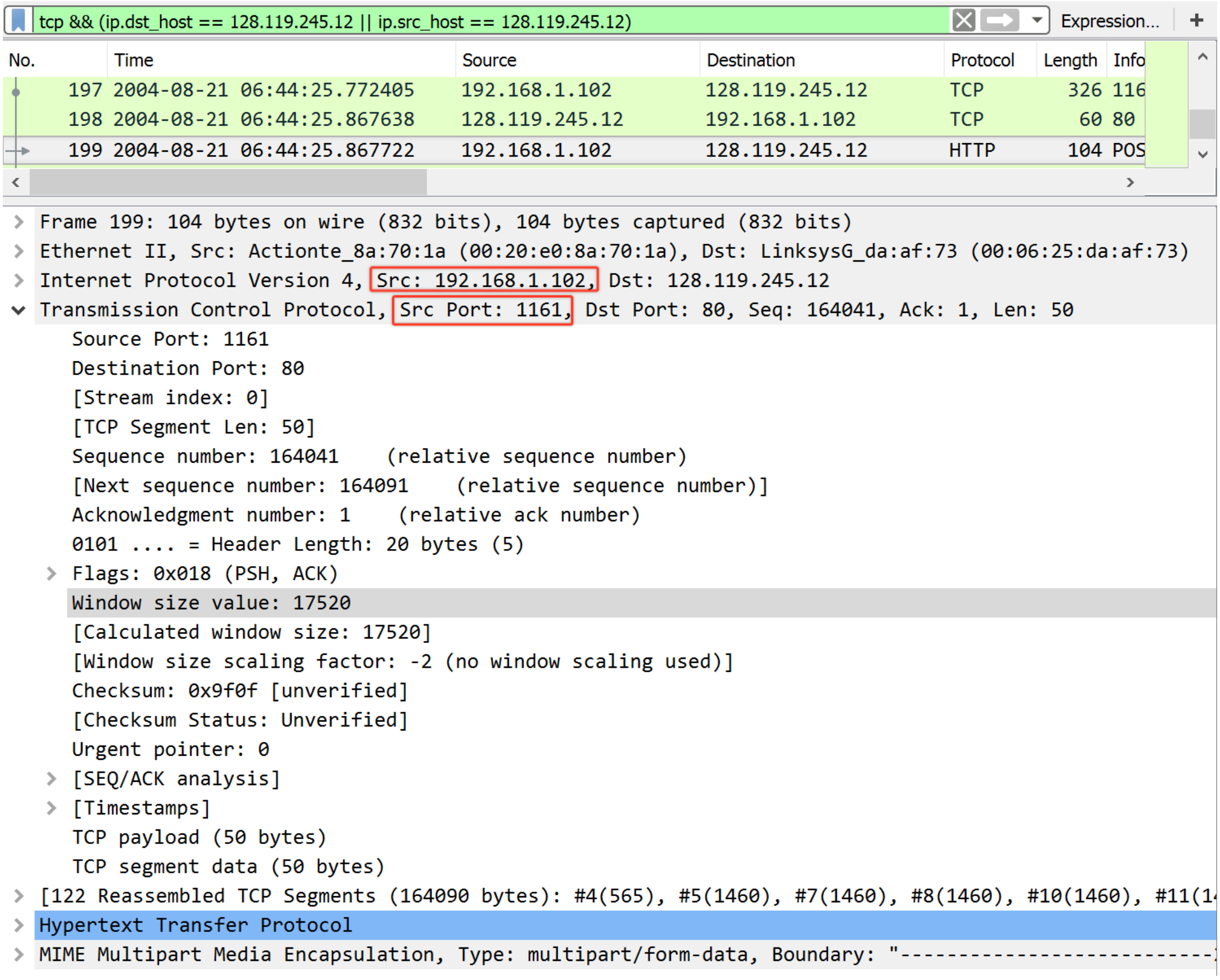
Joel Benjamin Castillo (jc5383)<br>
CS6843 - Computer Networking<br>
Prof. Rafail Portnoy

