

Marcos Ondruska
Student id: 2685885
Project 2

2) The IP address of gaia.cs.umass.edu is 128.119.245.12, and the port numbers are Port 80 for HTTP and Port 443 for HTTPS.

69	3.518002	192.168.7.75	128.119.245.12	TCP	78	53161 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=378263841 TSecr=0 SACK_PERM
70	3.518064	192.168.7.75	128.119.245.12	TCP	78	53162 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=753491732 TSecr=0 SACK_PERM
71	3.518170	192.168.7.75	128.119.245.12	TCP	78	53163 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=1010781211 TSecr=0 SACK_PERM
80	3.574439	128.119.245.12	192.168.7.75	TCP	66	80 → 53162 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
81	3.574439	128.119.245.12	192.168.7.75	TCP	66	80 → 53161 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
82	3.574439	128.119.245.12	192.168.7.75	TCP	66	443 → 53163 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
83	3.574513	192.168.7.75	128.119.245.12	TCP	54	53162 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
84	3.574551	192.168.7.75	128.119.245.12	TCP	54	53161 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
85	3.574552	192.168.7.75	128.119.245.12	TCP	54	53163 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0

3) The IP address of my client computer is 192.168.7.75, and the port numbers used are Port 80 for HTTP and Port 443 for HTTPS.

69	3.518002	192.168.7.75	128.119.245.12	TCP	78	53161 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=378263841 TSecr=0 SACK_PERM
70	3.518064	192.168.7.75	128.119.245.12	TCP	78	53162 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=753491732 TSecr=0 SACK_PERM
71	3.518170	192.168.7.75	128.119.245.12	TCP	78	53163 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=1010781211 TSecr=0 SACK_PERM
80	3.574439	128.119.245.12	192.168.7.75	TCP	66	80 → 53162 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
81	3.574439	128.119.245.12	192.168.7.75	TCP	66	80 → 53161 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
82	3.574439	128.119.245.12	192.168.7.75	TCP	66	443 → 53163 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
83	3.574513	192.168.7.75	128.119.245.12	TCP	54	53162 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
84	3.574551	192.168.7.75	128.119.245.12	TCP	54	53161 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
85	3.574552	192.168.7.75	128.119.245.12	TCP	54	53163 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0

4) The sequence number of the TCP SYN that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu is “0”. The [SYN] in the info section identifies the SYN segment.

69	3.518002	192.168.7.75	128.119.245.12	TCP	78	53161 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=378263841 TSecr=0 SACK_PERM
70	3.518064	192.168.7.75	128.119.245.12	TCP	78	53162 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=753491732 TSecr=0 SACK_PERM
71	3.518170	192.168.7.75	128.119.245.12	TCP	78	53163 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=1010781211 TSecr=0 SACK_PERM
80	3.574439	128.119.245.12	192.168.7.75	TCP	66	80 → 53162 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
81	3.574439	128.119.245.12	192.168.7.75	TCP	66	80 → 53161 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
82	3.574439	128.119.245.12	192.168.7.75	TCP	66	443 → 53163 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
83	3.574513	192.168.7.75	128.119.245.12	TCP	54	53162 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
84	3.574551	192.168.7.75	128.119.245.12	TCP	54	53161 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
85	3.574552	192.168.7.75	128.119.245.12	TCP	54	53163 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0

5) The sequence number of the SYNACK segment sent by gaia.cs.umass.edu is “0”. The value of the acknowledgement field in the SYNACK segment is “1”. Gaia.cs.umass.edu determined the acknowledgement value from the client sequence number (mine) and then adds 1 ($0 + 1 = 1$). The item that identifies the SYNACK is the [SYN, ACK] in the info section

