# 实验 2：TCP

班级 14012104 学号 2021303423 姓名 申 铭

## 一、实验内容

在实验二中，我们将研究 TCP 协议的行为。为此，我们将分析从你的计算机到远程服务器发送和接收一个 150 KB 的文件（《爱丽丝梦游仙境》的文本）的 TCP 段。具体包括：

**1、捕获从计算机到远程服务器的批量 TCP 传输**

**2、跟踪包的初步观察**

**3、TCP 基础**

**4、TCP 拥塞控制**

## 二、实验操作步骤及结果

### 1、捕获从计算机到远程服务器的批量 TCP 传输

① 浏览器访问 http://gaia.cs.umass.edu/wireshark-labs/alice.txt，检索《爱丽丝梦游仙境》的 ASCII 副本，将此文件存储在计算机的某个位置。

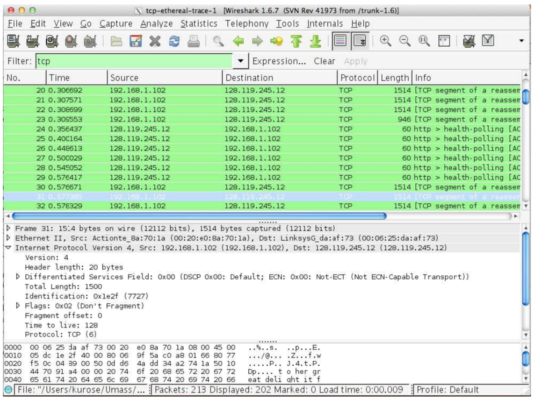
② 访问 http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html，点击页面中的“选择文件”按钮输入存储的文件的完整路径名。

③ 启动 Wireshark，开始数据包捕获，使过滤器筛选 TCP 协议的数据包。

④ 放回浏览器页面，点击“上传文件”按钮。

⑤ 当看到浏览器页面出现上传成功的提示时，停止数据包捕获。

**此时 Wireshark 中应该看到：**


tcp-ethereal-trace-1 [Wireshark 1.6.7 (SVN Rev 41973 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply

Filter: tcp

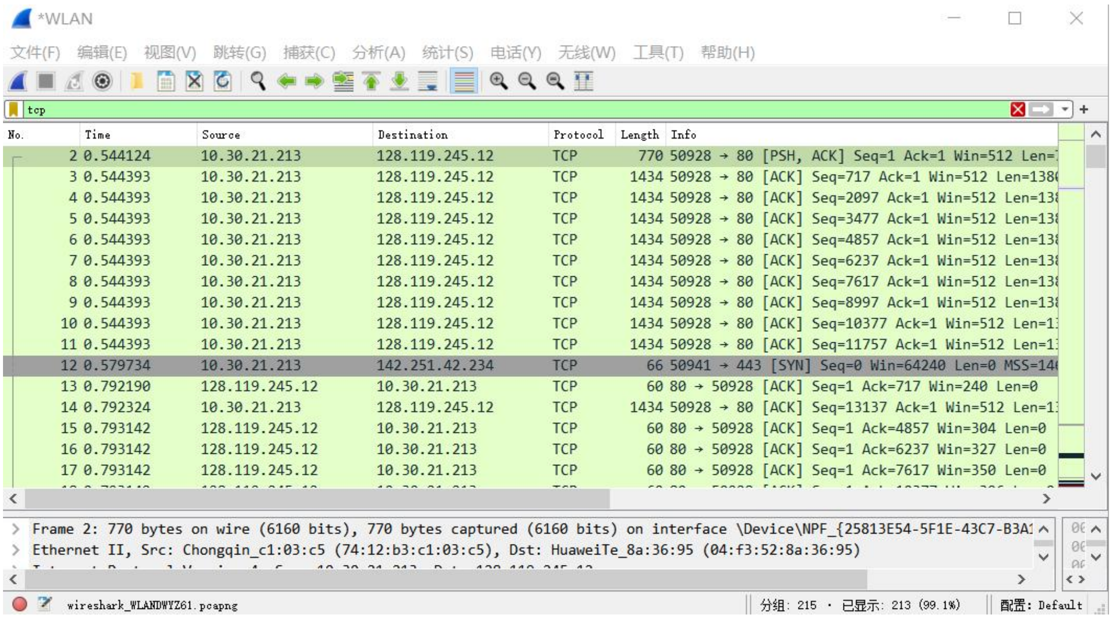
| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 20 | 0.306692 | 192.168.1.102 | 128.119.245.12 | TCP | 1514 | [TCP segment of a reassem |
| 21 | 0.307571 | 192.168.1.102 | 128.119.245.12 | TCP | 1514 | [TCP segment of a reassem |
| 22 | 0.308699 | 192.168.1.102 | 128.119.245.12 | TCP | 1514 | [TCP segment of a reassem |
| 23 | 0.309553 | 192.168.1.102 | 128.119.245.12 | TCP | 946 | [TCP segment of a reassem |
| 24 | 0.356437 | 128.119.245.12 | 192.168.1.102 | TCP | 60 | http > health-polling [AC |
| 25 | 0.400164 | 128.119.245.12 | 192.168.1.102 | TCP | 60 | http > health-polling [AC |
| 26 | 0.448613 | 128.119.245.12 | 192.168.1.102 | TCP | 60 | http > health-polling [AC |
| 27 | 0.500029 | 128.119.245.12 | 192.168.1.102 | TCP | 60 | http > health-polling [AC |
| 28 | 0.545052 | 128.119.245.12 | 192.168.1.102 | TCP | 60 | http > health-polling [AC |
| 29 | 0.576417 | 128.119.245.12 | 192.168.1.102 | TCP | 60 | http > health-polling [AC |
| 30 | 0.576671 | 192.168.1.102 | 128.119.245.12 | TCP | 1514 | [TCP segment of a reassem |
| 31 | 0.577385 | 192.168.1.102 | 128.119.245.12 | TCP | 1514 | [TCP segment of a reassem |
| 32 | 0.578329 | 192.168.1.102 | 128.119.245.12 | TCP | 1514 | [TCP segment of a reassem |

```
▷ Frame 31: 15:4 bytes on wire (12112 bits), 1514 bytes captured (12112 bits)
▷ Ethernet II, Src: Actionte_8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG_da:af:73 (00:06:25:da:af:73)
▽ Internet Protocol Version 4, Src: 192.168.1.102 (192.168.1.102), Dst: 128.119.245.12 (128.119.245.12)
    Version: 4
    Header length: 20 bytes
  ▷ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
    Total Length: 1500
    Identification: 0x1e2f (7727)
  ▷ Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (6)
```

```
0000  00 06 25 da af 73 00 20  e0 8a 70 1a 08 00 45 00   ..%..s.  ..p...E.
0010  05 dc 1e 2f 40 00 80 06  9f 5a c0 a8 01 66 80 77   .../@... .Z...f.w
0020  f5 0c 04 89 00 50 0d d6  4a dd 34 a2 74 1a 50 10   .....P.. J.4.t.P.
0030  44 70 91 a4 00 00 20 74  6f 20 68 65 72 20 67 72   Dp.... t o her gr
0040  65 61 74 20 64 65 6c 69  67 68 74 20 69 74 20 66   eat deli ght it f
```

File: "/Users/kurose/Umass/... Packets: 213 Displayed: 202 Marked: 0 Load time: 0:00.009 Profile: Default

我的 Wireshark 中显示如下图所示：



```
tcp

No.   Time       Source          Destination      Protocol  Length Info
    2 0.544124   10.30.21.213    128.119.245.12   TCP        770 50928 → 80 [PSH, ACK] Seq=1 Ack=1 Win=512 Len=
    3 0.544393   10.30.21.213    128.119.245.12   TCP       1434 50928 → 80 [ACK] Seq=717 Ack=1 Win=512 Len=138
    4 0.544393   10.30.21.213    128.119.245.12   TCP       1434 50928 → 80 [ACK] Seq=2097 Ack=1 Win=512 Len=13
    5 0.544393   10.30.21.213    128.119.245.12   TCP       1434 50928 → 80 [ACK] Seq=3477 Ack=1 Win=512 Len=13
    6 0.544393   10.30.21.213    128.119.245.12   TCP       1434 50928 → 80 [ACK] Seq=4857 Ack=1 Win=512 Len=13
    7 0.544393   10.30.21.213    128.119.245.12   TCP       1434 50928 → 80 [ACK] Seq=6237 Ack=1 Win=512 Len=13
    8 0.544393   10.30.21.213    128.119.245.12   TCP       1434 50928 → 80 [ACK] Seq=7617 Ack=1 Win=512 Len=13
    9 0.544393   10.30.21.213    128.119.245.12   TCP       1434 50928 → 80 [ACK] Seq=8997 Ack=1 Win=512 Len=13
   10 0.544393   10.30.21.213    128.119.245.12   TCP       1434 50928 → 80 [ACK] Seq=10377 Ack=1 Win=512 Len=1
   11 0.544393   10.30.21.213    128.119.245.12   TCP       1434 50928 → 80 [ACK] Seq=11757 Ack=1 Win=512 Len=1
   12 0.579734   10.30.21.213    142.251.42.234   TCP         66 50941 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=14
   13 0.792190   128.119.245.12  10.30.21.213     TCP         60 80 → 50928 [ACK] Seq=1 Ack=717 Win=240 Len=0
   14 0.792324   10.30.21.213    128.119.245.12   TCP       1434 50928 → 80 [ACK] Seq=13137 Ack=1 Win=512 Len=1
   15 0.793142   128.119.245.12  10.30.21.213     TCP         60 80 → 50928 [ACK] Seq=1 Ack=4857 Win=304 Len=0
   16 0.793142   128.119.245.12  10.30.21.213     TCP         60 80 → 50928 [ACK] Seq=1 Ack=6237 Win=327 Len=0
   17 0.793142   128.119.245.12  10.30.21.213     TCP         60 80 → 50928 [ACK] Seq=1 Ack=7617 Win=350 Len=0

> Frame 2: 770 bytes on wire (6160 bits), 770 bytes captured (6160 bits) on interface \Device\NPF_{25813E54-5F1E-43C7-B3A1
> Ethernet II, Src: Chongqin_c1:03:c5 (74:12:b3:c1:03:c5), Dst: HuaweiTe_8a:36:95 (04:f3:52:8a:36:95)

wireshark_WLANDWYZ61.pcapng     分组: 215 · 已显示: 213 (99.1%)     配置: Default
```

## 2、初步观察捕获的 TCP 包

### （1）实验步骤

① 在浏览器中打开链接，并下载文件：

http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip

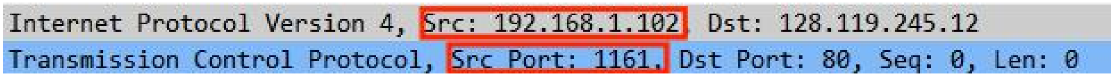
② 在 wireshark 中打开解压后文件中的 tcpethereal-trace-1

③ 根据 tcpethereal-trace-1 中的内容回答问题。

### （2）回答问题

1.What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu? To answer this question, it’s probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the “details of the selected packet header window” (refer to Figure 2 in the “Getting Started with Wireshark” Lab if you’re uncertain about the Wireshark windows.

