山东大学 计算机科学与技术 学院

新兴网络技术与实践 课程实验报告

学号：202300130183	姓名：宋浩宇	班级：23 级智能班

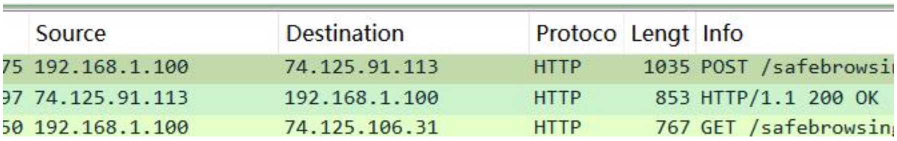
实验题目：Wireshark Lab: NAT

实验学时：2 | 实验日期：2025/4/16

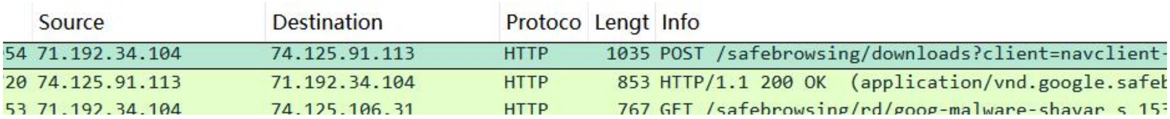
实验目的：学习 NAT

实验结果：

1. What is the IP address of the client?



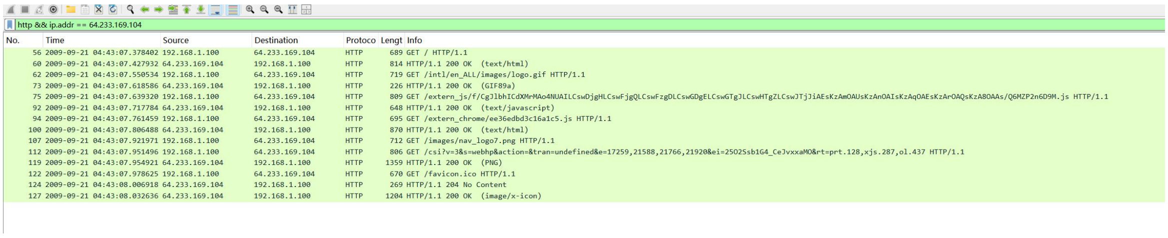
	Source	Destination	Protoco	Lengt	Info
75	192.168.1.100	74.125.91.113	HTTP	1035	POST /safebrowsi
97	74.125.91.113	192.168.1.100	HTTP	853	HTTP/1.1 200 OK
50	192.168.1.100	74.125.106.31	HTTP	767	GET /safebrowsin



	Source	Destination	Protoco	Lengt	Info
54	71.192.34.104	74.125.91.113	HTTP	1035	POST /safebrowsing/downloads?client=navclient-
20	74.125.91.113	71.192.34.104	HTTP	853	HTTP/1.1 200 OK (application/vnd.google.safeb
53	71.192.34.104	74.125.106.31	HTTP	767	GET /safebrowsing/rd/goog-malware-shavar_s_15

从这里可以看出来客户端 IP 的子网地址是 192.168.1.100，实际的发出地址是 71.192.34.104.

