Lab Exercise 4: Exploring TCP

zname : z5187292

Exercise 1: Understanding TCP using Wireshark

Question 1. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection? 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?

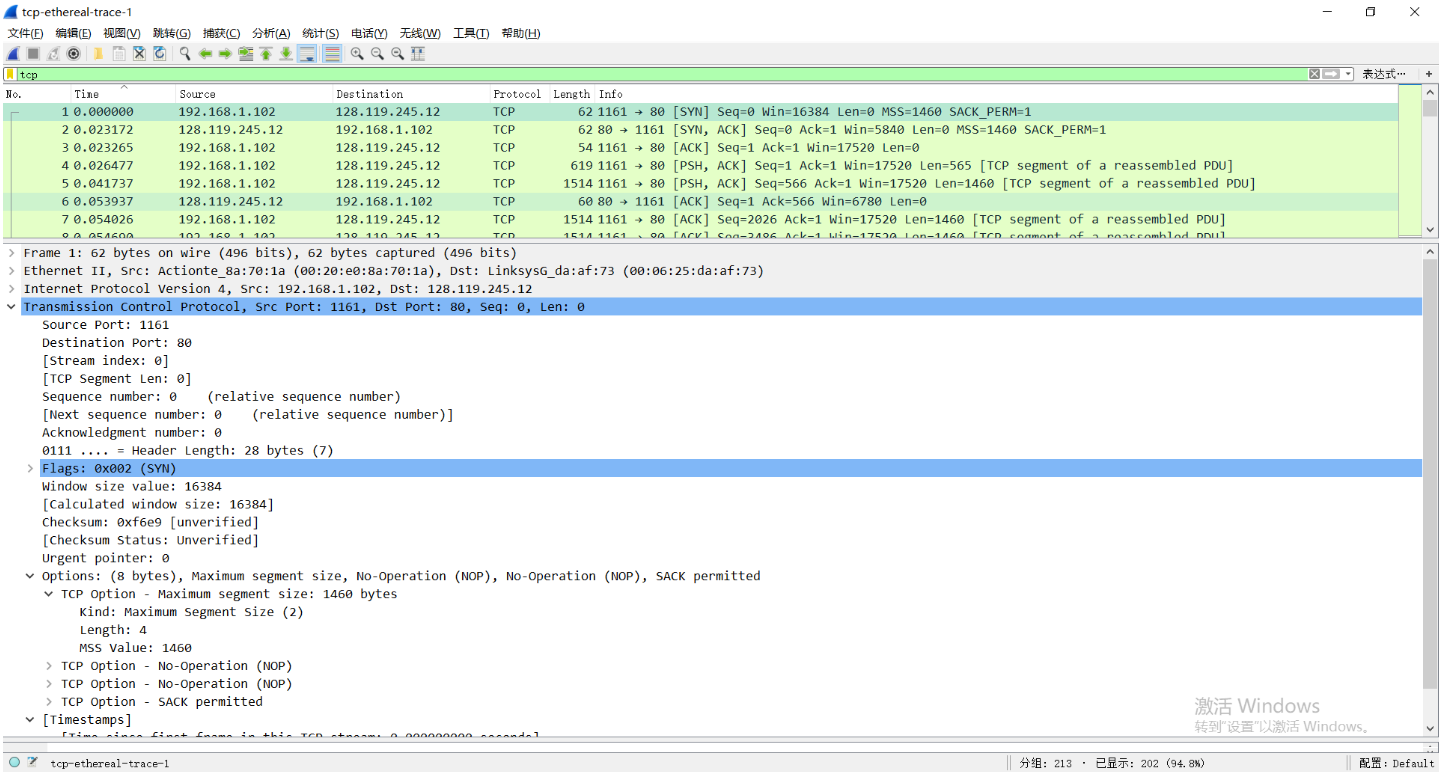
Answer:

As the picture show below,

the IP address of gaia.cs.umass.edu is 128.119.245.12.

For this connection, gaia.cs.umass.edu use port 80 to send and receive TCP segments.

The IP address of client computer(source) is 192.168.1.102, use port 1161 to transfer file.


tcp-ethereal-trace-1
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)
tcp
表达式…
激活 Windows
转到"设置"以激活 Windows。

```
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_PERM=1
    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=1460 SACK_PERM=1
    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 segment of a reassembled PDU]
    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 [TCP segment of a reassembled PDU]
    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 segment of a reassembled PDU]
    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 segment of a reassembled PDU]
```

