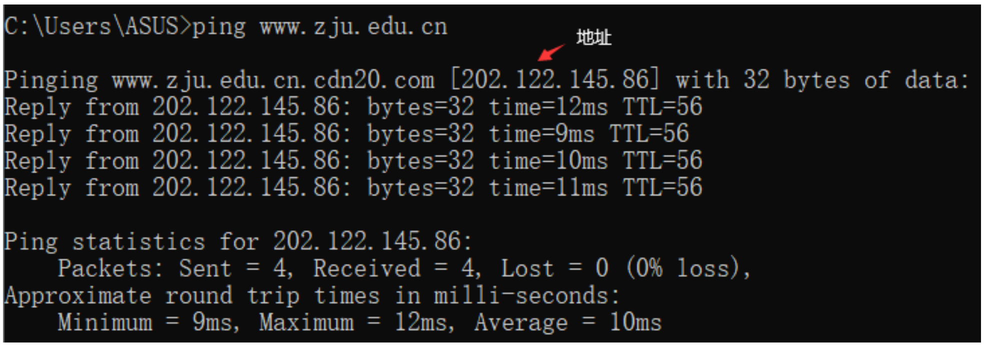
信息安全原理 HW4

姓名：姚熙源　　学号：3190300677

实验过程与结果分析

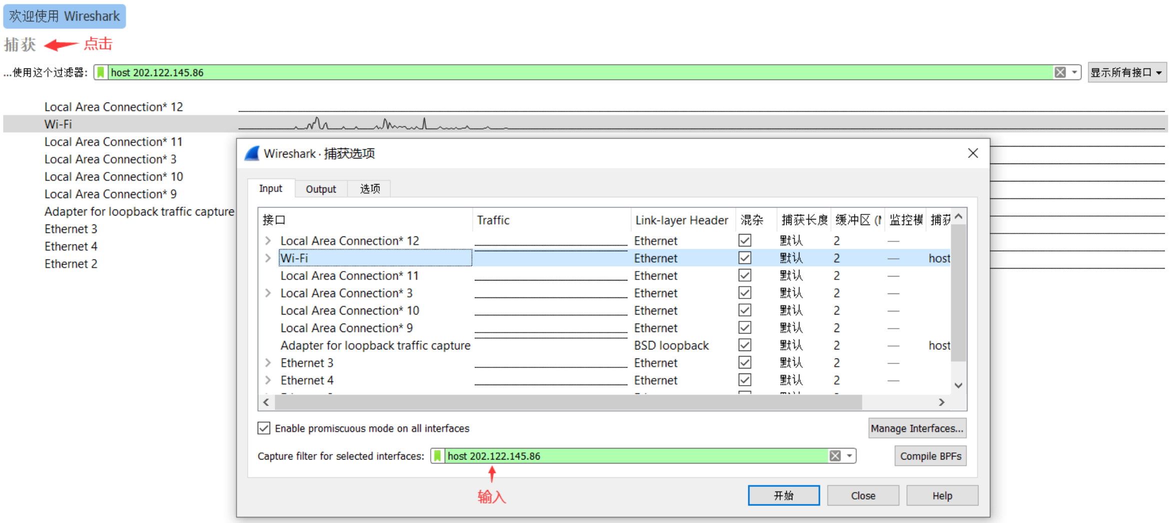
1. 下载并安装 Wireshark
2. 获取 www.zju.edu.cn 网站的服务器地址，在 cmd 中 ping 这个网址即可



```
C:\Users\ASUS>ping www.zju.edu.cn

Pinging www.zju.edu.cn.cdn20.com [202.122.145.86] with 32 bytes of data:
Reply from 202.122.145.86: bytes=32 time=12ms TTL=56
Reply from 202.122.145.86: bytes=32 time=9ms TTL=56
Reply from 202.122.145.86: bytes=32 time=10ms TTL=56
Reply from 202.122.145.86: bytes=32 time=11ms TTL=56

Ping statistics for 202.122.145.86:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 9ms, Maximum = 12ms, Average = 10ms
```

该网站(www.zju.edu.cn)的地址为 202.122.145.86

3. 打开 Wireshark，在界面中点击捕获，然后在下方的过滤器中输入 host 202.122.145.86，点击开始来抓取该网站的包。



…使用这个过滤器： host 202.122.145.86

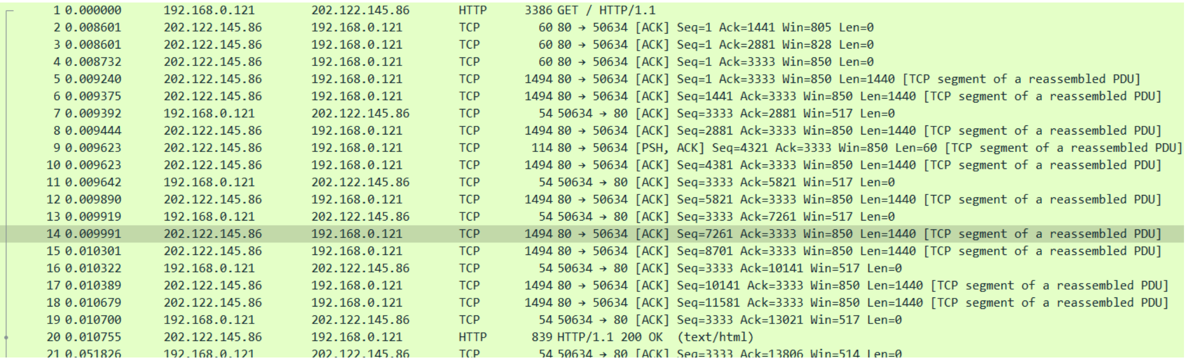
Local Area Connection* 12
Wi-Fi
Local Area Connection* 11
Local Area Connection* 3
Local Area Connection* 10
Local Area Connection* 9
Adapter for loopback traffic capture
Ethernet 3
Ethernet 4
Ethernet 2

