计算机网络第二次实验

朱浩泽 1911530 计算机科学与技术

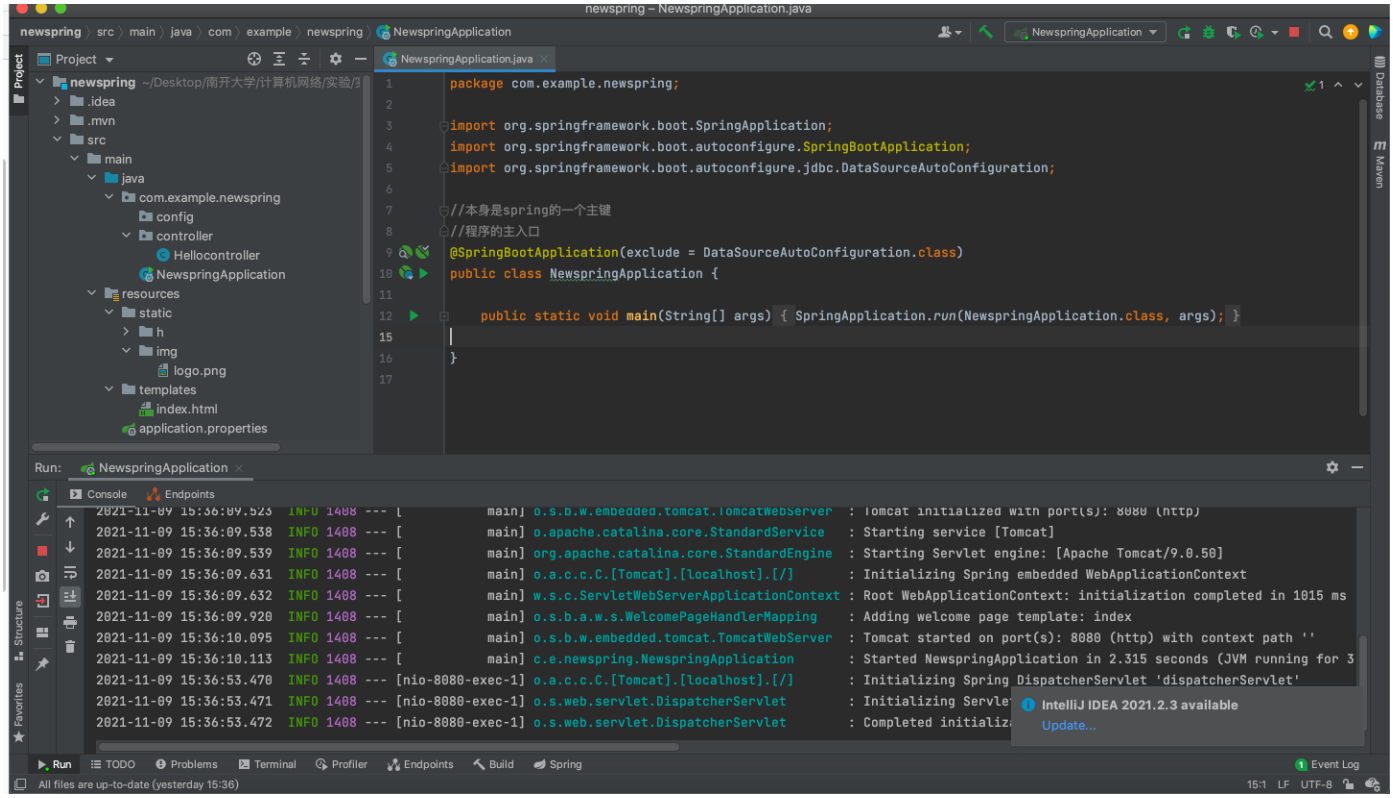
实验要求

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。
3. 提交实验报告。

Web服务器的搭建

平台：springboot

在springboot中搭建一个简单的Web服务器，其框架如下



启动该服务器后，在浏览器中输入localhost:8080即可进入该网站，该网站包括了姓名、学号、专业，以及自己设计的一张图片，其截图如下：