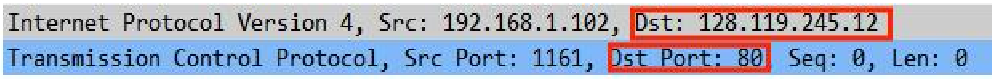
客户机端IP地址192.168.1.102；TCP端口号为1161。如下图所示：



```
Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 0, Len: 0
```

2.What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?

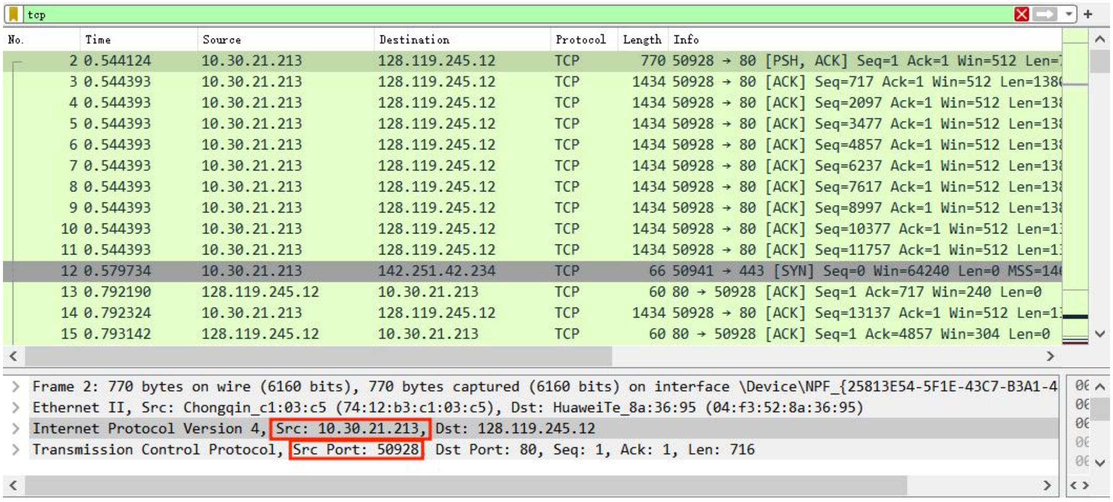
服务器端 IP 地址为 128.119.245.12；TCP 端口号为 80。如下图所示：



```
Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 0, Len: 0
```

3.What is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu?

打开刚才保存的捕获上传文件时捕获的 TCP 数据包。如下图所示：



```
tcp
No.  Time      Source          Destination     Protocol Length Info
   2 0.544124  10.30.21.213    128.119.245.12  TCP       770 50928 → 80 [PSH, ACK] Seq=1 Ack=1 Win=512 Len=
   3 0.544393  10.30.21.213    128.119.245.12  TCP      1434 50928 → 80 [ACK] Seq=717 Ack=1 Win=512 Len=138
   4 0.544393  10.30.21.213    128.119.245.12  TCP      1434 50928 → 80 [ACK] Seq=2097 Ack=1 Win=512 Len=13
   5 0.544393  10.30.21.213    128.119.245.12  TCP      1434 50928 → 80 [ACK] Seq=3477 Ack=1 Win=512 Len=13
   6 0.544393  10.30.21.213    128.119.245.12  TCP      1434 50928 → 80 [ACK] Seq=4857 Ack=1 Win=512 Len=13
   7 0.544393  10.30.21.213    128.119.245.12  TCP      1434 50928 → 80 [ACK] Seq=6237 Ack=1 Win=512 Len=13
   8 0.544393  10.30.21.213    128.119.245.12  TCP      1434 50928 → 80 [ACK] Seq=7617 Ack=1 Win=512 Len=13
   9 0.544393  10.30.21.213    128.119.245.12  TCP      1434 50928 → 80 [ACK] Seq=8997 Ack=1 Win=512 Len=13
  10 0.544393  10.30.21.213    128.119.245.12  TCP      1434 50928 → 80 [ACK] Seq=10377 Ack=1 Win=512 Len=1
  11 0.544393  10.30.21.213    128.119.245.12  TCP      1434 50928 → 80 [ACK] Seq=11757 Ack=1 Win=512 Len=1
  12 0.579734  10.30.21.213    142.251.42.234  TCP        66 50941 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=14
  13 0.792190  128.119.245.12  10.30.21.213    TCP        60 80 → 50928 [ACK] Seq=1 Ack=717 Win=240 Len=0
  14 0.792324  10.30.21.213    128.119.245.12  TCP      1434 50928 → 80 [ACK] Seq=13137 Ack=1 Win=512 Len=1
  15 0.793142  128.119.245.12  10.30.21.213    TCP        60 80 → 50928 [ACK] Seq=1 Ack=4857 Win=304 Len=0

> Frame 2: 770 bytes on wire (6160 bits), 770 bytes captured (6160 bits) on interface \Device\NPF_{25813E54-5F1E-43C7-B3A1-4
> Ethernet II, Src: Chongqin_c1:03:c5 (74:12:b3:c1:03:c5), Dst: HuaweiTe_8a:36:95 (04:f3:52:8a:36:95)
> Internet Protocol Version 4, Src: 10.30.21.213, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 50928  Dst Port: 80, Seq: 1, Ack: 1, Len: 716
```

my client IP 地址为 10.30.21.213；TCP 端口号为 50928。

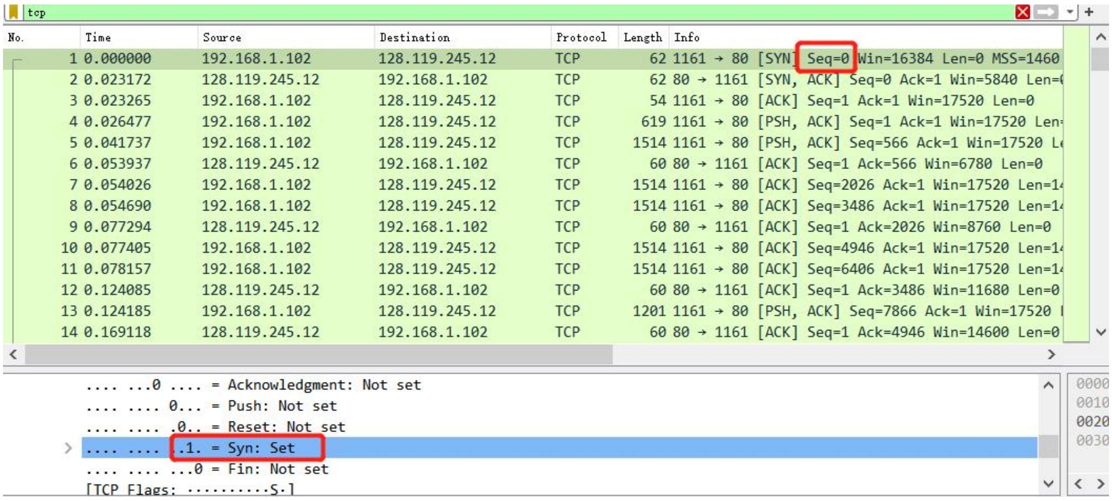
## 3、TCP 基础

**回答问题：**

4.What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in the segment that identifies the segment as a SYN segment?

用于启动客户端计算机与 gaia.cs.umass.edu 之间的 TCP 连接的 TCP SYN 段的序列号是 0。这个标志告诉接收端计算机，在建立连接时 SYN 段需要进行确认。当接收计算机确认 SYN 段时，它会向发送方计算机发送一个 SYN / ACK 段，表示可以建立连接。最后，发送方计算机将发送一个 ACK 段作为连接的最后确认。上述过程即为“三次握手”。

SYN 被置为 1 证明该段位 SYN 段。如下图所示：

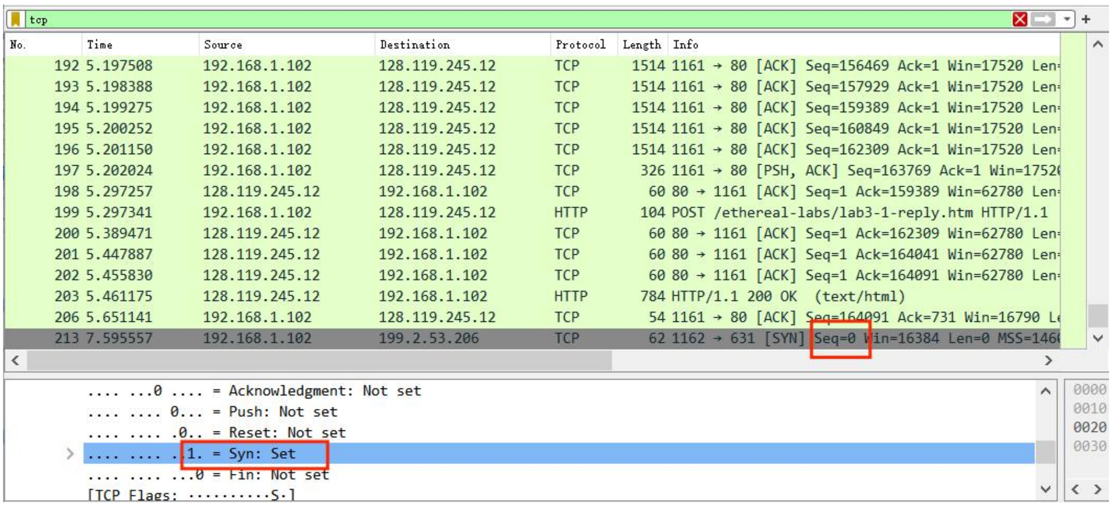


```
tcp
No.   Time       Source          Destination     Protocol  Length  Info
   1 0.000000   192.168.1.102   128.119.245.12  TCP         62 1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460
   2 0.023172   128.119.245.12  192.168.1.102   TCP         62 80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=
   3 0.023265   192.168.1.102   128.119.245.12  TCP         54 1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0
   4 0.026477   192.168.1.102   128.119.245.12  TCP        619 1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=
   5 0.041737   192.168.1.102   128.119.245.12  TCP       1514 1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=17520 L
   6 0.053937   128.119.245.12  192.168.1.102   TCP         60 80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
   7 0.054026   192.168.1.102   128.119.245.12  TCP       1514 1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1
   8 0.054690   192.168.1.102   128.119.245.12  TCP       1514 1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1
   9 0.077294   128.119.245.12  192.168.1.102   TCP         60 80 → 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0
  10 0.077405   192.168.1.102   128.119.245.12  TCP       1514 1161 → 80 [ACK] Seq=4946 Ack=1 Win=17520 Len=1
  11 0.078157   192.168.1.102   128.119.245.12  TCP       1514 1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1
  12 0.124085   128.119.245.12  192.168.1.102   TCP         60 80 → 1161 [ACK] Seq=1 Ack=3486 Win=11680 Len=0
  13 0.124185   192.168.1.102   128.119.245.12  TCP       1201 1161 → 80 [PSH, ACK] Seq=7866 Ack=1 Win=17520
  14 0.169118   128.119.245.12  192.168.1.102   TCP         60 80 → 1161 [ACK] Seq=1 Ack=4946 Win=14600 Len=0

     .... ...0 .... = Acknowledgment: Not set
     .... .... 0... = Push: Not set
     .... .... .0.. = Reset: Not set
   > .... .... ..1. = Syn: Set
     .... .... ...0 = Fin: Not set
     [TCP Flags: ··········S·]
```

5.What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did gaia.cs.umass.edu determine that value? What is it in the segment that identifies the segment as a SYNACK segment?