接口	Traffic	Link-layer Header	混杂	捕获长度	缓冲区 (I	监控模	捕获
Local Area Connection* 12		Ethernet	☑	默认	2	—	
Wi-Fi		Ethernet	☑	默认	2	—	host
Local Area Connection* 11		Ethernet	☑	默认	2	—	
Local Area Connection* 3		Ethernet	☑	默认	2	—	
Local Area Connection* 10		Ethernet	☑	默认	2	—	
Local Area Connection* 9		Ethernet	☑	默认	2	—	
Adapter for loopback traffic capture		BSD loopback	☑	默认	2	—	host
Ethernet 3		Ethernet	☑	默认	2	—	
Ethernet 4		Ethernet	☑	默认	2	—	

☑ Enable promiscuous mode on all interfaces

Capture filter for selected interfaces: host 202.122.145.86

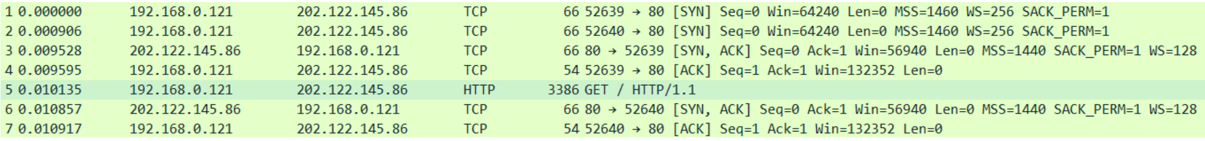
开始抓包后的界面：



```
 1 0.000000   192.168.0.121    202.122.145.86   HTTP   3386 GET / HTTP/1.1
 2 0.008601   202.122.145.86   192.168.0.121    TCP      60 80 → 50634 [ACK] Seq=1 Ack=1441 Win=805 Len=0
 3 0.008601   202.122.145.86   192.168.0.121    TCP      60 80 → 50634 [ACK] Seq=1 Ack=2881 Win=828 Len=0
 4 0.008732   202.122.145.86   192.168.0.121    TCP      60 80 → 50634 [ACK] Seq=1 Ack=3333 Win=850 Len=0
 5 0.009240   202.122.145.86   192.168.0.121    TCP    1494 80 → 50634 [ACK] Seq=1 Ack=3333 Win=850 Len=1440 [TCP segment of a reassembled PDU]
 6 0.009375   202.122.145.86   192.168.0.121    TCP    1494 80 → 50634 [ACK] Seq=1441 Ack=3333 Win=850 Len=1440 [TCP segment of a reassembled PDU]
 7 0.009392   192.168.0.121    202.122.145.86   TCP      54 50634 → 80 [ACK] Seq=3333 Ack=2881 Win=517 Len=0
 8 0.009444   202.122.145.86   192.168.0.121    TCP    1494 80 → 50634 [ACK] Seq=2881 Ack=3333 Win=850 Len=1440 [TCP segment of a reassembled PDU]
 9 0.009623   202.122.145.86   192.168.0.121    TCP     114 80 → 50634 [PSH, ACK] Seq=4321 Ack=3333 Win=850 Len=60 [TCP segment of a reassembled PDU]
10 0.009623   202.122.145.86   192.168.0.121    TCP    1494 80 → 50634 [ACK] Seq=4381 Ack=3333 Win=850 Len=1440 [TCP segment of a reassembled PDU]
11 0.009642   192.168.0.121    202.122.145.86   TCP      54 50634 → 80 [ACK] Seq=3333 Ack=5821 Win=517 Len=0
12 0.009890   202.122.145.86   192.168.0.121    TCP    1494 80 → 50634 [ACK] Seq=5821 Ack=3333 Win=850 Len=1440 [TCP segment of a reassembled PDU]
13 0.009919   192.168.0.121    202.122.145.86   TCP      54 50634 → 80 [ACK] Seq=3333 Ack=7261 Win=517 Len=0
14 0.009991   202.122.145.86   192.168.0.121    TCP    1494 80 → 50634 [ACK] Seq=7261 Ack=3333 Win=850 Len=1440 [TCP segment of a reassembled PDU]
15 0.010301   202.122.145.86   192.168.0.121    TCP    1494 80 → 50634 [ACK] Seq=8701 Ack=3333 Win=850 Len=1440 [TCP segment of a reassembled PDU]
16 0.010322   192.168.0.121    202.122.145.86   TCP      54 50634 → 80 [ACK] Seq=3333 Ack=10141 Win=517 Len=0
17 0.010389   202.122.145.86   192.168.0.121    TCP    1494 80 → 50634 [ACK] Seq=10141 Ack=3333 Win=850 Len=1440 [TCP segment of a reassembled PDU]
18 0.010679   202.122.145.86   192.168.0.121    TCP    1494 80 → 50634 [ACK] Seq=11581 Ack=3333 Win=850 Len=1440 [TCP segment of a reassembled PDU]
19 0.010700   192.168.0.121    202.122.145.86   TCP      54 50634 → 80 [ACK] Seq=3333 Ack=13021 Win=517 Len=0
20 0.010755   202.122.145.86   192.168.0.121    HTTP    839 HTTP/1.1 200 OK  (text/html)
21 0.051826   192.168.0.121    202.122.145.86   TCP      54 50634 → 80 [ACK] Seq=3333 Ack=13806 Win=514 Len=0
```

