

# LAB 3

TDTS04

hugho678 & viles222

1. What are the first and last packets for the POST request? Hint: Look for where the POST request is first initiated in the TCP stream; it often starts earlier in the trace than where 'POST' is explicitly labeled.

0930	44	70	1f	bd	09	60	50	47	53	54	20	2f	65	74	68	65	Dp	...	0	ST	/ethe
0940	72	65	61	6c	2d	6c	61	62	73	2f	6c	61	62	33	24	61					
0950	72	65	78	6c	79	6c	79	62	74	7d	29	48	54	54	50	2f					
0960	31	2c	31	6d	0a	48	6f	73	73	3a	28	67	61	69	61	61					
0970	63	73	2c	75	6d	61	73	73	2e	65	64	75	60	6b	5a	73					
0980	65	72	2d	41	67	65	6a	65	6a	20	4d	6f	7a	39	6c	6c					
0990	61	2f	35	2c	30	29	28	57	69	62	64	6f	77	73	30	26					
1000	55	3b	28	57	69	6a	64	6f	77	73	29	4e	54	29	25	30					
1010	31	3b	20	65	62	55	53	30	20	72	76	3a	31	2c	30	30					
1020	3c	32	29	2d	45	65	63	6b	6f	2f	32	30	39	33	30	32					
1030	30	30	20	4e	65	74	73	63	71	60	65	2f	37	2e	30	32					
1040	08	0a	41	63	63	65	78	74	3a	28	74	65	78	74	2f	78					
1050	6d	6c	2c	61	78	79	69	69	63	61	74	69	6f	6e	2f	78					
1060	6d	6c	2c	61	78	79	69	69	63	61	74	69	6f	6e	2f	78					

The first packet is of the Post request is frame 4 and the last one is frame 199.

No.	Time	Source	Destination	Protocol	Length	Info
181	4.921025	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=149737 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
182	4.921916	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=151197 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
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
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	5.201159	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 /etherreal-labs/lab3-1-reply.htm HTTP/1.1 (text/plain)
201	5.447887	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=162309 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 SACK_PERM=1
▶ 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: Linksys6_da:af:73 (00:06:25:da:af:73)						
▶ Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.119.245.12						
▶ Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 164041, Ack: 1, Len: 50						
Source Port: 1161						
Destination Port: 80						
[Stream index: 0]						
[Conversation completeness: Incomplete, DATA (15)]						
[TCP Segment Len: 50]						
Sequence Number: 164041 (relative sequence number)						
Sequence Number (raw): 232293053						
[Next Sequence Number: 164091 (relative sequence number)]						
Acknowledgment Number: 1 (relative ack number)						
Acknowledgment number (raw): 883061786						
0101 .... = Header Length: 20 bytes (5)						
Flags: 0x018 (PSH, ACK)						
Window: 17520						
[Calculated window size: 17520]						
[Window size scaling factor: -2 (no window scaling used)]						
Checksum: 0x9f0f [unverified]						
[Checksum Status: Unverified]						
Urgent Pointer: 0						
▶ [Timestamps]						
▶ [SEQ/ACK analysis]						
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(1460), #13(1147), #18(1460), #19(1460), #20(1460), #21(1460), #22(1460), #23(892), #30(1460), #31(1460), #32(1460), #33(1460),						
▶ Hypertext Transfer Protocol						
▶ MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "-----265001916915724"						
0000	00 06 25 da af 73 00 20	e0 8a 70 1a 00 00 45 00	. . . . .			
0010	00 5a 1e 9a 40 00 80 06	a4 71 c0 a8 01 66 80 77	. . . . .			
0020	f5 0c 04 89 00 50 0d d8	82 bd 34 a2 74 1a 50 18	. . . . .			
0030	44 70 9f 0f 00 00 0d 0a	2d 2d 2d 2d 2d 2d 2d 2d	Dp . . . . .			
0040	2d 2d 2d 2d 2d 2d 2d 2d	2d 2d 2d 2d 2d 2d 2d 2d	. . . . .			
0050	2d 2d 2d 2d 2d 32 36 35	30 39 31 39 31 36 39 31	-----265 00191691			
0060	35 37 32 34 2d 2d 0d 0a		5724----			
Frame (104 bytes) Reassembled TCP (164090 bytes)						
Transmission Control Protocol: Protocol Packets: 213 · Displayed: 202 (94.8%)						

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

The source ip address is 192.168.1.102 and the sending and receiving port number is 1161 as seen in blue.