69	3.518002	192.168.7.75	128.119.245.12	TCP	78	53161 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=378263841 TSecr=0 SACK_PERM
70	3.518064	192.168.7.75	128.119.245.12	TCP	78	53162 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=753491732 TSecr=0 SACK_PERM
71	3.518170	192.168.7.75	128.119.245.12	TCP	78	53163 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=1010781211 TSecr=0 SACK_PERM
80	3.574439	128.119.245.12	192.168.7.75	TCP	66	80 → 53162 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
81	3.574439	128.119.245.12	192.168.7.75	TCP	66	80 → 53161 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
82	3.574439	128.119.245.12	192.168.7.75	TCP	66	443 → 53163 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
83	3.574513	192.168.7.75	128.119.245.12	TCP	54	53162 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
84	3.574551	192.168.7.75	128.119.245.12	TCP	54	53161 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
85	3.574552	192.168.7.75	128.119.245.12	TCP	54	53163 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0

6) The sequence number of the TCP segment containing the HTTP POST is “1”.

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
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
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
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

<pre> > Frame 4: 619 bytes on wire (4952 bits), 619 bytes captured (4952 bits) > Ethernet II, Src: ActiontecEle_8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysGroup_da:af:73: > Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.119.245.12 0100 = Version: 4 0101 = Header Length: 20 bytes (5) > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT) Total Length: 605 Identification: 0x1e21 (7713) > 010. = Flags: 0x2, Don't fragment ...0 0000 0000 0000 = Fragment Offset: 0 Time to Live: 128 Protocol: TCP (6) Header Checksum: 0xa2e7 [validation disabled] [Header checksum status: Unverified] Source Address: 192.168.1.102 Destination Address: 128.119.245.12 > Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 1, Ack: 1, Len: 565 > Data (565 bytes) </pre>	<pre> 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 yethe 0040 72 65 61 6c 2d 6c 61 62 73 2f 6c 61 62 33 2d 31 real-lab sy[ab3-1 0050 2d 72 65 70 6c 79 2e 68 74 6d 20 48 54 50 2f -reply.htm 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 00 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 m!,appli cation/x 0100 6d 6c 2c 61 70 70 6c 69 63 61 74 69 6f 6e 2f 78 m!,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 m!;q=0.9 ,text/pl 0130 61 69 6e 30 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 6f 69 66 3b 71 3d 30 2e 32 2c 74 65 78 74 2f /gif;q=0.2 ,text/ 0170 63 73 73 2c 2a 2f 2a 3b 71 3d 30 2e 31 0d 0a 41 css,*/; q=0.1 .A 0180 63 63 65 70 74 2d 4c 61 6e 67 75 61 67 65 3a 20 ccept-L nguage: 0190 65 6e 2d 75 72 2c 20 65 6e 3b 71 3d 30 2e 35 30 en-us; e nq;q=0.50 01a0 0d 0a 41 63 63 65 70 74 2d 45 6e 63 6f 64 69 6e -Accept -Encodin 01b0 67 3a 20 6f 7a 69 70 2c 20 64 65 66 6c 61 74 65 g: gzip, deflate 01c0 2c 20 63 6f 6d 70 72 65 73 73 3b 71 3d 30 2e 39 , compre ss;q=0.9 01d0 0d 0a 41 63 63 65 70 74 2d 43 68 61 72 73 65 74 -Accept -Charset 01e0 3a 20 49 53 4f 20 38 30 35 39 2d 31 2c 20 75 74 ; ISO-88 59-1, ut 01f0 66 2d 38 3b 71 3d 30 2e 36 36 2c 20 2a 3b 71 3d f-0;q=0.66 ,*/q= </pre>
---	---

7) The sequence numbers of the first six segments starting with the HTTP POST are (1, 566, 2026, 3486, 4946, 6406). Each of the segments was sent at the following times (0.026477, 0.041737, 0.054026, 0.054690, 0.077405, 0.078157). Each ACK was received as follows respectively (0.053937, 0.077294, 0.124085, 0.124185, 0.169118, 0.217299). RTT for each of the 6 segments is as follows (0.053937-0.026477=0.02746, 0.077294-0.041737 = 0.035557, 0.124085-0.054026=0.070059, 0.124185-0.054690=0.069495, 0.169118-0.077405=0.091713, 0.217299-0.078157=0.139142).