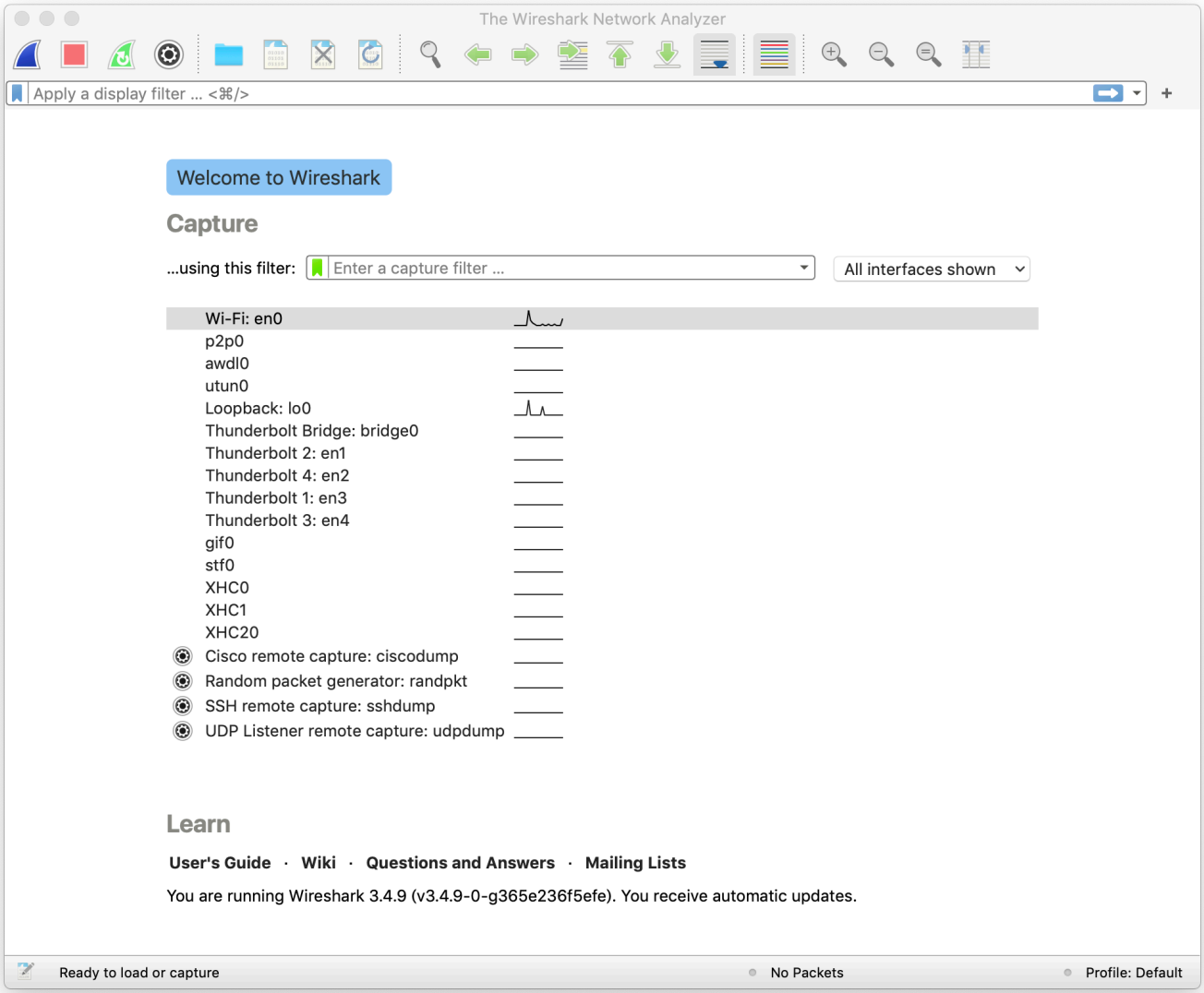
专业：计算机科学与技术

姓名：朱浩泽

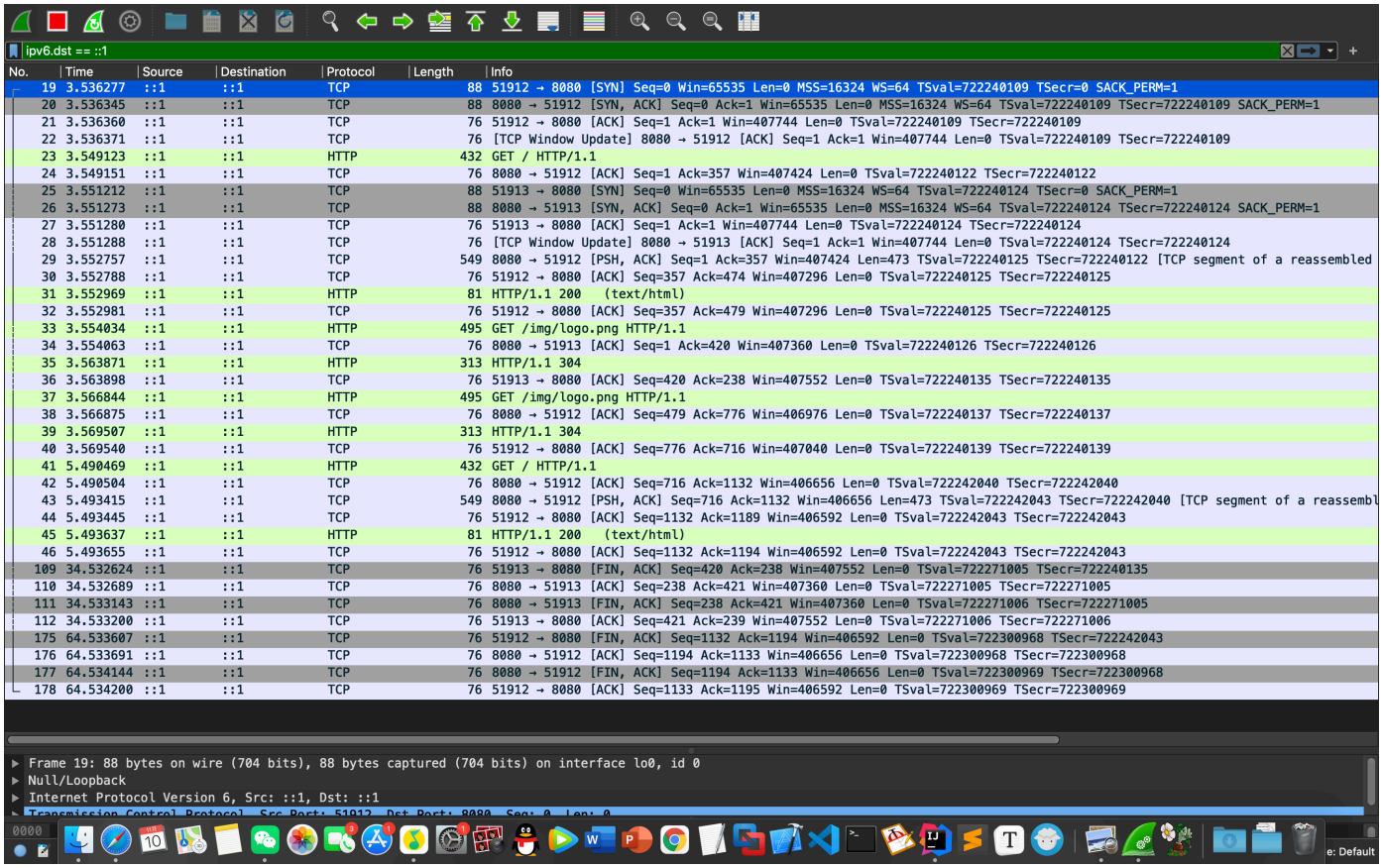
学号：1911530

捕获交互过程

我们在实验中首先开启Wireshark，并开启抓包，在浏览器中输入自己设计的网页的网址，观察Wireshark界面如下图所示：



可以看出，由于客户端和服务器都是本机，所以我们选择Loopback:lo0进入。进入后我们使用过滤器，取出本机端口与本机交互的数据包显示，以分析浏览器与Web服务器的交互过程。