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

The ip address of gaia.cs.umass.edu is 128.119.245.12 and the port being used is 80 as seen in red.

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?

A solid blue horizontal line.A solid blue horizontal line.

The sequence number for the TCP SYN segment is 232129012 and in the segment there is a syn flag set to identify it as a SYN segment as seen in blue.

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?

```

No.      Time            Source                Destination          Protocol Length Info
  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
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
Transmission Control Protocol, Src Port: 80, Dst Port: 1161, Seq: 0, Ack: 1, Len: 0
  Source Port: 80
  Destination Port: 1161
  [Stream index: 0]
  [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 0]
  Sequence Number: 0      (relative sequence number)
  Sequence Number (raw): 883061785
  [Next Sequence Number: 1      (relative sequence number)]
  Acknowledgment Number: 1      (relative ack number)
  Acknowledgment number (raw): 232129013
0111 .... = Header Length: 28 bytes (7)
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: 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
[Timestamps]
[SEQ/ACK analysis]

```

The sequence number sent by the server is 883061785. In the SYNACK segment the ACK value is 232129013, which is the TCP syn value incremented once as to identify it as a synack segment

6. What is the sequence number of the TCP segment containing the HTTP POST command?



The sequence number of the TCP segment containing the HTTP POST command is 232129013 as seen in blue.

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 269 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 270 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.*

Frame	Segment number	Time segment sent	ACK received	RTT	Estimated RTT
4	232129013	0,026477	0,053937	0,02746	0,02746
5	232129578	0,041737	0,077294	0,035557	0,02847
7	232131038	0,054026	0,124085	0,070059	0,03367
8	232132498	0,05469	0,169118	0,114428	0,04376
10	232133958	0,077405	0,217299	0,139894	0,05578
11	232135418	0,078157	0,267802	0,189645	0,07251

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

Frame 4 is 565 bytes as seen in green in question 6 and the rest of the segments are 1460 bytes long as seen in red.

```
[Stream index: 0]
[Conversation completeness: Incomplete, DATA (15)]
[TCP Segment Len: 1460]
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)
Window: 17520
[Calculated window size: 17520]
[Window size scaling factor: -2 (no window scaling used)]
```