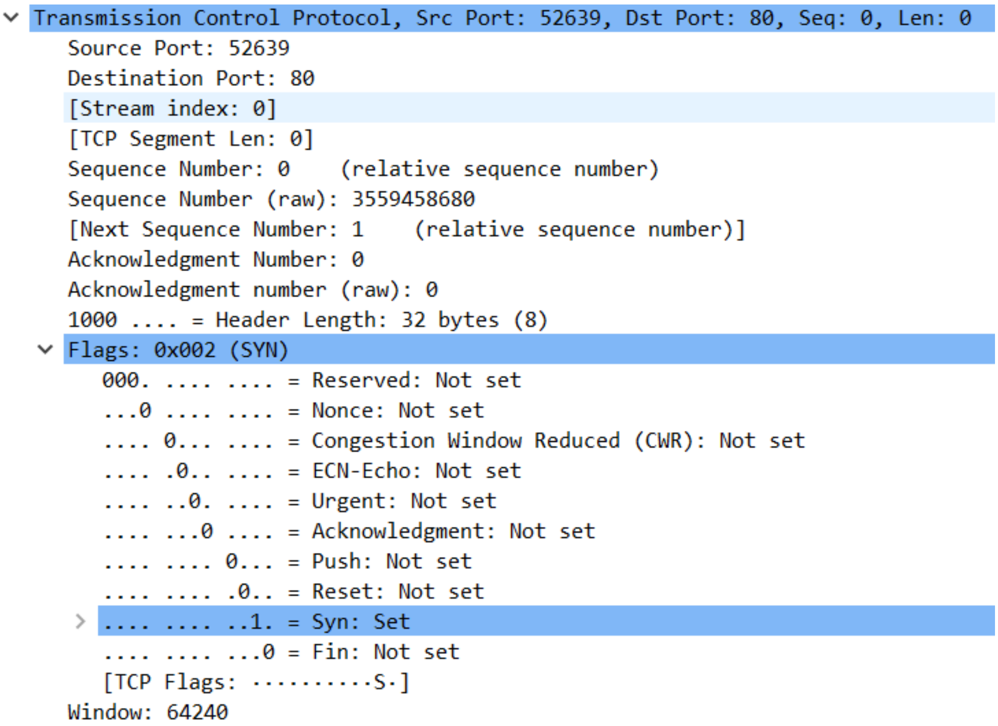
4. 对包进行分析

a. 建立 TCP（三次握手）



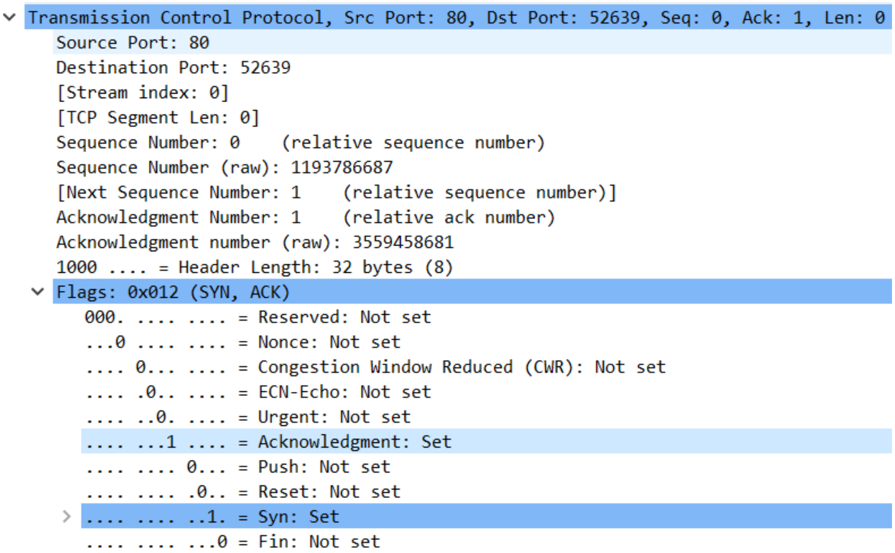
```
1 0.000000    192.168.0.121    202.122.145.86   TCP     66 52639 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2 0.000906    192.168.0.121    202.122.145.86   TCP     66 52640 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
3 0.009528    202.122.145.86   192.168.0.121    TCP     66 80 → 52639 [SYN, ACK] Seq=0 Ack=1 Win=56940 Len=0 MSS=1440 SACK_PERM=1 WS=128
4 0.009595    192.168.0.121    202.122.145.86   TCP     54 52639 → 80 [ACK] Seq=1 Ack=1 Win=132352 Len=0
5 0.010135    192.168.0.121    202.122.145.86   HTTP  3386 GET / HTTP/1.1
6 0.010857    202.122.145.86   192.168.0.121    TCP     66 80 → 52640 [SYN, ACK] Seq=0 Ack=1 Win=56940 Len=0 MSS=1440 SACK_PERM=1 WS=128
7 0.010917    192.168.0.121    202.122.145.86   TCP     54 52640 → 80 [ACK] Seq=1 Ack=1 Win=132352 Len=0
```