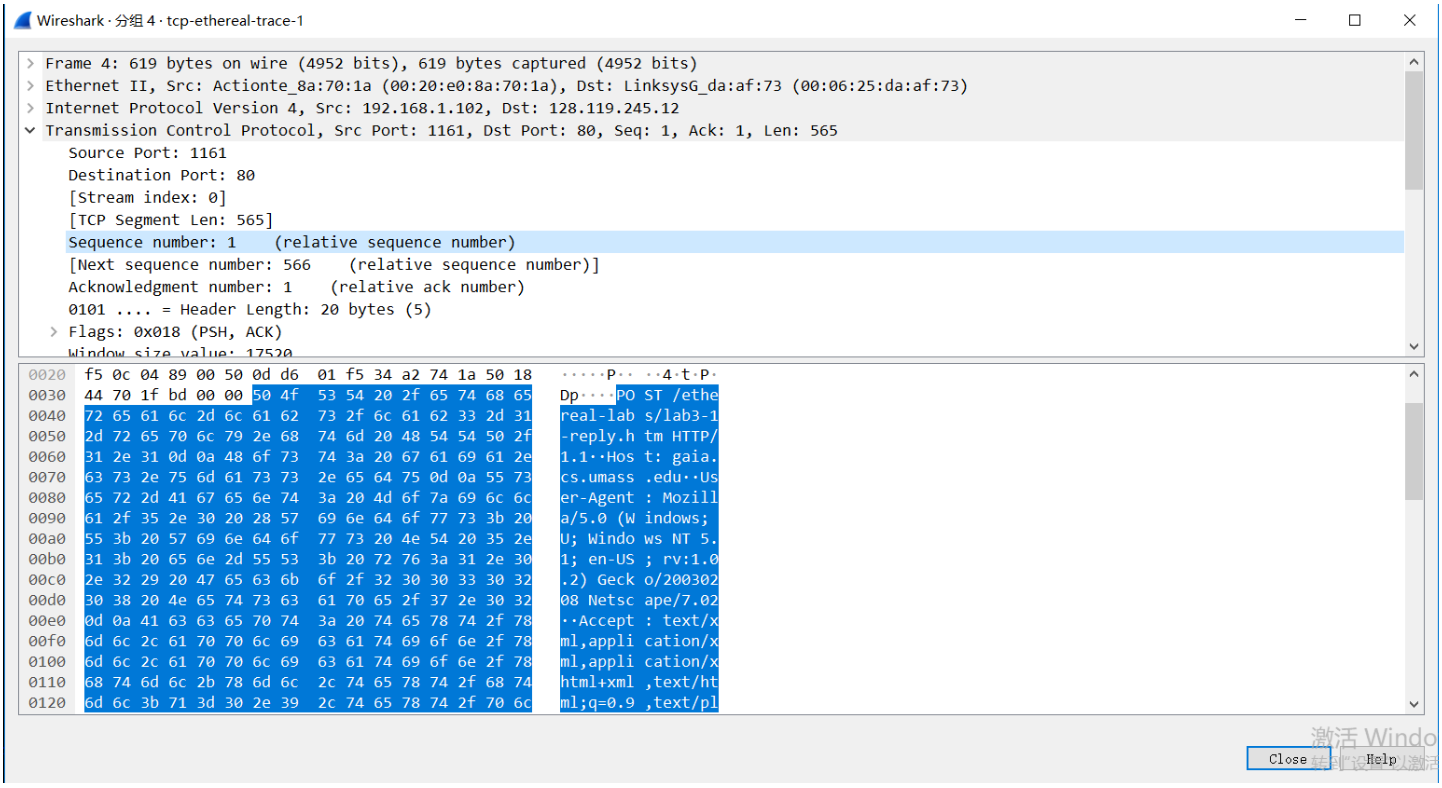
```
> Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)
> Ethernet II, Src: Actionte_8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG_da:af:73 (00:06:25:da:af:73)
> Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.119.245.12
v Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 1161
    Destination Port: 80
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    [Next sequence number: 0    (relative sequence number)]
    Acknowledgment number: 0
    0111 .... = Header Length: 28 bytes (7)
  > Flags: 0x002 (SYN)
    Window size value: 16384
    [Calculated window size: 16384]
    Checksum: 0xf6e9 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  v Options: (8 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted
    v TCP Option - Maximum segment size: 1460 bytes
        Kind: Maximum Segment Size (2)
        Length: 4
        MSS Value: 1460
    > TCP Option - No-Operation (NOP)
    > TCP Option - No-Operation (NOP)
    > TCP Option - SACK permitted
  v [Timestamps]
      [Time since first frame in this TCP stream: 0.000000000 seconds]
```

```
tcp-ethereal-trace-1        分组: 213 · 已显示: 202 (94.8%)        配置: Default
```

Question 2. 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 Ethereal window, looking for a segment with a "POST" within its DATA field.

Answer:

As the picture showed below, the sequence number of the TCP segment is 1.



```
> Frame 4: 619 bytes on wire (4952 bits), 619 bytes captured (4952 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, Dst: 128.119.245.12
v Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 1, Ack: 1, Len: 565
    Source Port: 1161
    Destination Port: 80
    [Stream index: 0]
    [TCP Segment Len: 565]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 566    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
    Window size value: 17520
```

```
0020  f5 0c 04 89 00 50 0d d6  01 f5 34 a2 74 1a 50 18   ·····P·· ··4·t·P·
0030  44 70 1f bd 00 00 50 4f  53 54 20 2f 65 74 68 65   Dp····PO ST /ethe
0040  72 65 61 6c 2d 6c 61 62  73 2f 6c 61 62 33 2d 31   real-lab s/lab3-1
0050  2d 72 65 70 6c 79 2e 68  74 6d 20 48 54 54 50 2f   -reply.h tm HTTP/
0060  31 2e 31 0d 0a 48 6f 73  74 3a 20 67 61 69 61 2e   1.1··Hos t: gaia.
0070  63 73 2e 75 6d 61 73 73  2e 65 64 75 0d 0a 55 73   cs.umass .edu··Us
0080  65 72 2d 41 67 65 6e 74  3a 20 4d 6f 7a 69 6c 6c   er-Agent : Mozill
0090  61 2f 35 2e 30 20 28 57  69 6e 64 6f 77 73 3b 20   a/5.0 (W indows; 
00a0  55 3b 20 57 69 6e 64 6f  77 73 20 4e 54 20 35 2e   U; Windo ws NT 5.
00b0  31 3b 20 65 6e 2d 55 53  3b 20 72 76 3a 31 2e 30   1; en-US ; rv:1.0
00c0  2e 32 29 20 47 65 63 6b  6f 2f 32 30 30 33 30 32   .2) Geck o/200302
00d0  30 38 20 4e 65 74 73 63  61 70 65 2f 37 2e 30 32   08 Netsc ape/7.02
00e0  0d 0a 41 63 63 65 70 74  3a 20 74 65 78 74 2f 78   ··Accept : text/x
00f0  6d 6c 2c 61 70 70 6c 69  63 61 74 69 6f 6e 2f 78   ml,appli cation/x
0100  6d 6c 2c 61 70 70 6c 69  63 61 74 69 6f 6e 2f 78   ml,appli cation/x
0110  68 74 6d 6c 2b 78 6d 6c  2c 74 65 78 74 2f 68 74   html+xml ,text/ht
0120  6d 6c 3b 71 3d 30 2e 39  2c 74 65 78 74 2f 70 6c   ml;q=0.9 ,text/pl
```