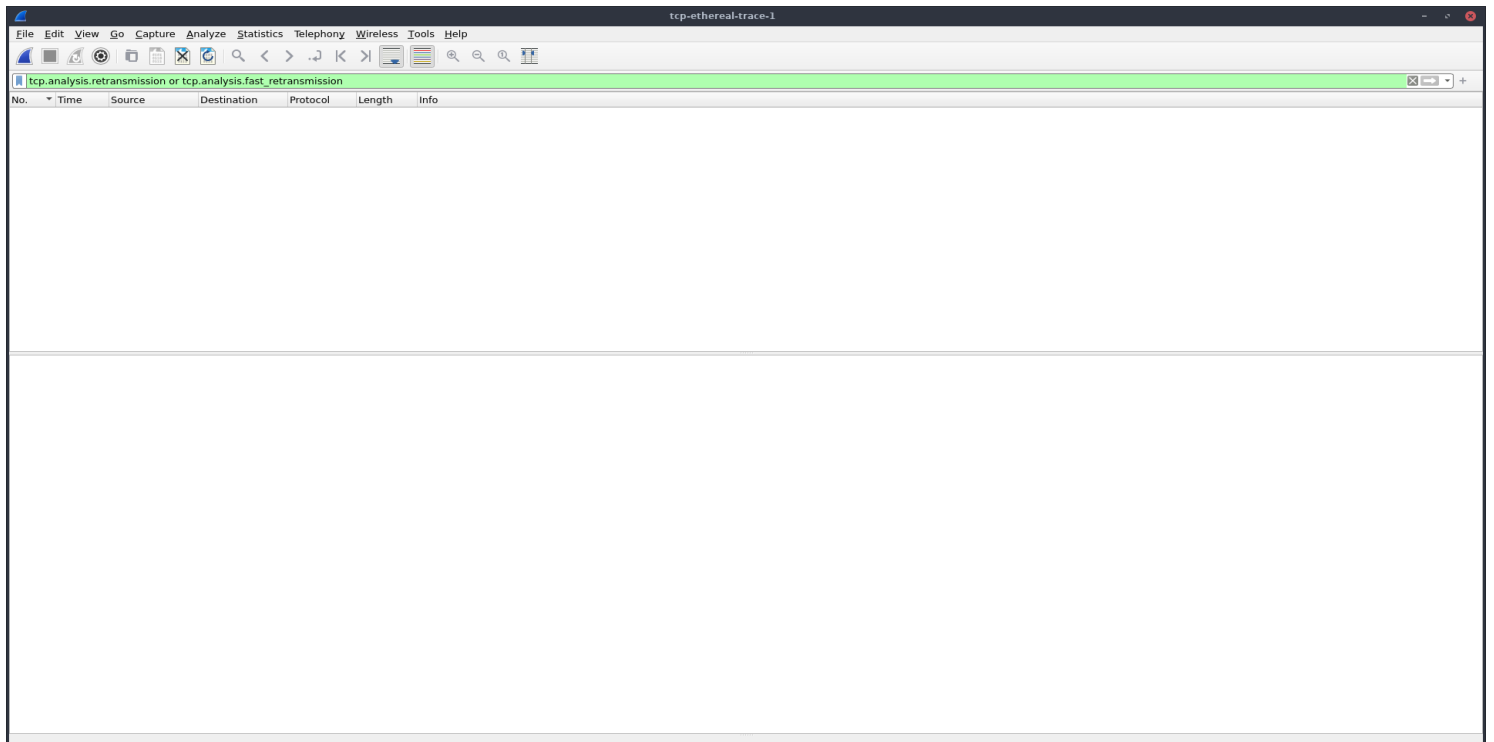


9. What is the minimum amount of available buffer space advertised at the receiver for the entire trace?

```
No.      Time      Source      Destination      Protocol Length Info
  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
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
Transmission Control Protocol, Src Port: 80, Dst Port: 1161, Seq: 0, Ack: 1, Len: 0
  Source Port: 80
  Destination Port: 1161
  [Stream index: 0]
  [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 0]
  Sequence Number: 0 (relative sequence number)
  Sequence Number (raw): 883061785
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 232129013
  0111 .... = Header Length: 28 bytes (7)
  Flags: 0x012 (SYN, ACK)
  Window: 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
    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
  [Timestamps]
  [SEQ/ACK analysis]
```

The minimum amount of available buffer space is found in the TCP SYN ACK in frame 2 where the window size is 5840 bytes.

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



There are no retransmitted segment in the trace file. To check in the trace we would look for duplicated segment numbers and duplicated ACKs.

11. How much data does the receiver typically acknowledge in an ACK? Identify instances in the packet trace where the receiver acknowledges two segments instead of one. Hint: Examine the sequence and ACK numbers to detect this pattern (see Table 3.2 on page 278 in the text).

The receiver typically ACKs 1460 bytes. An instance where the receiver acks two segments is for example in frame 80 where two 1460 packets are ACK ed once.