由上图可知道本机有两个端口分别为 52639 和 52640 都对网站的端口(80)建立了连接，我们以其中一个端口(52639)来分析与说明。由本机的 52639 端口发送请求到网站，我们可在下方的框架得到一些信息比如 source,destination,port 等信息，下面为 line1 的信息



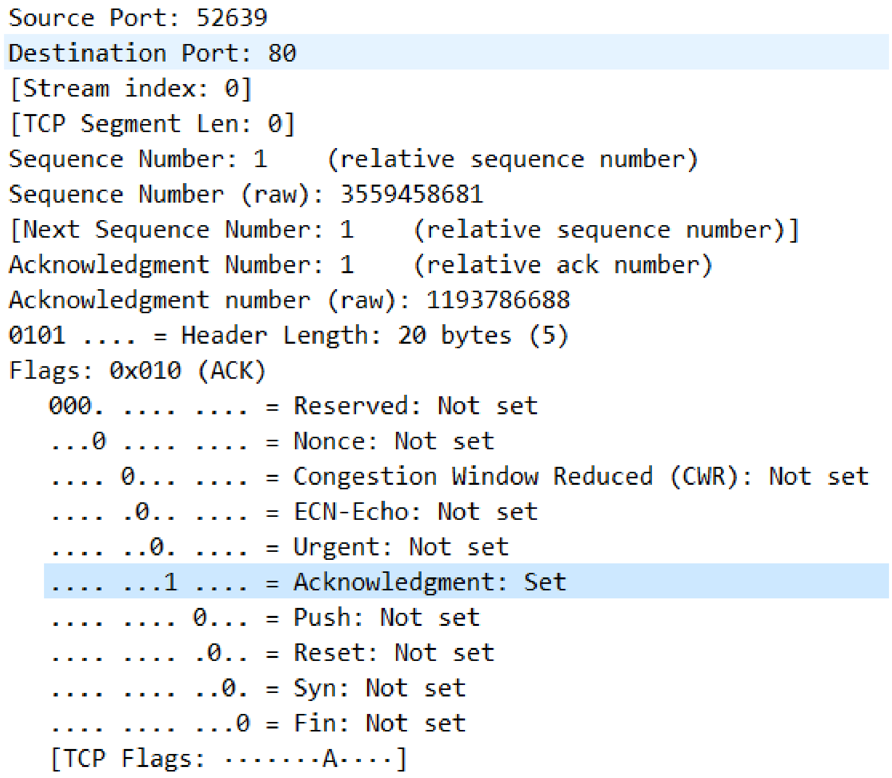
```
∨ Transmission Control Protocol, Src Port: 52639, Dst Port: 80, Seq: 0, Len: 0
     Source Port: 52639
     Destination Port: 80
     [Stream index: 0]
     [TCP Segment Len: 0]
     Sequence Number: 0    (relative sequence number)
     Sequence Number (raw): 3559458680
     [Next Sequence Number: 1    (relative sequence number)]
     Acknowledgment Number: 0
     Acknowledgment number (raw): 0
     1000 .... = Header Length: 32 bytes (8)
   ∨ Flags: 0x002 (SYN)
        000. .... .... = Reserved: Not set
        ...0 .... .... = Nonce: Not set
        .... 0... .... = Congestion Window Reduced (CWR): Not set
        .... .0.. .... = ECN-Echo: Not set
        .... ..0. .... = Urgent: Not set
        .... ...0 .... = Acknowledgment: Not set
        .... .... 0... = Push: Not set
        .... .... .0.. = Reset: Not set
      > .... .... ..1. = Syn: Set
        .... .... ...0 = Fin: Not set
        [TCP Flags: ··········S·]
     Window: 64240
```

其 Syn 已设置为 Set，表示建立新连接，发送 sequence(x）给服务器，然后再看看服务器对本机回复的这个 ACK 包(line3)和 Syn(sequence(y))来让本机确认序号有效(x+1)，其 Syn 和 Acknowledgment 都设置为 Set