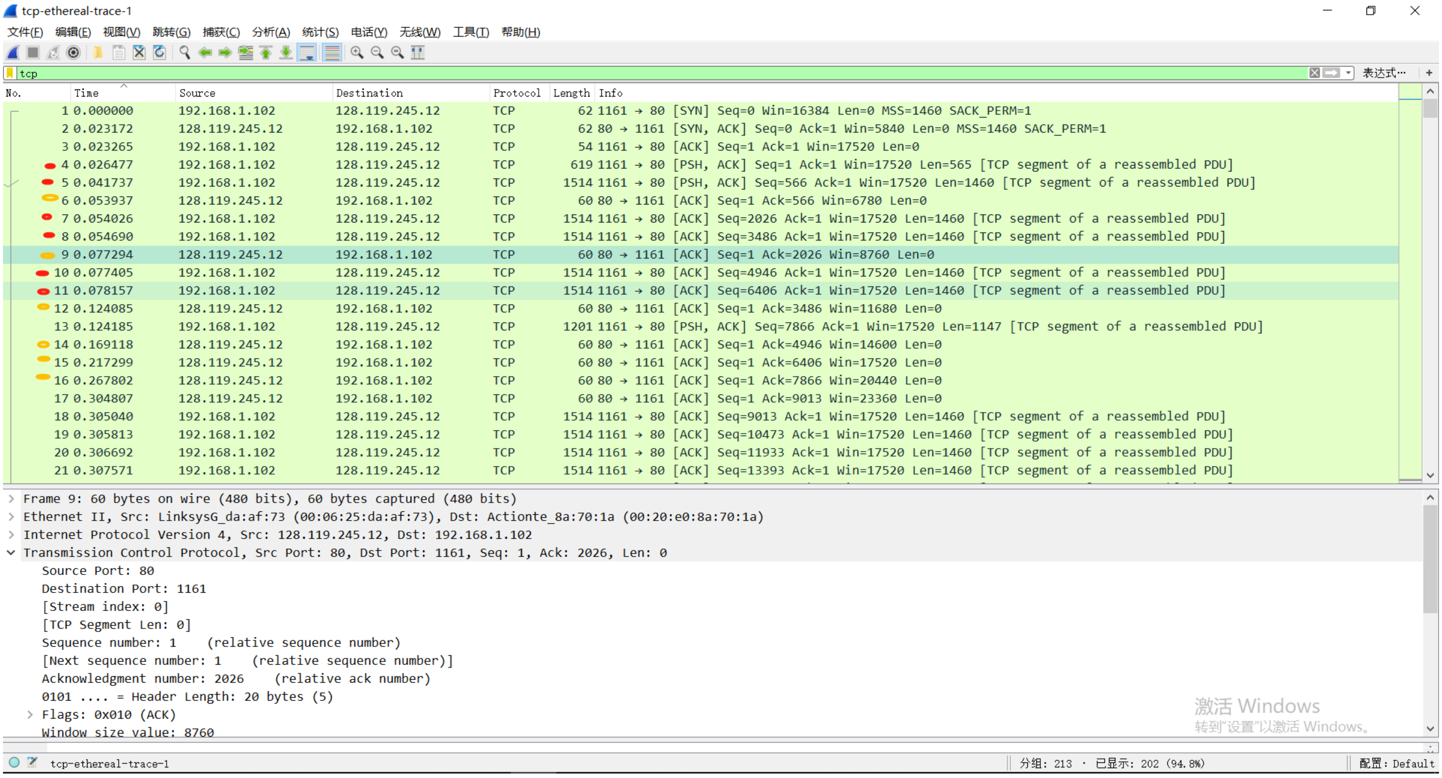
Question 3. Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection. What are the sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST) sent from the client to the web server (Do not consider the ACKs received from the server as part of these six segments)? 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 relevant parts of Section 3.5 or lecture slides) after the receipt of each ACK? Assume that the initial value of EstimatedRTT is equal to the measured RTT (SampleRTT) for the first segment, and then is computed using the EstimatedRTT equation for all subsequent segments. Set alpha to 0.125.

Answer:

As the graph show below, I make the table for indicating.

$$EstimatedRTT = 0.875 * EstimatedRTT + 0.125 * SampleRTT$$

sequence	sent time	ACK receive time	RTT	EstimatedRTT	Length
1	0.026477	0.053937	0.027460	0.027460	565
566	0.041737	0.077294	0.035557	0.028472	1460
2026	0.054026	0.124085	0.070059	0.033670	1460
3486	0.054690	0.169118	0.114428	0.043765	1460
4946	0.077405	0.217299	0.139894	0.055781	1460
6406	0.078157	0.267802	0.189645	0.072514	1460


tcp-ethereal-trace-1
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)
tcp
表达式…
激活 Windows
转到"设置"以激活 Windows。

```
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_PERM=1
   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=1460 SACK_PERM=1
   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 segment of a reassembled PDU]
   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 [TCP segment of a reassembled PDU]
   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 segment of a reassembled PDU]
   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 segment of a reassembled PDU]
   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 segment of a reassembled PDU]
  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 segment of a reassembled PDU]
  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 [TCP segment of a reassembled PDU]
  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=1460 [TCP segment of a reassembled PDU]
  19 0.305813  192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=10473 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
  20 0.306692  192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=11933 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
  21 0.307571  192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=13393 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
```

```
> Frame 9: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
> Ethernet II, Src: LinksysG_da:af:73 (00:06:25:da:af:73), Dst: Actionte_8a:70:1a (00:20:e0:8a:70:1a)
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.1.102
v Transmission Control Protocol, Src Port: 80, Dst Port: 1161, Seq: 1, Ack: 2026, Len: 0
    Source Port: 80
    Destination Port: 1161
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 1    (relative sequence number)]
    Acknowledgment number: 2026    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
    Window size value: 8760
```

```
tcp-ethereal-trace-1                    分组: 213 · 已显示: 202 (94.8%)        配置: Default
```

1552966513(1)

Question 4. What is the length of each of the first six TCP segments?