## Wireshark Lab - TCP

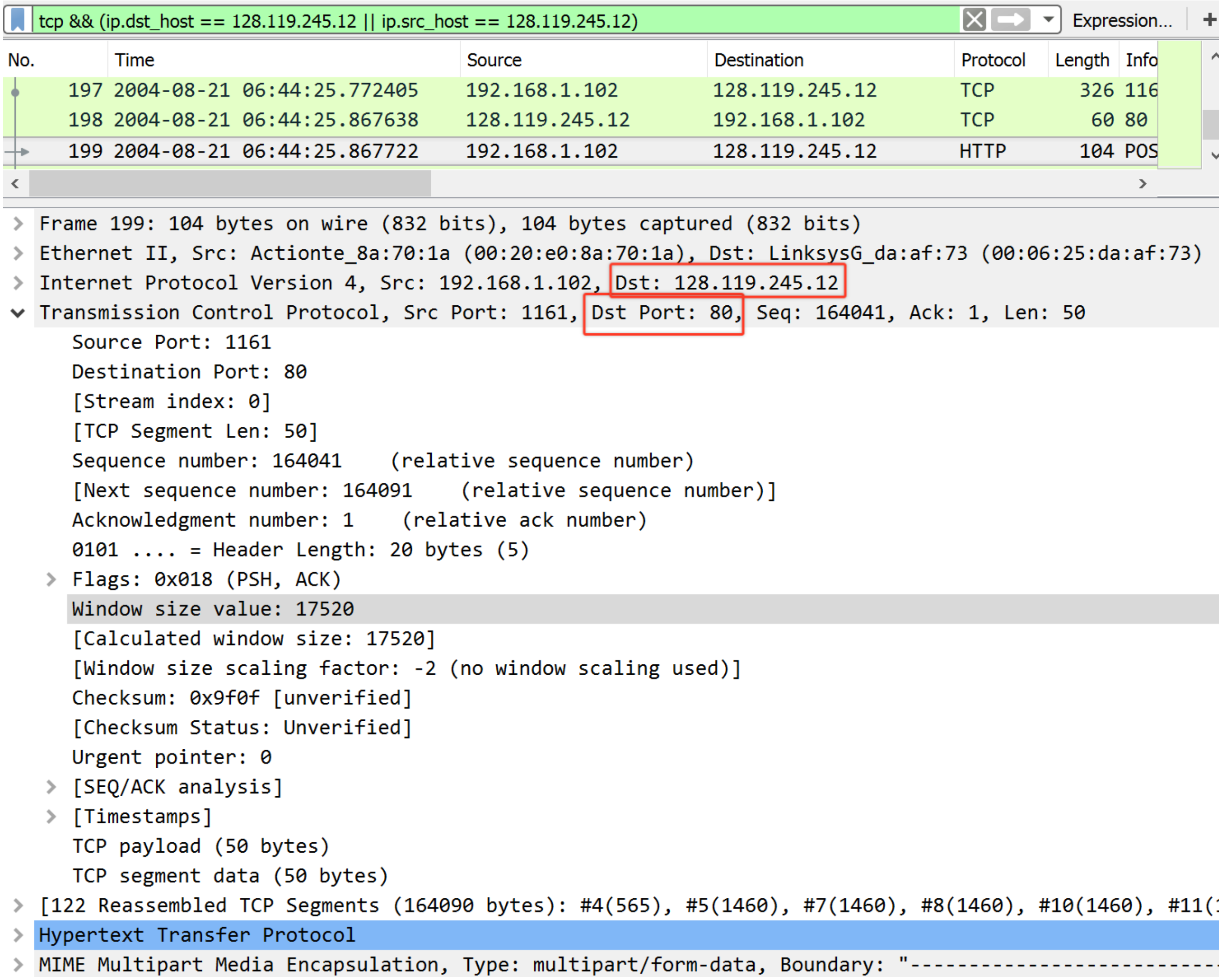
1. What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu? To answer this question, it's probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the "details 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.
   Source IP: `192.168.1.102`
   Source Port: `1161`



```
tcp && (ip.dst_host == 128.119.245.12 || ip.src_host == 128.119.245.12)

No.     Time                        Source          Destination     Protocol  Length  Info
    197 2004-08-21 06:44:25.772405  192.168.1.102   128.119.245.12  TCP          326  116
    198 2004-08-21 06:44:25.867638  128.119.245.12  192.168.1.102   TCP           60  80
    199 2004-08-21 06:44:25.867722  192.168.1.102   128.119.245.12  HTTP         104  POS

> Frame 199: 104 bytes on wire (832 bits), 104 bytes captured (832 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: 164041, Ack: 1, Len: 50
     Source Port: 1161
     Destination Port: 80
     [Stream index: 0]
     [TCP Segment Len: 50]
     Sequence number: 164041    (relative sequence number)
     [Next sequence number: 164091    (relative sequence number)]
     Acknowledgment number: 1    (relative ack number)
     0101 .... = Header Length: 20 bytes (5)
   > Flags: 0x018 (PSH, ACK)
     Window size value: 17520
     [Calculated window size: 17520]
     [Window size scaling factor: -2 (no window scaling used)]
     Checksum: 0x9f0f [unverified]
     [Checksum Status: Unverified]
     Urgent pointer: 0
   > [SEQ/ACK analysis]
   > [Timestamps]
     TCP payload (50 bytes)
     TCP segment data (50 bytes)
> [122 Reassembled TCP Segments (164090 bytes): #4(565), #5(1460), #7(1460), #8(1460), #10(1460), #11(1
> Hypertext Transfer Protocol
> MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "-------------------------
```

