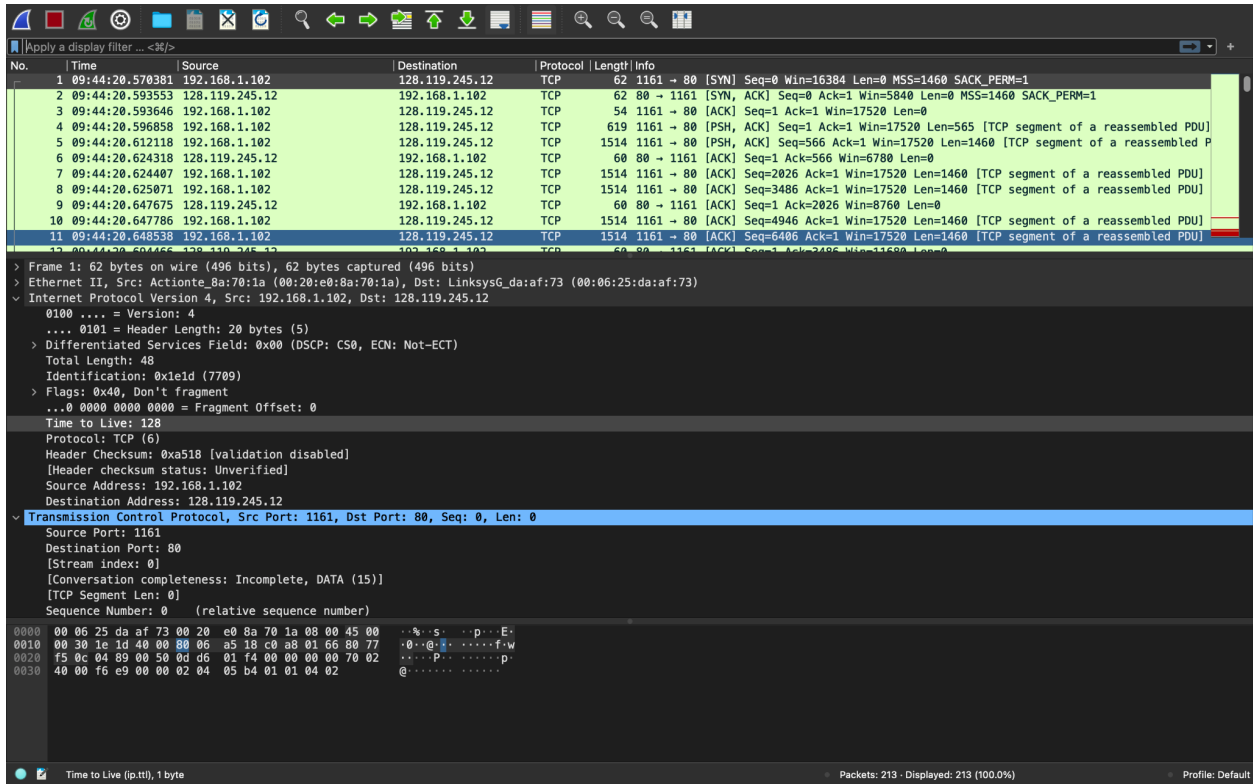


Foundation of Computer Networks - Wireshark Project

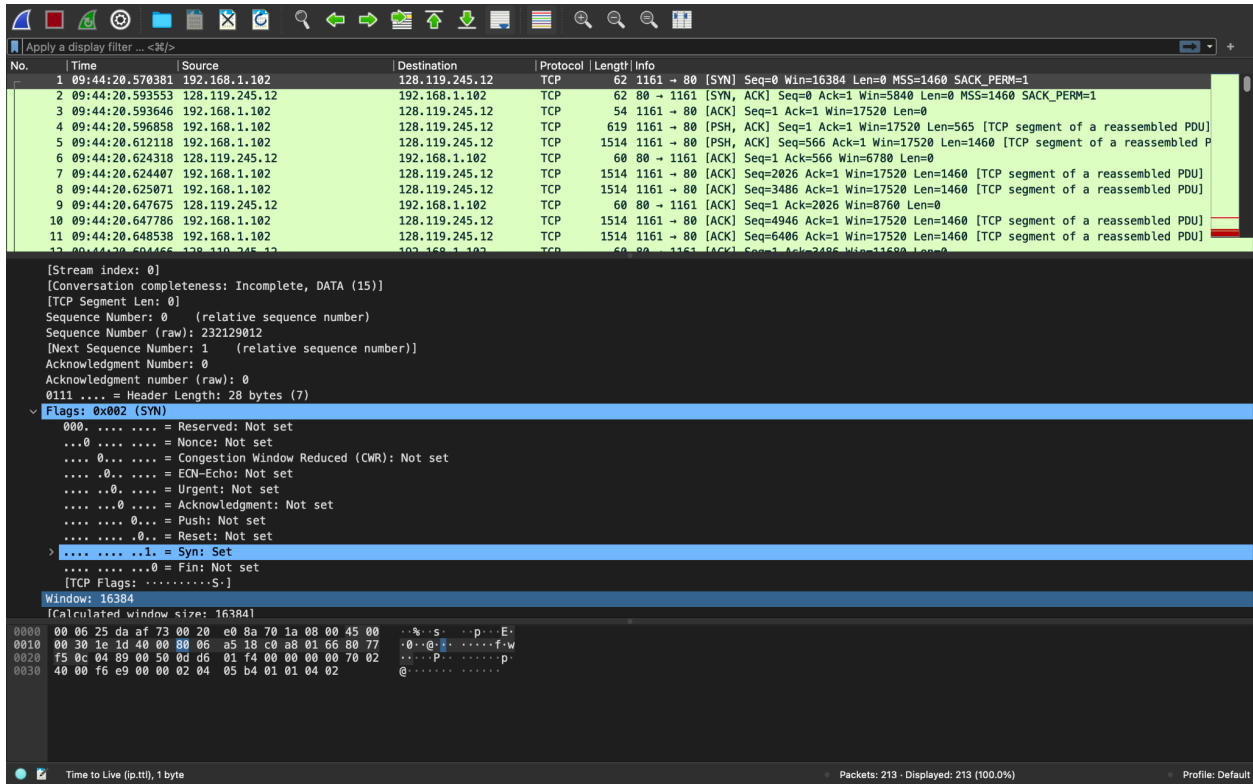
Bhavin Oza (bo2115)



- 1). client computer (Source) IP address: 192.168.1.102, Port: 1161