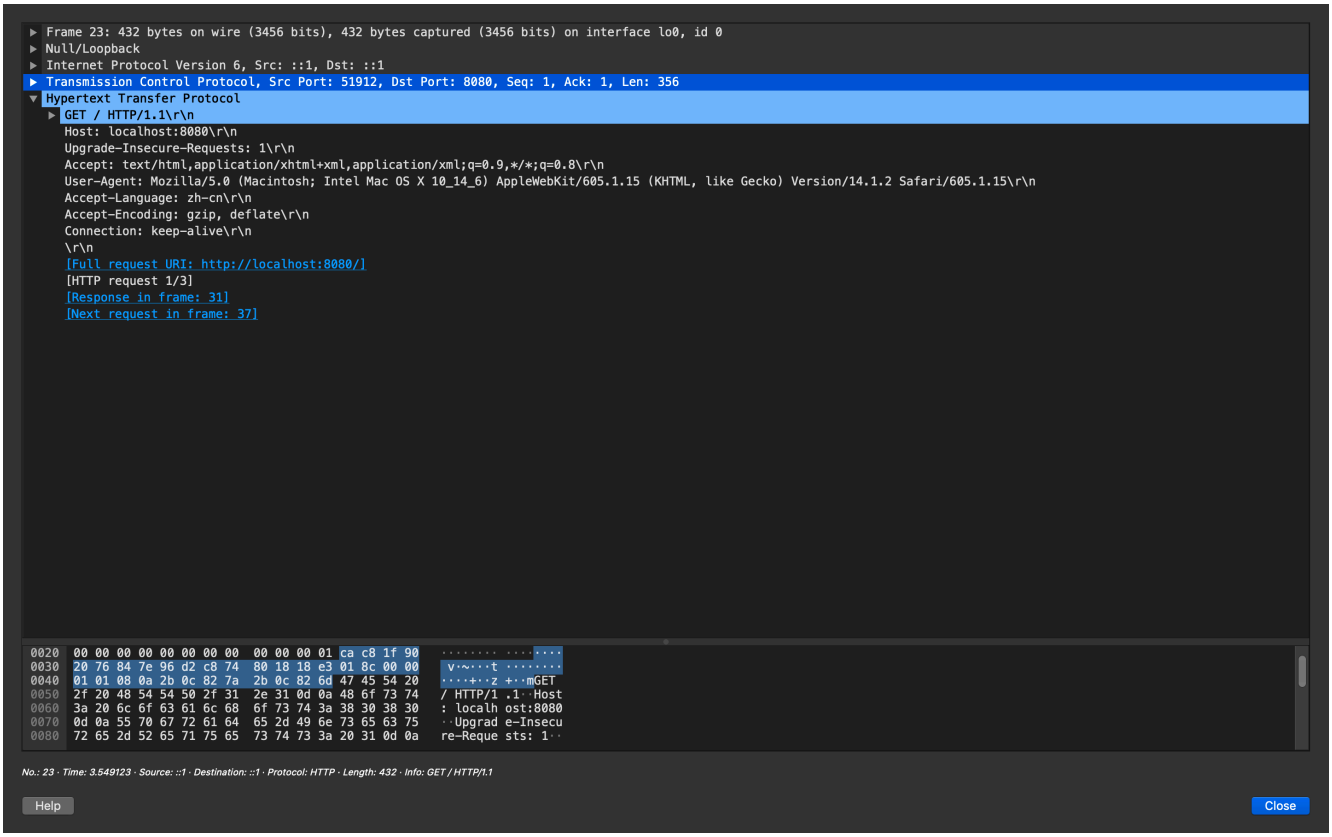


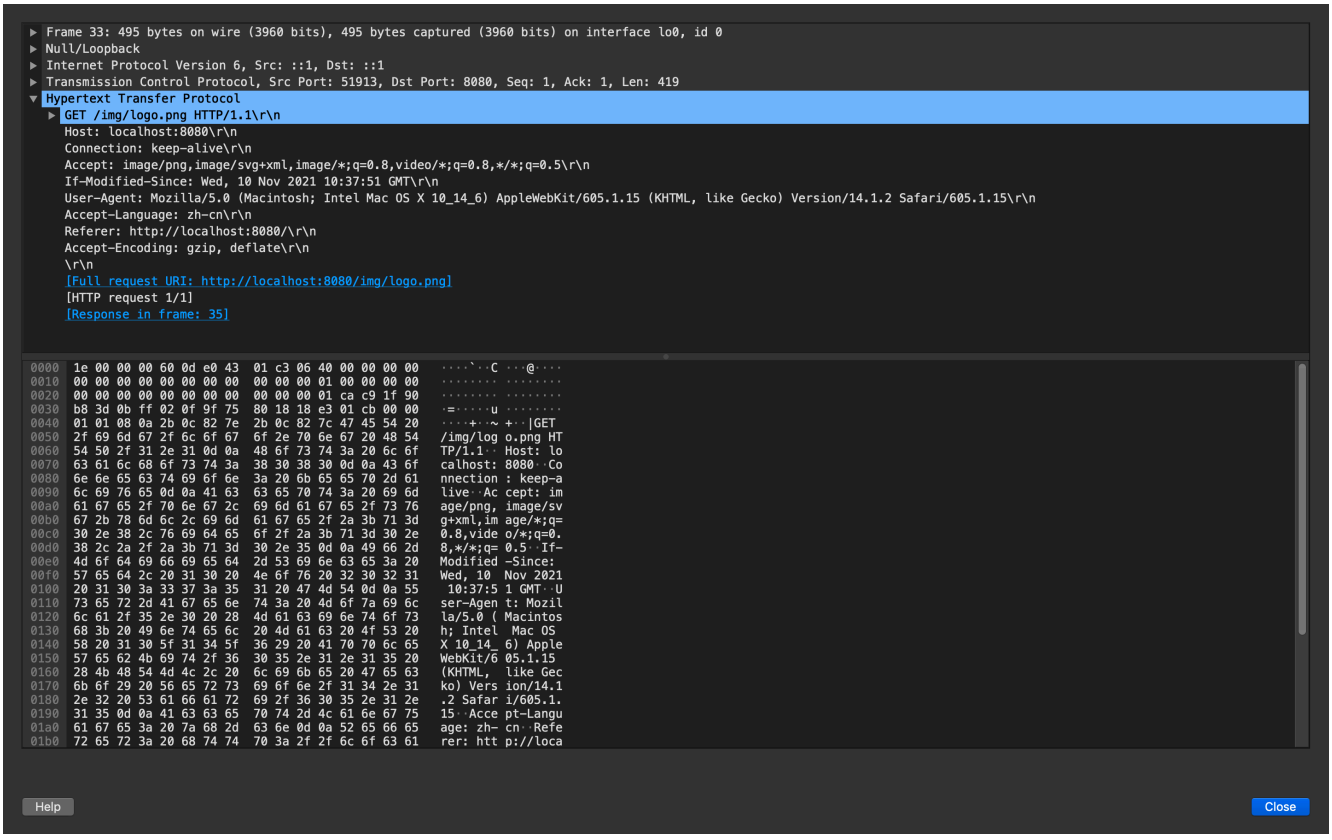
HTTP相关数据包

```
23 3.549123 ::1 ::1 HTTP 432 GET / HTTP/1.1
31 3.552969 ::1 ::1 HTTP 81 HTTP/1.1 200 (text/html)
33 3.554034 ::1 ::1 HTTP 495 GET /img/logo.png HTTP/1.1
35 3.563871 ::1 ::1 HTTP 313 HTTP/1.1 304
```