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?
   Destination IP: `128.119.245.12`

Destination Port: `80`

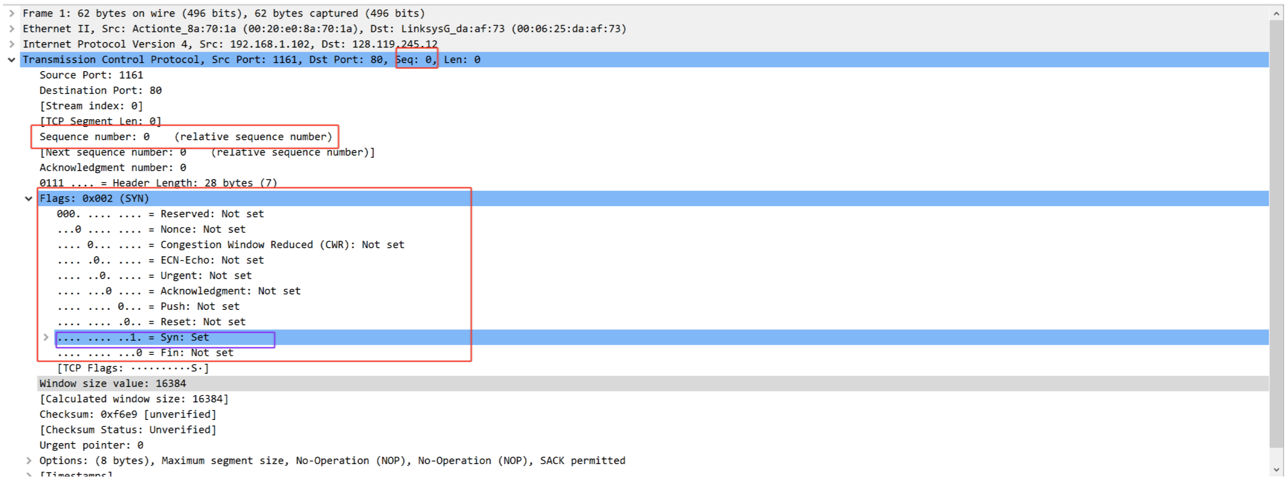


```
tcp && (ip.dst_host == 128.119.245.12 || ip.src_host == 128.119.245.12)

No.   Time                        Source          Destination     Protocol  Length  Info
197   2004-08-21 06:44:25.772405  192.168.1.102   128.119.245.12  TCP       326     116
198   2004-08-21 06:44:25.867638  128.119.245.12  192.168.1.102   TCP       60      80
199   2004-08-21 06:44:25.867722  192.168.1.102   128.119.245.12  HTTP      104     POS

> Frame 199: 104 bytes on wire (832 bits), 104 bytes captured (832 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: 164041, Ack: 1, Len: 50
     Source Port: 1161
     Destination Port: 80
     [Stream index: 0]
     [TCP Segment Len: 50]
     Sequence number: 164041    (relative sequence number)
     [Next sequence number: 164091    (relative sequence number)]
     Acknowledgment number: 1    (relative ack number)
     0101 .... = Header Length: 20 bytes (5)
   > Flags: 0x018 (PSH, ACK)
     Window size value: 17520
     [Calculated window size: 17520]
     [Window size scaling factor: -2 (no window scaling used)]
     Checksum: 0x9f0f [unverified]
     [Checksum Status: Unverified]
     Urgent pointer: 0
   > [SEQ/ACK analysis]
   > [Timestamps]
     TCP payload (50 bytes)
     TCP segment data (50 bytes)
> [122 Reassembled TCP Segments (164090 bytes): #4(565), #5(1460), #7(1460), #8(1460), #10(1460), #11(:
> Hypertext Transfer Protocol
> MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "--------------------------
```

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? I couldnt get this to work on my VM.
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?