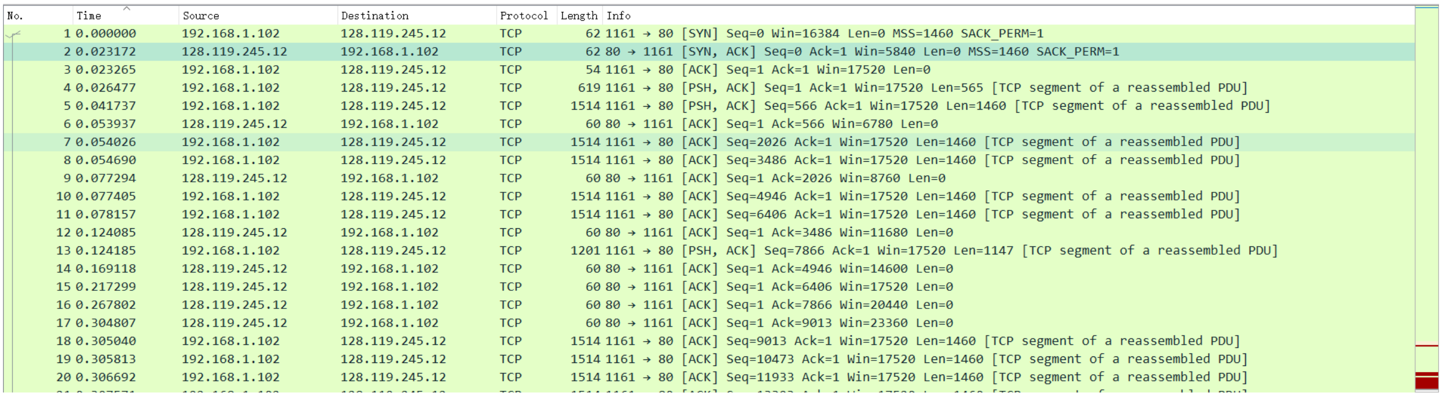
Answer:

As the table show in Quesion 3, we can know the length of each of first six TCP segments.

Question 5. What is the minimum amount of available buffer space advertised at the receiver for the entire trace? Does the lack of receiver buffer space ever throttle the sender?

Answer:

As the pictuer below show that, the minimun amount of available buffer space advertised at the receiver for the entire trace is 5840. It's not likely to throttle the sender because the buffer size always big than segment size.



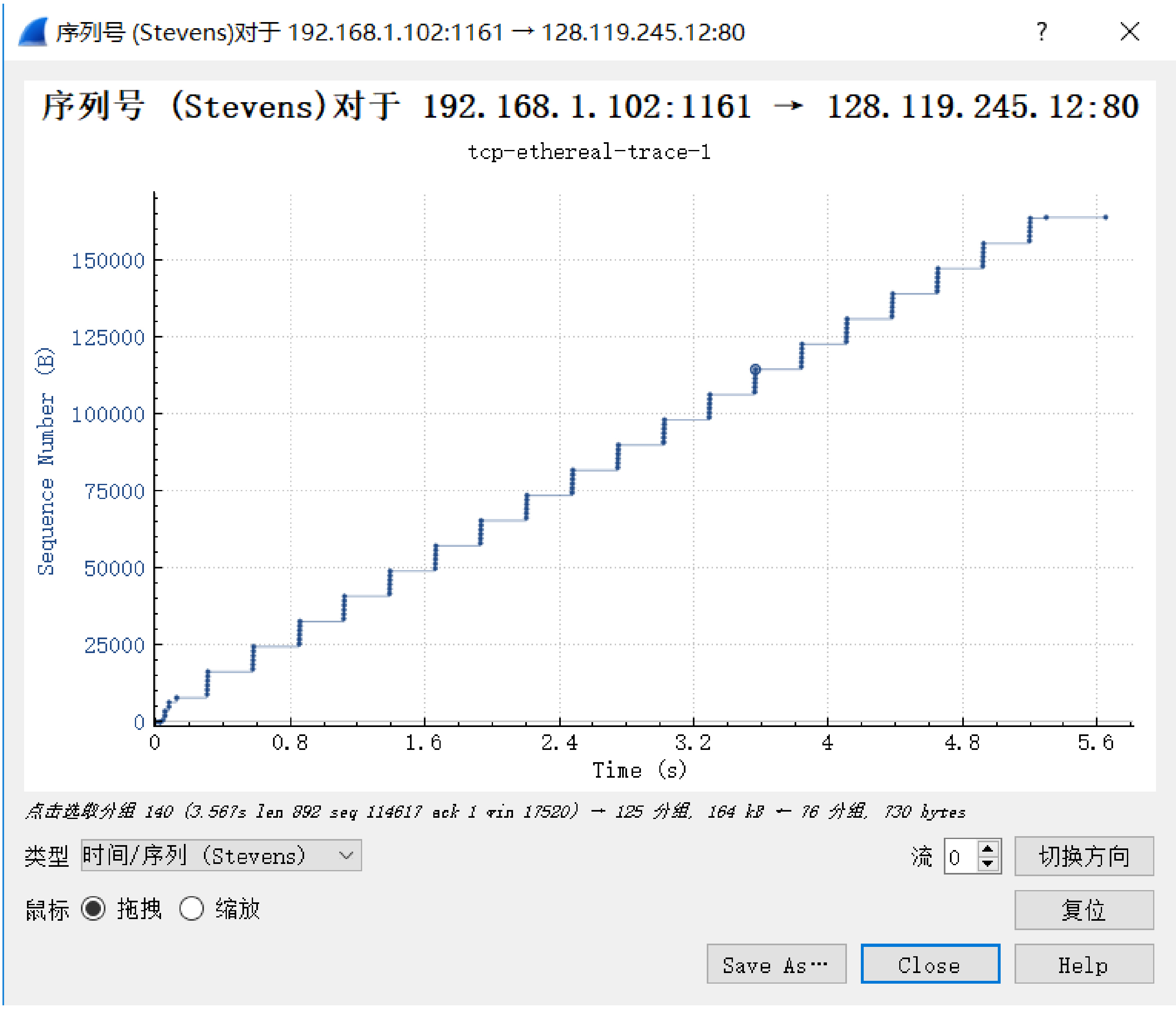
```
No.  Time      Source          Destination     Protocol Length Info
   1 0.000000  192.168.1.102   128.119.245.12  TCP        62 1161 → 80 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM=1
   2 0.023172  128.119.245.12  192.168.1.102   TCP        62 80 → 1161 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM=1
   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 segment of a reassembled PDU]
   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 [TCP segment of a reassembled PDU]
   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 segment of a reassembled PDU]
   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 segment of a reassembled PDU]
   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 segment of a reassembled PDU]
  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 segment of a reassembled PDU]
  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 [TCP segment of a reassembled PDU]
  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=1460 [TCP segment of a reassembled PDU]
  19 0.305813  192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=10473 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
  20 0.306692  192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=11933 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
```