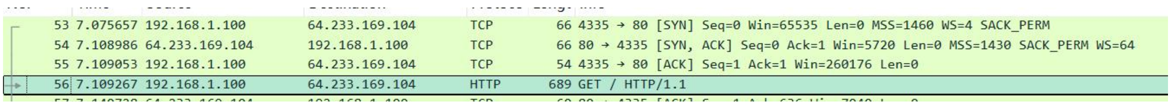
2. The client actually communicates with several different Google servers in order to implement “safe browsing.” (See extra credit section at the end of this lab). The main Google server that will serve up the main Google web page has IP address64.233.169.104. In order to display only those frames containing HTTP messages that are sent to/from this Google, server, enter the expression “http && ip.addr == 64.233.169.104” (without quotes) into the Filter: field in Wireshark .



```
http && ip.addr == 64.233.169.104
No.     Time                       Source          Destination     Protoco Lengt Info
     56 2009-09-21 04:43:07.378402 192.168.1.100   64.233.169.104  HTTP     689 GET / HTTP/1.1
     60 2009-09-21 04:43:07.427932 64.233.169.104  192.168.1.100   HTTP     814 HTTP/1.1 200 OK  (text/html)
     62 2009-09-21 04:43:07.550534 192.168.1.100   64.233.169.104  HTTP     719 GET /intl/en_ALL/images/logo.gif HTTP/1.1
     73 2009-09-21 04:43:07.618586 64.233.169.104  192.168.1.100   HTTP     226 HTTP/1.1 200 OK  (GIF89a)
     75 2009-09-21 04:43:07.639320 192.168.1.100   64.233.169.104  HTTP     809 GET /extern_js/f/CgJlbhICdXMrMAo4NUAILCswDjgHLCswFjgQLCswFzgDLCswGDgELCswGTgJLCswHTgZLCswJTjJiAEsKzAmOAUsKzAnOAIsKzAqOAEsKzArOAQsKzA8OAAs/Q6MZP2n6D9M.js HTTP/1.1
     92 2009-09-21 04:43:07.717784 64.233.169.104  192.168.1.100   HTTP     648 HTTP/1.1 200 OK  (text/javascript)
     94 2009-09-21 04:43:07.761459 192.168.1.100   64.233.169.104  HTTP     695 GET /extern_chrome/ee36edbd3c16a1c5.js HTTP/1.1
    100 2009-09-21 04:43:07.806488 64.233.169.104  192.168.1.100   HTTP     870 HTTP/1.1 200 OK  (text/html)
    107 2009-09-21 04:43:07.921971 192.168.1.100   64.233.169.104  HTTP     712 GET /images/nav_logo7.png HTTP/1.1
    112 2009-09-21 04:43:07.951496 192.168.1.100   64.233.169.104  HTTP     806 GET /csi?v=3&s=webhp&action=&tran=undefined&e=17259,21588,21766,21920&ei=25O2Ssb1G4_CeJvxxaMO&rt=prt.128,xjs.287,ol.437 HTTP/1.1
    119 2009-09-21 04:43:07.954921 64.233.169.104  192.168.1.100   HTTP    1359 HTTP/1.1 200 OK  (PNG)
    122 2009-09-21 04:43:07.978625 192.168.1.100   64.233.169.104  HTTP     670 GET /favicon.ico HTTP/1.1
    124 2009-09-21 04:43:08.006918 64.233.169.104  192.168.1.100   HTTP     269 HTTP/1.1 204 No Content
    127 2009-09-21 04:43:08.032636 64.233.169.104  192.168.1.100   HTTP    1204 HTTP/1.1 200 OK  (image/x-icon)
```

如图，这是过滤后的结果。

3. Consider now the HTTP GET sent from the client to the Google server (whose IP address is IP address 64.233.169.104) at time 7.109267. What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP GET?

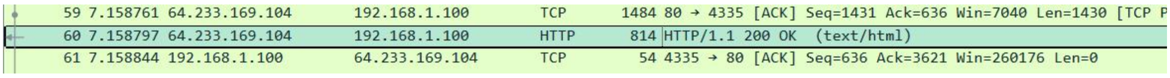


```
 53 7.075657 192.168.1.100    64.233.169.104   TCP     66 4335 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=4 SACK_PERM
 54 7.108986 64.233.169.104   192.168.1.100    TCP     66 80 → 4335 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430 SACK_PERM WS=64
 55 7.109053 192.168.1.100    64.233.169.104   TCP     54 4335 → 80 [ACK] Seq=1 Ack=1 Win=260176 Len=0
 56 7.109267 192.168.1.100    64.233.169.104   HTTP   689 GET / HTTP/1.1
```

如图，这个 HTTP GET 请求上方的是这三次握手的数据包。

源 IP 地址是 192.168.1.100，目的 IP 地址是 64.233.169.104，源端口号是 4335，目的端口号是 80

4. At what time4 is the corresponding 200 OK HTTP message received from the Google server? What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP 200 OK message?