   Sequence Number: `0`

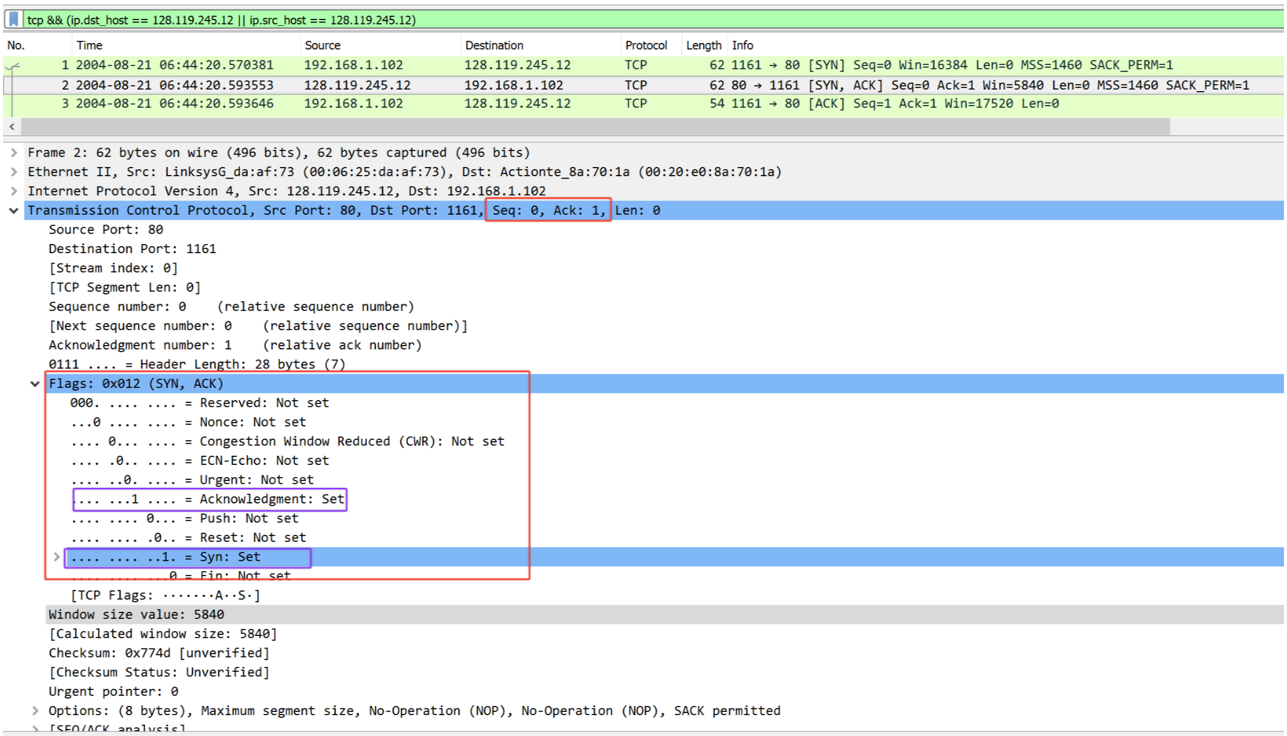
   The `SYN` flag is the only flag set in the TCP packet, identifying it as a `SYN` packet.



```
> 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)
   v 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 size value: 16384
     [Calculated window size: 16384]
     Checksum: 0xf6e9 [unverified]
     [Checksum Status: Unverified]
     Urgent pointer: 0
   > Options: (8 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted
   > [Timestamps]
```

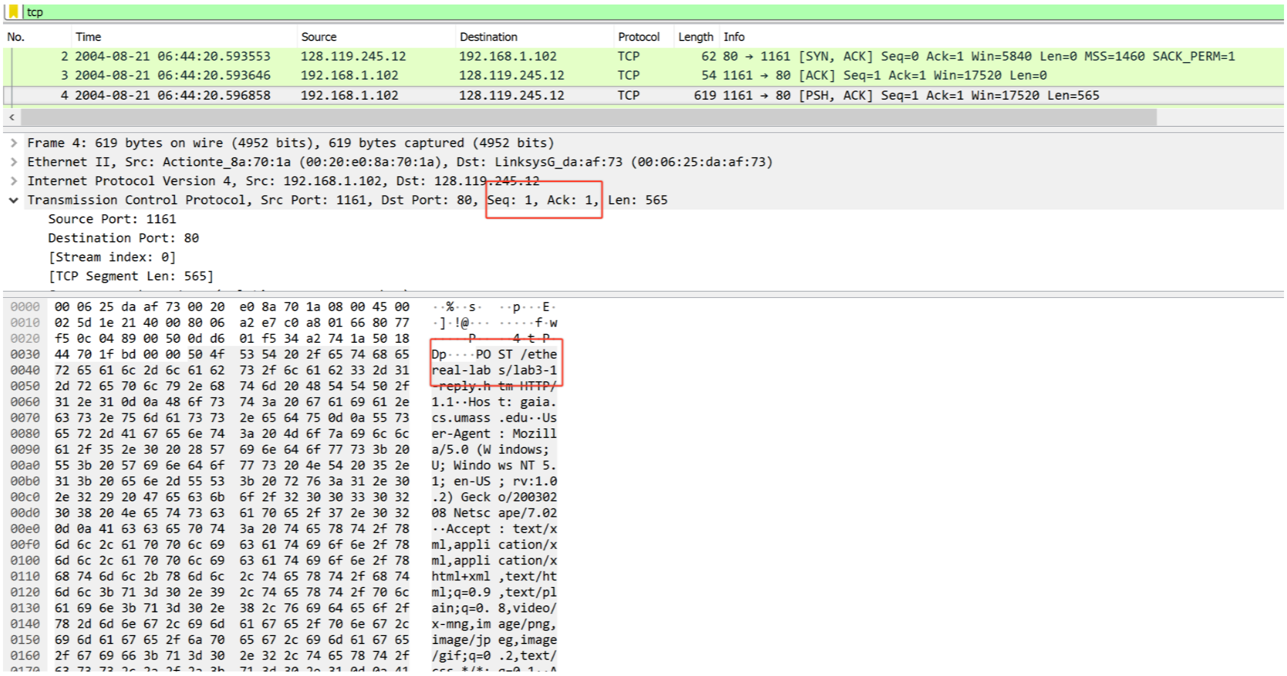
5. What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the Acknowledgementfield in the SYNACK segment? How did gaia.cs.umass.edu determine that value? What is it in the segment that identifies the segment as a SYNACK segment? Sequence Number: `0`