包含 HTTP POST 的 TCP 段的 sequence number 为 1。

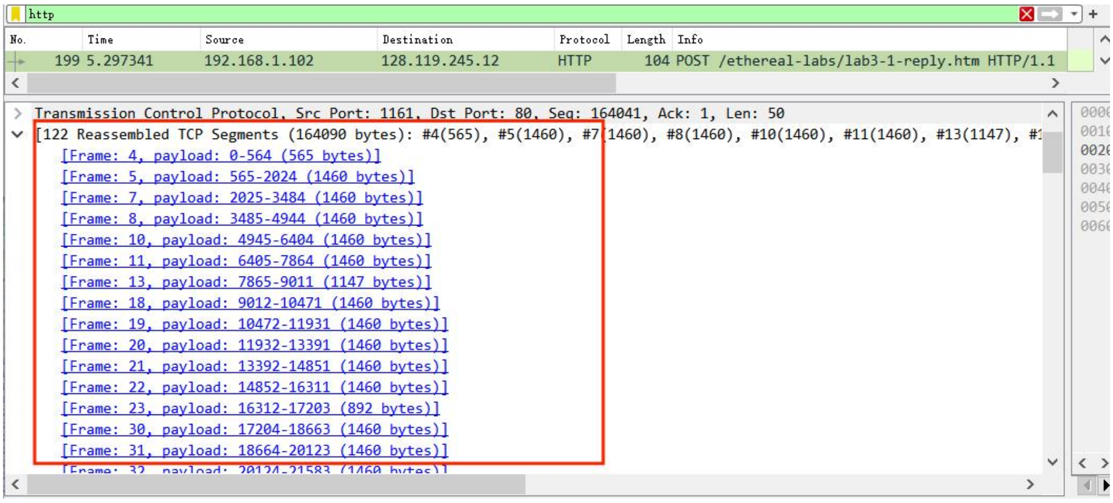
服务器将来自客户端计算机的 SYN 段的初始序列号加 1。来自客户端计算机的 SYN 段的初始序号为 0，ACK 的值为 SYN + 1，因此 SYN _ ACK 段的确认字段为 1。如下图所示：



```
tcp
No.   Time       Source          Destination     Protocol  Length  Info
 192 5.197508   192.168.1.102   128.119.245.12  TCP       1514 1161 → 80 [ACK] Seq=156469 Ack=1 Win=17520 Len=
 193 5.198388   192.168.1.102   128.119.245.12  TCP       1514 1161 → 80 [ACK] Seq=157929 Ack=1 Win=17520 Len=
 194 5.199275   192.168.1.102   128.119.245.12  TCP       1514 1161 → 80 [ACK] Seq=159389 Ack=1 Win=17520 Len=
 195 5.200252   192.168.1.102   128.119.245.12  TCP       1514 1161 → 80 [ACK] Seq=160849 Ack=1 Win=17520 Len=
 196 5.201150   192.168.1.102   128.119.245.12  TCP       1514 1161 → 80 [ACK] Seq=162309 Ack=1 Win=17520 Len=
 197 5.202024   192.168.1.102   128.119.245.12  TCP        326 1161 → 80 [PSH, ACK] Seq=163769 Ack=1 Win=1752
 198 5.297257   128.119.245.12  192.168.1.102   TCP         60 80 → 1161 [ACK] Seq=1 Ack=159389 Win=62780 Len=
 199 5.297341   192.168.1.102   128.119.245.12  HTTP       104 POST /ethereal-labs/lab3-1-reply.htm HTTP/1.1
 200 5.389471   128.119.245.12  192.168.1.102   TCP         60 80 → 1161 [ACK] Seq=1 Ack=162309 Win=62780 Len=
 201 5.447887   128.119.245.12  192.168.1.102   TCP         60 80 → 1161 [ACK] Seq=1 Ack=164041 Win=62780 Len=
 202 5.455830   128.119.245.12  192.168.1.102   TCP         60 80 → 1161 [ACK] Seq=1 Ack=164091 Win=62780 Len=
 203 5.461175   128.119.245.12  192.168.1.102   HTTP       784 HTTP/1.1 200 OK  (text/html)
 206 5.651141   192.168.1.102   128.119.245.12  TCP         54 1161 → 80 [ACK] Seq=164091 Ack=731 Win=16790 L
 213 7.595557   192.168.1.102   199.2.53.206    TCP         62 1162 → 631 [SYN] Seq=0 Win=16384 Len=0 MSS=146

     .... ...0 .... = Acknowledgment: Not set
     .... .... 0... = Push: Not set
     .... .... .0.. = Reset: Not set
   > .... .... ..1. = Syn: Set
     .... .... ...0 = Fin: Not set
     [TCP Flags: ··········S·]
```

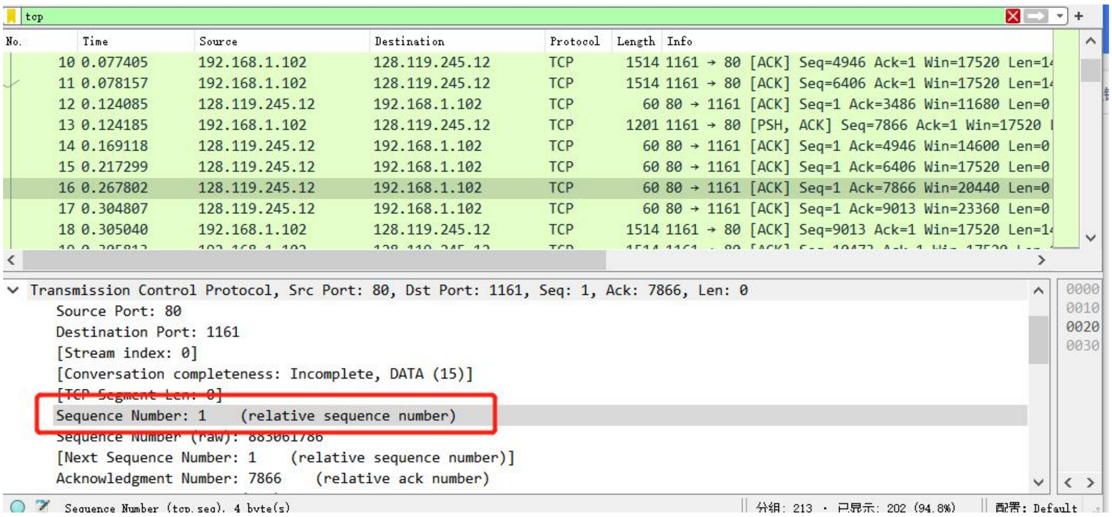
6.What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you’ll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a “POST” within its DATA field.

首先，筛选出 HTTP 报文，找到 HTTP POST，查看其 TCP 分组，如下图所示：



```
http
No.  Time      Source          Destination     Protocol Length Info
 199 5.297341  192.168.1.102   128.119.245.12  HTTP      104 POST /ethereal-labs/lab3-1-reply.htm HTTP/1.1

> Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 164041, Ack: 1, Len: 50
v [122 Reassembled TCP Segments (164090 bytes): #4(565), #5(1460), #7(1460), #8(1460), #10(1460), #11(1460), #13(1147), #1
    [Frame: 4, payload: 0-564 (565 bytes)]
    [Frame: 5, payload: 565-2024 (1460 bytes)]
    [Frame: 7, payload: 2025-3484 (1460 bytes)]
    [Frame: 8, payload: 3485-4944 (1460 bytes)]
    [Frame: 10, payload: 4945-6404 (1460 bytes)]
    [Frame: 11, payload: 6405-7864 (1460 bytes)]
    [Frame: 13, payload: 7865-9011 (1147 bytes)]
    [Frame: 18, payload: 9012-10471 (1460 bytes)]
    [Frame: 19, payload: 10472-11931 (1460 bytes)]
    [Frame: 20, payload: 11932-13391 (1460 bytes)]
    [Frame: 21, payload: 13392-14851 (1460 bytes)]
    [Frame: 22, payload: 14852-16311 (1460 bytes)]
    [Frame: 23, payload: 16312-17203 (892 bytes)]
    [Frame: 30, payload: 17204-18663 (1460 bytes)]
    [Frame: 31, payload: 18664-20123 (1460 bytes)]
    [Frame: 32, payload: 20124-21583 (1460 bytes)]
```

然后再筛选 TCP 报文，查看上述包含 HTTP POST 的 TCP 段的 sequence number：



```
tcp
No.  Time      Source          Destination     Protocol Length Info
  10 0.077405  192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=4946 Ack=1 Win=17520 Len=14
  11 0.078157  192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=14
  12 0.124085  128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=3486 Win=11680 Len=0
  13 0.124185  192.168.1.102   128.119.245.12  TCP      1201 1161 → 80 [PSH, ACK] Seq=7866 Ack=1 Win=17520 L
  14 0.169118  128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=4946 Win=14600 Len=0
  15 0.217299  128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=6406 Win=17520 Len=0
  16 0.267802  128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=7866 Win=20440 Len=0
  17 0.304807  128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=9013 Win=23360 Len=0
  18 0.305040  192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=9013 Ack=1 Win=17520 Len=14

v Transmission Control Protocol, Src Port: 80, Dst Port: 1161, Seq: 1, Ack: 7866, Len: 0
    Source Port: 80
    Destination Port: 1161
    [Stream index: 0]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 0]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 883061786
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 7866    (relative ack number)

Sequence Number (tcp.seq), 4 byte(s)        分组: 213 · 已显示: 202 (94.8%)        配置: Default
```

由图可知，包含 HTTP POST 的 TCP 段的 sequence number 为 1。

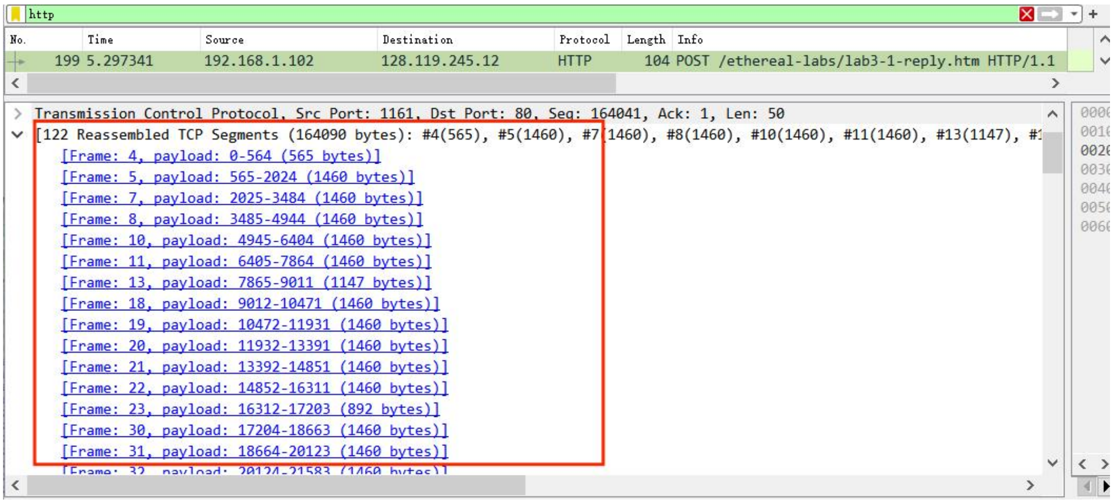
7. Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection. What are the sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST)? At what time was each segment sent? When was the ACK for each segment received? Given the difference between when each TCP segment was sent, and when its acknowledgement was received, what is the RTT value for each of the six segments? What is the EstimatedRTT value (see Section 3.5.3, page 239 in text) after the receipt of each ACK? Assume that the value of the EstimatedRTT is equal to the measured RTT for the first segment, and then is computed using the EstimatedRTT equation on page 239 for all subsequent segments.

