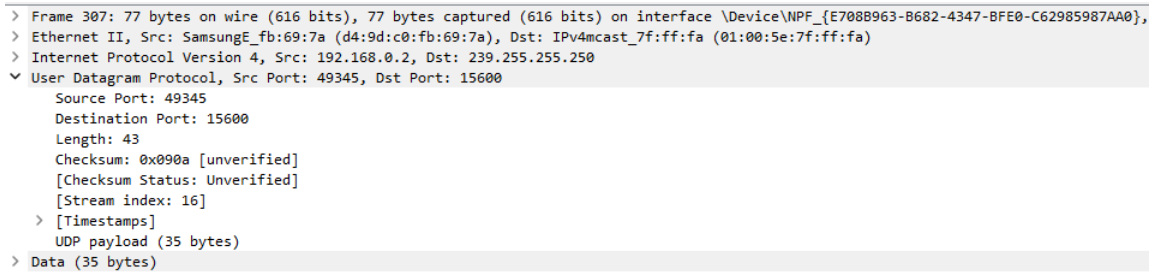
Problem 1

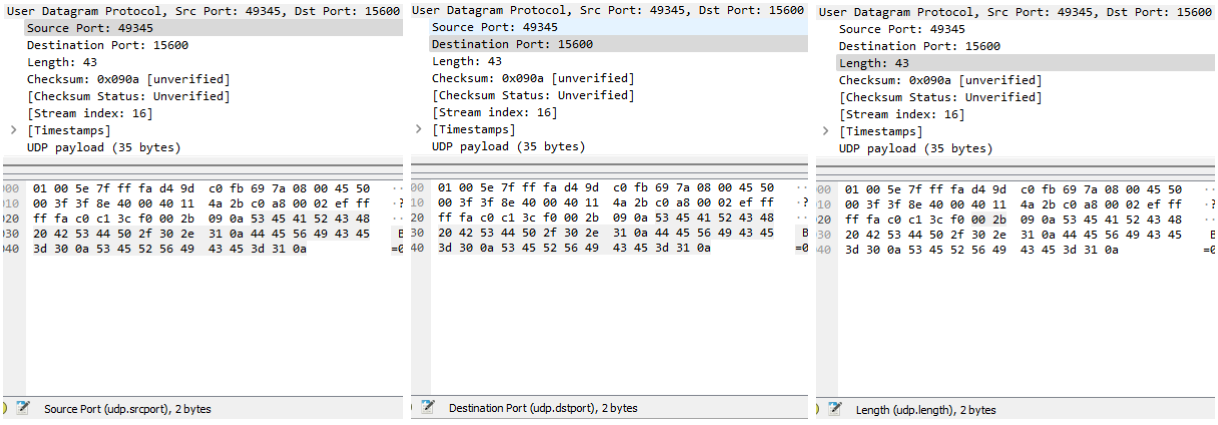
1. Select one UDP packet from your trace. From this packet, determine how many fields there are in the UDP header. Name these fields
2. By consulting the displayed information in Wireshark's packet content field for this packet, determine the length (in bytes) of each of the UDP header
3. The value in the Length field is the length of what? (You can consult the text for this answer). Verify your claim with your captured UDP packet
4. What is the maximum number of bytes that can be included in a UDP payload? (Hint: the answer to this question can be determined by your answer to b. above)
5. What is the largest possible source port number? (Hint: see the hint in d.)
6. What is the protocol number for UDP? Give your answer in both hexadecimal and decimal notation. To answer this question, you'll need to look into the Protocol field of the IP datagram containing this UDP segment.
7. Examine a pair of UDP packets in which your host sends the first UDP packet and the second UDP packet is a reply to this first UDP packet. (Hint: for a second packet to be sent in response to a first packet, the sender of the first packet should be the destination of the second packet). Describe the relationship between the port numbers in the two packets