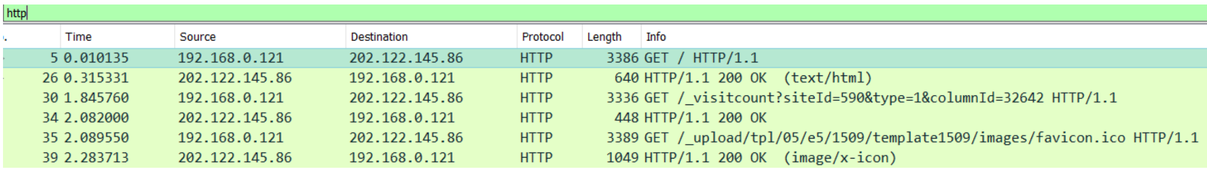
```
∨ Transmission Control Protocol, Src Port: 80, Dst Port: 52639, Seq: 0, Ack: 1, Len: 0
     Source Port: 80
     Destination Port: 52639
     [Stream index: 0]
     [TCP Segment Len: 0]
     Sequence Number: 0    (relative sequence number)
     Sequence Number (raw): 1193786687
     [Next Sequence Number: 1    (relative sequence number)]
     Acknowledgment Number: 1    (relative ack number)
     Acknowledgment number (raw): 3559458681
     1000 .... = Header Length: 32 bytes (8)
   ∨ Flags: 0x012 (SYN, ACK)
        000. .... .... = Reserved: Not set
        ...0 .... .... = Nonce: Not set
        .... 0... .... = Congestion Window Reduced (CWR): Not set
        .... .0.. .... = ECN-Echo: Not set
        .... ..0. .... = Urgent: Not set
        .... ...1 .... = Acknowledgment: Set
        .... .... 0... = Push: Not set
        .... .... .0.. = Reset: Not set
      > .... .... ..1. = Syn: Set
        .... .... ...0 = Fin: Not set
```

最后再进行最后一次的连接，由本机端口 52639 发送 ACK 包(y+1）到服务器端口来让服务器确认序号(y+1），其中 Acknowledgment 设置为 Set，然后 TCP 连接成功。



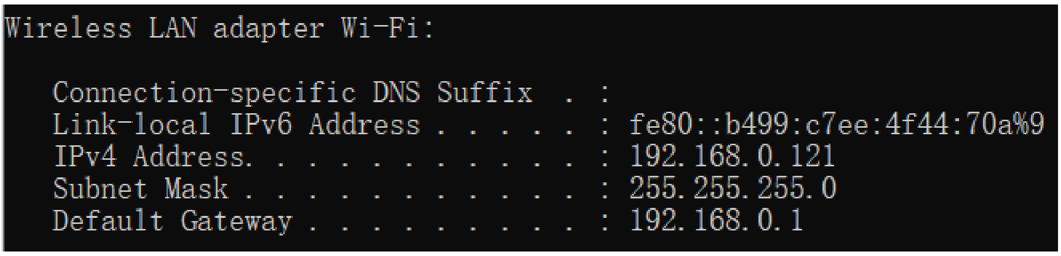
```
Source Port: 52639
Destination Port: 80
[Stream index: 0]
[TCP Segment Len: 0]
Sequence Number: 1    (relative sequence number)
Sequence Number (raw): 3559458681
[Next Sequence Number: 1    (relative sequence number)]
Acknowledgment Number: 1    (relative ack number)
Acknowledgment number (raw): 1193786688
0101 .... = Header Length: 20 bytes (5)
Flags: 0x010 (ACK)
   000. .... .... = Reserved: Not set
   ...0 .... .... = Nonce: Not set
   .... 0... .... = Congestion Window Reduced (CWR): Not set
   .... .0.. .... = ECN-Echo: Not set
   .... ..0. .... = Urgent: Not set
   .... ...1 .... = Acknowledgment: Set
   .... .... 0... = Push: Not set
   .... .... .0.. = Reset: Not set
   .... .... ..0. = Syn: Not set
   .... .... ...0 = Fin: Not set
   [TCP Flags: ·······A····]
```

b. HTTP 请求的连接，在上方的过滤器输入 http 可查看



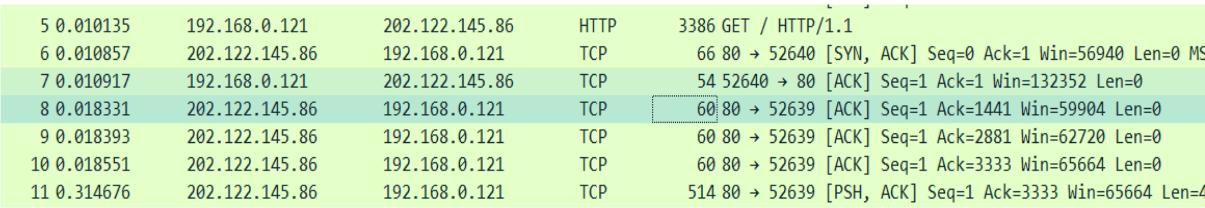
.	Time	Source	Destination	Protocol	Length	Info
5	0.010135	192.168.0.121	202.122.145.86	HTTP	3386	GET / HTTP/1.1
26	0.315331	202.122.145.86	192.168.0.121	HTTP	640	HTTP/1.1 200 OK (text/html)
30	1.845760	192.168.0.121	202.122.145.86	HTTP	3336	GET /_visitcount?siteId=590&type=1&columnId=32642 HTTP/1.1
34	2.082000	202.122.145.86	192.168.0.121	HTTP	448	HTTP/1.1 200 OK
35	2.089550	192.168.0.121	202.122.145.86	HTTP	3389	GET /_upload/tpl/05/e5/1509/template1509/images/favicon.ico HTTP/1.1
39	2.283713	202.122.145.86	192.168.0.121	HTTP	1049	HTTP/1.1 200 OK (image/x-icon)

