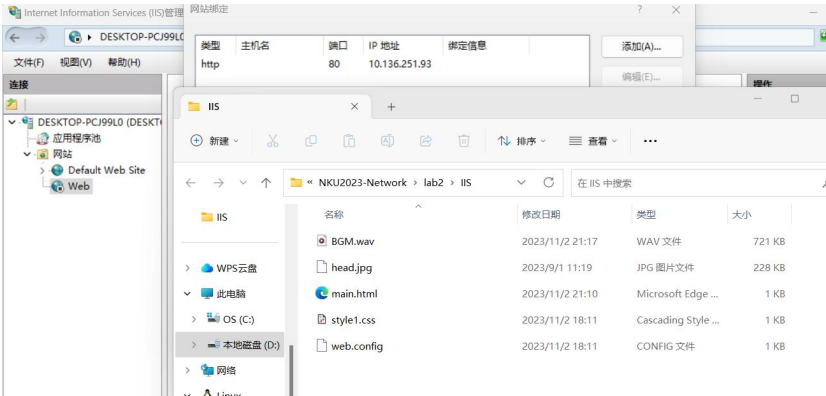
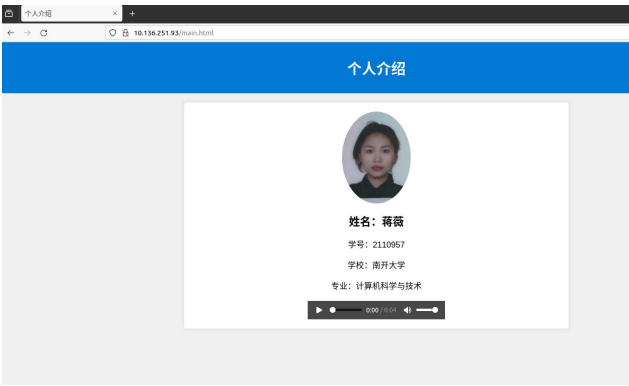


# 一、Web 服务器搭建（IIS）



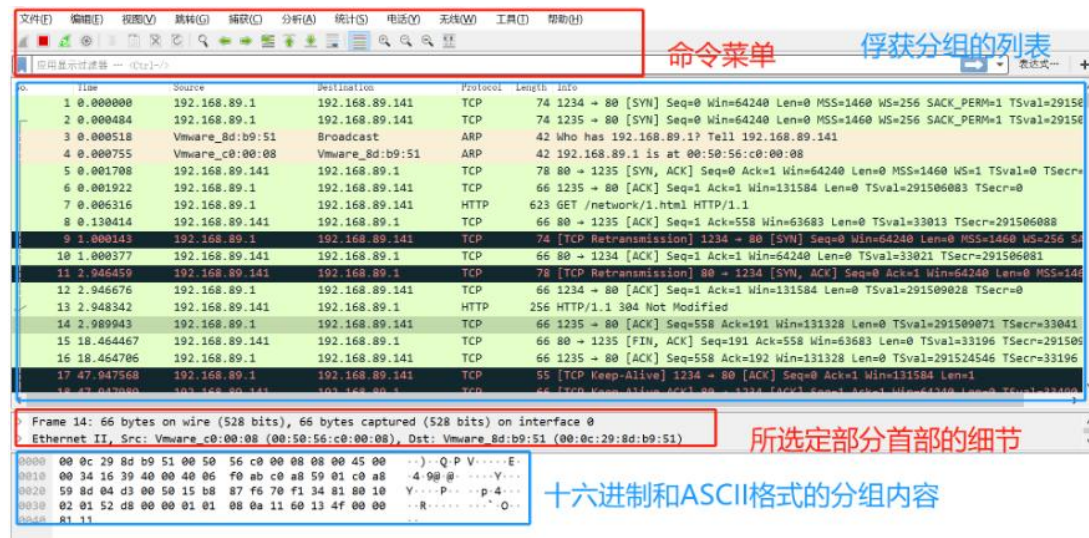
# 二、通过 Wireshark 捕获与 Web 服务器的交互过程

## 1、网页



```
1  <html>
2
3  <head>
4      <meta charset="UTF-8">
5      <title>个人介绍</title>
6      <link rel="stylesheet" href="style1.css">
7  </head>
8  <body>
9      <header>
10         <h1>个人介绍</h1>
11     </header>
12     <div class="container">
13         <div class="profile">
14             
15             <h2>姓名: 蒋薇</h2>
16             <p>学号: 2110957</p>
17             <p>学校: 南开大学</p>
18             <p>专业: 计算机科学与技术</p>
19         </div>
20         <div class="audio">
21             <audio controls>
22                 <source src="BGM.wav" type="audio/mpeg">
23             </audio>
24         </div>
25     </div>
26 </body>
27 </html>
28
```

## 2.wireshark



界面大致分为四个区：命令菜单区、俘获分组列表区、选定分组首部细节区、十六进制和 ASCII 格式分组内容区。

其中，命令菜单区的**应用显示过滤器**部分可以筛选显示的分组，例如：

服务器 IP 地址 .....: 10.136.251.93, 端口号 port = 80

浏览器访问，ip 地址：192.168.182.128, 端口号 port=57422

过滤器条件：

ip.addr == 192.168.182.128 && ip.addr == 10.136.251.93 && tcp.port == 80

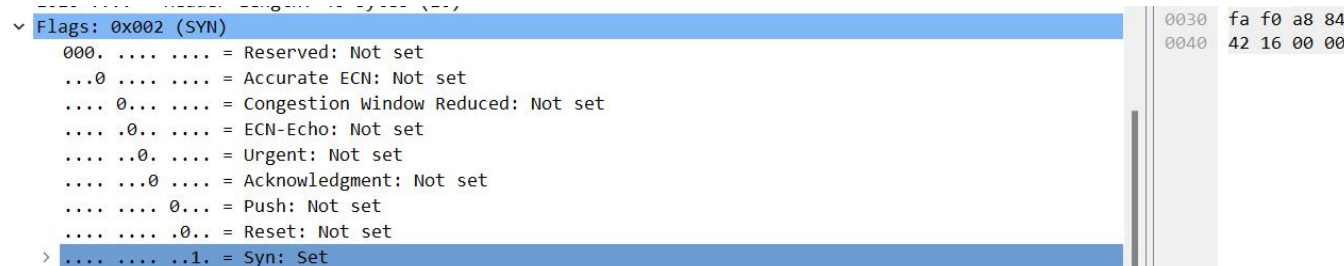
## 3.针对数据交互过程的数据包的关键属性说明

消息（4 个部分）

1. **Frame**，指的是==物理层的数据帧概况；
2. **Ethernet II**，第 0-13 个字节，表示数据层以太网帧头部信息，包含**目的地址、源地址和协议类型**
3. **Internet Protocol Version 4**，第 14-33 个字节，互联网层 IP 包头部信息
4. **Transmission Control Protocol**，第 34 个字节开始，传输层的数据段头部信息