Question 6. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?

Answer:

It doesn't have any retransmitted segments in the trace file because no two requests have same sequence from 192.168.1.102

to 128.119.245.12.



点击选择分组 140 (3.567s len 892 seq 114617 ack 1 win 17520) → 125 分组, 164 kB ← 76 分组, 730 bytes

类型 时间/序列 (Stevens)

流 0

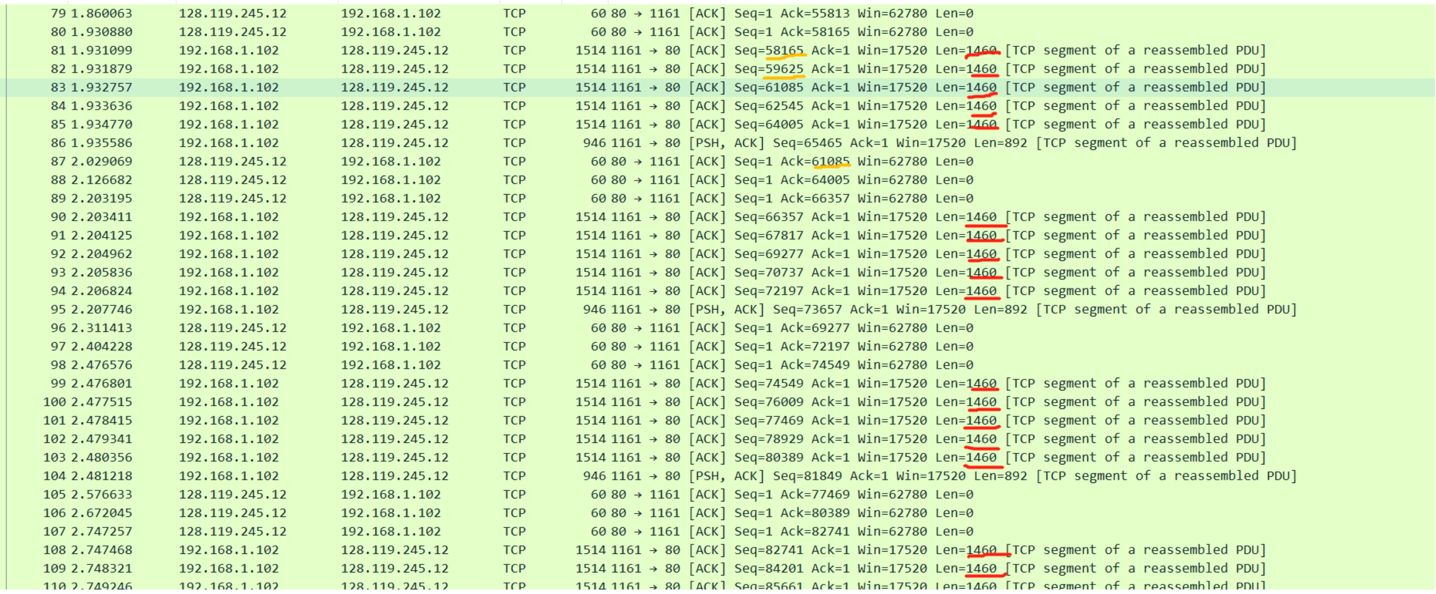
鼠标 ● 拖拽 ○ 缩放

Question 7. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (recall the discussion about delayed acks from the lecture notes or Section 3.5 of the text).