首先我们可以看出，springboot使用的是http1.1协议首先发送一个访问页面的get请求，客户端收到get请求后将所请求内容发送给客户端

- 访问页面的请求





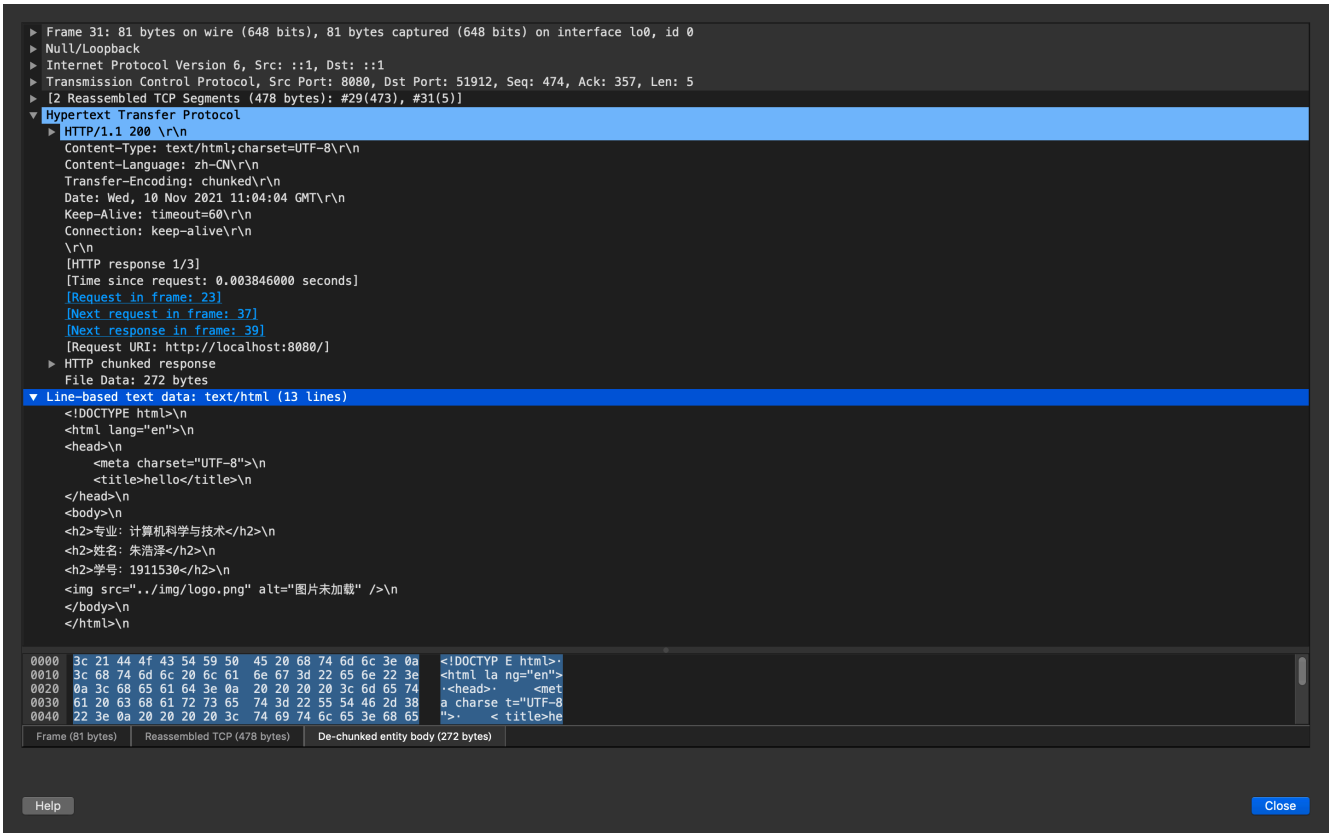
点开get请求，可以看出有关http协议的相关内容