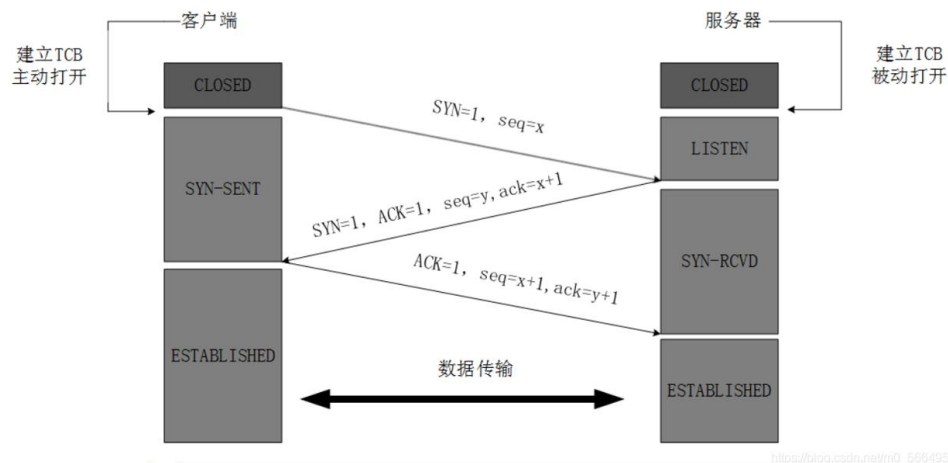
在 TCP 层，有个 FLAGS 字段，这个字段有以下几个标识：SYN, FIN, ACK, PSH, RST, URG。

其中，对于我们日常的分析有用的就是前面的五个字段。它们的含义是：SYN 表示建立连接，FIN 表示关闭连接，ACK 表示响应，PSH 表示有 DATA 数据传输，RST 表示连接重置。



**HTTP**，选取一条 HTTP 消息，消息格式为在原有 **TCP** 格式的基础上，增加超文本传输协议部分。

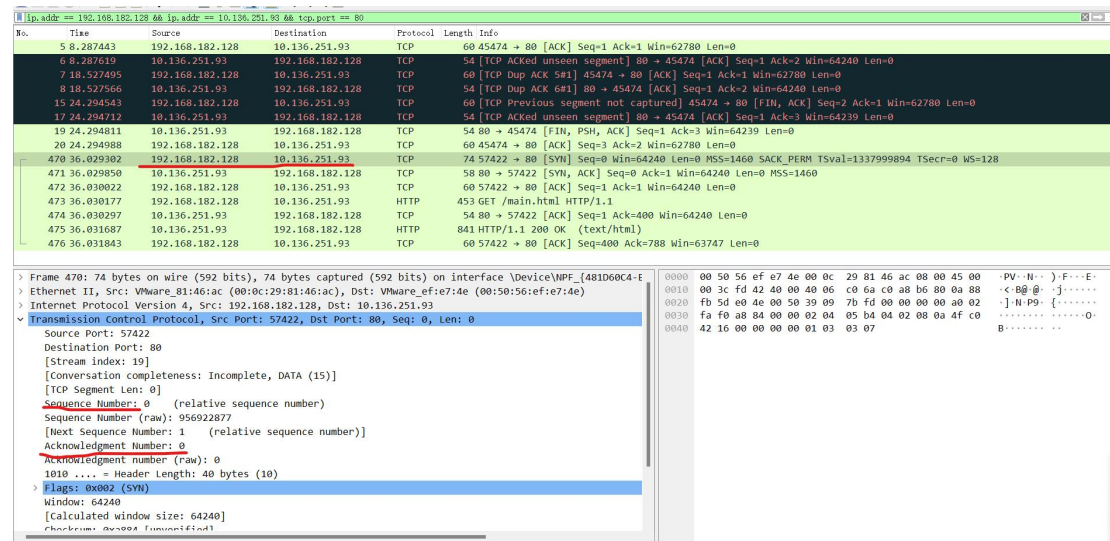
## 4.Tcp 三次握手：



SYN 表示建立连接，FIN 表示关闭连接，ACK 表示响应，PSH 表示有 DATA 数据传输，RST 表示连接重置。

### 1.第一次握手数据包

客户端发送一个 TCP，标志位为 SYN，序列号为 0，代表客户端请求建立连接。如下图。



SYN：标志位，表示请求建立连接，SYN

Seq = 0：初始建立连接值为 0，数据包的相对序列号从 0 开始，表示当前还没有发送数据

Ack = 0：初始建立连接值为 0，已经收到包的数量，表示当前没有接收到数据

### 2.第二次握手的数据包

服务器发回确认包，标志位为 SYN,ACK。将确认序号(Acknowledgement Number)设置为客户的 I S N 加 1 以.即 0+1=1

Wireshark packet capture showing a SYN-ACK packet (Frame 471) from 192.168.182.128 to 10.136.251.93. The packet details show the Transmission Control Protocol (TCP) segment with Sequence Number 0 and Acknowledgment Number 1. The packet length is 60 bytes.