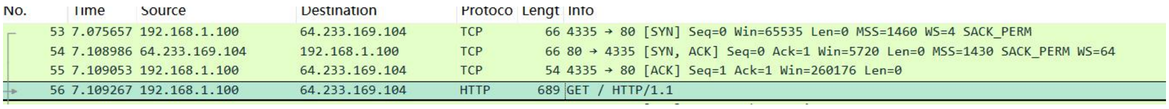


```
 59 7.158761 64.233.169.104   192.168.1.100    TCP    1484 80 → 4335 [ACK] Seq=1431 Ack=636 Win=7040 Len=1430 [TCP P
 60 7.158797 64.233.169.104   192.168.1.100    HTTP    814 HTTP/1.1 200 OK  (text/html)
 61 7.158844 192.168.1.100    64.233.169.104   TCP      54 4335 → 80 [ACK] Seq=636 Ack=3621 Win=260176 Len=0
```

接收时间是 7.158797，源 IP 地址是 64.233.169.104，目的地址是 192.168.1.100，源端口是 80，目的端口是 4335

5. Recall that before a GET command can be sent to an HTTP server, TCP must first set up a connection using the three-way SYN/ACK handshake. At what time is the client-to-server TCP SYN segment sent that sets up the connection used by the GET sent at time 7.109267? What are the source and destination IP addresses and

source and destination ports for the TCP SYN segment? What are the source and destination IP addresses and source and destination ports of the ACK sent in response to the SYN. At what time is this ACK received at the client? (Note: to find these segments you will need to clear the Filter expression you entered above in step 2. If you enter the filter “tcp”, only TCP segments will be displayed by Wireshark).



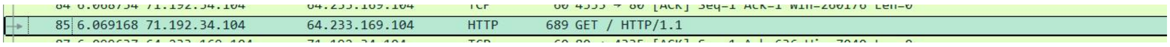
```
No.     Time     Source          Destination     Protoco Lengt Info
     53 7.075657 192.168.1.100   64.233.169.104  TCP        66 4335 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=4 SACK_PERM
     54 7.108986 64.233.169.104  192.168.1.100   TCP        66 80 → 4335 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430 SACK_PERM WS=64
     55 7.109053 192.168.1.100   64.233.169.104  TCP        54 4335 → 80 [ACK] Seq=1 Ack=1 Win=260176 Len=0
     56 7.109267 192.168.1.100   64.233.169.104  HTTP      689 GET / HTTP/1.1
```

如图，这个 HTTP GET 请求上方的是这三次握手的数据包。

第一个 TCP SYN 数据段是再 7.075657 发送的,源 IP 地址是 192.168.1.100，目的 IP 地址是 64.233.169.104，源端口号是 4335，目的端口号是 80

作为相应的 TCP ACK 数据段的源 IP 地址是 64.233.169.104，目的 IP 地址是 192.168.1.100，源端口号是 80，目的端口号是 4335，是在 7.108986 被客户端接收到的。

6. In the NAT_ISP_side trace file, find the HTTP GET message was sent from the client to the Google server at time 7.109267 (where t=7.109267 is time at which this was sent as recorded in the NAT_home_side trace file). At what time does this message appear in the NAT_ISP_side trace file? What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP GET (as recording in the NAT_ISP_side trace file)? Which of these fields are the same, and which are different, than in your answer to question 3 above?

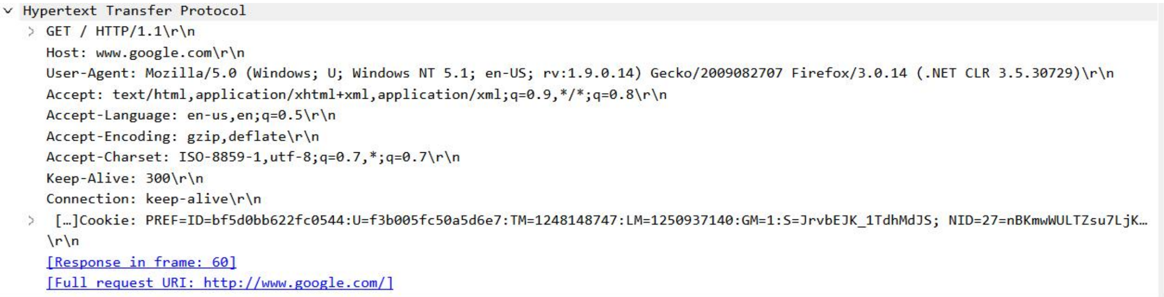


```
     84 6.068734 71.192.34.104   64.233.169.104  TCP        60 4335 → 80 [ACK] Seq=1 Ack=1 Win=260176 Len=0
     85 6.069168 71.192.34.104   64.233.169.104  HTTP      689 GET / HTTP/1.1
```

