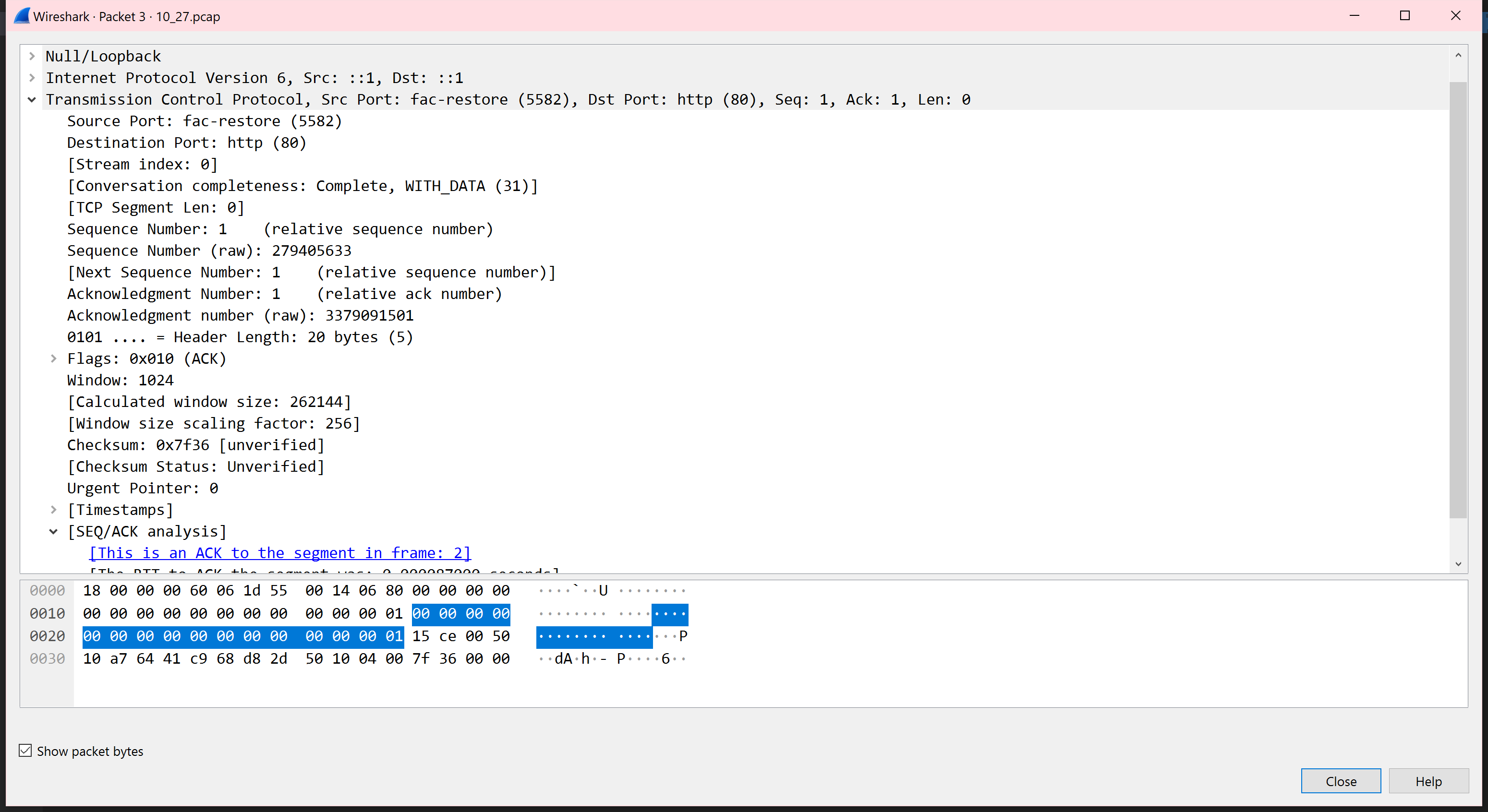
第二次握手：Server收到数据包后由标志位SYN=1知道Client请求建立连接，Server将标志位SYN和ACK都置为1，ACK确认号=J+1，随机产生一个序号seq=K，并将该数据包发送给Client以确认连接请求，Server进入SYN\_RCVD状态。展示如下。



可以看到这个数据包从服务器端80端口发往浏览器5582端口。它的确认号原始值为279405633，是上面提到的J+1，也就是说“279405633-1=279405632被我收到了”。此时服务器生成了一个新的随机的序列号K，3379091500。要注意的，在整个TCP流中，浏览器和服务器各自作为发送端的时候，使用不同的起始序列号，分别为K和J。