No.	Time	Source	Destination	Protocol	Length	Info
5	0.287443	192.168.182.128	10.136.251.93	TCP	60	45474 → 80 [ACK] Seq=1 Ack=1 Win=62780 Len=0
6	0.287619	10.136.251.93	192.168.182.128	TCP	54	[TCP ACKed unseen segment] 80 → 45474 [ACK] Seq=1 Ack=2 Win=64240 Len=0
7	18.527495	192.168.182.128	10.136.251.93	TCP	60	[TCP Dup ACK 5#1] 45474 → 80 [ACK] Seq=1 Ack=1 Win=62780 Len=0
8	18.527566	10.136.251.93	192.168.182.128	TCP	54	[TCP Dup ACK 6#1] 80 → 45474 [ACK] Seq=1 Ack=2 Win=64240 Len=0
15	24.294543	192.168.182.128	10.136.251.93	TCP	60	[TCP Previous segment not captured] 45474 → 80 [FIN, ACK] Seq=2 Ack=1 Win=62780 Len=0
17	24.294712	10.136.251.93	192.168.182.128	TCP	54	[TCP ACKed unseen segment] 80 → 45474 [ACK] Seq=1 Ack=3 Win=64239 Len=0
19	24.294811	10.136.251.93	192.168.182.128	TCP	54	80 → 45474 [FIN, PSH, ACK] Seq=1 Ack=3 Win=64239 Len=0
20	24.294988	192.168.182.128	10.136.251.93	TCP	60	45474 → 80 [ACK] Seq=3 Ack=2 Win=62780 Len=0
470	36.029302	192.168.182.128	10.136.251.93	TCP	74	57422 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1337999894 TSecr=0 WS=128
471	36.029850	10.136.251.93	192.168.182.128	TCP	58	80 → 57422 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
472	36.030022	192.168.182.128	10.136.251.93	TCP	60	57422 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
473	36.030177	192.168.182.128	10.136.251.93	HTTP	453	GET /main.html HTTP/1.1
474	36.030297	10.136.251.93	192.168.182.128	TCP	54	80 → 57422 [ACK] Seq=1 Ack=400 Win=64240 Len=0
475	36.031687	10.136.251.93	192.168.182.128	HTTP	841	HTTP/1.1 200 OK (text/html)
476	36.031843	192.168.182.128	10.136.251.93	TCP	60	57422 → 80 [ACK] Seq=400 Ack=788 Win=63747 Len=0

Packet 471 details:

- Source Port: 80
- Destination Port: 57422
- [Stream index: 19]
- [Conversation completeness: Incomplete, DATA (15)]
- [TCP Segment Len: 0]
- Sequence Number: 0 (relative sequence number)
- Sequence Number (raw): 1591048935
- [Next Sequence Number: 1 (relative sequence number)]
- Acknowledgment Number: 1 (relative ack number)
- Acknowledgment number (raw): 956922878
- 010 .... = Header Length: 24 bytes (6)
- Flags: 0x012 (SYN, ACK)
- Window: 64240
- [calculated window size: 64240]

Flag = SYN,ACK

Seq = 0 : 初始建立值为 0, 表示当前还没有发送数据

Ack = 1: 表示当前端成功接收的数据位数, 虽然客户端没有发送任何有效数据, 确认号还是被加 1, 因为包含 SYN 或 FIN 标志位。(并不会对有效数据的计数产生影响, 因为含有 SYN 或 FIN 标志位的包并不携带有效数据)

### 3.第三次握手的数据包

客户端再次发送确认包(ACK) SYN 标志位为 0,ACK 标志位为 1.并且把服务器发来的 ACK 的序号字段+1,放在确定字段中发送给对方.并且在数据段放写 ISN 的+1,

Wireshark packet capture showing an ACK packet (Frame 472) from 192.168.182.128 to 10.136.251.93. The packet details show the Transmission Control Protocol (TCP) segment with Sequence Number 1 and Acknowledgment Number 1. The packet length is 60 bytes.

No.	Time	Source	Destination	Protocol	Length	Info
5	0.287443	192.168.182.128	10.136.251.93	TCP	60	45474 → 80 [ACK] Seq=1 Ack=1 Win=62780 Len=0
6	0.287619	10.136.251.93	192.168.182.128	TCP	54	[TCP ACKed unseen segment] 80 → 45474 [ACK] Seq=1 Ack=2 Win=64240 Len=0
7	18.527495	192.168.182.128	10.136.251.93	TCP	60	[TCP Dup ACK 5#1] 45474 → 80 [ACK] Seq=1 Ack=1 Win=62780 Len=0
8	18.527566	10.136.251.93	192.168.182.128	TCP	54	[TCP Dup ACK 6#1] 80 → 45474 [ACK] Seq=1 Ack=2 Win=64240 Len=0
15	24.294543	192.168.182.128	10.136.251.93	TCP	60	[TCP Previous segment not captured] 45474 → 80 [FIN, ACK] Seq=2 Ack=1 Win=62780 Len=0
17	24.294712	10.136.251.93	192.168.182.128	TCP	54	[TCP ACKed unseen segment] 80 → 45474 [ACK] Seq=1 Ack=3 Win=64239 Len=0
19	24.294811	10.136.251.93	192.168.182.128	TCP	54	80 → 45474 [FIN, PSH, ACK] Seq=1 Ack=3 Win=64239 Len=0
20	24.294988	192.168.182.128	10.136.251.93	TCP	60	45474 → 80 [ACK] Seq=3 Ack=2 Win=62780 Len=0
470	36.029302	192.168.182.128	10.136.251.93	TCP	74	57422 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1337999894 TSecr=0 WS=128
471	36.029850	10.136.251.93	192.168.182.128	TCP	58	80 → 57422 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
472	36.030022	192.168.182.128	10.136.251.93	TCP	60	57422 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
473	36.030177	192.168.182.128	10.136.251.93	HTTP	453	GET /main.html HTTP/1.1
474	36.030297	10.136.251.93	192.168.182.128	TCP	54	80 → 57422 [ACK] Seq=1 Ack=400 Win=64240 Len=0
475	36.031687	10.136.251.93	192.168.182.128	HTTP	841	HTTP/1.1 200 OK (text/html)
476	36.031843	192.168.182.128	10.136.251.93	TCP	60	57422 → 80 [ACK] Seq=400 Ack=788 Win=63747 Len=0

Packet 472 details:

- Source Port: 57422
- Destination Port: 80
- [Stream index: 19]
- [Conversation completeness: Incomplete, DATA (15)]
- [TCP Segment Len: 0]
- Sequence Number: 1 (relative sequence number)
- Sequence Number (raw): 956922878
- [Next Sequence Number: 1 (relative sequence number)]
- Acknowledgment Number: 1 (relative ack number)
- Acknowledgment number (raw): 1591048936
- 0101 .... = Header Length: 20 bytes (5)
- Flags: 0x010 (ACK)
- Window: 64240
- [calculated window size: 64240]

Flag = ACK : 标志位, 表示已经收到记录

Seq = 1 : 表示当前已经发送 1 个数据

Ack = 1: 表示当前端成功接收的数据位数, 虽然服务端没有发送任何有效数据, 确认号还是被加 1, 因为包含 SYN 或 FIN 标志位 (并不会对有效数据的计数产生影响, 因为含有 SYN 或 FIN 标志位的包并不携带有效数据)。

## 6.http



No.	Time	Source	Destination	Protocol	Length	Info
10	1.626747	10.136.148.155	10.136.148.155	HTTP	522	GET /main.html HTTP/1.1
12	1.627067	10.136.148.155	10.136.148.155	HTTP	843	HTTP/1.1 200 OK (text/html)
17	1.638030	10.136.148.155	10.136.148.155	HTTP	429	GET /style1.css HTTP/1.1
19	1.638213	10.136.148.155	10.136.148.155	HTTP	462	GET /head.jpg HTTP/1.1
21	1.638492	10.136.148.155	10.136.148.155	HTTP	736	HTTP/1.1 200 OK (text/css)
28	1.639735	10.136.148.155	10.136.148.155	HTTP	36882	HTTP/1.1 200 OK (JPEG JFIF image)
30	1.694221	10.136.148.155	10.136.148.155	HTTP	434	GET /BGM.wav HTTP/1.1
54	1.792117	10.136.148.155	10.136.148.155	HTTP	16513	HTTP/1.1 200 Partial Content (audio/wav)

> Frame 10: 522 bytes on wire (4176 bits), 522 bytes captured (4176 bits) on interface \Device\NPF_{...} id 0	0020 f6 c4
> Null/loopback	0030 2f 6d
> Internet Protocol Version 4, Src: 10.136.148.155, Dst: 10.136.148.155	0040 31 2e
> Transmission Control Protocol, Src Port: 51887, Dst Port: 80, Seq: 1, Ack: 478	0050 36 2e
> Hypertext Transfer Protocol	0060 63 74
GET /main.html HTTP/1.1	0070 65 0d
Host: 10.136.148.155	0080 75 72
Connection: keep-alive	0090 0a 55
Upgrade-Insecure-Requests: 1	00a0 69 6c
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Safari/537.36 Edg/117.0.2045.6	00b0 73 20
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7	00c0 30 20
Accept-Encoding: gzip, deflate	00d0 69 74
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6	00e0 2c 20
[Full request URI: http://10.136.148.155/main.html]	00f0 72 6f
[HTTP request 1/2]	0100 01 66
	0110 2f 31
	0120 41 63
	0130 6c 2c

名称	修改日期	类型	大小
BGM.wav	2023/11/2 21:17	WAV 文件	721
head.jpg	2023/9/1 11:19	JPG 图片文件	228
main.html	2023/11/2 21:10	Microsoft Edge ...	1
style1.css	2023/11/2 18:11	Cascading Style ...	1
web.config	2023/11/2 18:11	CONFIG 文件	1

```

GET /static/log/2.0/css/log.css?ver=1509012186 HTTP/1.1
Host: static02.babytreeimg.com
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.102 Safari/537.36
Accept: text/css,*/*;q=0.1
Referer: http://www.babytree.com/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
[Full request URI: http://static02.babytreeimg.com/static/log/2.0/css/log.css?ver=1509012186]
HTTP request 1/61

```

GET 为请求方式，后面跟请求的内容（这个地方可以看成是一个网页），协议版本  
 Host: static02.babytreeimg.com 请求的主机名  
 Connection: keep-alive 客户端与服务端指定的请求，响应有关选项（保持连接）  
 User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.102 Safari/537.36 发送请求的操作  
 Accept: text/css,\*/\*;q=0.1 客户端可识别的内容类型列表，用于指定客户端接受哪些类型的信息  
 Referer: http://www.babytree.com/ 判断来源页面  
 Accept-Encoding: gzip, deflate 客户端可识别的数据编码  
 Accept-Language: zh-CN,zh;q=0.9 浏览器所支持的语言类型  
 [Full request URI: http://static02.babytreeimg.com/static/log/2.0/css/log.css?ver=1509012186]  
 HTTP request 1/61

```

HTTP/1.1 200 OK
Expires: Sat, 16 Nov 2019 13:47:11 GMT
Date: Fri, 16 Nov 2018 13:47:11 GMT
Server: nginx
Content-Type: image/png
Content-Length: 3173
Last-Modified: Thu, 12 May 2016 03:32:32 GMT
ETag: "5733f950-c65"
Cache-Control: max-age=31536000
Access-Control-Allow-Origin: *
Accept-Ranges: bytes
Age: 1
X-Via: 1.1 PShnaywtoh149:3 (Cdn Cache Server V2.0), 1.1 PSbjlgwtub21:1 (Cdn Cache Server V2.0)
Connection: keep-alive
[HTTP response 4/6]
[Time since request: 0.009049000 seconds]
[Prev request in frame: 695]
[Prev response in frame: 696]
[Request in frame: 1000]
[Next request in frame: 1458]