The value of the ACK field in the SYNACK segment is 1. This is determined by adding 1 to the original sequence number of the SYN segment on the server. Both the SYN and ACK flags are set to 1, marking the segment as a SYNACK segment.



```
tcp && (ip.dst_host == 128.119.245.12 || ip.src_host == 128.119.245.12)

No.  Time                        Source          Destination     Protocol  Length  Info
   1 2004-08-21 06:44:20.570381  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 2004-08-21 06:44:20.593553  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 2004-08-21 06:44:20.593646  192.168.1.102   128.119.245.12  TCP          54 1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0

> Frame 2: 62 bytes on wire (496 bits), 62 bytes captured (496 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: 0, Ack: 1, Len: 0
    Source Port: 80
    Destination Port: 1161
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    [Next sequence number: 0    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    0111 .... = Header Length: 28 bytes (7)
  v 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
      [TCP Flags: ·······A··S·]
    Window size value: 5840
    [Calculated window size: 5840]
    Checksum: 0x774d [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > Options: (8 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted
  > [SEQ/ACK analysis]
```

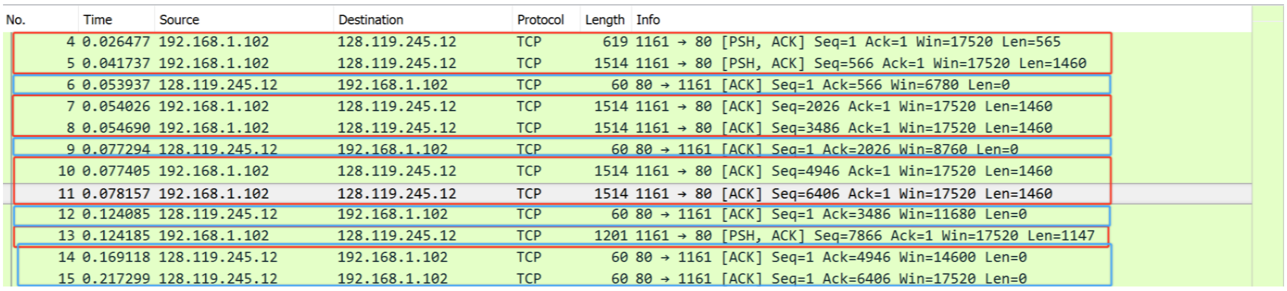
6. What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you'll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a "POST" within its DATA field.
   Sequence Number: 1