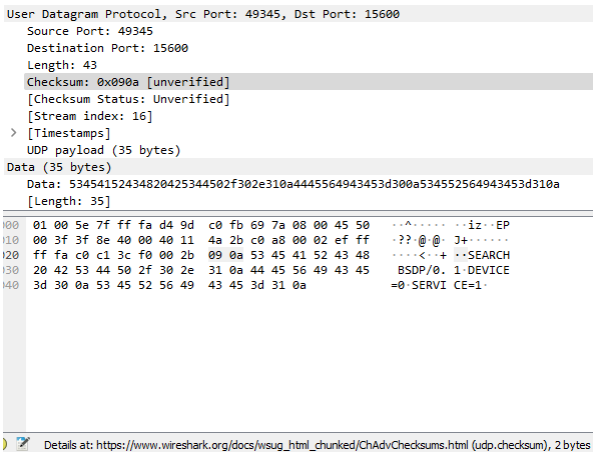
Solution

1) 4 Header Fields: Source Port, Destination Port, Length and Checksum

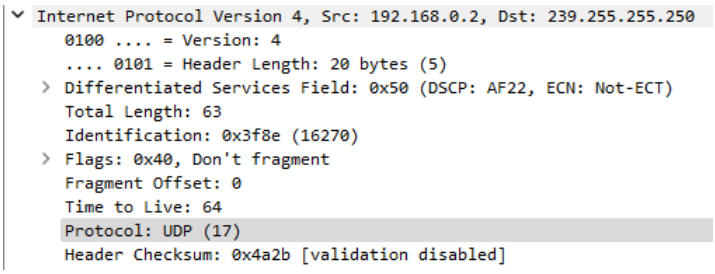


2) They are all 2 bytes. See bottom of these images.





3) Length is the sum of the 8 header field bytes + the encapsulated data bytes (of which there are 35). 35 + 8 = 43
4) $2^{16} - 1 - 8 = 65535 - 8 = 65527$ bytes. Subtracting 8 because of header field bytes.
5) $2^{16} - 1 = 65535$
6) Decimal: 17, Hex: 0x11



7) The source port and destination port are swapped. This is because first, 192.168.0.18 sends a packet to 213.179.197.152. Then, 213.179.197.152 sends a packet back to 192.168.0.18.

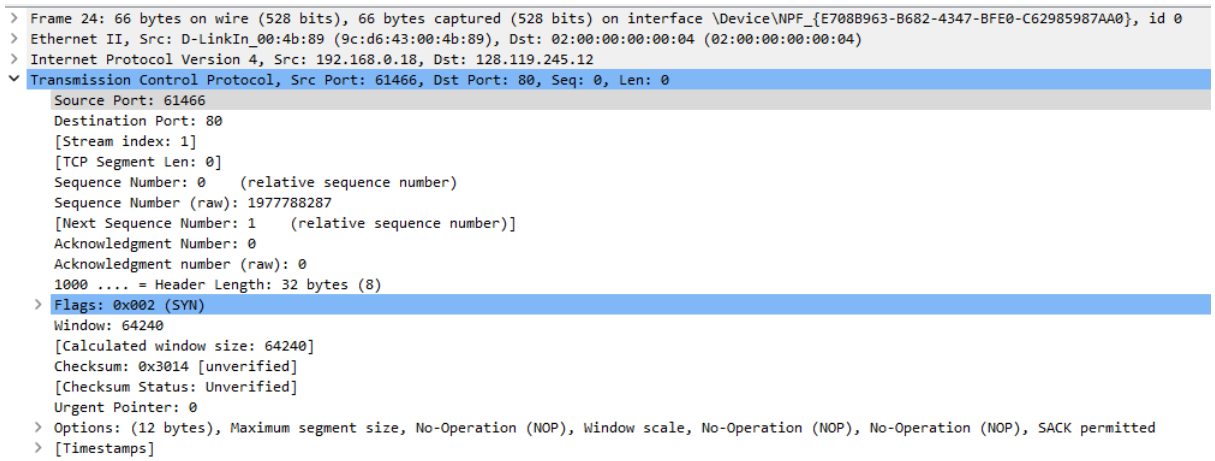
7 0.021917	192.168.0.18	213.179.197.152	UDP
8 0.023199	213.179.197.152	192.168.0.18	UDP