一个是 GET 请求，一个是 RESPONSE 发送回我的 ip 地址，在 line5 中可知道 source 是我的 ip 地址（在 cmd 中输入 ipconfig 便可查看自己的 ip 地址），而 destination 是网站的服务器地址。在 line26，由服务器地址发送 RESPONSE 信息回给我的地址显示“OK”。



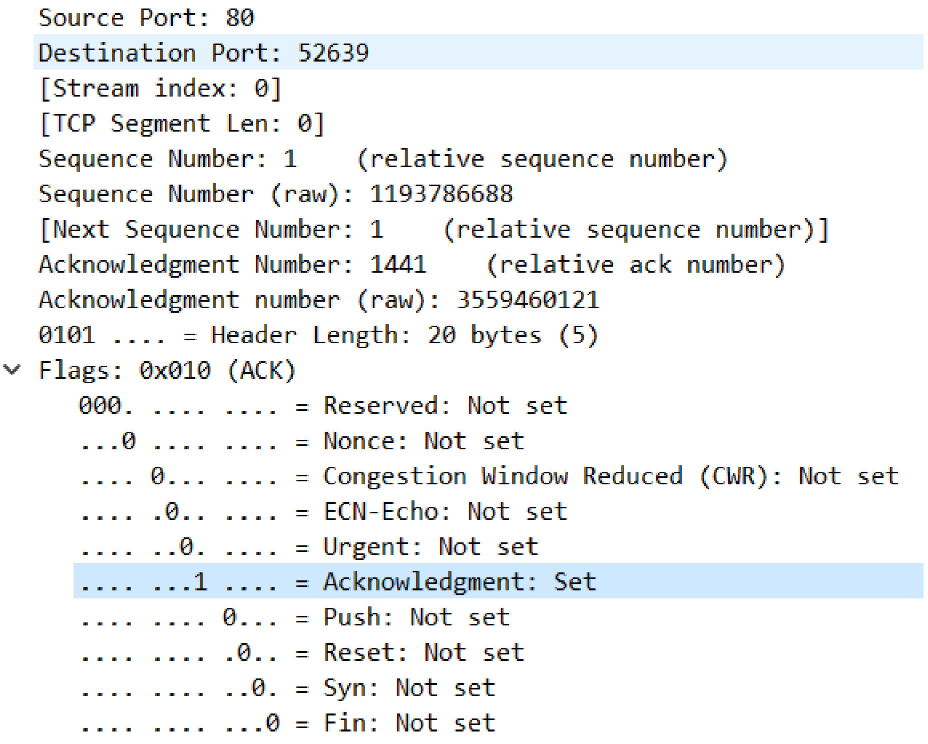
```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::b499:c7ee:4f44:70a%9
   IPv4 Address. . . . . . . . . . . : 192.168.0.121
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1
```

由本机发送 GET 请求，网站服务器端口在接收到请求后便会发送一个 ACK 包回给本机



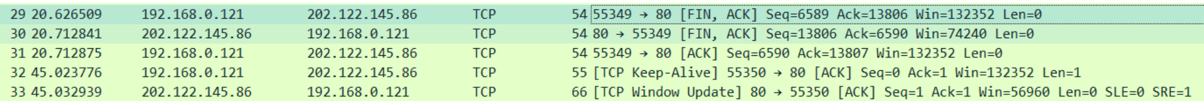
5	0.010135	192.168.0.121	202.122.145.86	HTTP	3386	GET / HTTP/1.1
6	0.010857	202.122.145.86	192.168.0.121	TCP	66	80 → 52640 [SYN, ACK] Seq=0 Ack=1 Win=56940 Len=0 MS
7	0.010917	192.168.0.121	202.122.145.86	TCP	54	52640 → 80 [ACK] Seq=1 Ack=1 Win=132352 Len=0
8	0.018331	202.122.145.86	192.168.0.121	TCP	60	80 → 52639 [ACK] Seq=1 Ack=1441 Win=59904 Len=0
9	0.018393	202.122.145.86	192.168.0.121	TCP	60	80 → 52639 [ACK] Seq=1 Ack=2881 Win=62720 Len=0
10	0.018551	202.122.145.86	192.168.0.121	TCP	60	80 → 52639 [ACK] Seq=1 Ack=3333 Win=65664 Len=0
11	0.314676	202.122.145.86	192.168.0.121	TCP	514	80 → 52639 [PSH, ACK] Seq=1 Ack=3333 Win=65664 Len=4

本机接收到的信息：



```
Source Port: 80
Destination Port: 52639
[Stream index: 0]
[TCP Segment Len: 0]
Sequence Number: 1    (relative sequence number)
Sequence Number (raw): 1193786688
[Next Sequence Number: 1    (relative sequence number)]
Acknowledgment Number: 1441    (relative ack number)
Acknowledgment number (raw): 3559460121
0101 .... = Header Length: 20 bytes (5)
Flags: 0x010 (ACK)
   000. .... .... = Reserved: Not set
   ...0 .... .... = Nonce: Not set
   .... 0... .... = Congestion Window Reduced (CWR): Not set
   .... .0.. .... = ECN-Echo: Not set
   .... ..0. .... = Urgent: Not set
   .... ...1 .... = Acknowledgment: Set
   .... .... 0... = Push: Not set
   .... .... .0.. = Reset: Not set
   .... .... ..0. = Syn: Not set
   .... .... ...0 = Fin: Not set
```