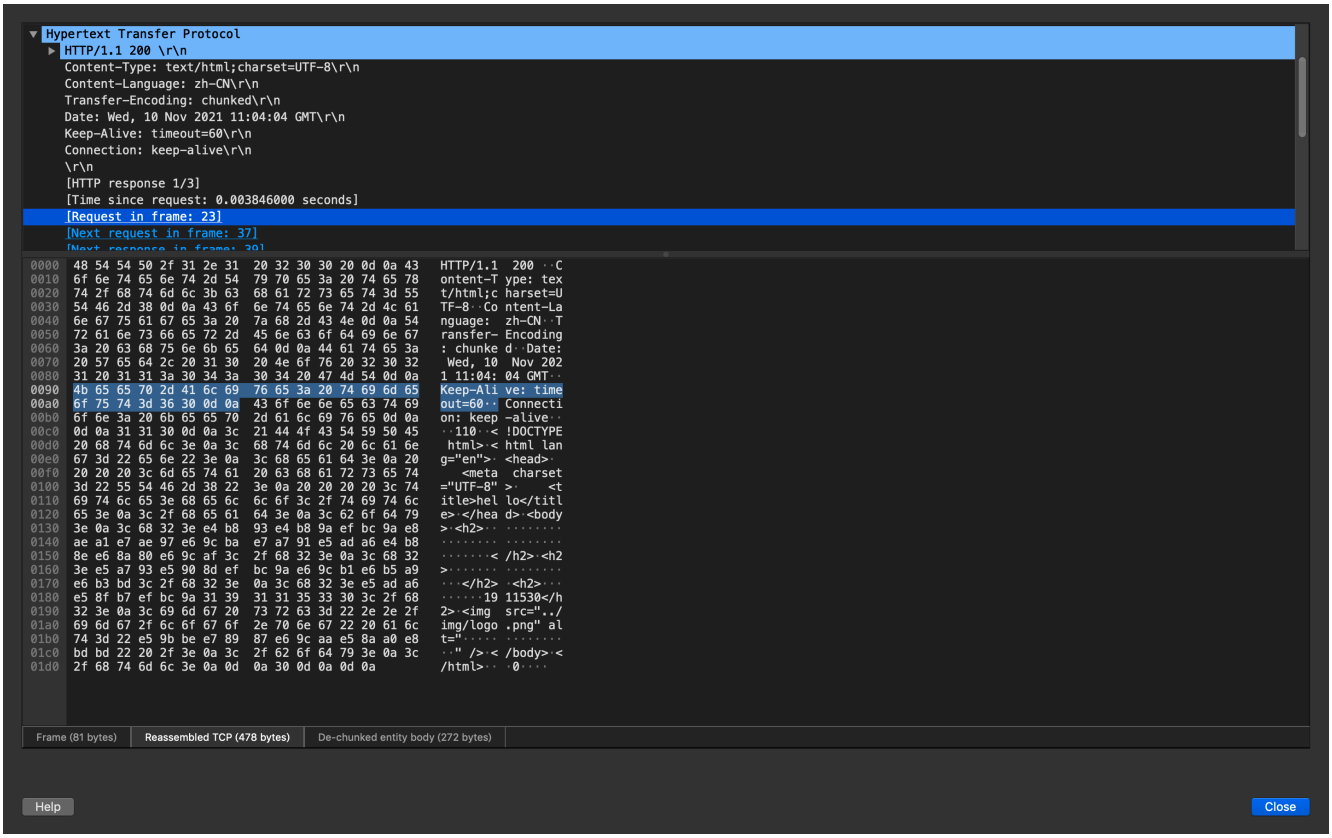
GET / HTTP/1.1\r\n 称为请求行，由三个部分组成：请求方法、URI、HTTP协议版本，他们之间用空格分隔。该部分位于数据首行，基本格式为：该部分的请求方法字段（GET）给出了请求类型，URI给出请求的资源位置），最后HTTP协议版本给出HTTP的版本号。