先查看 HTTP POST 中的分组序号，然后查看各个 TCP 分组，TCP 连接中的前六个段的 Sequence number，发送时间、ACK 接受时间、RTT、和 EstimatedRTT 如

下表所示：

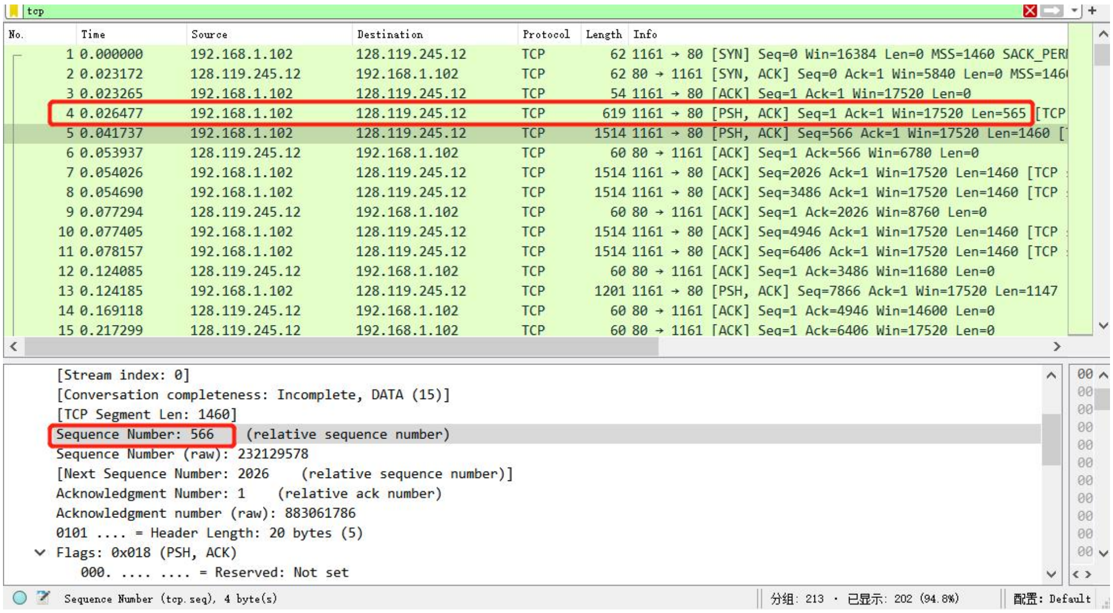
| Segment | Length | Sequence number | Sent time | ACK received time | RTT | Estimated RTT |
|---|---|---|---|---|---|---|
| 1 | 565 | 1 | 0.026477 | 0.053937 | 0.02746 | 0.02746 |
| 2 | 1460 | 566 | 0.041737 | 0.077294 | 0.035557 | 0.028472 |
| 3 | 1460 | 2026 | 0.054026 | 0.124085 | 0.070059 | 0.033670 |
| 4 | 1460 | 3486 | 0.054690 | 0.169118 | 0.11443 | 0.043765 |
| 5 | 1460 | 4946 | 0.077405 | 0.217299 | 0.13989 | 0.055781 |
| 6 | 1460 | 6406 | 0.078157 | 0.267802 | 0.18964 | 0.072513 |

查看分组的截图如下：



```
No.   Time       Source          Destination     Protocol  Length  Info
  199 5.297341   192.168.1.102   128.119.245.12  HTTP      104 POST /ethereal-labs/lab3-1-reply.htm HTTP/1.1
```

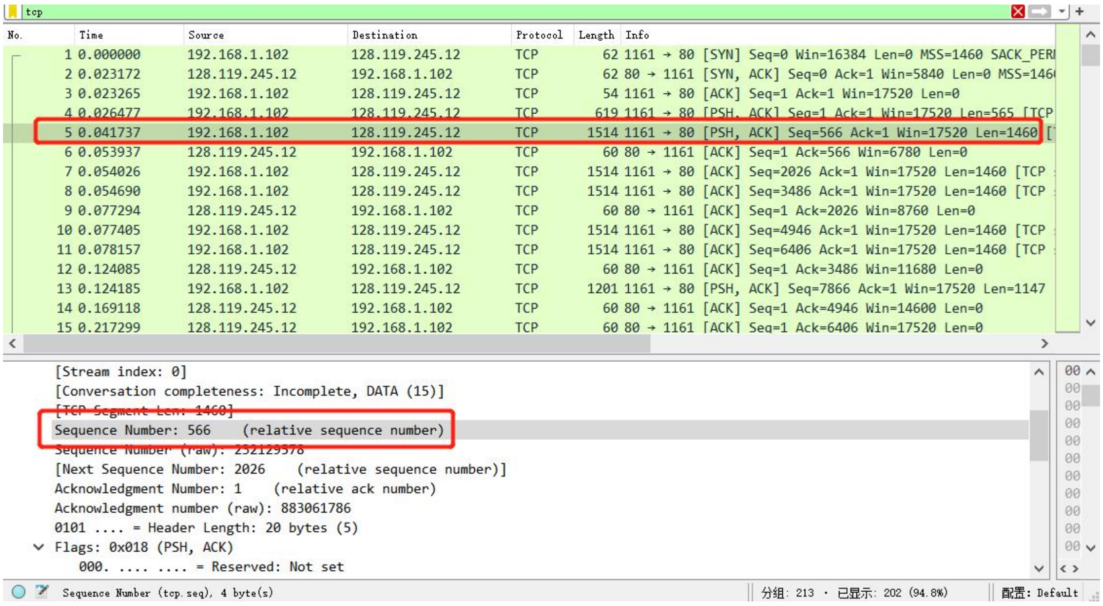
```
> Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 164041, Ack: 1, Len: 50
v [122 Reassembled TCP Segments (164090 bytes): #4(565), #5(1460), #7(1460), #8(1460), #10(1460), #11(1460), #13(1147), #1
    [Frame: 4, payload: 0-564 (565 bytes)]
    [Frame: 5, payload: 565-2024 (1460 bytes)]
    [Frame: 7, payload: 2025-3484 (1460 bytes)]
    [Frame: 8, payload: 3485-4944 (1460 bytes)]
    [Frame: 10, payload: 4945-6404 (1460 bytes)]
    [Frame: 11, payload: 6405-7864 (1460 bytes)]
    [Frame: 13, payload: 7865-9011 (1147 bytes)]
    [Frame: 18, payload: 9012-10471 (1460 bytes)]
    [Frame: 19, payload: 10472-11931 (1460 bytes)]
    [Frame: 20, payload: 11932-13391 (1460 bytes)]
    [Frame: 21, payload: 13392-14851 (1460 bytes)]
    [Frame: 22, payload: 14852-16311 (1460 bytes)]
    [Frame: 23, payload: 16312-17203 (892 bytes)]
    [Frame: 30, payload: 17204-18663 (1460 bytes)]
    [Frame: 31, payload: 18664-20123 (1460 bytes)]
    [Frame: 32, payload: 20124-21583 (1460 bytes)]
```



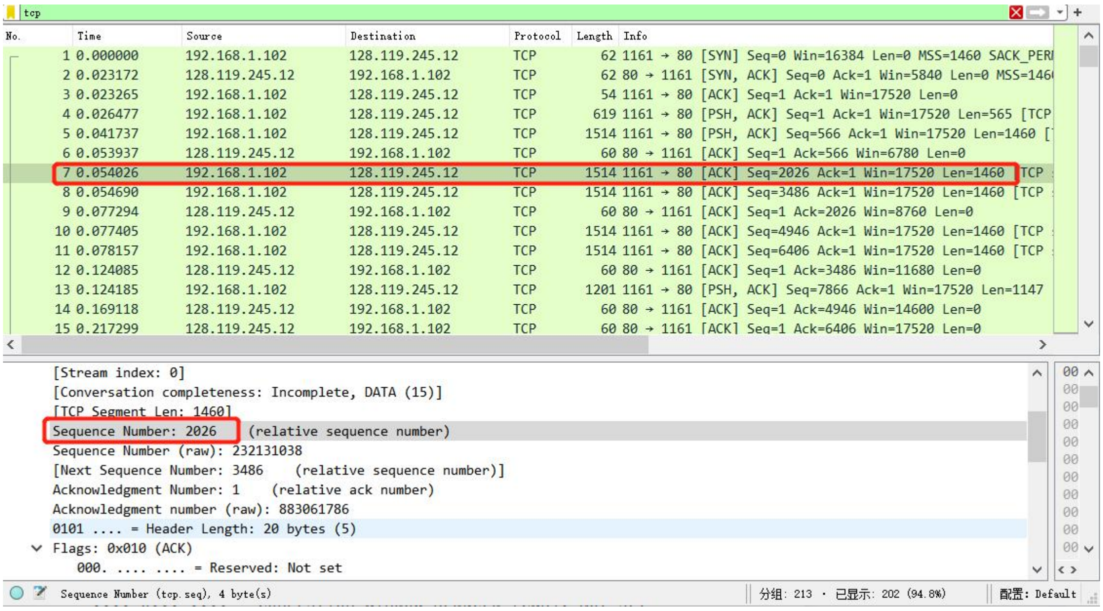
```
No.  Time       Source          Destination     Protocol  Length  Info
   1 0.000000   192.168.1.102   128.119.245.12  TCP         62 1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERI
   2 0.023172   128.119.245.12  192.168.1.102   TCP         62 80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=146
   3 0.023265   192.168.1.102   128.119.245.12  TCP         54 1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0
   4 0.026477   192.168.1.102   128.119.245.12  TCP        619 1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565 [TCP
   5 0.041737   192.168.1.102   128.119.245.12  TCP       1514 1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=1460 [
   6 0.053937   128.119.245.12  192.168.1.102   TCP         60 80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
   7 0.054026   192.168.1.102   128.119.245.12  TCP       1514 1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460 [TCP
   8 0.054690   192.168.1.102   128.119.245.12  TCP       1514 1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460 [TCP
   9 0.077294   128.119.245.12  192.168.1.102   TCP         60 80 → 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0
  10 0.077405   192.168.1.102   128.119.245.12  TCP       1514 1161 → 80 [ACK] Seq=4946 Ack=1 Win=17520 Len=1460 [TCP
  11 0.078157   192.168.1.102   128.119.245.12  TCP       1514 1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460 [TCP
  12 0.124085   128.119.245.12  192.168.1.102   TCP         60 80 → 1161 [ACK] Seq=1 Ack=3486 Win=11680 Len=0
  13 0.124185   192.168.1.102   128.119.245.12  TCP       1201 1161 → 80 [PSH, ACK] Seq=7866 Ack=1 Win=17520 Len=1147
  14 0.169118   128.119.245.12  192.168.1.102   TCP         60 80 → 1161 [ACK] Seq=1 Ack=4946 Win=14600 Len=0
  15 0.217299   128.119.245.12  192.168.1.102   TCP         60 80 → 1161 [ACK] Seq=1 Ack=6406 Win=17520 Len=0
```

```
    [Stream index: 0]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 1460]
    Sequence Number: 566    (relative sequence number)
    Sequence Number (raw): 232129578
    [Next Sequence Number: 2026    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 883061786
    0101 .... = Header Length: 20 bytes (5)
  v Flags: 0x018 (PSH, ACK)
      000. .... .... = Reserved: Not set
```