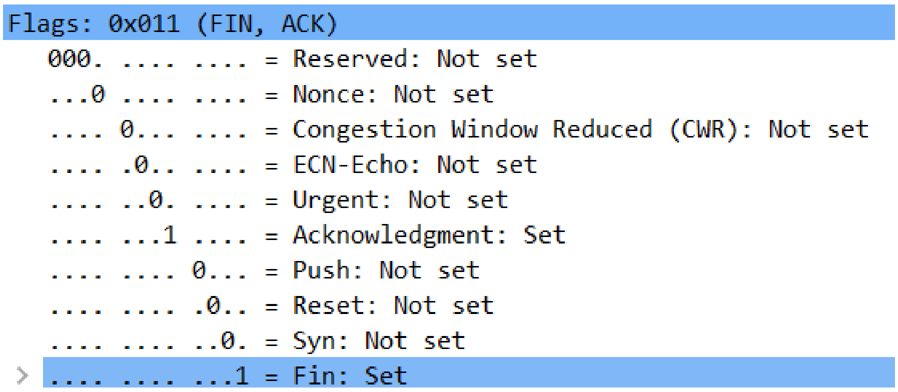
c. 断开连接



```
29 20.626509   192.168.0.121    202.122.145.86   TCP   54 55349 → 80 [FIN, ACK] Seq=6589 Ack=13806 Win=132352 Len=0
30 20.712841   202.122.145.86   192.168.0.121    TCP   54 80 → 55349 [FIN, ACK] Seq=13806 Ack=6590 Win=74240 Len=0
31 20.712875   192.168.0.121    202.122.145.86   TCP   54 55349 → 80 [ACK] Seq=6590 Ack=13807 Win=132352 Len=0
32 45.023776   192.168.0.121    202.122.145.86   TCP   55 [TCP Keep-Alive] 55350 → 80 [ACK] Seq=0 Ack=1 Win=132352 Len=1
33 45.032939   202.122.145.86   192.168.0.121    TCP   66 [TCP Window Update] 80 → 55350 [ACK] Seq=1 Ack=1 Win=56960 Len=0 SLE=0 SRE=1
```

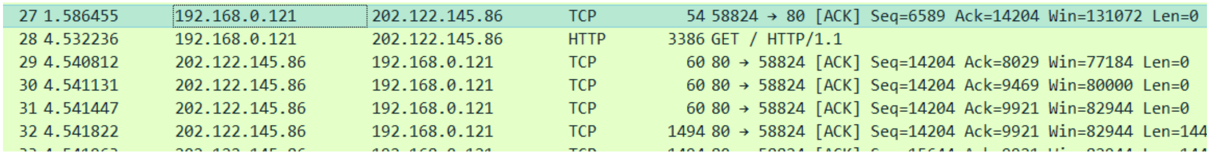
（由于我不小心点到了刷新网页所以图中的端口可能与之前的不同）

当我关掉 www.zju.edu.cn 网站时，本机的端口向服务器断开连接，随后服务器端口也断开了连接。Fin 设置为 Set，表示断开连接。



```
Flags: 0x011 (FIN, ACK)
   000. .... .... = Reserved: Not set
   ...0 .... .... = Nonce: Not set
   .... 0... .... = Congestion Window Reduced (CWR): Not set
   .... .0.. .... = ECN-Echo: Not set
   .... ..0. .... = Urgent: Not set
   .... ...1 .... = Acknowledgment: Set
   .... .... 0... = Push: Not set
   .... .... .0.. = Reset: Not set
   .... .... ..0. = Syn: Not set
   .... .... ...1 = Fin: Set
```

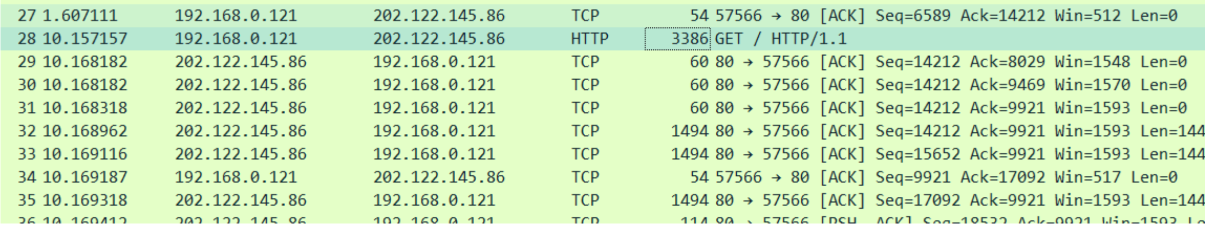
5. 刷新页面并重新抓包(hw4_Refresh.pcapng)



```
27 1.586455   192.168.0.121    202.122.145.86   TCP      54 58824 → 80 [ACK] Seq=6589 Ack=14204 Win=131072 Len=0
28 4.532236   192.168.0.121    202.122.145.86   HTTP   3386 GET / HTTP/1.1
29 4.540812   202.122.145.86   192.168.0.121    TCP      60 80 → 58824 [ACK] Seq=14204 Ack=8029 Win=77184 Len=0
30 4.541131   202.122.145.86   192.168.0.121    TCP      60 80 → 58824 [ACK] Seq=14204 Ack=9469 Win=80000 Len=0
31 4.541447   202.122.145.86   192.168.0.121    TCP      60 80 → 58824 [ACK] Seq=14204 Ack=9921 Win=82944 Len=0
32 4.541822   202.122.145.86   192.168.0.121    TCP    1494 80 → 58824 [ACK] Seq=14204 Ack=9921 Win=82944 Len=144
```