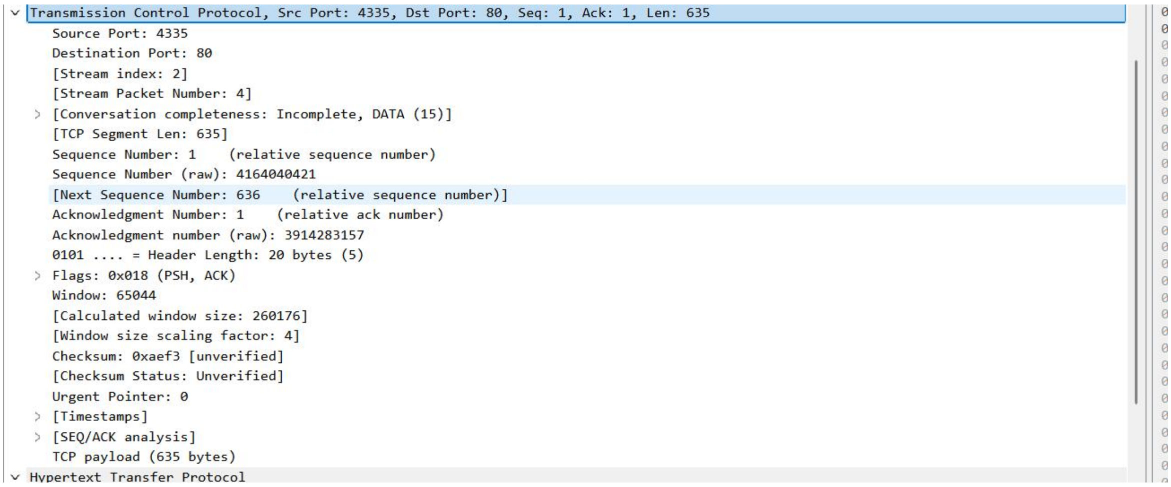
这是对应的那个 HTTP GET，时间为 6.069168，源 IP 地址是 71.192.34.104，目的 IP 地址是 64.233.169.104，源端口号是 4335，目的端口号是 80.

其中发送时间和源 IP 地址和 NAT_home_side 的不同。

7. Are any fields in the HTTP GET message changed? Which of the following fields in the IP datagram carrying the HTTP GET are changed: Version, Header Length, Flags, Checksum. If any of these fields have changed, give a reason (in one sentence) stating why this field needed to change.

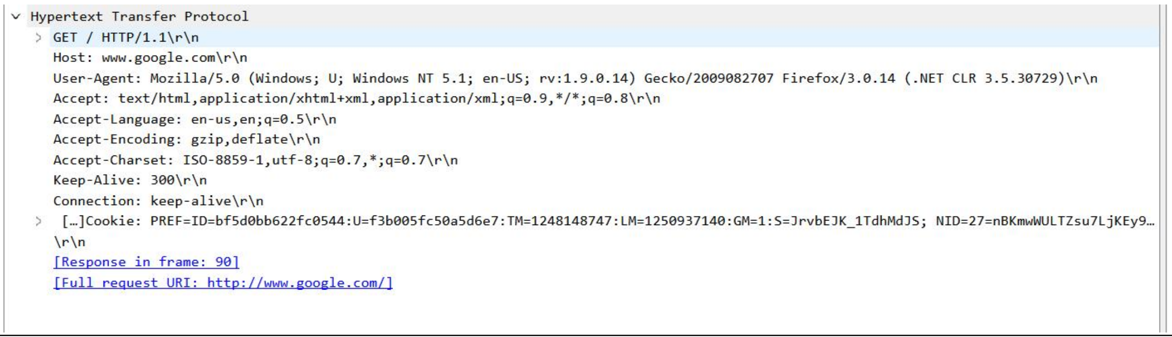


```
v Hypertext Transfer Protocol
  > GET / HTTP/1.1\r\n
    Host: www.google.com\r\n
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.14) Gecko/2009082707 Firefox/3.0.14 (.NET CLR 3.5.30729)\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Language: en-us,en;q=0.5\r\n
    Accept-Encoding: gzip,deflate\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
    Keep-Alive: 300\r\n
    Connection: keep-alive\r\n
  >  […]Cookie: PREF=ID=bf5d0bb622fc0544:U=f3b005fc50a5d6e7:TM=1248148747:LM=1250937140:GM=1:S=JrvbEJK_1TdhMdJS; NID=27=nBKmwWULTZsu7LjK…
    \r\n
    [Response in frame: 60]
    [Full request URI: http://www.google.com/]
```