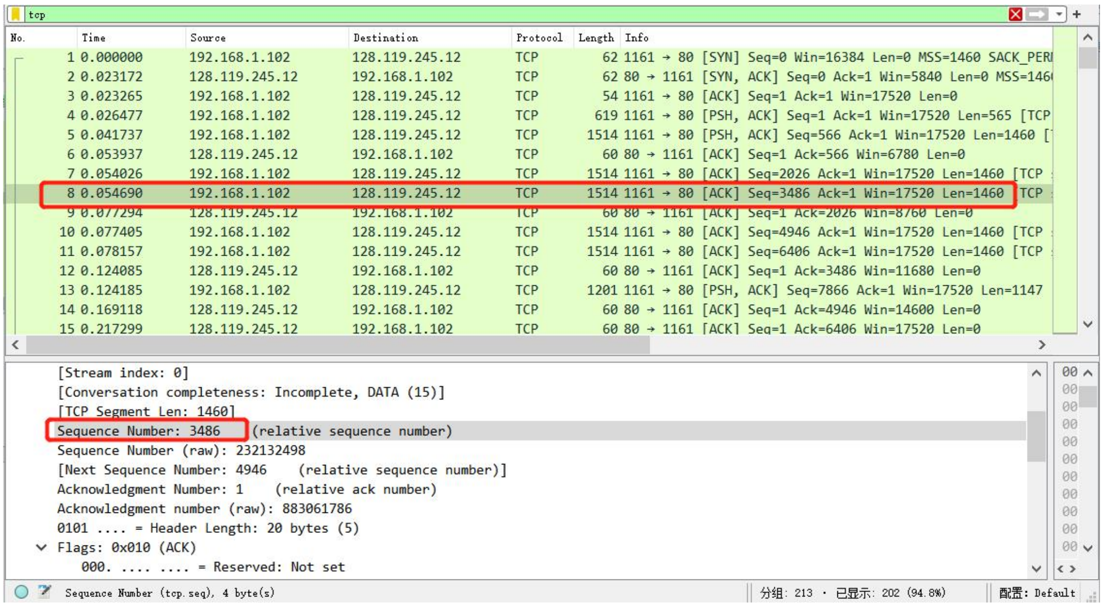
Sequence Number (tcp.seq), 4 byte(s) | 分组: 213 · 已显示: 202 (94.8%) | 配置: Default



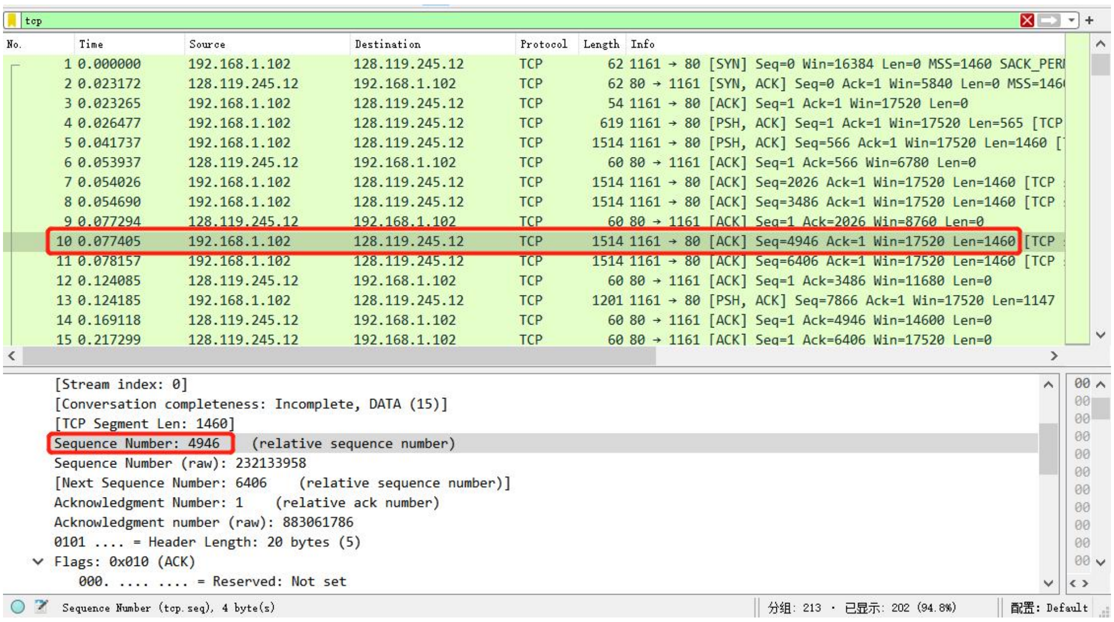
```
tcp
No.  Time      Source          Destination     Protocol Length Info
   1 0.000000  192.168.1.102   128.119.245.12  TCP        62 1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PER
   2 0.023172  128.119.245.12  192.168.1.102   TCP        62 80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=146
   3 0.023265  192.168.1.102   128.119.245.12  TCP        54 1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0
   4 0.026477  192.168.1.102   128.119.245.12  TCP       619 1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565 [TCP
   5 0.041737  192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=1460 [
   6 0.053937  128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
   7 0.054026  192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460 [TCP
   8 0.054690  192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460 [TCP
   9 0.077294  128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0
  10 0.077405  192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=4946 Ack=1 Win=17520 Len=1460 [TCP
  11 0.078157  192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460 [TCP
  12 0.124085  128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=3486 Win=11680 Len=0
  13 0.124185  192.168.1.102   128.119.245.12  TCP      1201 1161 → 80 [PSH, ACK] Seq=7866 Ack=1 Win=17520 Len=1147
  14 0.169118  128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=4946 Win=14600 Len=0
  15 0.217299  128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=6406 Win=17520 Len=0

    [Stream index: 0]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 1460]
    Sequence Number: 566    (relative sequence number)
    Sequence Number (raw): 232129578
    [Next Sequence Number: 2026    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 883061786
    0101 .... = Header Length: 20 bytes (5)
  ∨ Flags: 0x018 (PSH, ACK)
      000. .... .... = Reserved: Not set

Sequence Number (tcp.seq), 4 byte(s)        分组: 213 · 已显示: 202 (94.8%)        配置: Default
```



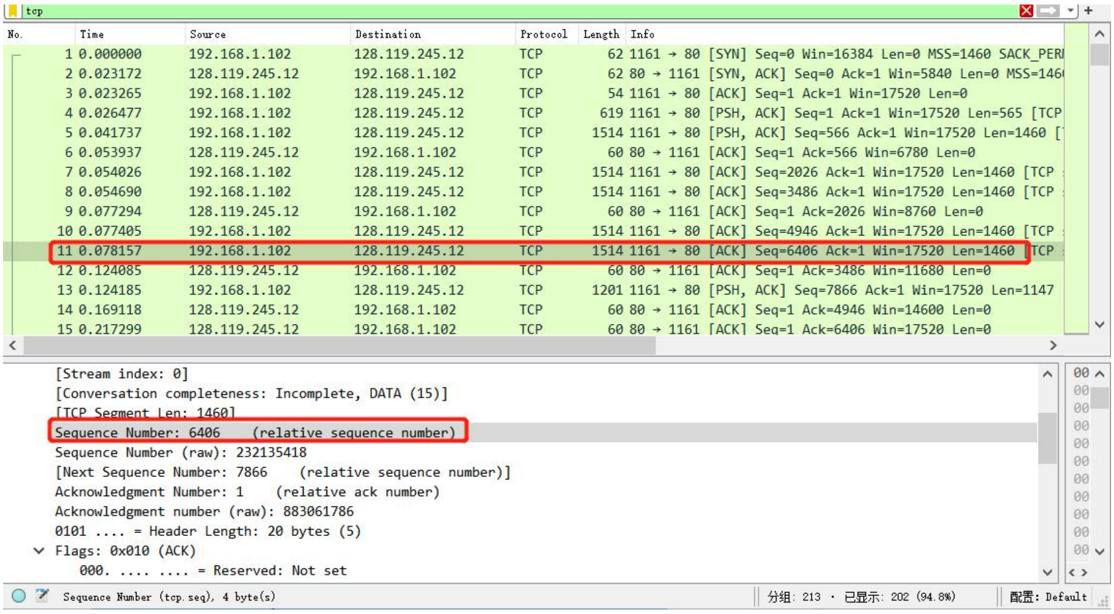
```
tcp
No.  Time      Source          Destination     Protocol Length Info
   1 0.000000  192.168.1.102   128.119.245.12  TCP        62 1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PER
   2 0.023172  128.119.245.12  192.168.1.102   TCP        62 80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=146
   3 0.023265  192.168.1.102   128.119.245.12  TCP        54 1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0
   4 0.026477  192.168.1.102   128.119.245.12  TCP       619 1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565 [TCP
   5 0.041737  192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=1460 [
   6 0.053937  128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
   7 0.054026  192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460 [TCP
   8 0.054690  192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460 [TCP
   9 0.077294  128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0
  10 0.077405  192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=4946 Ack=1 Win=17520 Len=1460 [TCP
  11 0.078157  192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460 [TCP
  12 0.124085  128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=3486 Win=11680 Len=0
  13 0.124185  192.168.1.102   128.119.245.12  TCP      1201 1161 → 80 [PSH, ACK] Seq=7866 Ack=1 Win=17520 Len=1147
  14 0.169118  128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=4946 Win=14600 Len=0
  15 0.217299  128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=6406 Win=17520 Len=0

    [Stream index: 0]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 1460]
    Sequence Number: 2026    (relative sequence number)
    Sequence Number (raw): 232131038
    [Next Sequence Number: 3486    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 883061786
    0101 .... = Header Length: 20 bytes (5)
  ∨ Flags: 0x010 (ACK)
      000. .... .... = Reserved: Not set

Sequence Number (tcp.seq), 4 byte(s)        分组: 213 · 已显示: 202 (94.8%)        配置: Default
```



```
tcp
No.  Time      Source          Destination     Protocol Length Info
   1 0.000000  192.168.1.102   128.119.245.12  TCP        62 1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PER
   2 0.023172  128.119.245.12  192.168.1.102   TCP        62 80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=146
   3 0.023265  192.168.1.102   128.119.245.12  TCP        54 1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0
   4 0.026477  192.168.1.102   128.119.245.12  TCP       619 1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565 [TCP
   5 0.041737  192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=1460 [
   6 0.053937  128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
   7 0.054026  192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460 [TCP
   8 0.054690  192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460 [TCP
   9 0.077294  128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0
  10 0.077405  192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=4946 Ack=1 Win=17520 Len=1460 [TCP
  11 0.078157  192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460 [TCP
  12 0.124085  128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=3486 Win=11680 Len=0
  13 0.124185  192.168.1.102   128.119.245.12  TCP      1201 1161 → 80 [PSH, ACK] Seq=7866 Ack=1 Win=17520 Len=1147
  14 0.169118  128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=4946 Win=14600 Len=0
  15 0.217299  128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=6406 Win=17520 Len=0

    [Stream index: 0]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 1460]
    Sequence Number: 3486    (relative sequence number)
    Sequence Number (raw): 232132498
    [Next Sequence Number: 4946    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 883061786
    0101 .... = Header Length: 20 bytes (5)
  ∨ Flags: 0x010 (ACK)
      000. .... .... = Reserved: Not set

Sequence Number (tcp.seq), 4 byte(s)        分组: 213 · 已显示: 202 (94.8%)        配置: Default
```



```
tcp
No.   Time       Source          Destination     Protocol Length Info
    1 0.000000   192.168.1.102   128.119.245.12  TCP        62 1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PER
    2 0.023172   128.119.245.12  192.168.1.102   TCP        62 80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=146
    3 0.023265   192.168.1.102   128.119.245.12  TCP        54 1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0
    4 0.026477   192.168.1.102   128.119.245.12  TCP       619 1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565 [TCP
    5 0.041737   192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=1460 [
    6 0.053937   128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
    7 0.054026   192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460 [TCP
    8 0.054690   192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460 [TCP
    9 0.077294   128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0
   10 0.077405   192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=4946 Ack=1 Win=17520 Len=1460 [TCP
   11 0.078157   192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460 [TCP
   12 0.124085   128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=3486 Win=11680 Len=0
   13 0.124185   192.168.1.102   128.119.245.12  TCP      1201 1161 → 80 [PSH, ACK] Seq=7866 Ack=1 Win=17520 Len=1147
   14 0.169118   128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=4946 Win=14600 Len=0
   15 0.217299   128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=6406 Win=17520 Len=0

    [Stream index: 0]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 1460]
    Sequence Number: 4946    (relative sequence number)
    Sequence Number (raw): 232133958
    [Next Sequence Number: 6406    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 883061786
    0101 .... = Header Length: 20 bytes (5)
  ∨ Flags: 0x010 (ACK)
       000. .... .... = Reserved: Not set

Sequence Number (tcp.seq), 4 byte(s)          分组: 213 · 已显示: 202 (94.8%)          配置: Default
```



```
tcp
No.   Time       Source          Destination     Protocol Length Info
    1 0.000000   192.168.1.102   128.119.245.12  TCP        62 1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PER
    2 0.023172   128.119.245.12  192.168.1.102   TCP        62 80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=146
    3 0.023265   192.168.1.102   128.119.245.12  TCP        54 1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0
    4 0.026477   192.168.1.102   128.119.245.12  TCP       619 1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565 [TCP
    5 0.041737   192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=1460 [
    6 0.053937   128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
    7 0.054026   192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460 [TCP
    8 0.054690   192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460 [TCP
    9 0.077294   128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0
   10 0.077405   192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=4946 Ack=1 Win=17520 Len=1460 [TCP
   11 0.078157   192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460 [TCP
   12 0.124085   128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=3486 Win=11680 Len=0
   13 0.124185   192.168.1.102   128.119.245.12  TCP      1201 1161 → 80 [PSH, ACK] Seq=7866 Ack=1 Win=17520 Len=1147
   14 0.169118   128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=4946 Win=14600 Len=0
   15 0.217299   128.119.245.12  192.168.1.102   TCP        60 80 → 1161 [ACK] Seq=1 Ack=6406 Win=17520 Len=0

    [Stream index: 0]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 1460]
    Sequence Number: 6406    (relative sequence number)
    Sequence Number (raw): 232135418
    [Next Sequence Number: 7866    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 883061786
    0101 .... = Header Length: 20 bytes (5)
  ∨ Flags: 0x010 (ACK)
       000. .... .... = Reserved: Not set

Sequence Number (tcp.seq), 4 byte(s)          分组: 213 · 已显示: 202 (94.8%)          配置: Default
```