此时Flags标志位的SYN和ACK都被置1，这说明是第二次握手的数据包。

第三次握手：Client收到确认后，检查确认号是否为J+1，ACK是否为1，如果正确则将标志位ACK置为1，确认号=K+1，并将该数据包发送给Server，Server检查确认号是否为K+1，ACK是否为1，如果正确则连接建立成功，Client和Server进入ESTABLISHED状态，完成三次握手，随后Client与Server之间可以开始传输数据了。截图如下。



注意到客户端发送的此数据包序列号是J+1，确认的是K+1（表示K及之前的被我收到了）。Flags标志位的ACK被置1。Window大小此时为262144bytes，也就是客户端说“我现在最多只能接受262144bytes”。

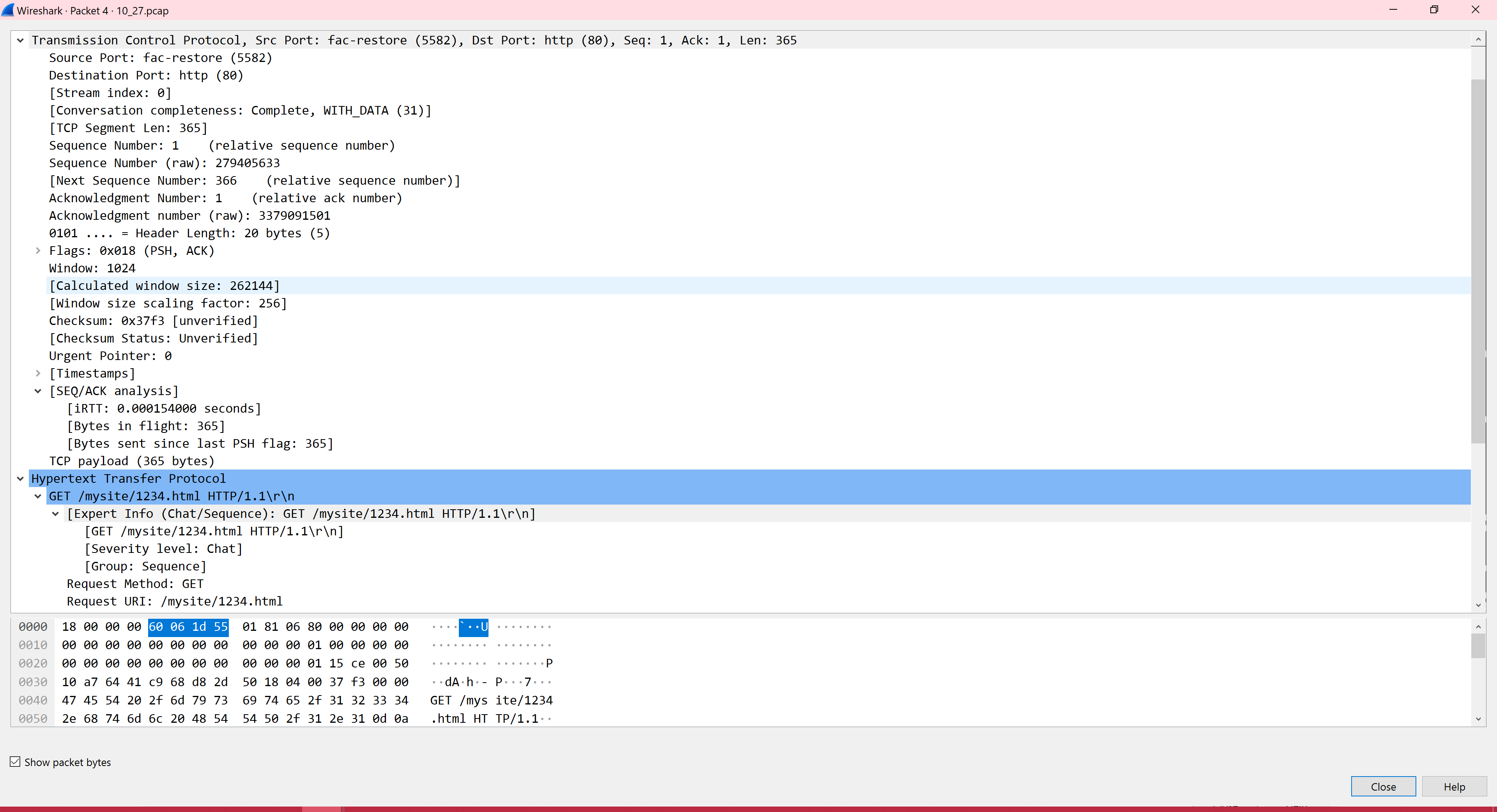
以上三个数据包中的数据段都是没有的。

值得注意一点，只是发送ACK并不占用序列号，而序列号一般是跟着要发送的字节走的，可以留意一下，后面的数据包浏览器将从J+1开始发送，服务器将从K+1开始发送。

HTTP请求和响应：

1

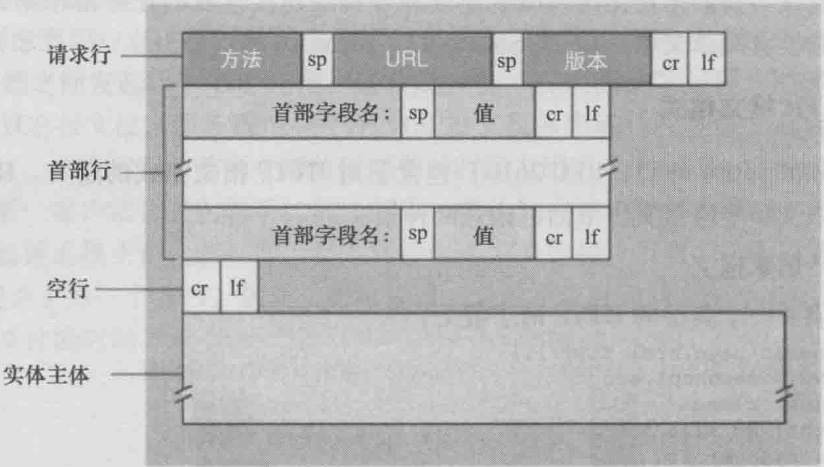
建立连接后，浏览器发送的第一个数据包包含一个HTTP GET请求。截图如下。



TCP首部部分的序列号为J+1，确认序列号为K+1。Flags字段给出了ACK和PSH，PSH表示接收方应该尽快将这个报文交给应用层。

TCP层将HTTP报文放在了TCP数据部分，长度365bytes。

HTTP请求报文格式如下：



img

请求报文由三部分组成：请求行(request line)、首部行(header line)、实体体(entity body)。

请求行有三个字段：方法、URL、HTTP版本

（1）方法：可以取不同的值，包括GET、POST、HEAD、PUT和DELETE等。该报文使用GET方法拉取HTML网页。

（2）URL：请求对象的标识。这里请求对象标识就是：/mysite/1234.html