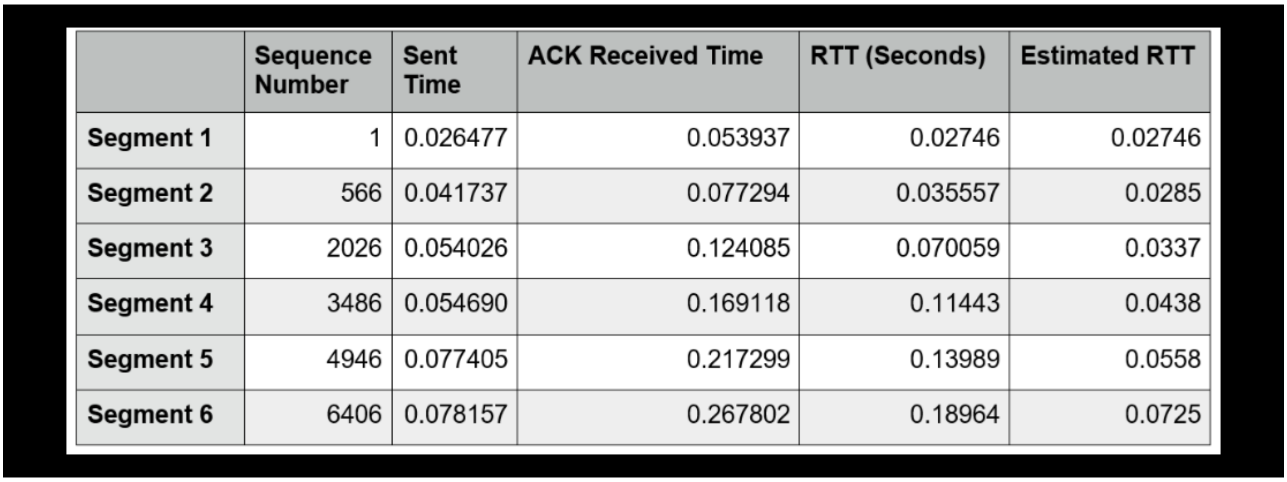
```
tcp

No.  Time                        Source          Destination     Protocol  Length  Info
   2 2004-08-21 06:44:20.593553  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 2004-08-21 06:44:20.593646  192.168.1.102   128.119.245.12  TCP          54 1161 → 80 [ACK] Seq=1 Ack=1 Win=17520 Len=0
   4 2004-08-21 06:44:20.596858  192.168.1.102   128.119.245.12  TCP         619 1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565

> 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]

0000  00 06 25 da af 73 00 20  e0 8a 70 1a 08 00 45 00   ··%··s·  ··p···E·
0010  02 5d 1e 21 40 00 80 06  a2 e7 c0 a8 01 66 80 77   ·]·!@···  ·····f·w
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
0130  61 69 6e 3b 71 3d 30 2e  38 2c 76 69 64 65 6f 2f   ain;q=0. 8,video/
0140  78 2d 6d 6e 67 2c 69 6d  61 67 65 2f 70 6e 67 2c   x-mng,im age/png,
0150  69 6d 61 67 65 2f 6a 70  65 67 2c 69 6d 61 67 65   image/jp eg,image
0160  2f 67 69 66 3b 71 3d 30  2e 32 2c 74 65 78 74 2f   /gif;q=0 .2,text/
```

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 EstimatedRTTvalue (see Section 3.5.3, page 242 in text) after the receipt of each ACK? Assume that the value of the EstimatedRTTis equal to the measured RTT for the first segment, and then is computed

using the EstimatedRTTequation on page 242 for all subsequent segments.Note:Wireshark has a nice feature that allows you to plot the RTT for each of the TCP segments sent. Select a TCP segment in the “listing of captured packets” window that is being sent from the client to the gaia.cs.umass.edu server. Then select: Statistics->TCP Stream Graph->Round Trip Time Graph.



```
No.   Time     Source          Destination     Protocol Length Info
    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=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
    8 0.054690 192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460
    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
   11 0.078157 192.168.1.102   128.119.245.12  TCP      1514 1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460
   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
```



| | Sequence Number | Sent Time | ACK Received Time | RTT (Seconds) | Estimated RTT |
|---|---|---|---|---|---|
| **Segment 1** | 1 | 0.026477 | 0.053937 | 0.02746 | 0.02746 |
| **Segment 2** | 566 | 0.041737 | 0.077294 | 0.035557 | 0.0285 |
| **Segment 3** | 2026 | 0.054026 | 0.124085 | 0.070059 | 0.0337 |
| **Segment 4** | 3486 | 0.054690 | 0.169118 | 0.11443 | 0.0438 |
| **Segment 5** | 4946 | 0.077405 | 0.217299 | 0.13989 | 0.0558 |
| **Segment 6** | 6406 | 0.078157 | 0.267802 | 0.18964 | 0.0725 |

```
EstimatedRTT = .875 * EstimatedRTT + .125 * SampleRTT
Segment 1: RTT for Segment 1 = .02746
Segment 2: .875 * .02746 + .125 * .035557 = .0285s
Segment 3: .875 * .0285 + .125 * .070059 = .0337s
Segment 4: .875 * .0377 + .125 * .11443 = .0438s
Segment 5: .875 * .0438 + .125 * .13989 = .0558s
Segment 6: .875 * .0558 + .125 * .19064 = .0725s
```