使用 239 页的 EspatedRTT 方程对所有分组进行计算：

EstimatedRTT = 0.875 * Last EstimatedRTT + 0.125 * sample RTT

After Segment 1 : EstimatedRTT = 0.02746

After Segment 2 : EstimatedRTT = 0.875 * 0.02746 + 0.125*0.035557 = 0.028472

After Segment 3 : EstimatedRTT = 0.875 * 0.028472 + 0.125*0. 070059 = 0.033670

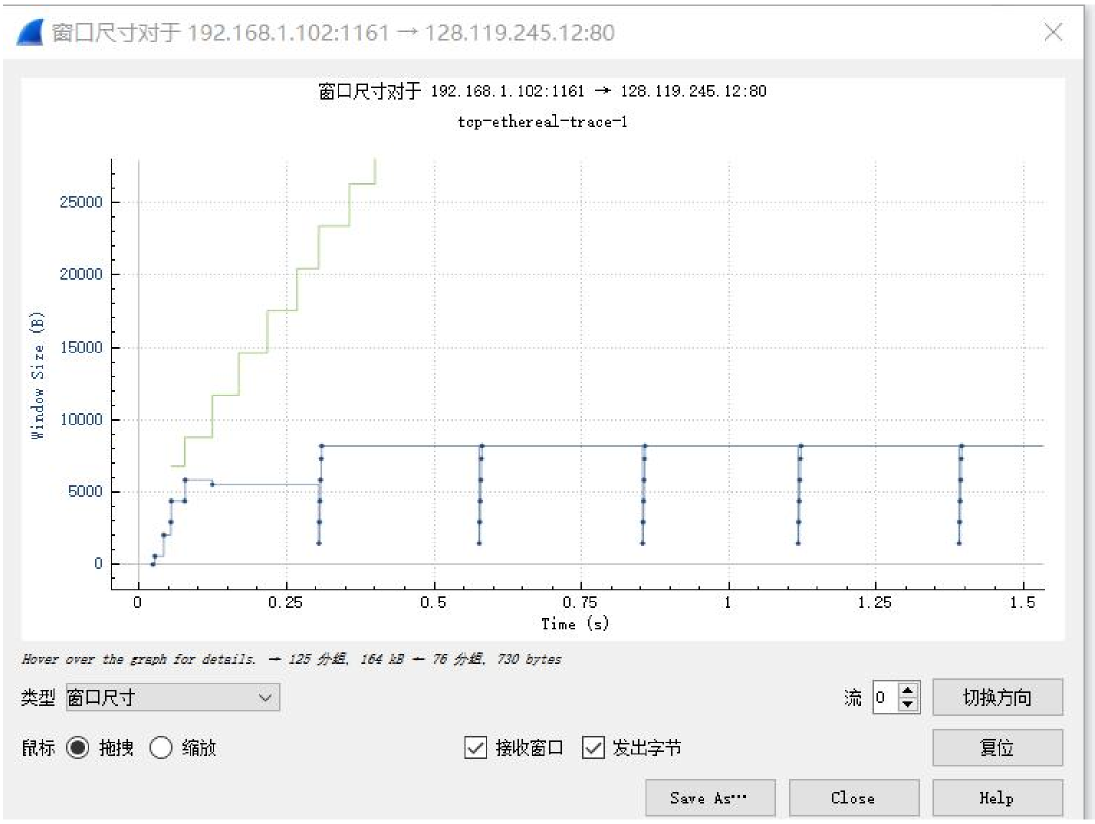
After Segment 4 : EstimatedRTT = 0.875 * 0.033670 + 0.125*0.11443 = 0.043765

After Segment 5 : EstimatedRTT = 0.875 * 0.043765 + 0.125*0.13989 = 0.055781

After Segment 6 : EstimatedRTT = 0.875 * 0.055781 + 0.125*0.18964 = 0.072513

9.What is the minimum amount of available buffer space advertised at the received for the entire trace? Does the lack of receiver buffer space ever throttle the sender?

通过查看串口尺寸统计图可以确定最小窗口尺寸分组的分组中序号，然后在根据分组序号找到分组，最小窗口尺寸为 5840 字节。



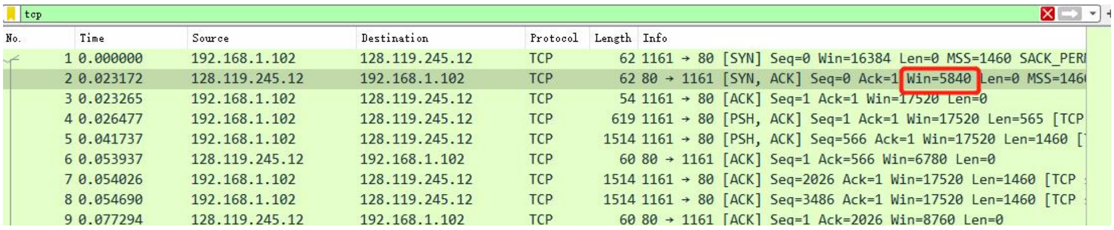
Hover over the graph for details. → 125 分组, 164 kB ← 76 分组, 730 bytes

类型 窗口尺寸

流 0

鼠标 ● 拖拽 ○ 缩放

☑ 接收窗口 ☑ 发出字节



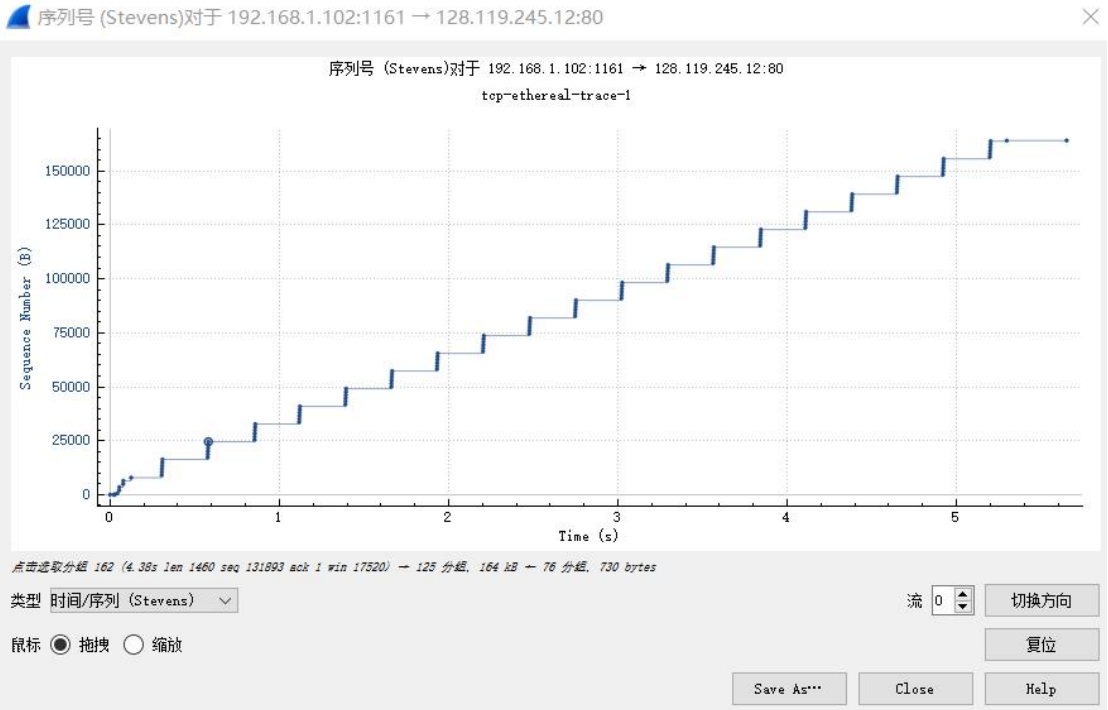
tcp

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 1 | 0.000000 | 192.168.1.102 | 128.119.245.12 | TCP | 62 | 1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERI |
| 2 | 0.023172 | 128.119.245.12 | 192.168.1.102 | TCP | 62 | 80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=146 |
| 3 | 0.023265 | 192.168.1.102 | 128.119.245.12 | TCP | 54 | 1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0 |
| 4 | 0.026477 | 192.168.1.102 | 128.119.245.12 | TCP | 619 | 1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565 [TCP |
| 5 | 0.041737 | 192.168.1.102 | 128.119.245.12 | TCP | 1514 | 1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=1460 [ |
| 6 | 0.053937 | 128.119.245.12 | 192.168.1.102 | TCP | 60 | 80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0 |
| 7 | 0.054026 | 192.168.1.102 | 128.119.245.12 | TCP | 1514 | 1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460 [TCP |
| 8 | 0.054690 | 192.168.1.102 | 128.119.245.12 | TCP | 1514 | 1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460 [TCP |
| 9 | 0.077294 | 128.119.245.12 | 192.168.1.102 | TCP | 60 | 80 → 1161 [ACK] Seq=1 Ack=2026 Win=8760 Len=0 |

缺少接收器缓冲区空间会限制发送方传送 TCP 区段。TCP 使用接收方通知窗口大小来限制发送方流量，当接收方缓冲空间不足时（win 为 0），接收方会向发送方发送一个零窗口通知，表示缓冲区已满，无法继续接收数据，此时，发送方需要停止发送数据，避免丢包和网络阻塞。

10.Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?