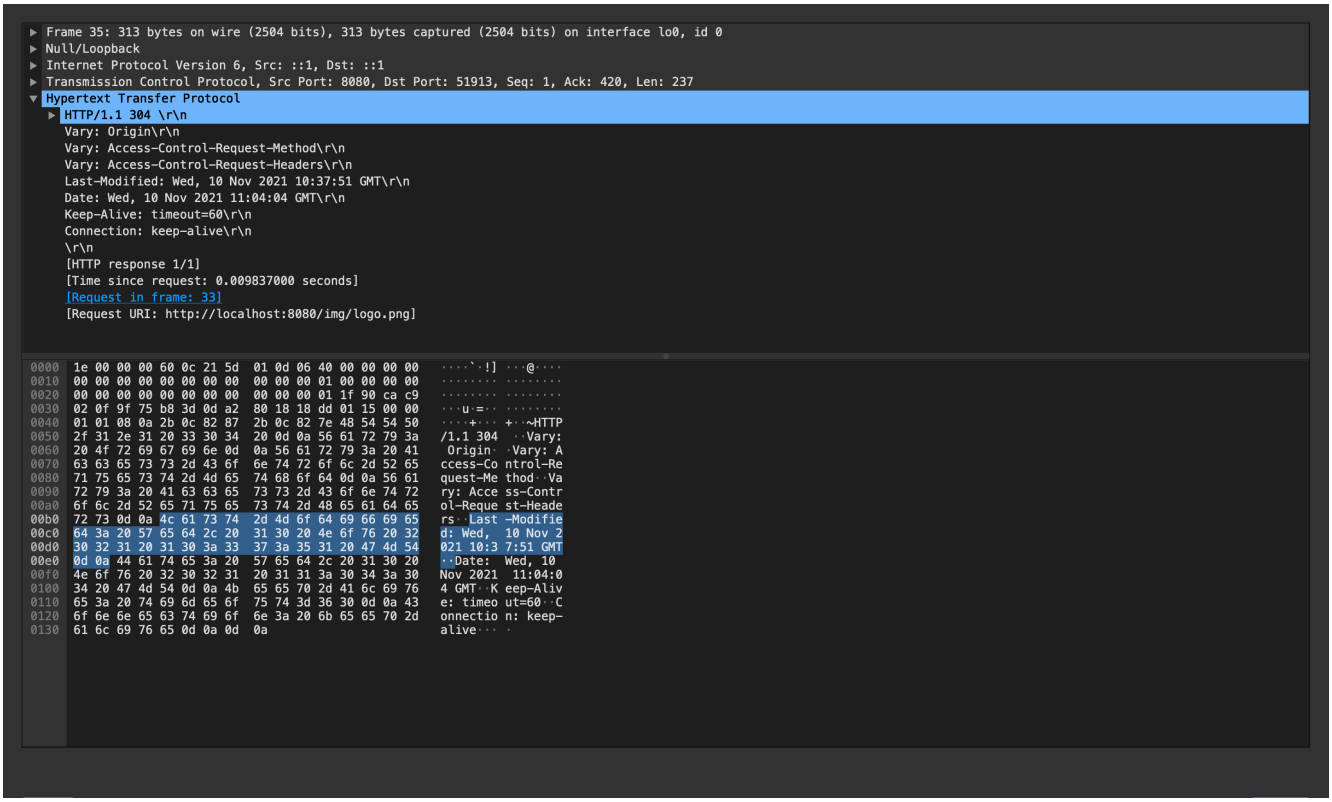
接下来的部分称为请求头部，其紧跟着请求行，该部分主要是用于描述请求正文，主要是用于说明请求源、连接类型、以及一些Cookie信息等。

每一行都以\r\n结尾。

- 返回数据







响应报文由状态行、响应头部、空行和响应体4个部分构成。

首先是状态行，主要给出响应HTTP协议的版本号、响应返回状态码、响应描述；响应头部主要是返回一些服务器的基本信息，以及一些Cookie值等；响应体为所请求的数据（可以为html文档内容或图片等多种格式数据）。

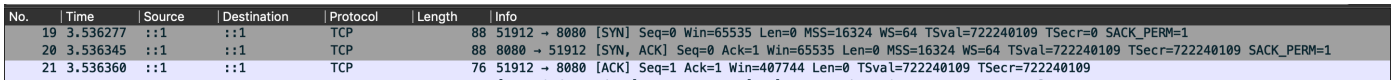
TCP相关数据包

TCP协议提供的是按序、可靠的服务，是一种面向连接的传输方式，即其发送数据之前发送方和接收放需要握手，断开链接时需要四次挥手。

TCP的三次握手

- TCP三次握手过程
  1. 第一次握手： 客户端发送syn包(seq=x)到服务器，并进入SYN_SEND状态，等待服务器确认
  2. 第二次握手： 服务器收到syn包，必须确认客户的SYN(ack=x+1)，同时自己也发送一个SYN包(seq=y)，即SYN+ACK包，此时服务器进入SYN_RECV状态
  3. 第三次握手： 客户端收到服务器的SYN+ACK包，向服务器发送确认包ACK(ack=y+1)，此包发送完毕，客户端和服务器进入ESTABLISHED状态，完成三次握手

握手过程中传送的包里不包含数据，三次握手完毕后，客户端与服务器才正式开始传送数据。理想状态下，TCP连接一旦建立，在通信双方中的任何一方主动关闭连接之前，TCP连接都将被一直保持下去。