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

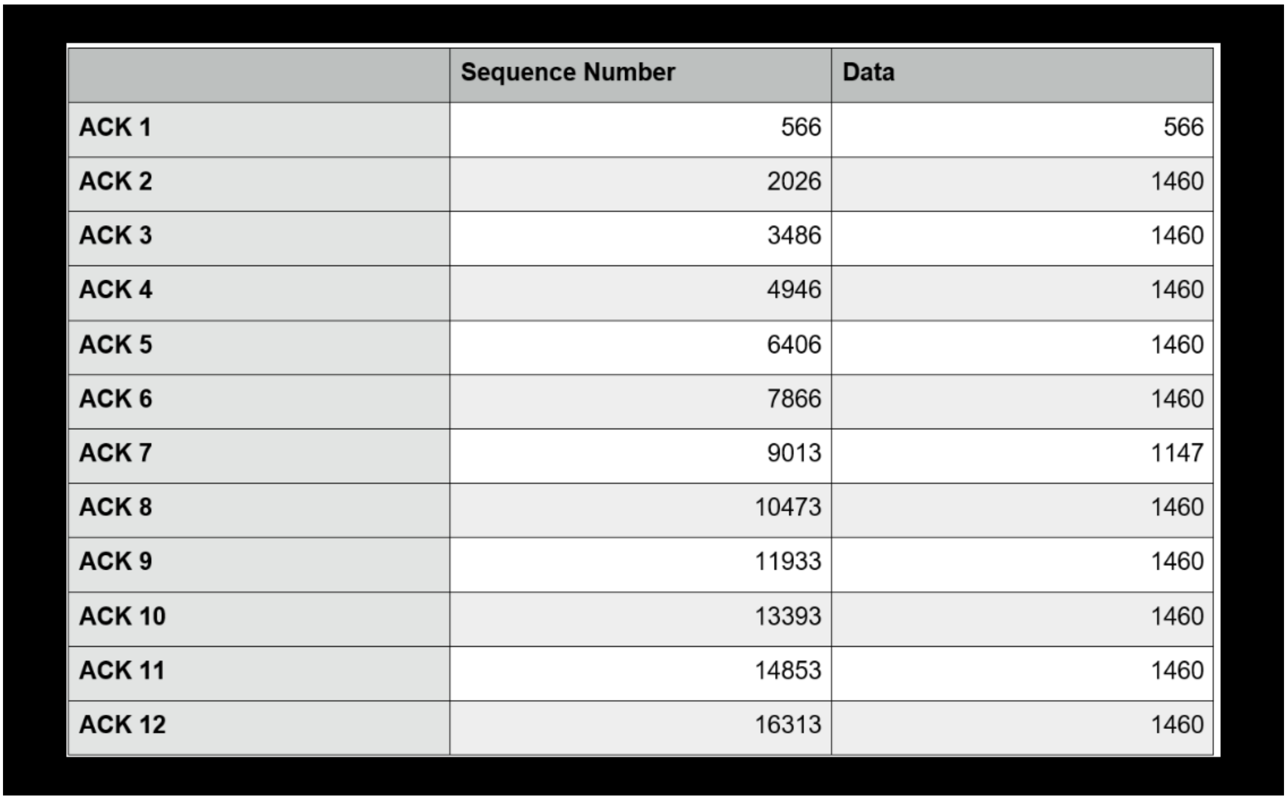
```
Segment 1: 565 bytes
Segment 2: 1460 bytes
Segment 3: 1460 bytes
Segment 4: 1460 bytes
Segment 5: 1460 bytes
Segment 6: 1460 bytes
```

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? Minimum Amount of Buffer Space: `17536 bytes` No, the sender is never throttled.

10. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question? No there are not. I checked to see if the sequence numbers were in order. If a

segment was retransmitted, the number of that segment would be smaller than the neighboring segments.

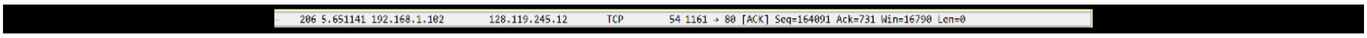
11. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (see Table 3.2 on page 250 in the text). `1460 bytes`. The difference between each ACK is about 1460



| | Sequence Number | Data |
|---|---|---|
| ACK 1 | 566 | 566 |
| ACK 2 | 2026 | 1460 |
| ACK 3 | 3486 | 1460 |
| ACK 4 | 4946 | 1460 |
| ACK 5 | 6406 | 1460 |
| ACK 6 | 7866 | 1460 |
| ACK 7 | 9013 | 1147 |
| ACK 8 | 10473 | 1460 |
| ACK 9 | 11933 | 1460 |
| ACK 10 | 13393 | 1460 |
| ACK 11 | 14853 | 1460 |
| ACK 12 | 16313 | 1460 |