通过查看序列号流图可知，没有重新发送的段。如下图所示：



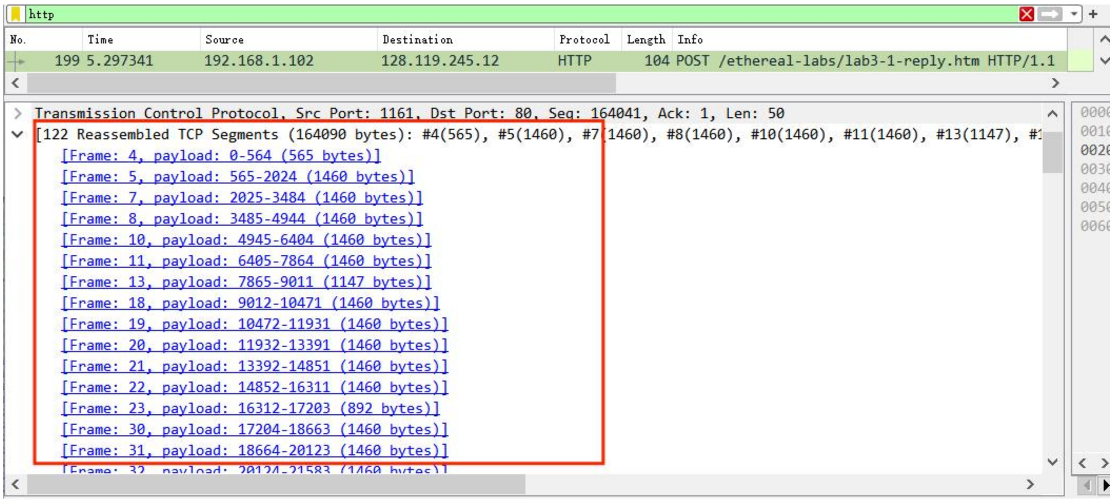
点击选取分组 162 (4.38s len 1460 seq 131893 ack 1 win 17520) → 125 分组, 164 kB ← 76 分组, 730 bytes

类型 时间/序列 (Stevens)　　流 0

鼠标 ● 拖拽 ○ 缩放

11.How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (see Table 3.2 on page 247 in the text).

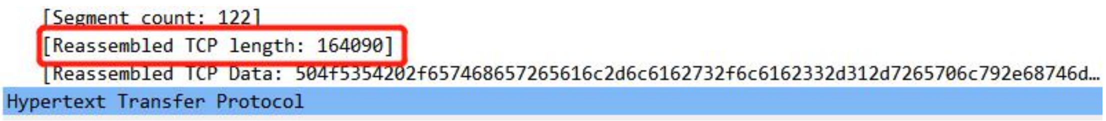
查看 HTTP POST 可知，服务器在这两个 ack 之间接收到的数据是 1460 字节。在某些情况下，接收器每隔 2920 字节（1460*2 字节）进行一次确认。如下图所示：



```
http
No.   Time       Source          Destination     Protocol  Length  Info
199   5.297341   192.168.1.102   128.119.245.12  HTTP      104     POST /ethereal-labs/lab3-1-reply.htm HTTP/1.1

> Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 164041, Ack: 1, Len: 50
v [122 Reassembled TCP Segments (164090 bytes): #4(565), #5(1460), #7(1460), #8(1460), #10(1460), #11(1460), #13(1147), #1
    [Frame: 4, payload: 0-564 (565 bytes)]
    [Frame: 5, payload: 565-2024 (1460 bytes)]
    [Frame: 7, payload: 2025-3484 (1460 bytes)]
    [Frame: 8, payload: 3485-4944 (1460 bytes)]
    [Frame: 10, payload: 4945-6404 (1460 bytes)]
    [Frame: 11, payload: 6405-7864 (1460 bytes)]
    [Frame: 13, payload: 7865-9011 (1147 bytes)]
    [Frame: 18, payload: 9012-10471 (1460 bytes)]
    [Frame: 19, payload: 10472-11931 (1460 bytes)]
    [Frame: 20, payload: 11932-13391 (1460 bytes)]
    [Frame: 21, payload: 13392-14851 (1460 bytes)]
    [Frame: 22, payload: 14852-16311 (1460 bytes)]
    [Frame: 23, payload: 16312-17203 (892 bytes)]
    [Frame: 30, payload: 17204-18663 (1460 bytes)]
    [Frame: 31, payload: 18664-20123 (1460 bytes)]
    [Frame: 32, payload: 20124-21583 (1460 bytes)]
```

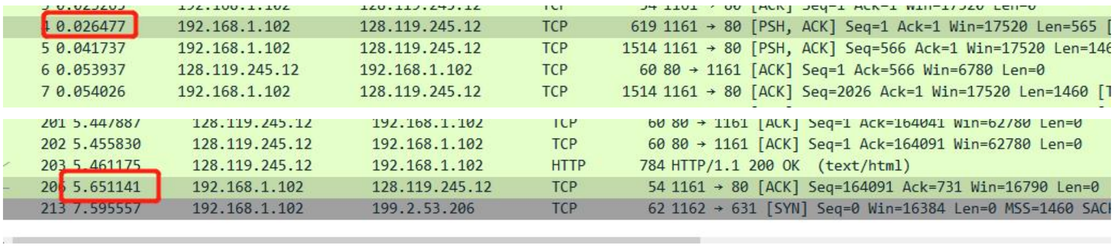
12.What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

通过查看 HTTP POST 可知总数据量为 164090 字节；



```
[Segment count: 122]
[Reassembled TCP length: 164090]
[Reassembled TCP Data: 504f5354202f657468657265616c2d6c6162732f6c6162332d312d7265706c792e68746d…
Hypertext Transfer Protocol
```

查看第一个和最后一个分组的发送时间可知：总传输时间为 5.297341 秒。



```
  3 0.025263     192.168.1.102     128.119.245.12     TCP       54 1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0
  4 0.026477     192.168.1.102     128.119.245.12     TCP      619 1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565 [
  5 0.041737     192.168.1.102     128.119.245.12     TCP     1514 1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=146
  6 0.053937     128.119.245.12    192.168.1.102      TCP       60 80 → 1161 [ACK] Seq=1 Ack=566 Win=6780 Len=0
  7 0.054026     192.168.1.102     128.119.245.12     TCP     1514 1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460 [T

201 5.447887     128.119.245.12    192.168.1.102      TCP       60 80 → 1161 [ACK] Seq=1 Ack=164041 Win=62780 Len=0
202 5.455830     128.119.245.12    192.168.1.102      TCP       60 80 → 1161 [ACK] Seq=1 Ack=164091 Win=62780 Len=0
203 5.461175     128.119.245.12    192.168.1.102      HTTP     784 HTTP/1.1 200 OK  (text/html)
206 5.651141     192.168.1.102     128.119.245.12     TCP       54 1161 → 80 [ACK] Seq=164091 Ack=731 Win=16790 Len=0
213 7.595557     192.168.1.102     199.2.53.206       TCP       62 1162 → 631 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACI
```

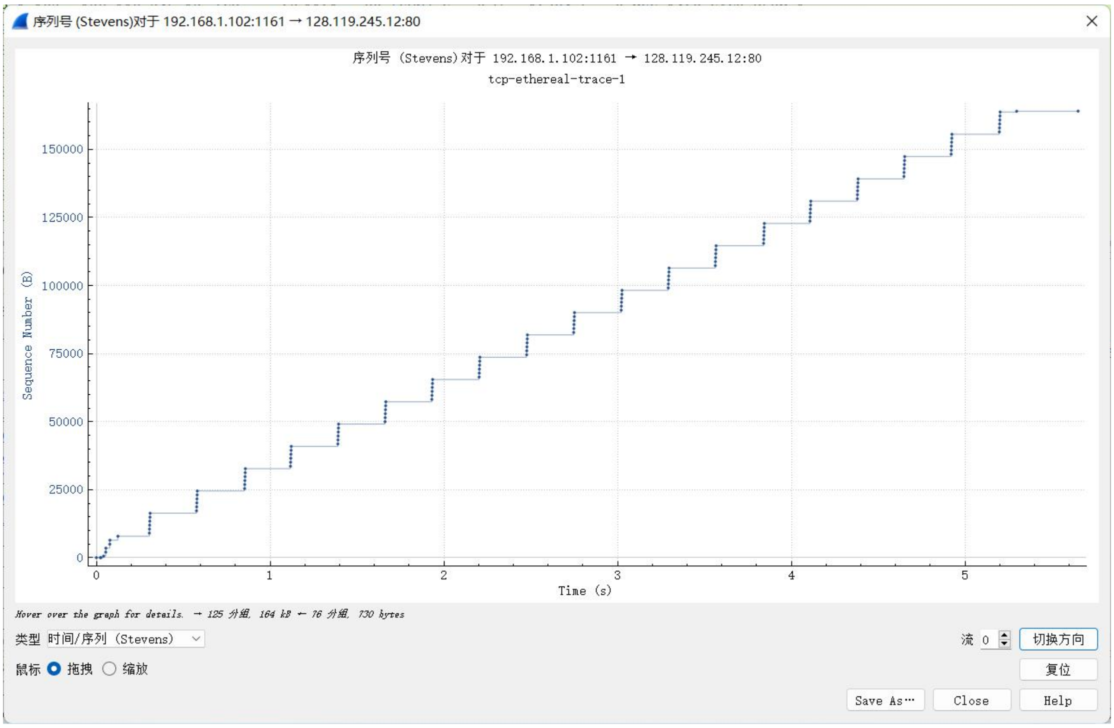
TCP 平均吞吐量 ＝ 传输数据的比特数 F ÷ 接收方接收所有数据所用时间 T ＝164090/5.297341＝30975.91791806493 字节/秒。