（3）HTTP版本：这里HTTP版本为1.1。

首部行由多组键值对（首部字段名：首部字段值）组成。这里包括：

User-Agent：浏览器类型。

Host：初始URL中的主机和端口。

Accept：浏览器可接受的MIME类型，也就是数据类型信息，这里是text/html, application/xhtml+xml, image/jxr, \*/\*。

Accept-Encoding：浏览器能够进行解码的数据编码方式，比如gzip。Servlet能够向支持gzip的浏览器返回经gzip编码的HTML页面。许多情形下这可以减少5到10倍的下载时间。

Accept-Language：浏览器所希望的语言种类。

Connection：表示是否需要持久连接，这里是HTTP1.1默认就是持久连接。如果Servlet看到这里的值为“Keep-Alive”，或者看到请求使用的是HTTP 1.1（HTTP 1.1默认进行持久连接），它就可以利用持久连接的优点，当页面包含多个元素时（例如Applet，图片），显著地减少下载所需要的时间。要实现这一点，Servlet需要在应答中发送一个Content-Length头，最简单的实现方法是：先把内容写入ByteArrayOutputStream，然后在正式写出内容之前计算它的大小。

2

服务器对上一个请求的响应。

实验完成后，保存Wireshark PCAP文件到本地。

参考资料：

[Wireshark抓包实验*yuminglc的博客-CSDN博客*wireshark抓包实验](https://blog.csdn.net/vic_to_ry/article/details/111603584)

[简述TCP的三次握手过程 - 98剑南春 - 博客园 (cnblogs.com)](https://.cnblogs.com/jiannanchun/p/15398343.html)

[HTTP请求和响应详解 - 木木-林 - 博客园 (cnblogs.com)](https://.cnblogs.com/mumu-lin/p/15621755.html)