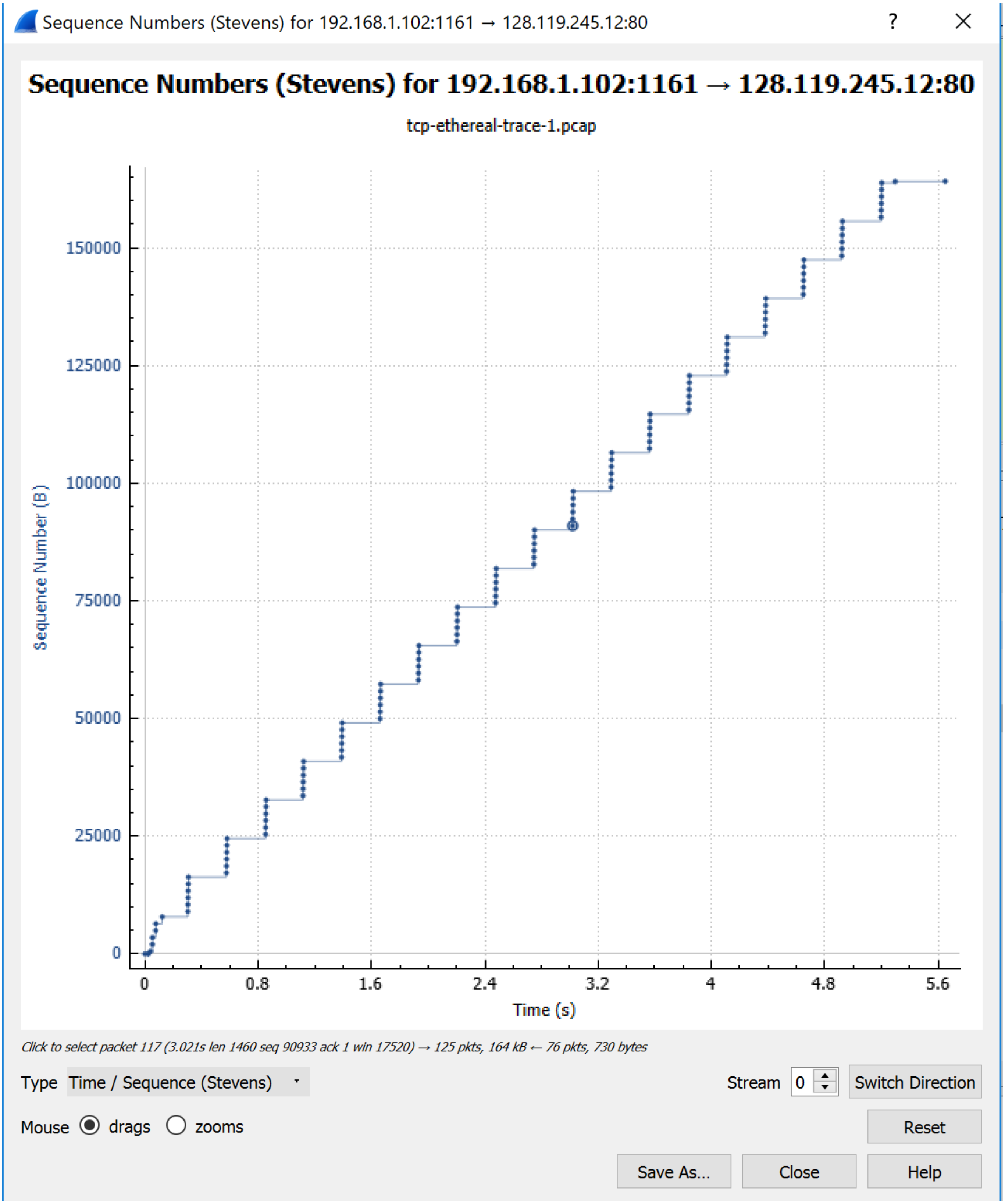
12. 
```
 What is the throughput (bytes transferred per unit time) for the TCP
connection?  Explain how you calculated this value
```

`Throughput` = `164090/5.6511` = `29036 B/s` The total amount of data is 1 less tha final ACK # divided by the total connection time.



```
206 5.651141 192.168.1.102    128.119.245.12    TCP    54 1161 → 80 [ACK] Seq=164091 Ack=731 Win=16790 Len=0
```

1. 
```
 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.
```



Sequence Numbers (Stevens) for 192.168.1.102:1161 → 128.119.245.12:80

**Sequence Numbers (Stevens) for 192.168.1.102:1161 → 128.119.245.12:80**

tcp-ethereal-trace-1.pcap

*Click to select packet 117 (3.021s len 1460 seq 90933 ack 1 win 17520) → 125 pkts, 164 kB ← 76 pkts, 730 bytes*

Type Time / Sequence (Stevens)

Stream 0

Mouse ● drags ○ zooms

TCP slowstart starts at the beginning of the transmission with the HTTP POST segment. The window size can be lower bound is estimated by looking at the amount of unACKd data. This is the amount of data that has not been successfully transferred (transferred and confirmed). The size of this window has to be greater that 8192 bytes, since the amount of unACKd data never exceeds 8192 bytes. That means we can't determine when slowstart ends and congestion control begins because data is not sent in a way that would cause congestion control to start. AKA the application doesn't send more than 8192 bytes at a time. According to the text, TCP behavior assumes that the sender is always sending data and does not wait for an ACK. In this case, it looks like the application waits for an ACK before sending additional data, even if there is still space in the transmission window. This means that the transmission is complete before the end of the slowstart phase, and that the transfer is slow. 1. 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 I wasn't able to get the trace working properly on my computer.