## 4、TCP 拥塞控制

### （1）实验步骤

① 使用 TCP 图形工具查看时间序列图：在 Wireshark 的“捕获数据包列表”窗口中选择一个 TCP 段。然后选择菜单：统计数据->TCP 流图->时间-序列-图。

② 打开 trace tcp-etherealtrace-1，利用其中的分组回答下面的问题。我的 Wireshark 中 trace tcp-etherealtrace-1 对应的时间序列图如下图所示：



Hover over the graph for details. → 125 分组, 164 kB ← 76 分组, 730 bytes

类型 时间/序列 (Stevens)

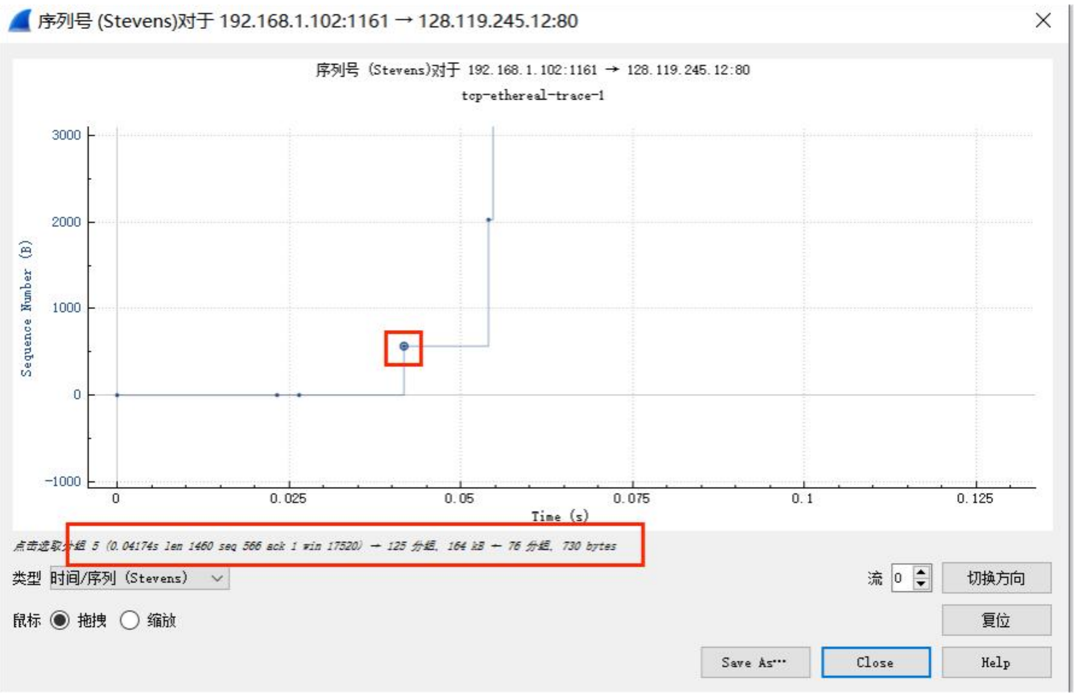
鼠标 ● 拖拽 ○ 缩放

流 0

## （2）回答问题

13.Use the *Time-Sequence-Graph(Stevens)* plotting tool to view the sequence number versus time plot of segments being sent from the client to the gaia.cs.umass.edu server. Can you identify where TCP's slowstart phase begins and ends, and where congestion avoidance takes over? Comment on ways in which the measured data differs from the idealized behavior of TCP that we've studied in the text.

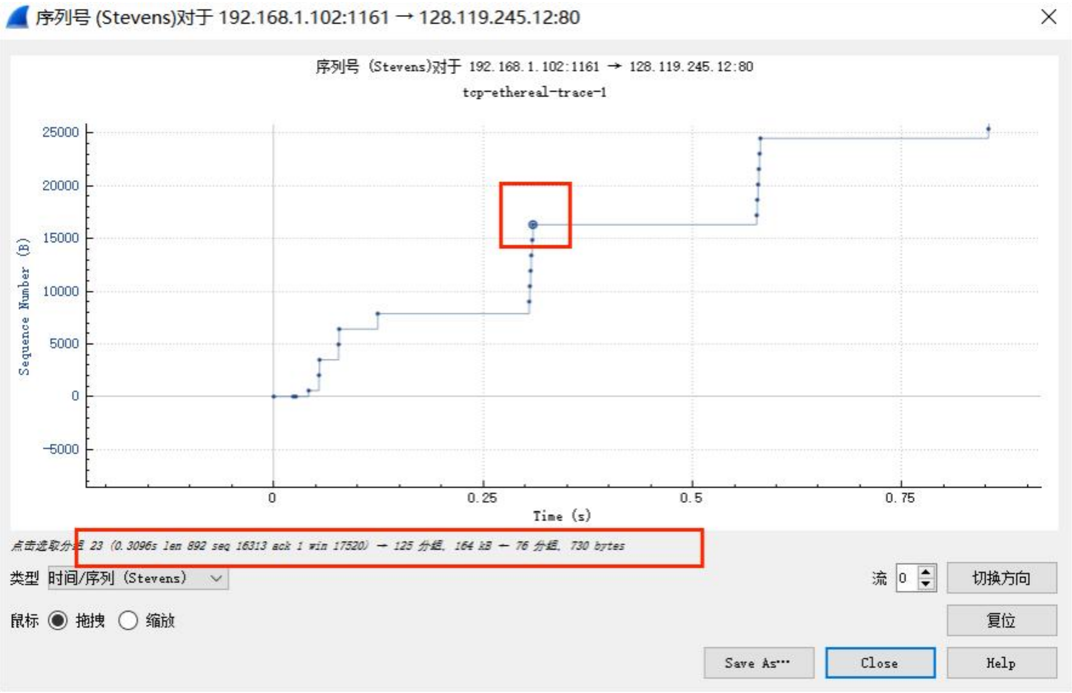
由 trace tcp-etherealtrace-1 对应的时间序列图可知，分组 5 开始慢启动,分组 23 处进入拥塞避免阶段。如下图所示：



点击选取分组 5 (0.04174s len 1460 seq 566 ack 1 win 17520) → 125 分组, 164 kB ← 76 分组, 730 bytes

类型 时间/序列 (Stevens)

鼠标 ● 拖拽 ○ 缩放



点击选取分组 23 (0.3096s len 892 seq 16313 ack 1 win 17520) → 125 分组, 164 kB ← 76 分组, 730 bytes

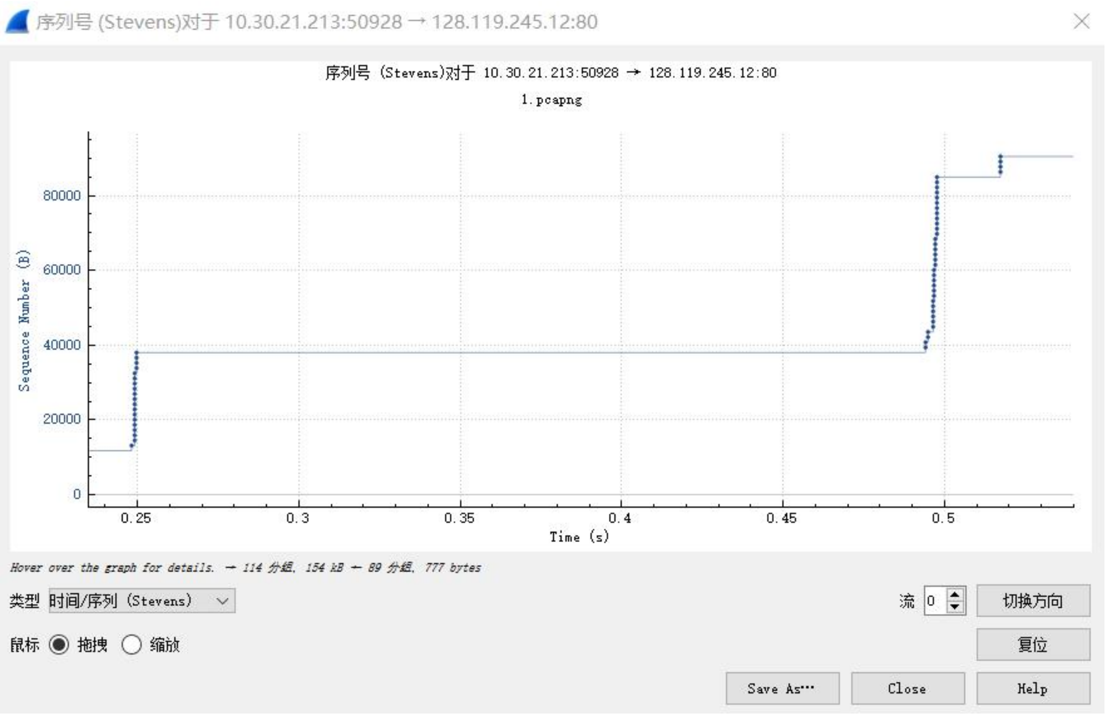
类型 时间/序列 (Stevens)

鼠标 ● 拖拽 ○ 缩放

慢启动是 TCP 在拥塞控制方面的措施之一，但是对于一些数据量较小的小文件，在网络畅通的情况下发送非常快，甚至可能在慢启动结束之前就已经发送完毕，这是这次测量的数据与 TCP 理想行为的不同之处。

14. Answer each of two questions above for the trace that you have gathered when you transferred a file from your computer to gaia.cs.umass.edu

打开gaia.cs.umass.edu上传文件是捕获的TCP分组对应的时间序列图，如下图所示：



序列号 (Stevens)对于 10.30.21.213:50928 → 128.119.245.12:80

序列号 (Stevens)对于 10.30.21.213:50928 → 128.119.245.12:80
1.pcapng

Hover over the graph for details. → 114 分组, 154 kB ← 89 分组, 777 bytes

类型 时间/序列 (Stevens)

流 0

鼠标 ● 拖拽 ○ 缩放

根据图像可以看出，慢启动阶段即从 HTTP POST 报文段发出时开始，但是无法判断什么时候慢启动结束，以及无法判断拥塞避免阶段何时开始。因为慢启动阶段和拥塞避免阶段的鉴定取决于发送方拥塞窗口的大小，而拥塞窗口的大小并不能从时间-序号图(time-sequence-graph)直接获得。

## 三、实验中存在问题及分析。

首先，有了第一次实验的基础，对于 wireshark 软件逐渐熟悉，在解决问题时逐渐得心应手。然而实验中我依旧遇到了不少问题，由于 TCP 反馈太多，在寻找过程中出现了问题，但最后通过与同学讨论得以解决，再比如抓不到数据包，抓到了之后没有找到需要的信息等. 后来不得已使用了题目给出的包,总算能够将后续的题目写完了. 可能是实验时的网络环境不佳的缘故，今后将争取在良好的条件下进行实验.

在解决问题时，由于 wireshark 新版本与老版本的功能差异，以为自己新版本捕捉的信息与老版本捕捉的信息有差异，是直接新版本出现了错误。通过对 wireshark 版本方面的学习成果认识到版本信息的差异，并解决了心中的困惑。由于本次是对已有 zip 文件中的 trace 文件进行捕获，所以避免了在浏览器中多条网页的相互影响，做起来遇到的问题较少。

总之，本次实验虽然内容较多，但是过程比较顺利，也培养了我逐步解决问题的耐心。