- 2). gaia.cs.umass.edu (Destination) IP address: 128.119.245.12, Port: 80
- 3). Using trace file provided in wireshark labs

Foundation of Computer Networks - Wireshark Project

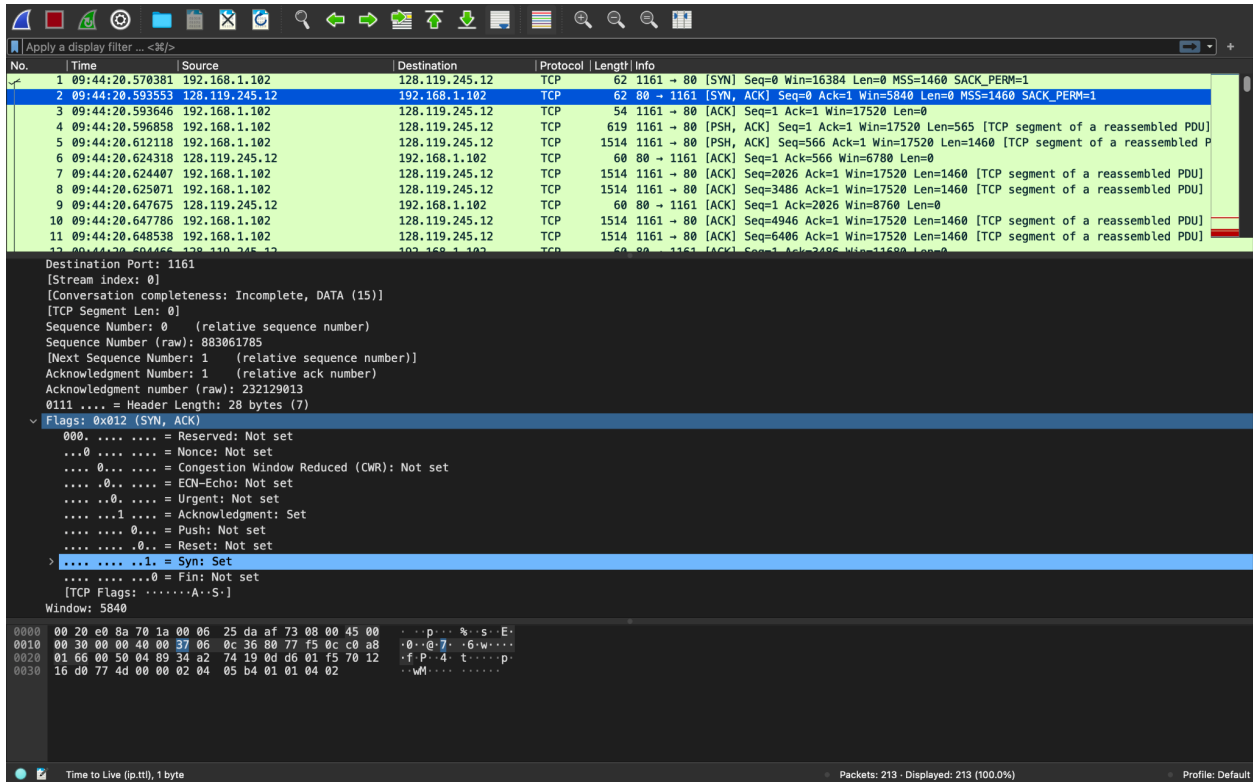
Bhavin Oza (bo2115)