Answer:

As the picture showed below, 1460 bytes of data is sent in an ACK. And among that graph, there is a case that No.87 acking

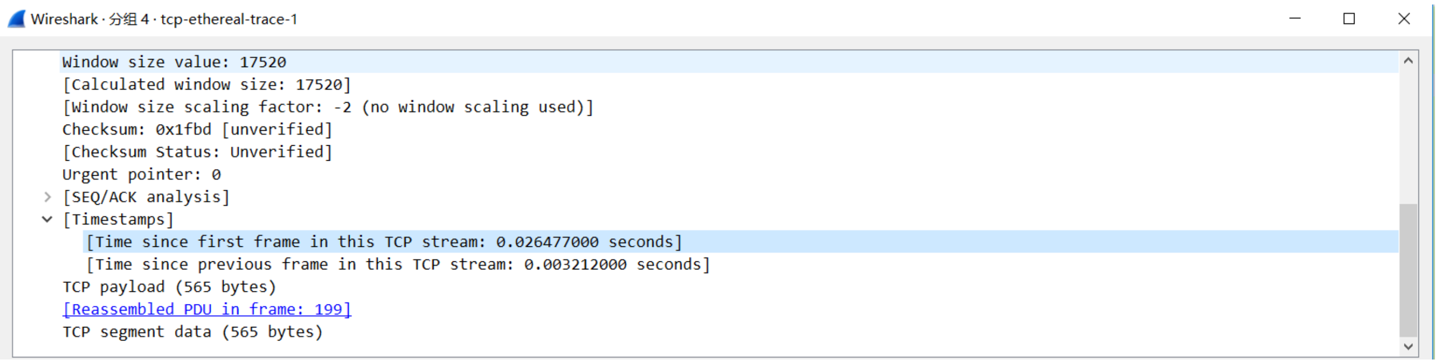
No.81 and No.82 received, and there are more cases in trace file.