Problem 2

1. What is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu?
2. 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 identities the segment as a SYN segment?
3. 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 identities the segment as a SYNACK segment?
4. 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

5. What is the length of each of the first six TCP segments?
6. 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?
7. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question
8. 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 the table on slide 18 of Lecture 2 in Transport Layer)
9. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

Solution 1) Source Port: 61466, Source IP: 192.168.0.18



2) Sequence Number is 0. I know this is the TCP SYN segment because the SYN flag is set.

```
> Frame 24: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{E708B963-B682-4347-BFE0-C62985987AA0}, id 0
> Ethernet II, Src: D-LinkIn_00:4b:89 (9c:d6:43:00:4b:89), Dst: 02:00:00:00:00:04 (02:00:00:00:00:04)
> Internet Protocol Version 4, Src: 192.168.0.18, Dst: 128.119.245.12
v Transmission Control Protocol, Src Port: 61466, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 61466
    Destination Port: 80
    [Stream index: 1]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 1977788287
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    1000 .... = Header Length: 32 bytes (8)
  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: 64240
    [Calculated window size: 64240]
    Checksum: 0x3014 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted
  > [Timestamps]
```

3) Sequence Number is 0. I know this is the TCP SYNACK segment because both SYN and ACK flags are set. The value of the ACK field is 1. The value is determined by adding 1 to the initial sequence number of SYN segment from the client computer. $0 + 1 = 1$

```
> Transmission Control Protocol, Src Port: 61466, Dst Port: 80, Seq: 1, Ack: 1, Len: 1460
v Data (1460 bytes)
    Data: 504f5354202f7769726573686861726b2d6c6162732f6c6162332d312d7265706c792e6874…
    [Length: 1460]
```

```
0000  02 00 00 00 00 04 9c d6  43 00 4b 89 08 00 45 00   ······ C·K···E·
0010  05 dc 25 45 40 00 80 06  99 98 c0 a8 00 12 80 77   ··%E@··· ·····w
0020  f5 0c f0 1a 00 50 75 e2  a7 80 58 fb c4 84 50 10   ·····Pu· ··X···P·
0030  02 01 63 d7 00 00 50 4f  53 54 20 2f 77 69 72 65   ··c···PO ST /wire
0040  73 68 61 72 6b 2d 6c 61  62 73 2f 6c 61 62 33 2d   shark-la bs/lab3-
0050  31 2d 72 65 70 6c 79 2e  68 74 6d 20 48 54 54 50   1-reply. htm HTTP
0060  2f 31 2e 31 0d 0a 48 6f  73 74 3a 20 67 61 69 61   /1.1··Ho st: gaia
0070  2e 63 73 2e 75 6d 61 73  73 2e 65 64 75 0d 0a 55   .cs.umas s.edu··U
0080  73 65 72 2d 41 67 65 6e  74 3a 20 4d 6f 7a 69 6c   ser-Agen t: Mozil
0090  6c 61 2f 35 2e 30 20 28  57 69 6e 64 6f 77 73 20   la/5.0 ( Windows
00a0  4e 54 20 31 30 2e 30 3b  20 57 69 6e 36 34 3b 20   NT 10.0;  Win64;
00b0  78 36 34 3b 20 72 76 3a  39 33 2e 30 29 20 47 65   x64; rv: 93.0) Ge
00c0  63 6b 6f 2f 32 30 31 30  30 31 30 31 20 46 69 72   cko/2010 0101 Fir
00d0  65 66 6f 78 2f 39 33 2e  30 0d 0a 41 63 63 65 70   efox/93. 0··Accep
00e0  74 3a 20 74 65 78 74 2f  68 74 6d 6c 2c 61 70 70   t: text/ html,app
00f0  6c 69 63 61 74 69 6f 6e  2f 78 68 74 6d 6c 2b 78   lication /xhtml+x
0100  6d 6c 2c 61 70 70 6c 69  63 61 74 69 6f 6e 2f 78   ml,appli cation/x
0110  6d 6c 3b 71 3d 30 2e 39  2c 69 6d 61 67 65 2f 61   ml;q=0.9 ,image/a
0120  76 69 66 2c 69 6d 61 67  65 2f 77 65 62 70 2c 2a   vif,imag e/webp,*
0130  2f 2a 3b 71 3d 30 2e 38  0d 0a 41 63 63 65 70 74   /*;q=0.8 ··Accept
0140  2d 4c 61 6e 67 75 61 67  65 3a 20 65 6e 2d 55 53   -Languag e: en-US
0150  2c 65 6e 3b 71 3d 30 2e  35 0d 0a 41 63 63 65 70   ,en;q=0. 5··Accep
0160  74 2d 45 6e 63 6f 64 69  6e 67 3a 20 67 7a 69 70   t-Encodi ng: gzip
0170  2c 20 64 65 66 6c 61 74  65 0d 0a 43 6f 6e 74 65   , deflat e··Conte
0180  6e 74 2d 54 79 70 65 3a  20 6d 75 6c 74 69 70 61   nt-Type:  multipa
0190  72 74 2f 66 6f 72 6d 2d  64 61 74 61 3b 20 62 6f   rt/form- data; bo
```