4). Sequence number of the TCP SYN segment is 0, since it is used to initiate the TCP connection between client and server. The SYN flag is set to 1 and it is shown in the flag section, thus signifying the segment is SYN segment

Foundation of Computer Networks - Wireshark Project

Bhavin Oza (bo2115)



5).

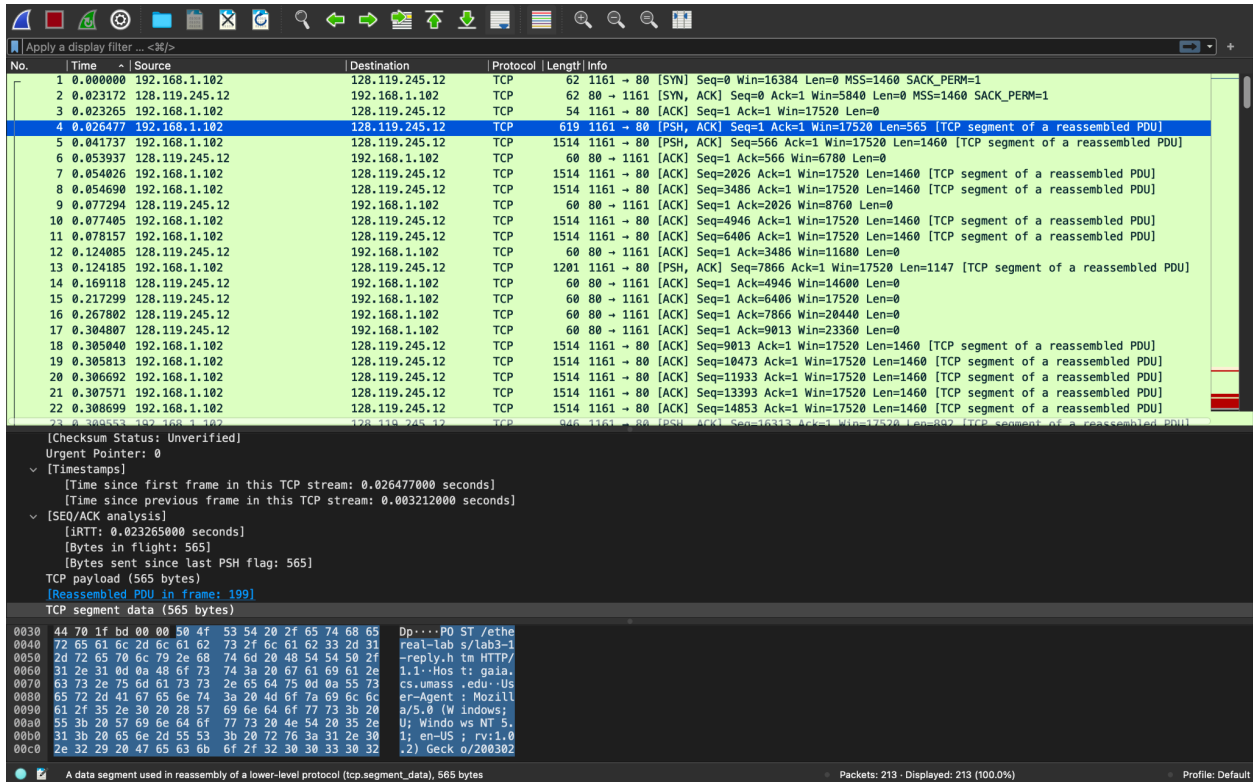
In SYNACK, SYN sent by server is 0 and ACK sent is 1

The server can determine the value of ACK by adding 1 to the initial sequence of SYN received from the client computer.