Line28 是刷新后所得到的新包，说明了刷新页面并不会再重新对 TCP 进行连接，而是直接发送 HTTP 的 GET 请求到服务器上。

6. 在文本框重新输入网址(hw4_EnterAgain.pcapng)

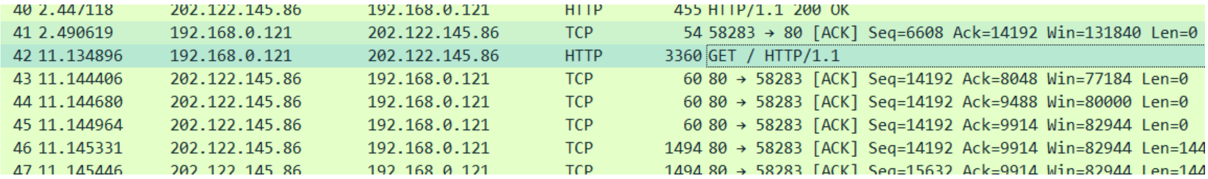


```
27 1.607111    192.168.0.121    202.122.145.86   TCP     54 57566 → 80 [ACK] Seq=6589 Ack=14212 Win=512 Len=0
28 10.157157   192.168.0.121    202.122.145.86   HTTP  3386 GET / HTTP/1.1
29 10.168182   202.122.145.86   192.168.0.121    TCP     60 80 → 57566 [ACK] Seq=14212 Ack=8029 Win=1548 Len=0
30 10.168182   202.122.145.86   192.168.0.121    TCP     60 80 → 57566 [ACK] Seq=14212 Ack=9469 Win=1570 Len=0
31 10.168318   202.122.145.86   192.168.0.121    TCP     60 80 → 57566 [ACK] Seq=14212 Ack=9921 Win=1593 Len=0
32 10.168962   202.122.145.86   192.168.0.121    TCP   1494 80 → 57566 [ACK] Seq=14212 Ack=9921 Win=1593 Len=144
33 10.169116   202.122.145.86   192.168.0.121    TCP   1494 80 → 57566 [ACK] Seq=15652 Ack=9921 Win=1593 Len=144
34 10.169187   192.168.0.121    202.122.145.86   TCP     54 57566 → 80 [ACK] Seq=9921 Ack=17092 Win=517 Len=0
35 10.169318   202.122.145.86   192.168.0.121    TCP   1494 80 → 57566 [ACK] Seq=17092 Ack=9921 Win=1593 Len=144
36 10.169412   202.122.145.86   192.168.0.121    TCP    114 80 → 57566 [PSH, ACK] Seq=18532 Ack=9921 Win=1593 Le
```

Line28 是重新输入网址后得到的新包，说明了重新输入网址也不会断开端口的连接，而是又再发送 HTTP 的 GET 请求，与刷新界面一样。

重新输入该网址会将本机的端口断开连接并连接上新的断开然后再重新与 TCP 进行连接。

7. 在已有此网页的情况下，再新建一个标签页输入相同的网址（hw4_EnterAnotherTab.pcapng）



```
40 2.447118    202.122.145.86   192.168.0.121    HTTP   455 HTTP/1.1 200 OK
41 2.490619    192.168.0.121    202.122.145.86   TCP     54 58283 → 80 [ACK] Seq=6608 Ack=14192 Win=131840 Len=0
42 11.134896   192.168.0.121    202.122.145.86   HTTP  3360 GET / HTTP/1.1
43 11.144406   202.122.145.86   192.168.0.121    TCP     60 80 → 58283 [ACK] Seq=14192 Ack=8048 Win=77184 Len=0
44 11.144680   202.122.145.86   192.168.0.121    TCP     60 80 → 58283 [ACK] Seq=14192 Ack=9488 Win=80000 Len=0
45 11.144964   202.122.145.86   192.168.0.121    TCP     60 80 → 58283 [ACK] Seq=14192 Ack=9914 Win=82944 Len=0
46 11.145331   202.122.145.86   192.168.0.121    TCP   1494 80 → 58283 [ACK] Seq=14192 Ack=9914 Win=82944 Len=144
47 11.145446   202.122.145.86   192.168.0.121    TCP   1494 80 → 58283 [ACK] Seq=15632 Ack=9914 Win=82944 Len=144
```

Line42 是重新输入该网址后得到的新包，可以看到的是并没有端口的断开连接而是直接发送 HTTP 请求到服务器上。