The equation on Page 242 of the textbook for Estimated Round Trip Time is:

EstimatedRTT = (1 – alpha) * EstimatedRTT + (alpha * sample RTT)

The sample value for alpha in the book is 0.125, therefore I will use the same value as per instructions. Estimated RTT for the first segment will equal its RTT, all others are EstimatedRTT.

Therefore Estimated RTT for the segments are (0.027460,

(1-0.125) * 0.027460 + 0.125*0.035557 = 0.028498,

(1-0.125) * 0.028498 + 0.125*0.070059 = 0.033607,

(1-0.125) * 0.033607 + 0.125*0.069495 = 0.037684,

(1-0.125) * 0.037684 + 0.125*0.091713 = 0.044731,

(1-0.125) * 0.044731 + 0.125*0.139142 = 0.056836)

8)

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
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
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
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
19	0.305813	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=10473 Ack=1 Win=17520 Len=1460
20	0.306692	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=11933 Ack=1 Win=17520 Len=1460
21	0.307571	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=13393 Ack=1 Win=17520 Len=1460
22	0.308699	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=14853 Ack=1 Win=17520 Len=1460
23	0.309553	192.168.1.102	128.119.245.12	TCP	946	1161 → 80 [PSH, ACK] Seq=16313 Ack=1 Win=17520 Len=892
24	0.356437	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=10473 Win=26280 Len=0
25	0.400164	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=11933 Win=29200 Len=0
26	0.448613	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=13393 Win=32120 Len=0

The lengths of the first six segments are:
(565, 1460, 1460, 1460, 1460, 1460)

9)

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
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
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
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
19	0.305813	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=10473 Ack=1 Win=17520 Len=1460
20	0.306692	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=11933 Ack=1 Win=17520 Len=1460
21	0.307571	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=13393 Ack=1 Win=17520 Len=1460
22	0.308699	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=14853 Ack=1 Win=17520 Len=1460
23	0.309553	192.168.1.102	128.119.245.12	TCP	946	1161 → 80 [PSH, ACK] Seq=16313 Ack=1 Win=17520 Len=892
24	0.356437	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=10473 Win=26280 Len=0
25	0.400164	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=11933 Win=29200 Len=0
26	0.448613	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=13393 Win=32120 Len=0

The initial window size is initially advertised as 5840 bytes in packet no 2. This 5840 looks to be the minimum amount of available buffer space advertised.

It does not seem that there is throttling due to a lack of available buffer space. The Advertised window space starts at 5840 bytes and steadily increases to a maximum of 62780 over time. There don't seem to be any instances where the window size drops to a very low value or zero, and the client doesn't seem to be affected as they continue to send full sized segments (1460 bytes) all the way through the trace. There also don't seem to be any long pauses in the data transmission.

10) There seem to be no duplicate sequence numbers in any of the transmissions, there also seem to be no duplicate ACKs that might seem like packet loss. I have also checked