```

HTTP/1.1 200 OK 状态行，200表示客户端请求成功  
 Expires: Sat, 16 Nov 2019 13:47:11 GMT 表示在这个日期之后，响应过期  
 Date: Fri, 16 Nov 2018 13:47:11 GMT 响应时间  
 Server: nginx 服务器信息  
 Content-Type: image/png 告诉客户端实际返回的内容类型  
 Content-Length: 3173 消息主体的大小  
 Last-Modified: Thu, 12 May 2016 03:32:32 GMT 请求资源的最后修改时间  
 ETag: "5733f950-c65" 资源的特定版本的标识符  
 Cache-Control: max-age=31536000 缓存机制，这里的max-age表示设置缓存存储的最大周期  
 Access-Control-Allow-Origin: \* 指定该响应的资源是否被允许与给定的origin共享，\*表示允许所有资源可以访问，也可以跟一个路径  
 Accept-Ranges: bytes 用于标识下载中断时，可以尝试中断了的下载，值一般是0，或byte,0表示不支持  
 Age: 1 Age消息头的值通常接近于0，表示此消息对象刚从原始服务器获取不久；其他的值则是表示代理服务器当前的系统时间与此应答消息中的通用消息头 Date 的值之差。  
 X-Via: 1.1 PShnaywtoh149:3 (Cdn Cache Server V2.0), 1.1 PSbjlgwtub21:1 (Cdn Cache Server V2.0) 代理服务器信息  
 Connection: keep-alive 连接状态  
 [HTTP response 4/6]  
 [Time since request: 0.009049000 seconds]  
 [Prev request in frame: 695]  
 [Prev response in frame: 696]  
 [Request in frame: 1000]  
 [Next request in frame: 1458]

## 报文传输各层简要介绍

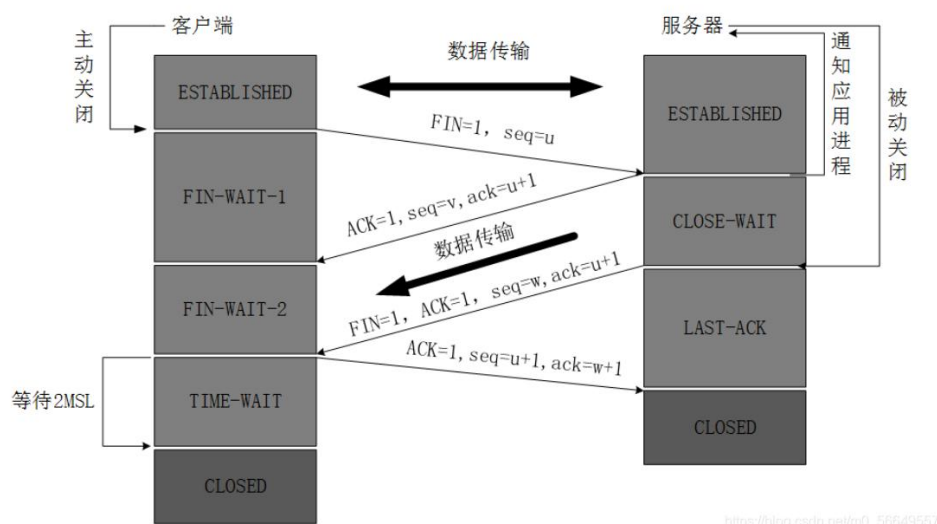
- Frame: 物理层的数据帧概况
- Ethernet II: 数据链路层以太网帧头部信息
- Internet Protocol Version 4: 互联网层 IP 包头部信息
- Transmission Control Protocol: 传输层 T 的数据段头部信息，此处是 TCP
- Hypertext Transfer Protocol: 应用层的信息，此处是 HTTP 协议

Get 请求，返回协议码 200 表示请求成功；若出现协议码 304 为有缓存且未修改。  
Style.css 文件美化页面，head.jpg 为图片文件，BGM.wav 为音频文件

音频文件，响应码 206 表示 Partial Content，表示服务器成功处理了部分 GET 请求。这通常用于断点续传或者分块下载时，客户端只请求资源的一部分。服务器会在应头中包含 Content-Range 字段，指示返回的是资源的哪一部分。

TCP 三次握手，建立了连接，发起 http 的一个连接，开始进行数据交互，中间发的都是正常的数据包，直到客户端发完数据包了，客户端发起一个 fin, ack 包开始四次挥手

## 5、Tcp 四次挥手



No.	Time	Source	Destination	Protocol	Length	Info
2405	8.420281	10.136.251.93	35.224.170.84	TCP	66	61596 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2536	8.734682	35.224.170.84	10.136.251.93	TCP	60	80 → 61596 [SYN, ACK] Seq=0 Ack=1 Win=65320 Len=0 MSS=1420
2537	8.734984	10.136.251.93	35.224.170.84	TCP	54	61596 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
2538	8.736035	10.136.251.93	35.224.170.84	HTTP	141	GET / HTTP/1.1
2572	9.046220	35.224.170.84	10.136.251.93	TCP	60	80 → 61596 [ACK] Seq=1 Ack=88 Win=64153 Len=0
2573	9.046220	35.224.170.84	10.136.251.93	TCP	202	HTTP/1.1 204 No Content
2574	9.046220	35.224.170.84	10.136.251.93	TCP	60	80 → 61596 [FIN, ACK] Seq=149 Ack=88 Win=64153 Len=0
2575	9.046431	10.136.251.93	35.224.170.84	TCP	54	61596 → 80 [ACK] Seq=88 Ack=150 Win=64092 Len=0
2576	9.047146	10.136.251.93	35.224.170.84	TCP	54	61596 → 80 [FIN, ACK] Seq=88 Ack=150 Win=64092 Len=0
2680	9.368236	35.224.170.84	10.136.251.93	TCP	60	80 → 61596 [ACK] Seq=150 Ack=89 Win=64152 Len=0
3253	11.268509	10.136.251.93	110.242.69.174	TCP	54	61568 → 80 [FIN, ACK] Seq=1 Ack=1 Win=513 Len=0
3254	11.269603	110.242.69.174	10.136.251.93	TCP	60	80 → 61568 [FIN, ACK] Seq=1 Ack=1 Win=980 Len=0
3255	11.269722	10.136.251.93	110.242.69.174	TCP	54	61568 → 80 [ACK] Seq=2 Ack=2 Win=513 Len=0
3258	11.279983	110.242.69.174	10.136.251.93	TCP	60	80 → 61568 [ACK] Seq=2 Ack=2 Win=980 Len=0
4529	17.417047	10.136.251.93	34.107.221.82	TCP	66	61597 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
4542	17.544450	10.136.251.93	10.136.128.1	TCP	66	61599 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM

> Frame 2680: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF\_{B42EC848-94...}

> Ethernet II, Src: IETF-VRRP-VRID 00 (00:00:5e:00:01:00), Dst: 8a:0c:da:48:45:aa (8a:0c:da:48:45:aa)

> Internet Protocol Version 4, Src: 35.224.170.84, Dst: 10.136.251.93

> Transmission Control Protocol, Src Port: 80, Dst Port: 61596, Seq: 150, Ack: 89, Len: 0

0000 8a 0c da 48 45 aa 00 00 5e 00 01 00 00 00 45 00 ...3HE

0010 00 28 f8 24 40 00 33 06 7b 91 23 e0 aa 54 0a 88 ...(-\$0

0020 fb 5d 00 50 f0 9c cf d6 27 9d 26 bf 6b c3 50 10 ...]-P-

0030 fa 98 66 3e 00 00 00 00 00 00 00 00 00 00 00 ...f>.