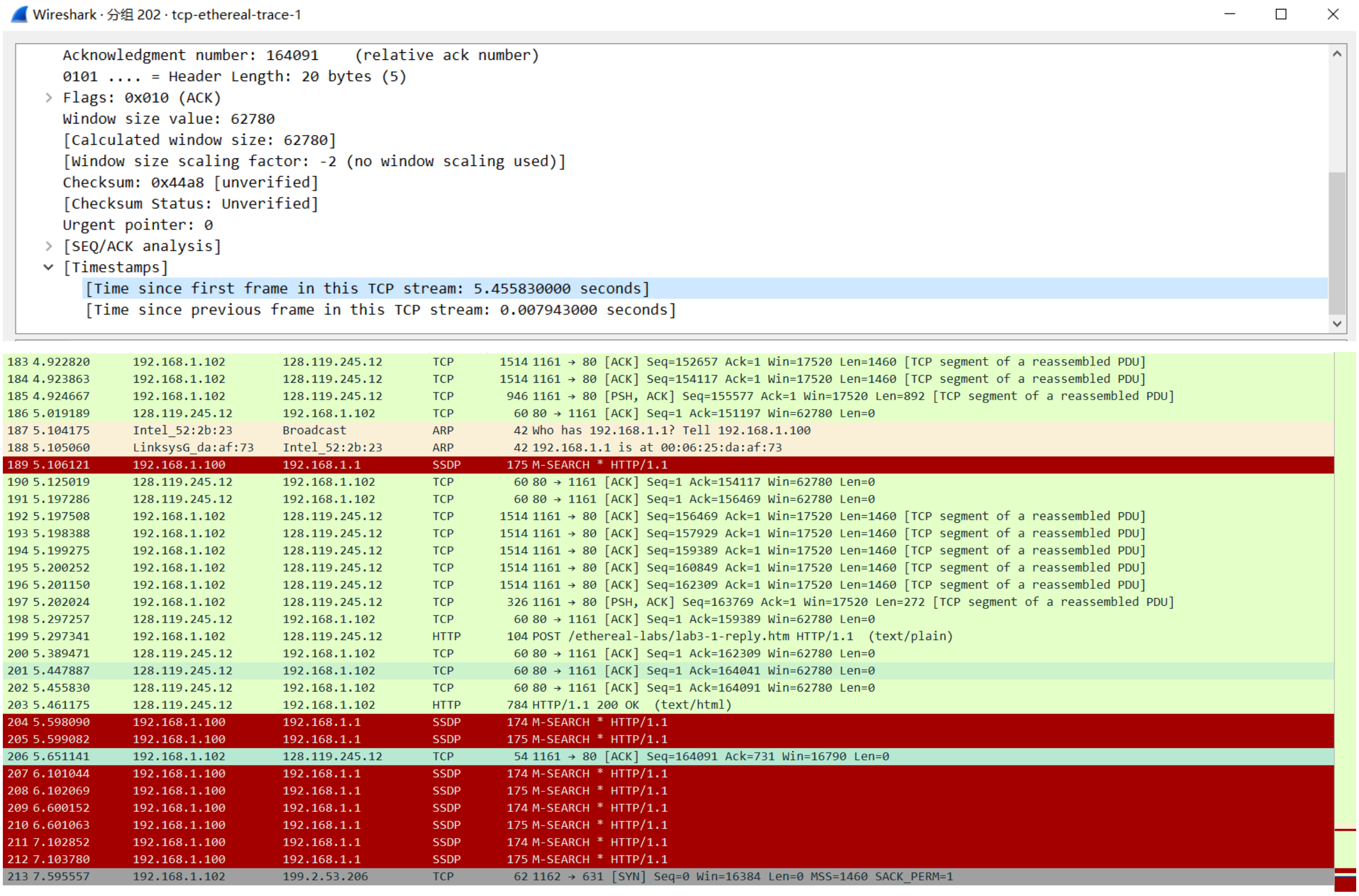
```
 79 1.860063    128.119.245.12    192.168.1.102     TCP      60 80 → 1161 [ACK] Seq=1 Ack=55813 Win=62780 Len=0
 80 1.930880    128.119.245.12    192.168.1.102     TCP      60 80 → 1161 [ACK] Seq=1 Ack=58165 Win=62780 Len=0
 81 1.931099    192.168.1.102     128.119.245.12    TCP    1514 1161 → 80 [ACK] Seq=58165 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
 82 1.931879    192.168.1.102     128.119.245.12    TCP    1514 1161 → 80 [ACK] Seq=59625 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
 83 1.932757    192.168.1.102     128.119.245.12    TCP    1514 1161 → 80 [ACK] Seq=61085 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
 84 1.933636    192.168.1.102     128.119.245.12    TCP    1514 1161 → 80 [ACK] Seq=62545 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
 85 1.934770    192.168.1.102     128.119.245.12    TCP    1514 1161 → 80 [ACK] Seq=64005 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
 86 1.935586    192.168.1.102     128.119.245.12    TCP     946 1161 → 80 [PSH, ACK] Seq=65465 Ack=1 Win=17520 Len=892 [TCP segment of a reassembled PDU]
 87 2.029069    128.119.245.12    192.168.1.102     TCP      60 80 → 1161 [ACK] Seq=1 Ack=61085 Win=62780 Len=0
 88 2.126682    128.119.245.12    192.168.1.102     TCP      60 80 → 1161 [ACK] Seq=1 Ack=64005 Win=62780 Len=0
 89 2.203195    128.119.245.12    192.168.1.102     TCP      60 80 → 1161 [ACK] Seq=1 Ack=66357 Win=62780 Len=0
 90 2.203411    192.168.1.102     128.119.245.12    TCP    1514 1161 → 80 [ACK] Seq=66357 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
 91 2.204125    192.168.1.102     128.119.245.12    TCP    1514 1161 → 80 [ACK] Seq=67817 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
 92 2.204962    192.168.1.102     128.119.245.12    TCP    1514 1161 → 80 [ACK] Seq=69277 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
 93 2.205836    192.168.1.102     128.119.245.12    TCP    1514 1161 → 80 [ACK] Seq=70737 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
 94 2.206824    192.168.1.102     128.119.245.12    TCP    1514 1161 → 80 [ACK] Seq=72197 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
 95 2.207746    192.168.1.102     128.119.245.12    TCP     946 1161 → 80 [PSH, ACK] Seq=73657 Ack=1 Win=17520 Len=892 [TCP segment of a reassembled PDU]
 96 2.311413    128.119.245.12    192.168.1.102     TCP      60 80 → 1161 [ACK] Seq=1 Ack=69277 Win=62780 Len=0
 97 2.404228    128.119.245.12    192.168.1.102     TCP      60 80 → 1161 [ACK] Seq=1 Ack=72197 Win=62780 Len=0
 98 2.476576    128.119.245.12    192.168.1.102     TCP      60 80 → 1161 [ACK] Seq=1 Ack=74549 Win=62780 Len=0
 99 2.476801    192.168.1.102     128.119.245.12    TCP    1514 1161 → 80 [ACK] Seq=74549 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
100 2.477515    192.168.1.102     128.119.245.12    TCP    1514 1161 → 80 [ACK] Seq=76009 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
101 2.478415    192.168.1.102     128.119.245.12    TCP    1514 1161 → 80 [ACK] Seq=77469 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
102 2.479341    192.168.1.102     128.119.245.12    TCP    1514 1161 → 80 [ACK] Seq=78929 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
103 2.480356    192.168.1.102     128.119.245.12    TCP    1514 1161 → 80 [ACK] Seq=80389 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
104 2.481218    192.168.1.102     128.119.245.12    TCP     946 1161 → 80 [PSH, ACK] Seq=81849 Ack=1 Win=17520 Len=892 [TCP segment of a reassembled PDU]
105 2.576633    128.119.245.12    192.168.1.102     TCP      60 80 → 1161 [ACK] Seq=1 Ack=77469 Win=62780 Len=0
106 2.672045    128.119.245.12    192.168.1.102     TCP      60 80 → 1161 [ACK] Seq=1 Ack=80389 Win=62780 Len=0
107 2.747257    128.119.245.12    192.168.1.102     TCP      60 80 → 1161 [ACK] Seq=1 Ack=82741 Win=62780 Len=0
108 2.747468    192.168.1.102     128.119.245.12    TCP    1514 1161 → 80 [ACK] Seq=82741 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
109 2.748321    192.168.1.102     128.119.245.12    TCP    1514 1161 → 80 [ACK] Seq=84201 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
110 2.749246    192.168.1.102     128.119.245.12    TCP    1514 1161 → 80 [ACK] Seq=85661 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
```