tcp-ethereal-trace-1						
Arkiv Redigera Visa Kör Fånga Analysera Statistik Telefonj Trådlöst Verktyg Hjälp						
tcp						
No.	Time	Source	Destination	Protocol	Length	Info
70	1.584980	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=47621 Win=62780 Len=0
71	1.661513	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=49973 Win=62780 Len=0
72	1.661734	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=49973 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
73	1.662474	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=51433 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
74	1.663315	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=52893 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
75	1.664198	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=54353 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
76	1.665254	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=55813 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
77	1.666151	192.168.1.102	128.119.245.12	TCP	946	1161 → 80 [PSH, ACK] Seq=57273 Ack=1 Win=17520 Len=892 [TCP segment of a reassembled PDU]
78	1.758227	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=52893 Win=62780 Len=0
79	1.860863	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]
<div> <div>Frame 80: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)</div> <div> <div>Ethernet II, Src: LinksysG_da:af:73 (00:06:25:da:af:73), Dst: Actionte_8a:70:1a (00:20:e0:8a:70:1a)</div> <div>Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.1.102</div> <div>Transmission Control Protocol, Src Port: 80, Dst Port: 1161, Seq: 1, Ack: 58165, Len: 0</div> </div> </div>						
<div> <div>0000 00 20 e0 8a 70 1a 00 06 25 da af 73 08 00 45 00 . . . . . E</div> <div>0010 00 28 58 93 40 00 37 06 d3 aa 80 77 f5 0c c0 a8 . (X . 0 7 . . . . . W . . . . .</div> <div>0020 01 66 00 50 04 89 34 a2 74 1a 0d d6 e5 29 50 10 . f . P . . 4 . . t . . . . . ) P .</div> <div>0030 f5 3c e2 6f 00 00 2d bd 00 00 01 01 . &lt; . o . . . . . . . . . .</div> </div>						

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

Throughout = total sent bytes in tcp connection / time = 29036,773 bytes per second.

time = 0.026477 – 5.65111

total sent bytes == 164090

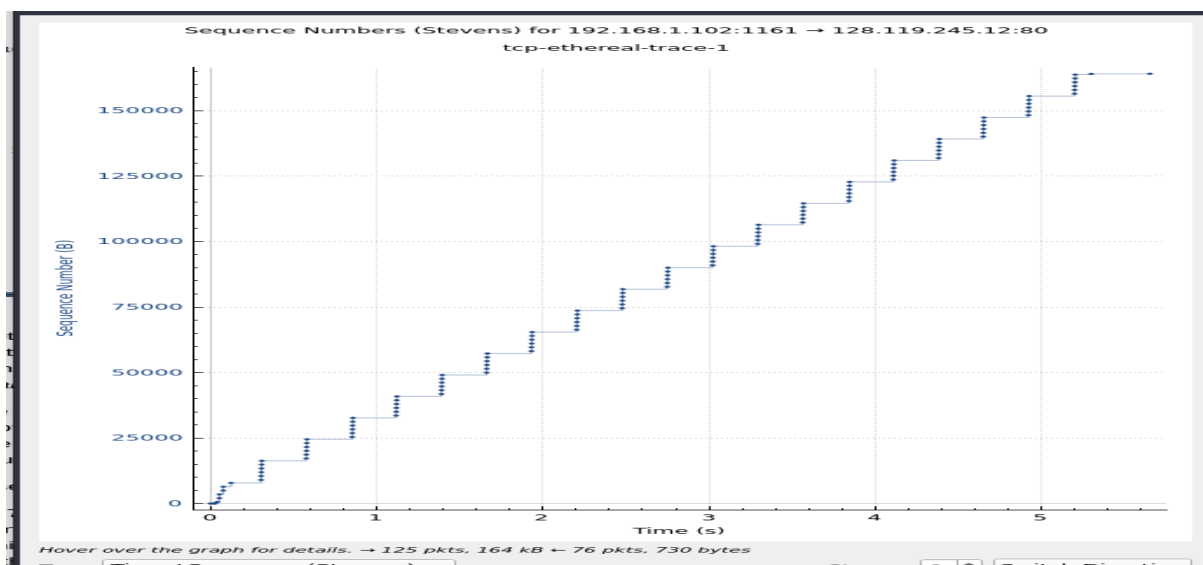
**Task A:** Now, based on questions 1-12, please write two paragraphs explaining and discussing your observations from the above questions. One paragraph should describe and discuss the connections at a high level. The second paragraph should discuss the impact of RTT estimates, packet losses, and interpreted packet loss events. Note that your answer may benefit from explaining and/or referring to some of your observations from the practice questions explicitly. Note that, similar to previous assignments, you are expected to convince us that you understand these aspects of TCP.

**Discussing the connections at a highlevel**, the tcp connection begins in frame 4 and ends in frame 199. With a local computer transferring data to gaia.umass.edu through http port 80. After the syn ack is completed the tcp connection is realized and the client starts sending 560 and 1460 bytes size packes of data over to the server.