### 1.客户端发的第一个释放连接的请求

No.	Time	Source	Destination	Protocol	Length	Info
2405	8.420281	10.136.251.93	35.224.170.84	TCP	66	61596 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2536	8.734682	35.224.170.84	10.136.251.93	TCP	60	80 → 61596 [SYN, ACK] Seq=0 Ack=1 Win=65320 Len=0 MSS=1420
2537	8.734984	10.136.251.93	35.224.170.84	TCP	54	61596 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
2538	8.736035	10.136.251.93	35.224.170.84	HTTP	141	GET / HTTP/1.1
2572	9.046220	35.224.170.84	10.136.251.93	TCP	60	80 → 61596 [ACK] Seq=1 Ack=88 Win=64153 Len=0
2573	9.046220	35.224.170.84	10.136.251.93	HTTP	202	HTTP/1.1 204 No Content
2574	9.046220	35.224.170.84	10.136.251.93	TCP	60	80 → 61596 [FIN, ACK] Seq=149 Ack=88 Win=64153 Len=0
2575	9.046431	10.136.251.93	35.224.170.84	TCP	54	61596 → 80 [ACK] Seq=88 Ack=150 Win=64092 Len=0
2576	9.047146	10.136.251.93	35.224.170.84	TCP	54	61596 → 80 [FIN, ACK] Seq=88 Ack=150 Win=64092 Len=0
2680	9.368236	35.224.170.84	10.136.251.93	TCP	60	80 → 61596 [ACK] Seq=150 Ack=80 Win=64152 Len=0
3253	11.268509	10.136.251.93	110.242.69.174	TCP	54	61568 → 80 [FIN, ACK] Seq=1 Ack=1 Win=513 Len=0
3254	11.269603	110.242.69.174	10.136.251.93	TCP	60	80 → 61568 [FIN, ACK] Seq=1 Ack=1 Win=980 Len=0
3255	11.269722	10.136.251.93	110.242.69.174	TCP	54	61568 → 80 [ACK] Seq=2 Ack=2 Win=513 Len=0
3258	11.279983	110.242.69.174	10.136.251.93	TCP	60	80 → 61568 [ACK] Seq=2 Ack=2 Win=980 Len=0
4529	17.417047	10.136.251.93	34.107.221.82	TCP	66	61597 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
4542	17.544450	10.136.251.93	10.136.128.1	TCP	66	61599 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM

> Frame 2574: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{B42EC848-0000-0000-0000-000000000000} (00:00:00:00:00:00:00:00), Dst: 8a:0c:4a:48:45:aa (8a:0c:4a:48:45:aa)	0000 8a 0c 4a 48 45 aa 00 00 5e 00 01 08 08 00 45 00 --JHE-- ^...
> Ethernet II, Src: IETF-VRRP-VRID_08 (00:00:5e:00:01:08), Dst: 8a:0c:4a:48:45:aa (8a:0c:4a:48:45:aa)	0010 00 28 f8 23 40 00 33 06 7b 92 23 e0 aa 54 0a 88 --(-#03- {-#-
> Internet Protocol Version 4, Src: 35.224.170.84, Dst: 10.136.251.93	0020 fb 5d 00 50 f0 9c cf d6 27 9c 26 bf 6b c2 50 11 --:]P-... ';&-
> Transmission Control Protocol, Src Port: 80, Dst Port: 61596, Seq: 149, Ack: 88, Len: 0	0030 fa 99 66 3e 00 00 00 00 00 00 00 00 00 00 00 00 --:f-... ..-

Source Port: 80	
Destination Port: 61596	
[Stream index: 5]	
[Conversation completeness: Complete, WITH_DATA (31)]	
[TCP Segment Len: 0]	
Sequence Number: 149 (relative sequence number)	
Sequence Number (raw): 3486918556	
[Next Sequence Number: 150 (relative sequence number)]	
Acknowledgment Number: 88 (relative ack number)	
Acknowledgment number (raw): 650079170	
0101 .... = Header Length: 20 bytes (5)	
> Flags: 0x011 (FIN, ACK)	
Window: 64153	
[calculated window size: 64153]	

tcp 报文是一个可靠的协议，它的每一个数据包都要进行确认，每发一个数据包都有一个 ack 包。表示每发一个包，都要确认。Sequence Number,序号; Acknowledgment Number,确认号; Acknowledgment: Set, 对上一个报文的确认包; Fin:Set,fin 位被标记, 为释放连接的请求报文。