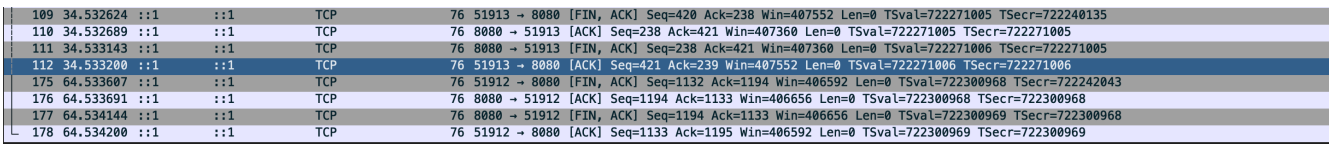


No.	Time	Source	Destination	Protocol	Length	Info
19	3.536277	::1	::1	TCP	88	51912 → 8080 [SYN] Seq=0 Win=65535 Len=0 MSS=16324 WS=64 TSval=722240109 TSecr=0 SACK_PERM=1
20	3.536345	::1	::1	TCP	88	8080 → 51912 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=16324 WS=64 TSval=722240109 TSecr=722240109 SACK_PERM=1
21	3.536360	::1	::1	TCP	76	51912 → 8080 [ACK] Seq=1 Ack=1 Win=407744 Len=0 TSval=722240109 TSecr=722240109

- 如上图所示，即为tcp的三次握手过程。首先由51912向8080端口发送了一条syn包，并令seq = x = 0（本机初始序列号），报文长度为0，滑动窗口为65535，最大窗口长度为163245，窗口扩大因子为64，此时浏览进入了SYN_SEND状态，这是第一次握手；然后服务器8080端口接收到了浏览器发送的syn包后，确认客户的syn，使ack = x + 1 = 1，seq = y = 0，对序列号为1（ack）之前的报文进行确认，同时向客户发送一个（syn，ack）包，其中报文长度为0，滑动窗口为65535，最大窗口长度为163245，这是第二次握手；客户端接受到了服务器发送的（syn，ack）包后，客户端向服务器返回确认包，其中ack = y + 1 = 1，滑动窗口为407744，报文长度为0，这是第三次握手。至此三次握手完成，服务器和客户端建立了联系。

TCP的四次挥手

- TCP四次挥手过程
  1. 第一次挥手： Client发送一个FIN，用来关闭Client到Server的数据传送，Client进入FIN_WAIT_1状态。
  2. 第二次挥手： Server收到FIN后，发送一个ACK给Client，确认序号为收到序号+1（与SYN相同，一个FIN占用一个序号），Server进入CLOSE_WAIT状态。
  3. 第三次挥手： Server发送一个FIN，用来关闭Server到Client的数据传送，Server进入LAST_ACK状态。
  4. 第四次挥手： Client收到FIN后，Client进入TIME_WAIT状态，接着发送一个ACK给Server，确认序号为收到序号+1，Server进入CLOSED状态，完成四次挥手。



109	34.532624	::1	::1	TCP	76	51913 → 8080 [FIN, ACK] Seq=420 Ack=238 Win=407552 Len=0 TSval=722271005 TSecr=722240135
110	34.532689	::1	::1	TCP	76	8080 → 51913 [ACK] Seq=238 Ack=421 Win=407360 Len=0 TSval=722271005 TSecr=722271005
111	34.533143	::1	::1	TCP	76	8080 → 51913 [FIN, ACK] Seq=238 Ack=421 Win=407360 Len=0 TSval=722271006 TSecr=722271005
112	34.533200	::1	::1	TCP	76	51913 → 8080 [ACK] Seq=421 Ack=239 Win=407552 Len=0 TSval=722271006 TSecr=722271006
175	64.533607	::1	::1	TCP	76	51912 → 8080 [FIN, ACK] Seq=1132 Ack=1194 Win=406592 Len=0 TSval=722300968 TSecr=722242043
176	64.533691	::1	::1	TCP	76	8080 → 51912 [ACK] Seq=1194 Ack=1133 Win=406656 Len=0 TSval=722300968 TSecr=722300968
177	64.534144	::1	::1	TCP	76	8080 → 51912 [FIN, ACK] Seq=1194 Ack=1133 Win=406656 Len=0 TSval=722300969 TSecr=722300968
178	64.534200	::1	::1	TCP	76	51912 → 8080 [ACK] Seq=1133 Ack=1195 Win=406592 Len=0 TSval=722300969 TSecr=722300969

- 如上图所示，即为tcp四次挥手过程（由于浏览器开启了多线程，我们只对一个线程做分析）。首先，51912向8080端口发送了一条请求结束的报文（fin，ack），报文序列号为seq = x = 1132，ack = y = 1194，报文长度为0，这是第一次挥手；然后服务器8080端口接收到了浏览器发送的fin包后，对收到的1194号前的报文进行确认，其报文序列号为seq = y = 1194，ack = x + 1 = 1132 + 1 = 1133，这是第二次挥手；然后服务器（8080端口）再

向浏览器（51912）发送一个fin包，其内容与前一个基本相同分，用来关闭服务器端到客户端的数据传送，这是第三次挥手；客户端对收到的fin包进行确认，其报文序列号为seq = y = 1133，ack = 1194 + 1 = 1195，客户端进入关闭状态，这是第四次挥手。至此tcp的四次挥手结束，客户端和服务器断开连接，访问结束。