4) Sequence Number is 1 as seen in the TCP header of the HTTP POST

```
> Frame 193: 1095 bytes on wire (8760 bits), 1095 bytes captured (8760 bits) on interface \Device\NPF_{E708B963-B
> Ethernet II, Src: D-LinkIn_00:4b:89 (9c:d6:43:00:4b:89), Dst: 02:00:00:00:00:04 (02:00:00:00:00:04)
> Internet Protocol Version 4, Src: 192.168.0.18, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 61466, Dst Port: 80, Seq: 151838, Ack: 1, Len: 1041
> [105 Reassembled TCP Segments (152878 bytes): #27(1460), #28(1460), #29(1460), #30(1460), #31(1460), #32(1460),
v Hypertext Transfer Protocol
  v POST /wireshark-labs/lab3-1-reply.htm HTTP/1.1\r\n
    > [Expert Info (Chat/Sequence): POST /wireshark-labs/lab3-1-reply.htm HTTP/1.1\r\n]
      Request Method: POST
      Request URI: /wireshark-labs/lab3-1-reply.htm
      Request Version: HTTP/1.1
    Host: gaia.cs.umass.edu\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:93.0) Gecko/20100101 Firefox/93.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Content-Type: multipart/form-data; boundary=---------------------------136608768182377867818066231l2\r\n
  > Content-Length: 152357\r\n
    Origin: null\r\n
```