Question 8. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

Answer:



```
    Window size value: 17520
    [Calculated window size: 17520]
    [Window size scaling factor: -2 (no window scaling used)]
    Checksum: 0x1fbd [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > [SEQ/ACK analysis]
  ⌄ [Timestamps]
      [Time since first frame in this TCP stream: 0.026477000 seconds]
      [Time since previous frame in this TCP stream: 0.003212000 seconds]
    TCP payload (565 bytes)
    [Reassembled PDU in frame: 199]
    TCP segment data (565 bytes)
```



```
    Acknowledgment number: 164091    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
    Window size value: 62780
    [Calculated window size: 62780]
    [Window size scaling factor: -2 (no window scaling used)]
    Checksum: 0x44a8 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > [SEQ/ACK analysis]
  v [Timestamps]
      [Time since first frame in this TCP stream: 5.455830000 seconds]
      [Time since previous frame in this TCP stream: 0.007943000 seconds]
```

```
183 4.922820   192.168.1.102     128.119.245.12   TCP    1514 1161 → 80 [ACK] Seq=152657 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
184 4.923863   192.168.1.102     128.119.245.12   TCP    1514 1161 → 80 [ACK] Seq=154117 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
185 4.924667   192.168.1.102     128.119.245.12   TCP     946 1161 → 80 [PSH, ACK] Seq=155577 Ack=1 Win=17520 Len=892 [TCP segment of a reassembled PDU]
186 5.019189   128.119.245.12    192.168.1.102    TCP      60 80 → 1161 [ACK] Seq=1 Ack=151197 Win=62780 Len=0
187 5.104175   Intel_52:2b:23    Broadcast        ARP      42 Who has 192.168.1.1? Tell 192.168.1.100
188 5.105060   LinksysG_da:af:73 Intel_52:2b:23   ARP      42 192.168.1.1 is at 00:06:25:da:af:73
189 5.106121   192.168.1.100     192.168.1.1      SSDP    175 M-SEARCH * HTTP/1.1
190 5.125019   128.119.245.12    192.168.1.102    TCP      60 80 → 1161 [ACK] Seq=1 Ack=154117 Win=62780 Len=0
191 5.197286   128.119.245.12    192.168.1.102    TCP      60 80 → 1161 [ACK] Seq=1 Ack=156469 Win=62780 Len=0
192 5.197508   192.168.1.102     128.119.245.12   TCP    1514 1161 → 80 [ACK] Seq=156469 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
193 5.198388   192.168.1.102     128.119.245.12   TCP    1514 1161 → 80 [ACK] Seq=157929 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
194 5.199275   192.168.1.102     128.119.245.12   TCP    1514 1161 → 80 [ACK] Seq=159389 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
195 5.200252   192.168.1.102     128.119.245.12   TCP    1514 1161 → 80 [ACK] Seq=160849 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
196 5.201150   192.168.1.102     128.119.245.12   TCP    1514 1161 → 80 [ACK] Seq=162309 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
197 5.202024   192.168.1.102     128.119.245.12   TCP     326 1161 → 80 [PSH, ACK] Seq=163769 Ack=1 Win=17520 Len=272 [TCP segment of a reassembled PDU]
198 5.297257   128.119.245.12    192.168.1.102    TCP      60 80 → 1161 [ACK] Seq=1 Ack=159389 Win=62780 Len=0
199 5.297341   192.168.1.102     128.119.245.12   HTTP    104 POST /ethereal-labs/lab3-1-reply.htm HTTP/1.1  (text/plain)
200 5.389471   128.119.245.12    192.168.1.102    TCP      60 80 → 1161 [ACK] Seq=1 Ack=162309 Win=62780 Len=0
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)
204 5.598090   192.168.1.100     192.168.1.1      SSDP    174 M-SEARCH * HTTP/1.1
205 5.599082   192.168.1.100     192.168.1.1      SSDP    175 M-SEARCH * HTTP/1.1
206 5.651141   192.168.1.102     128.119.245.12   TCP      54 1161 → 80 [ACK] Seq=164091 Ack=731 Win=16790 Len=0
207 6.101044   192.168.1.100     192.168.1.1      SSDP    174 M-SEARCH * HTTP/1.1
208 6.102069   192.168.1.100     192.168.1.1      SSDP    175 M-SEARCH * HTTP/1.1
209 6.600152   192.168.1.100     192.168.1.1      SSDP    174 M-SEARCH * HTTP/1.1
210 6.601063   192.168.1.100     192.168.1.1      SSDP    175 M-SEARCH * HTTP/1.1
211 7.102852   192.168.1.100     192.168.1.1      SSDP    174 M-SEARCH * HTTP/1.1
212 7.103780   192.168.1.100     192.168.1.1      SSDP    175 M-SEARCH * HTTP/1.1
213 7.595557   192.168.1.102     199.2.53.206     TCP      62 1162 → 631 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM=1
```

$$Throughput = \frac{Number\ of\ bytes\ recevied}{Cost\ time}$$

From the group below, the sequence number of last TCP response message is 164091. But TCP is cumulative acknowledge, so we know 164090 bytes has received.

Then cost time = 5.45583-0.026477 = 5.429353s,

So,

$$Throughout = \frac{164090}{5.429353} = 30222.75 bytes/s$$

Exercise 2: TCP Connection Management

Question 1. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and server?

Answer:

As the picture showed in the question, the sequence number is 2818463618.

Question 2. What is the sequence number of the SYNACK segment sent by the server to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did the server determine that value?

Answer:

The sequence number of the SYNACK segment is 1247095790.

The value of the Acknowledgement field in the SYNACK segment is 2818463619. The value is get from the initial serial number sent by the client + 1.

Question 3. What is the sequence number of the ACK segment sent by the client computer in response to the SYNACK? What is the value of the Acknowledgment field in this ACK segment? Does this segment contain any data?

Answer:

The sequence number of the ACK segment sent by the client computer in response to the SYNACK is 2818463619.

The value of the Acknowledgment field in this ACK segment is 1247095791.

This segment contain 2818463652-2818463619=33bytes data, because in No.301 ACK field is 2818463652.

Question 4. Who has done the active close? client or the server? how you have determined this? What type of closure has been performed? 3 Segment (FIN/FINACK/ACK), 4 Segment (FIN/ACK/FIN/ACK) or Simultaneous close?

Answer:

Both server and client do the active close at the same time. From the given graph, we find that they both generate the FIN segment before receiving the FIN segment from the other. We can determine it by No.304 and No.305. Therefore, the type is Simltaneous close.

Question 5. How many data bytes have been transferred from the client to the server and from the server to the client during the whole duration of the connection? What relationship does this have with the Initial Sequence Number and the final ACK received from the other side?

Answer: The number of data transferred between server and client can determinded by the last sequence number and Initial sequence number of two sides. So the data transferred from client to server is 2818463652-2818463619=33bytes, and the data transferred from server to client is 124709831 - 1247095791 = 40 bytes. According to TCP connecting and disconecting process, the relationship is (final ACK received from the other side - 1) - (Initial sequence number + 1) = number of transferred data.