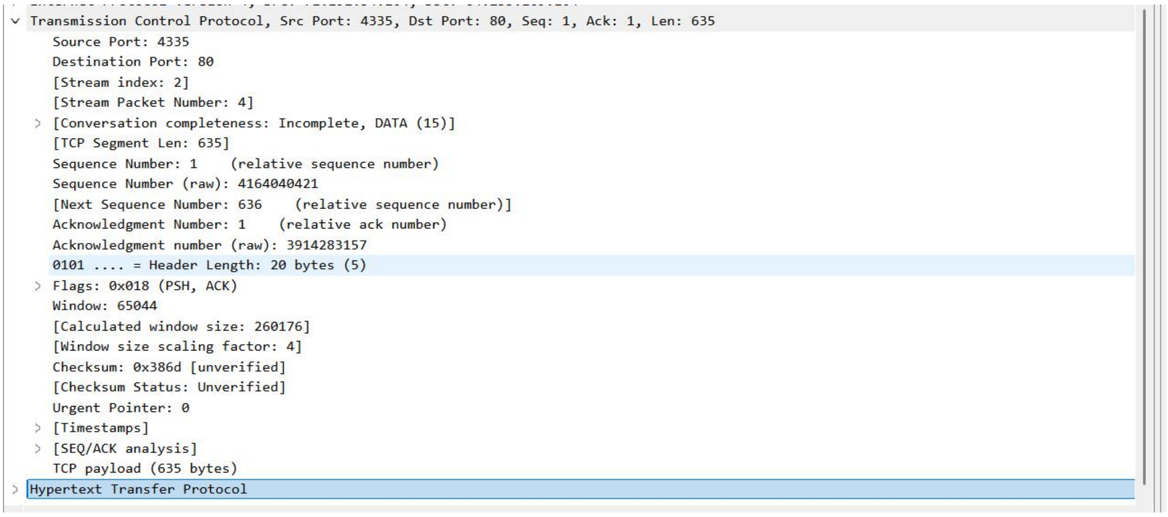


```
v Transmission Control Protocol, Src Port: 4335, Dst Port: 80, Seq: 1, Ack: 1, Len: 635
    Source Port: 4335
    Destination Port: 80
    [Stream index: 2]
    [Stream Packet Number: 4]
  > [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 635]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 4164040421
    [Next Sequence Number: 636    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 3914283157
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
    Window: 65044
    [Calculated window size: 260176]
    [Window size scaling factor: 4]
    Checksum: 0xaef3 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > [Timestamps]
  > [SEQ/ACK analysis]
    TCP payload (635 bytes)
v Hypertext Transfer Protocol
```

这是 homeside 的



```
v Hypertext Transfer Protocol
  > GET / HTTP/1.1\r\n
    Host: www.google.com\r\n
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.14) Gecko/2009082707 Firefox/3.0.14 (.NET CLR 3.5.30729)\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Language: en-us,en;q=0.5\r\n
    Accept-Encoding: gzip,deflate\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
    Keep-Alive: 300\r\n
    Connection: keep-alive\r\n
  >  […]Cookie: PREF=ID=bf5d0bb622fc0544:U=f3b005fc50a5d6e7:TM=1248148747:LM=1250937140:GM=1:S=JrvbEJK_1TdhMdJS; NID=27=nBKmwWULTZsu7LjKEy9…
    \r\n
    [Response in frame: 90]
    [Full request URI: http://www.google.com/]
```