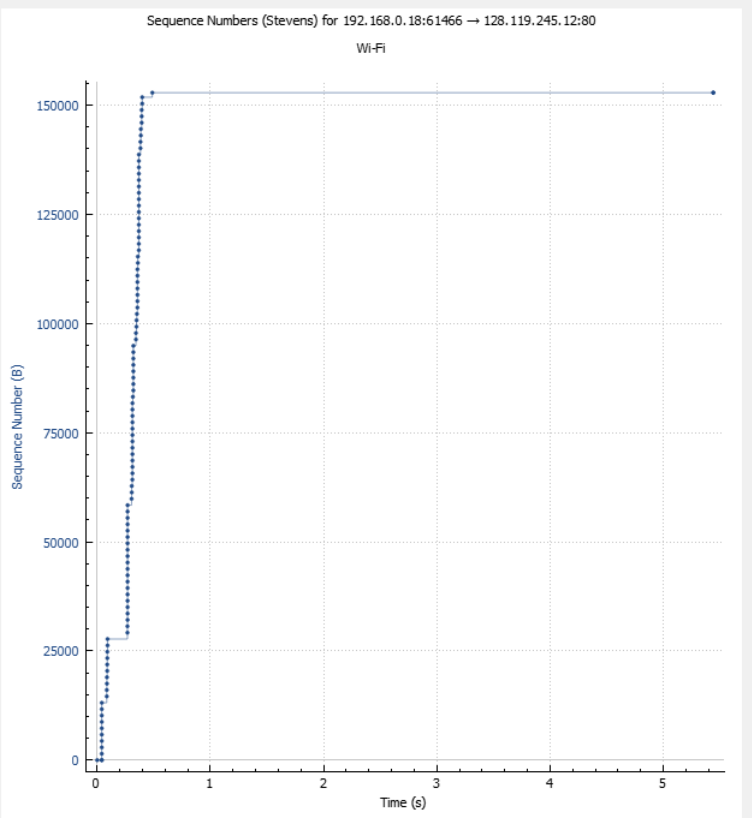
5) The first 6 TCP segments have length 1514. The first TCP segments are from the handshake process.

No.	Time	Source	Destination	Protocol	Length	Info
16	3.251477	192.168.0.18	34.210.55.11	TCP	55	61457 → 443 [ACK] Seq=1 Ack=1 Win=510 Len=1 [TCP segment of a reassembled PDU]
23	3.350362	34.210.55.11	192.168.0.18	TCP	66	443 → 61457 [ACK] Seq=1 Ack=2 Win=120 Len=0 SLE=1 SRE=2
24	3.650048	192.168.0.18	128.119.245.12	TCP	66	61466 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
25	3.690258	128.119.245.12	192.168.0.18	TCP	66	80 → 61466 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
26	3.690337	192.168.0.18	128.119.245.12	TCP	54	61466 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
27	3.690550	192.168.0.18	128.119.245.12	TCP	1514	61466 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=1460
28	3.690550	192.168.0.18	128.119.245.12	TCP	1514	61466 → 80 [ACK] Seq=1461 Ack=1 Win=131328 Len=1460
29	3.690550	192.168.0.18	128.119.245.12	TCP	1514	61466 → 80 [ACK] Seq=2921 Ack=1 Win=131328 Len=1460
30	3.690550	192.168.0.18	128.119.245.12	TCP	1514	61466 → 80 [ACK] Seq=4381 Ack=1 Win=131328 Len=1460
31	3.690550	192.168.0.18	128.119.245.12	TCP	1514	61466 → 80 [ACK] Seq=5841 Ack=1 Win=131328 Len=1460
32	3.690550	192.168.0.18	128.119.245.12	TCP	1514	61466 → 80 [ACK] Seq=7301 Ack=1 Win=131328 Len=1460

6) Minimum Window size is 29200, seen in the first ACK from the server. The window size grows to 131328 (see above image). There is no throttling here.

```
v Transmission Control Protocol, Src Port: 80, Dst Port: 61466, Seq: 0, Ack: 1, Len: 0
    Source Port: 80
    Destination Port: 61466
    [Stream index: 1]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 1492894851
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 1977788288
    1000 .... = Header Length: 32 bytes (8)
  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
    v .... .... ..1. = Syn: Set
      > [Expert Info (Chat/Sequence): Connection establish acknowledge (SYN+ACK): server port 80]
      .... .... ...0 = Fin: Not set
      [TCP Flags: ·····A··S·]
    Window: 29200
```

7) There are no re-transmitted segments in the trace file. The segment number increases linearly with no re-transmissions. I checked the Time Sequence TCP Stream Graph (Stevens)



8) The receiver typically acknowledges 1460 bytes of data. This is shown in the difference in sequence number between 2 acknowledgements. Here you can see that taking the difference of any 2 consecutive acknowledgements = 1460.