Discussing the impact of RTT estimates, packet losses and interpreted packet loss events

Our RTT estimated did deviate by a significant bit from the real values. In the TCP connection there were no packet losses or packet loss events as there are no ack duplicates found at all in the trace.

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 any indications of packet loss in this communication. What patterns in the sequence numbers suggest lost packets?



14. Explain the relationship between (i) the congestion window (cwnd), (ii) the receiver advertised window (rwnd), (iii) the number of unacknowledged bytes, and (iv) the effective window at the sender (i.e., the window effectively limiting the data transmission).

(i). The congestion window is variable the sender maintains for congestion control which is used to give the sender an idea of how much free buffer space is available at the receiver and will dynamically adjust for network conditions. While the rwnd (ii) is the amount of buffer space available at the receiver and is advertised in the tcp header. It is used to prevent the sender from sending too much data at once. (iii) The number of unacknowledged bytes is the amount of data the sender has sent but has not yet been acknowledged for by the receiver. The sender will keep the amount unacknowledged data less than the value of rwnd to prevent itself from overflowing the buffer of the receiver. (iv) The effective window at the sender is the smaller value of the congestion window and the rwnd minus the unacknowledged bytes and is the value limiting the transmission.

15. Is it **generally** possible to find the congestion window size (cwnd) and how it changes with time, from the captured trace files? If so, please explain how. If not, please explain when and when not. Motivate your answer and give examples.

It is now generally possible to find the cwnd value from a captured trace as it is a variable maintained at the sender and is not transmitted explicitly in the tcp package headers. Yet it can be estimated its behaviour from observations as retransmissions, duplicate acknowledgements or packet losses where the cwnd adjusts and from which one can infer the cwnd indirectly.

16. What is the throughput of each of the connections in bps (bits per second)? What is the total bandwidth of the host on which the clients are running? Discuss the TCP fairness for this case.

Connection	Total transferred bytes	Duration (in seconds)	RTT (in milliseconds)	average throughput of connection
1	165095720	521	12	2535059,0403071
2	165842766	521	12	2546529,99616123
3	165458792	514	12	2575234,11673152
4	163235772	512	12	2550558,9375

17. What is the throughput of each of the connections in bps (bits per second)? What is the total bandwidth of the host on which the clients are running? Discuss the TCP fairness for this case.

Connection	Total transferred bytes	Duration (in seconds)	RTT (in milliseconds)	average throughput of connection in mega bits per second	Total bandwidth of host in Mbits per sec
1	261319130	90	13	23,2250123	93,8862172
2	175995832	90	35	15,6379925	519062
3	151894552	90	68	140223	13,4915443
4	140388568	90	73	442732	12,4688701
5	108610702	90	49	830737	9,64903126
6	70644690	90	33	075803	6,27722635
7	65744938	90	135	033821	5,83524162
8	43212876	90	326	644921	3,82728127
9	39222524	90	322	006621	3,47401731
				58256	

A tcp connection is considered fair if each connection gets and equal share of the bandwidth.

18. Discuss the TCP fairness for this case. How does it differ from the previous cases, and how is it affected by the use of BitTorrent?

Connection	Total transferred bytes	Duration (in seconds)	RTT (in milliseconds)
1	108851134	58	40
2	90435681	58	36
3	57971584	53	100
4	32000012	29	68
5	32557334	35	31
6	27199361	31	33
7	26329578	31	122
8	38834490	56	146
9	23571761	35	74
10	36252962	55	66

For this case it is harder to discuss tcp fairness as the RTT and the duration differs a lot for most of the connection. Connections 1 and 2, and 4 and 9 have similar durations and similar RTT but different amounts of transferred bytes hinting on TCP unfairness. This might be the cause of the use of bittorrents use of multiple TCP connections at once which might make the connections compete with each single connections tcp running at the same time.

For all of these questions you must take a closer look at the relationships between the characteristics of the different connections and discuss your findings in the context of the different experiments. You are expected to show that you understand the concept of TCP fairness and how the different scenarios may impact the throughput relationships that you observe and those that you may expect in general. To help the discussion you may for example want to create a scatter plot that show the estimated round trip time (RTT) and throughput against each other (for the different connections). You also want to carefully examine and discuss the above throughput equation and how it may apply to each scenario.