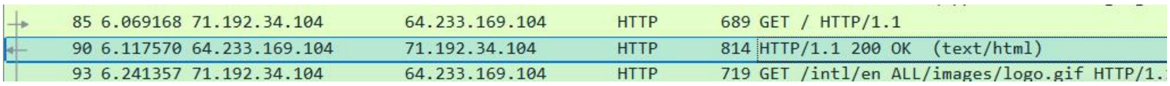


```
v Transmission Control Protocol, Src Port: 4335, Dst Port: 80, Seq: 1, Ack: 1, Len: 635
      Source Port: 4335
      Destination Port: 80
      [Stream index: 2]
      [Stream Packet Number: 4]
  >   [Conversation completeness: Incomplete, DATA (15)]
      [TCP Segment Len: 635]
      Sequence Number: 1    (relative sequence number)
      Sequence Number (raw): 4164040421
      [Next Sequence Number: 636    (relative sequence number)]
      Acknowledgment Number: 1    (relative ack number)
      Acknowledgment number (raw): 3914283157
      0101 .... = Header Length: 20 bytes (5)
  >   Flags: 0x018 (PSH, ACK)
      Window: 65044
      [Calculated window size: 260176]
      [Window size scaling factor: 4]
      Checksum: 0x386d [unverified]
      [Checksum Status: Unverified]
      Urgent Pointer: 0
  >   [Timestamps]
  >   [SEQ/ACK analysis]
      TCP payload (635 bytes)
> Hypertext Transfer Protocol
```

这是 ISPside 的

可以看出没有任何一个字段被改变。

8. In the NAT_ISP_side trace file, at what time is the first 200 OK HTTP message received from the Google server? What are the source and destination IP addresses and TCP source and destination ports on the IP datagram carrying this HTTP 200 OK message? Which of these fields are the same, and which are different than your answer to question 4 above?



```
85 6.069168 71.192.34.104     64.233.169.104    HTTP    689 GET / HTTP/1.1
90 6.117570 64.233.169.104    71.192.34.104     HTTP    814 HTTP/1.1 200 OK  (text/html)
93 6.241357 71.192.34.104     64.233.169.104    HTTP    719 GET /intl/en ALL/images/logo.gif HTTP/1.
```

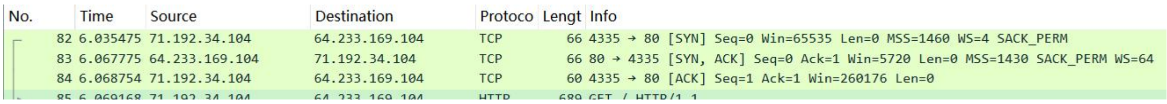
第一个 HTTP OK 消息是在 6.117570 被接收到的。

源 IP 地址是 64.233.169.104，目的地址是 71.192.34.104，源端口号是 80，目的端口号是 4335

这些字段与 homeside 相比，目的地址不同，接收时间不同。

9. In the NAT_ISP_side trace file, at what time were the client-to-server TCP SYN segment and the server-to-client TCP ACK segment corresponding to the segments in question 5 above captured? What are the source and destination IP addresses and source and destination ports for these two segments? Which of these fields are the same, and which are different than your answer to

question 5 above?



```
No.      Time     Source           Destination      Protoco Lengt Info
      82 6.035475 71.192.34.104    64.233.169.104   TCP        66 4335 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=4 SACK_PERM
      83 6.067775 64.233.169.104   71.192.34.104    TCP        66 80 → 4335 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430 SACK_PERM WS=64
      84 6.068754 71.192.34.104    64.233.169.104   TCP        60 4335 → 80 [ACK] Seq=1 Ack=1 Win=260176 Len=0
      85 6.069168 71.192.34.104    64.233.169.104   HTTP      689 GET / HTTP/1.1
```

如图，这是三次握手的 TCP 数据包，捕获时间分别为 6.035475、6.067775，源 IP 地址分别是 71.192.34.104、64.233.169.104，目标地址分别为 64.233.169.104，源端口号分别是 4335、80，目标端口号分别是 80、4335.可以看出和 homeside 相比主要是客户端 IP 地址不同

10. Using your answers to 1-8 above, fill in the NAT translation table entries for HTTP connection considered in questions 1-8 above.

192.168.1.100:4335=>71.192.34.104:4335
71.192.34.104:4335=>192.168.1.100:4335

问题及收获：

NAT（网络地址转换）能够将私有网络中的内部 IP 地址转换为公共 IP 地址，从而允许多个设备共享一个公共 IP 地址访问互联网，同时增强了网络安全性和隐私保护。