## 2.服务器给客户端回应确认消息

No.	Time	Source	Destination	Protocol	Length	Info
2405	8.420281	10.136.251.93	35.224.170.84	TCP	66	61596 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2536	8.734682	35.224.170.84	10.136.251.93	TCP	60	80 → 61596 [SYN, ACK] Seq=0 Ack=1 Win=65320 Len=0 MSS=1420
2537	8.734984	10.136.251.93	35.224.170.84	TCP	54	61596 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
2538	8.736035	10.136.251.93	35.224.170.84	HTTP	141	GET / HTTP/1.1
2572	9.046220	35.224.170.84	10.136.251.93	TCP	60	80 → 61596 [ACK] Seq=1 Ack=88 Win=64153 Len=0
2573	9.046220	35.224.170.84	10.136.251.93	HTTP	202	HTTP/1.1 204 No Content
2574	9.046220	35.224.170.84	10.136.251.93	TCP	60	80 → 61596 [FIN, ACK] Seq=149 Ack=88 Win=64153 Len=0
2575	9.046431	10.136.251.93	35.224.170.84	TCP	54	61596 → 80 [ACK] Seq=88 Ack=150 Win=64092 Len=0
2576	9.047146	10.136.251.93	35.224.170.84	TCP	54	61596 → 80 [FIN, ACK] Seq=88 Ack=150 Win=64092 Len=0
2680	9.368236	35.224.170.84	10.136.251.93	TCP	60	80 → 61596 [ACK] Seq=150 Ack=80 Win=64152 Len=0
3253	11.268509	10.136.251.93	110.242.69.174	TCP	54	61568 → 80 [FIN, ACK] Seq=1 Ack=1 Win=513 Len=0
3254	11.269603	110.242.69.174	10.136.251.93	TCP	60	80 → 61568 [FIN, ACK] Seq=1 Ack=1 Win=980 Len=0
3255	11.269722	10.136.251.93	110.242.69.174	TCP	54	61568 → 80 [ACK] Seq=2 Ack=2 Win=513 Len=0
3258	11.279983	110.242.69.174	10.136.251.93	TCP	60	80 → 61568 [ACK] Seq=2 Ack=2 Win=980 Len=0
4529	17.417047	10.136.251.93	34.107.221.82	TCP	66	61597 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
4542	17.544450	10.136.251.93	10.136.128.1	TCP	66	61599 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM

> Frame 2575: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{B42EC848-0000-0000-0000-000000000000} (00:00:00:00:00:00:00:00), Dst: IETF-VRRP-VRID_08 (00:00:5e:00:01:08)	0000 00 00 5e 00 01 08 0a 0c 4a 48 45 aa 08 00 45 00 --^.....
> Ethernet II, Src: 8a:0c:4a:48:45:aa (8a:0c:4a:48:45:aa), Dst: IETF-VRRP-VRID_08 (00:00:5e:00:01:08)	0010 00 28 c4 06 40 00 00 06 00 00 0a 88 fb 5d 23 e0 --(L@-...
> Internet Protocol Version 4, Src: 10.136.251.93, Dst: 35.224.170.84	0020 aa 54 f0 9c 00 50 26 bf 6b c2 cf d6 27 9d 50 10 --T...P&-
> Transmission Control Protocol, Src Port: 61596, Dst Port: 80, Seq: 88, Ack: 150, Len: 0	0030 fa 5c d4 34 00 00 00 00 00 00 00 00 00 00 00 00 --\4...

Source Port: 61596	
Destination Port: 80	
[Stream index: 5]	
[Conversation completeness: Complete, WITH_DATA (31)]	
[TCP Segment Len: 0]	
Sequence Number: 88 (relative sequence number)	
Sequence Number (raw): 650079170	
[Next Sequence Number: 88 (relative sequence number)]	
Acknowledgment Number: 150 (relative ack number)	
Acknowledgment number (raw): 3486918557	
0101 .... = Header Length: 20 bytes (5)	
> Flags: 0x010 (ACK)	
Window: 64092	
[calculated window size: 64092]	

只有 ack 位被标记了，其它位没有被标记，因为这就是一个确认消息。

ack 包的序号是 150，是因为客户端发的 fin，ack 包希望下一个包的序号是 150（确认号是 150）

## 3.服务器发给客户端释放连接的请求



No.	Time	Source	Destination	Protocol	Length	Info
2405	8.420281	10.136.251.93	35.224.170.84	TCP	66	61596 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2536	8.734682	35.224.170.84	10.136.251.93	TCP	60	80 → 61596 [SYN, ACK] Seq=0 Ack=1 Win=65320 Len=0 MSS=1420
2537	8.734984	10.136.251.93	35.224.170.84	TCP	54	61596 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
2538	8.736035	10.136.251.93	35.224.170.84	HTTP	141	GET / HTTP/1.1
2572	9.046220	35.224.170.84	10.136.251.93	TCP	60	80 → 61596 [ACK] Seq=1 Ack=88 Win=64153 Len=0
2573	9.046220	35.224.170.84	10.136.251.93	HTTP	202	HTTP/1.1 204 No Content
2574	9.046220	35.224.170.84	10.136.251.93	TCP	60	80 → 61596 [FIN, ACK] Seq=149 Ack=88 Win=64153 Len=0
2575	9.046431	10.136.251.93	35.224.170.84	TCP	54	61596 → 80 [ACK] Seq=88 Ack=150 Win=64092 Len=0
2576	9.047146	10.136.251.93	35.224.170.84	TCP	54	61596 → 80 [FIN, ACK] Seq=88 Ack=150 Win=64092 Len=0
2680	9.368236	35.224.170.84	10.136.251.93	TCP	60	80 → 61596 [ACK] Seq=150 Ack=89 Win=64152 Len=0
3253	11.268509	10.136.251.93	110.242.69.174	TCP	54	61568 → 80 [FIN, ACK] Seq=1 Ack=1 Win=513 Len=0
3254	11.269603	110.242.69.174	10.136.251.93	TCP	60	80 → 61568 [FIN, ACK] Seq=1 Ack=1 Win=980 Len=0
3255	11.269722	10.136.251.93	110.242.69.174	TCP	54	61568 → 80 [ACK] Seq=2 Ack=2 Win=513 Len=0
3258	11.279983	110.242.69.174	10.136.251.93	TCP	60	80 → 61568 [ACK] Seq=2 Ack=2 Win=980 Len=0
4529	17.417047	10.136.251.93	34.107.221.82	TCP	66	61597 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
4542	17.544450	10.136.251.93	10.136.128.1	TCP	66	61599 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM

> Frame 2576: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{B42EC848-0010-00-28-4C} 00 00 5e 00 01 08 8a 0c 4a 0010 00 28 4c 07 40 00 80 06 000020 aa 54 f0 9c 00 50 26 bf 6b0030 fa 5c d4 34 00 00
---