24	3.650048	192.168.0.18	128.119.245.12	TCP	66 61466 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
25	3.690258	128.119.245.12	192.168.0.18	TCP	66 80 → 61466 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
26	3.690337	192.168.0.18	128.119.245.12	TCP	54 61466 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
27	3.690550	192.168.0.18	128.119.245.12	TCP	1514 61466 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=1460
28	3.690550	192.168.0.18	128.119.245.12	TCP	1514 61466 → 80 [ACK] Seq=1461 Ack=1 Win=131328 Len=1460
29	3.690550	192.168.0.18	128.119.245.12	TCP	1514 61466 → 80 [ACK] Seq=2921 Ack=1 Win=131328 Len=1460
30	3.690550	192.168.0.18	128.119.245.12	TCP	1514 61466 → 80 [ACK] Seq=4381 Ack=1 Win=131328 Len=1460
31	3.690550	192.168.0.18	128.119.245.12	TCP	1514 61466 → 80 [ACK] Seq=5841 Ack=1 Win=131328 Len=1460
32	3.690550	192.168.0.18	128.119.245.12	TCP	1514 61466 → 80 [ACK] Seq=7301 Ack=1 Win=131328 Len=1460
33	3.690550	192.168.0.18	128.119.245.12	TCP	1514 61466 → 80 [ACK] Seq=8761 Ack=1 Win=131328 Len=1460
34	3.690550	192.168.0.18	128.119.245.12	TCP	1514 61466 → 80 [ACK] Seq=10221 Ack=1 Win=131328 Len=1460

4381-2921 = 1460
2921-1461 = 1460
1460-1 = 1460

When the receiver is ACKing every other received segment it can be seen in the image below, segments #40 and #43. The Acknowledgement numbers are 4381 and 2921 with the difference being 1460 bytes.

9) We can see the last acknowledgment sent with the FIN flag. The last segment has ACK #152879 and the first segment has ACK #1. Total of 152879-1 = 152878 bytes.

249	9.090681	128.119.245.12	192.168.0.18	TCP	54 80 → 61466 [FIN, ACK] Seq=778 Ack=152879 Win=250368 Len=0

From this image we can also see the time of 9.090681.

No.	Time	Source	Destination	Protocol	Length	Info
16	3.251477	192.168.0.18	34.210.55.11	TCP	55	61457 → 443 [ACK] Seq=1 Ack=1 Win=510 Len=1 [TCP segment of a reassembled PDU]
23	3.350362	34.210.55.11	192.168.0.18	TCP	66	443 → 61457 [ACK] Seq=1 Ack=2 Win=120 Len=0 SLE=1 SRE=2
24	3.650048	192.168.0.18	128.119.245.12	TCP	66	61466 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
25	3.690258	128.119.245.12	192.168.0.18	TCP	66	80 → 61466 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
26	3.690337	192.168.0.18	128.119.245.12	TCP	54	61466 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
27	3.690550	192.168.0.18	128.119.245.12	TCP	1514	61466 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=1460
28	3.690550	192.168.0.18	128.119.245.12	TCP	1514	61466 → 80 [ACK] Seq=1461 Ack=1 Win=131328 Len=1460
29	3.690550	192.168.0.18	128.119.245.12	TCP	1514	61466 → 80 [ACK] Seq=2921 Ack=1 Win=131328 Len=1460
30	3.690550	192.168.0.18	128.119.245.12	TCP	1514	61466 → 80 [ACK] Seq=4381 Ack=1 Win=131328 Len=1460
31	3.690550	192.168.0.18	128.119.245.12	TCP	1514	61466 → 80 [ACK] Seq=5841 Ack=1 Win=131328 Len=1460
32	3.690550	192.168.0.18	128.119.245.12	TCP	1514	61466 → 80 [ACK] Seq=7301 Ack=1 Win=131328 Len=1460

From this image we can see the start time of 3.650048.

$$\text{Throughput} = \frac{152879 - 1}{9.090681 - 3.650048}$$

$$= \frac{152878}{5.440633}$$

$$= 28099.3 \text{ bytes/sec}$$