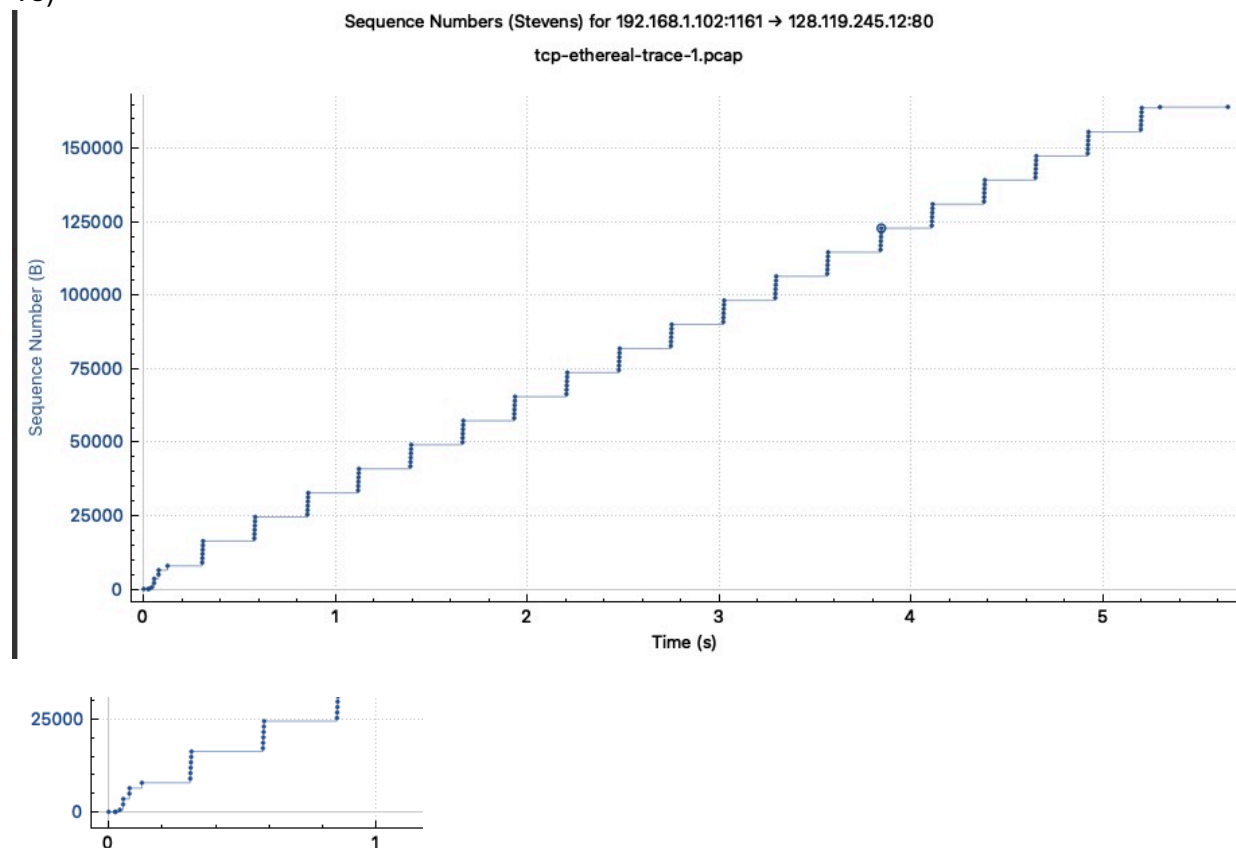
Wireshark for retransmissions at Analyze-> Expert information, and there seem to be no retransmissions there. So in short, I checked for Duplicate Sequence numbers, Duplicate ACK's, and Wireshark's Expert Analysis for any flags.

11) The receiver typically acknowledges 1460 bytes in an ACK, which represents the MSS (maximum segment size) of the sender's packets. I don't see any cases where the receiver is ACKing every other received segment.

12) Packet 202 shows that the seq ends with a total of 164091 bytes of data sent. The total time of the transfer would be calculated by taking the time at packet 202 of 5.45583 and subtracting the start time in packet 4 of 0.026477. So total time would be $5.45583 - 0.026477 = 5.429353$ seconds. To calculate throughput you would use the formula:

Throughput = total data transferred / total time
= $164091 / 5.429353$
= approx 30223 bytes/second

13)



Given the graph and the zoom in of the graph, the slow start phase starts at 0 seconds and seems to end just before 0.1 seconds. This is where the sequence

numbers increase in smaller sizes initially and then end up increasing quickly. The congestion window increases exponentially during this phase. When slow start ends (just before 0.1 seconds) congestion avoidance takes over.

The biggest differences from the idealized behavior of TCP that is in the material is that all examples seemed to have a situation where there was packet loss and then the congestion window was reset. There doesn't seem to be a reset at all and potentially another slow start. The graph itself also seem very linear, which probably shows that the MSS of the client doesn't overwhelm the server at all.