> Ethernet II, Src: 8a:0c:4a:48:45:aa (8a:0c:4a:48:45:aa), Dst: IETF-VRRP-VRID_08 (00:00:5e:00:01:08)
> Internet Protocol Version 4, Src: 10.136.251.93, Dst: 35.224.170.84
> Transmission Control Protocol, Src Port: 61596, Dst Port: 80, Seq: 88, Ack: 150, Len: 0
Source Port: 61596
Destination Port: 80
[Stream index: 5]
[Conversation completeness: Complete, WITH_DATA (31)]
[TCP Segment Len: 0]
Sequence Number: 88 (relative sequence number)
Sequence Number (raw): 650079170
[Next Sequence Number: 89 (relative sequence number)]
Acknowledgment Number: 150 (relative ack number)
Acknowledgment number (raw): 3486918557
0101 .... = Header Length: 20 bytes (5)
> Flags: 0x011 (FIN, ACK)
Window: 64092
[Calculated window size: 64092]
Window size scaling factor: 2 (window scaling used)

再次发送一个 fin, ack 包，表示我也要释放连接；fin, ack 和 ack 包的序号、确认号是一样的；因为是一个服务器发的，所以确认号和序号是一样的；fin 位被置位，也是要释放连接到。

#### 4.客户端发送确认消息

No.	Time	Source	Destination	Protocol	Length	Info
2405	8.420281	10.136.251.93	35.224.170.84	TCP	66	61596 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2536	8.734682	35.224.170.84	10.136.251.93	TCP	60	80 → 61596 [SYN, ACK] Seq=0 Ack=1 Win=65320 Len=0 MSS=1420
2537	8.734984	10.136.251.93	35.224.170.84	TCP	54	61596 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
2538	8.736035	10.136.251.93	35.224.170.84	HTTP	141	GET / HTTP/1.1
2572	9.046220	35.224.170.84	10.136.251.93	TCP	60	80 → 61596 [ACK] Seq=1 Ack=88 Win=64153 Len=0
2573	9.046220	35.224.170.84	10.136.251.93	HTTP	202	HTTP/1.1 204 No Content
2574	9.046220	35.224.170.84	10.136.251.93	TCP	60	80 → 61596 [FIN, ACK] Seq=149 Ack=88 Win=64153 Len=0
2575	9.046431	10.136.251.93	35.224.170.84	TCP	54	61596 → 80 [ACK] Seq=88 Ack=150 Win=64092 Len=0
2576	9.047146	10.136.251.93	35.224.170.84	TCP	54	61596 → 80 [FIN, ACK] Seq=88 Ack=150 Win=64092 Len=0
2680	9.368236	35.224.170.84	10.136.251.93	TCP	60	80 → 61596 [ACK] Seq=150 Ack=89 Win=64152 Len=0
3253	11.268509	10.136.251.93	110.242.69.174	TCP	54	61568 → 80 [FIN, ACK] Seq=1 Ack=1 Win=513 Len=0
3254	11.269603	110.242.69.174	10.136.251.93	TCP	60	80 → 61568 [FIN, ACK] Seq=1 Ack=1 Win=980 Len=0
3255	11.269722	10.136.251.93	110.242.69.174	TCP	54	61568 → 80 [ACK] Seq=2 Ack=2 Win=513 Len=0
3258	11.279983	110.242.69.174	10.136.251.93	TCP	60	80 → 61568 [ACK] Seq=2 Ack=2 Win=980 Len=0
4529	17.417047	10.136.251.93	34.107.221.82	TCP	66	61597 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
4542	17.544450	10.136.251.93	10.136.128.1	TCP	66	61599 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM

> Frame 2680: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{B42EC848-0010-00-28-4C} 00 00 5e 00 01 08 8a 0c 4a 0010 00 28 4c 00 33 06 7b 91 23 e0 aa 54 8a 880020 fb 5d 00 50 f0 9c cf d6 27 9d 26 bf 6b c3 50 100030 fa 98 66 3e 00 00 00 00 00 00 00
--

> Ethernet II, Src: IETF-VRRP-VRID_08 (00:00:5e:00:01:08), Dst: 8a:0c:4a:48:45:aa (8a:0c:4a:48:45:aa)
> Internet Protocol Version 4, Src: 35.224.170.84, Dst: 10.136.251.93
> Transmission Control Protocol, Src Port: 80, Dst Port: 61596, Seq: 150, Ack: 89, Len: 0
Source Port: 80
Destination Port: 61596
[Stream index: 5]
[Conversation completeness: Complete, WITH_DATA (31)]
[TCP Segment Len: 0]
Sequence Number: 150 (relative sequence number)
Sequence Number (raw): 3486918557
[Next Sequence Number: 150 (relative sequence number)]
Acknowledgment Number: 89 (relative ack number)
Acknowledgment number (raw): 650079171
0101 .... = Header Length: 20 bytes (5)
> Flags: 0x010 (ACK)
Window: 64152
[Calculated window size: 64152]

客户端发了一个 ack 包。这个包的 ack 号等于上一个包的序列号加 1。

### 三、总结

1.当第一次请求页面并请求成功时，页面会返回状态码 200 表示请求成功，并同时返回页面的 html 内容；当再次请求且页面没有修改时会返回 304 表示页面未修改可以直接使用浏览器缓存的内容

2.传输流程整体：

客户端发出请求，服务器在局域网内（因为使用的是本地的虚拟机）发送查找网卡请求并找到 ip 对应的物理地址；

三次握手建立连接；



进行页面请求，服务器返回 html 内容;  
请求并返回文字、图片、音频等内容;  
四次挥手断开连接。

HTTP 1.1 与 HTTP 1.0 是两个不同版本的 HTTP 协议,

持久连接为其重要特性, HTTP 1.1 引入了持久连接 (Persistent Connections) 的概念, 允许多个请求和响应在同一个 TCP 连接上进行, 减少了连接的建立和关闭的开销;

2.虚拟主机支持: HTTP 1.1 支持虚拟主机 (Virtual Hosting), 允许在同一个服务器上运行多个网站, 通过 Host 头字段来区分不同的网站;

3.缓存控制: HTTP 1.1 引入了更强大的缓存控制机制, 包括 Cache-Control 头字段和 ETag 头字段;