Both SYN flag and ACK flag in the segment are set to be 1, thus signifying the segment is a SYN_ACK segment.

Foundation of Computer Networks - Wireshark Project

Bhavin Oza (bo2115)



6). SEQ number of TCP segment (Segment no. 4) containing POST command is 1

7). The 6 segments starting Segment no.4 with their SEQ and ACK are as follows:

Segment no	SEQ	ACK Segment no	Sent time	ACK Received Time	RTT (Seconds)
4	1	6	0.026477	0.059337	0.02746
5	566	9	0.041737	0.077294	0.035557
7	2026	12	0.054026	0.124085	0.070059
8	3486	14	0.054690	0.169118	0.11443
10	4946	15	0.077405	0.217299	0.13989
11	6406	16	0.078157	0.267802	0.18964

8). The length for each 6 TCP segment:

First one: 565 bytes

Length of each of the rest of the segments: 1460 bytes

Foundation of Computer Networks - Wireshark Project

Bhavin Oza (bo2115)

Apply a display filter: ... <36/>

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]
22	0.308699	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=14853 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
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 [TCP segment of a reassembled PDU]

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
[Timestamps]
[SEQ/ACK analysis]

0000 00 20 e0 8a 70 1a 00 06 25 da af 73 08 00 45 00 ... p...%s..E
0010 00 30 00 00 40 00 37 06 0c 36 80 77 f5 0c c0 a8 0 @.7. 6w...
0020 01 66 00 50 04 89 34 a2 74 19 00 d6 01 f5 70 12 f-P-4: t....p
0030 10 d0 77 4d 00 00 02 04 05 b4 01 01 04 02 W.....

The window size value from the TCP header (tcp.window_size_value), 2 bytes

Packets: 213 - Displayed: 213 (100.0%)

Profile: Default

Apply a display filter: ... <36/>

No.	Time	Source	Destination	Protocol	Length	Info
37	0.672796	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=20125 Win=43800 Len=0
38	0.730684	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=21585 Win=46720 Len=0
39	0.772990	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=23045 Win=49640 Len=0
40	0.820622	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=24505 Win=52560 Len=0
41	0.853186	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=25397 Win=52560 Len=0
42	0.853405	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=25397 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
43	0.854076	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=26857 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
44	0.855036	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=28317 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
45	0.855878	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=29777 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
46	0.856802	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=31237 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
47	0.857683	192.168.1.102	128.119.245.12	TCP	946	1161 → 80 [PSH, ACK] Seq=32697 Ack=1 Win=17520 Len=892 [TCP segment of a reassembled PDU]
48	0.899423	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=26857 Win=55480 Len=0
49	0.949545	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=28317 Win=58400 Len=0
50	0.994715	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=29777 Win=61320 Len=0
51	1.039500	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=31237 Win=62780 Len=0
52	1.117097	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=1 Ack=33589 Win=62780 Len=0
53	1.117333	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=33589 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
54	1.118133	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=35049 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
55	1.119029	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=36509 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
56	1.119858	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=37969 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
57	1.120902	192.168.1.102	128.119.245.12	TCP	1514	1161 → 80 [ACK] Seq=39429 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
58	1.121891	192.168.1.102	128.119.245.12	TCP	946	1161 → 80 [PSH, ACK] Seq=40889 Ack=1 Win=17520 Len=892 [TCP segment of a reassembled PDU]

Acknowledgment Number: 31237 (relative ack number)
Acknowledgment number (raw): 232160249
0101 ... = Header Length: 20 bytes (5)
Flags: 0x010 (ACK)
Window: 62780
[Calculated window size: 62780]
[Window size scaling factor: -2 (no window scaling used)]
Checksum: 0xbab0 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
[Timestamps]
[SEQ/ACK analysis]

0000 00 20 e0 8a 70 1a 00 06 25 da af 73 08 00 45 00 ... p...%s..E
0010 00 28 58 88 40 00 37 06 b3 b5 80 77 f5 0c c0 a8 (X@.7. 6w...
0020 01 66 00 50 04 89 34 a2 74 1a 00 d6 7b f9 50 10 f-P-4: t....p
0030 f5 3c 4b a0 00 00 11 0b 0b 00 cd fd K.....

The window size value from the TCP header (tcp.window_size_value), 2 bytes

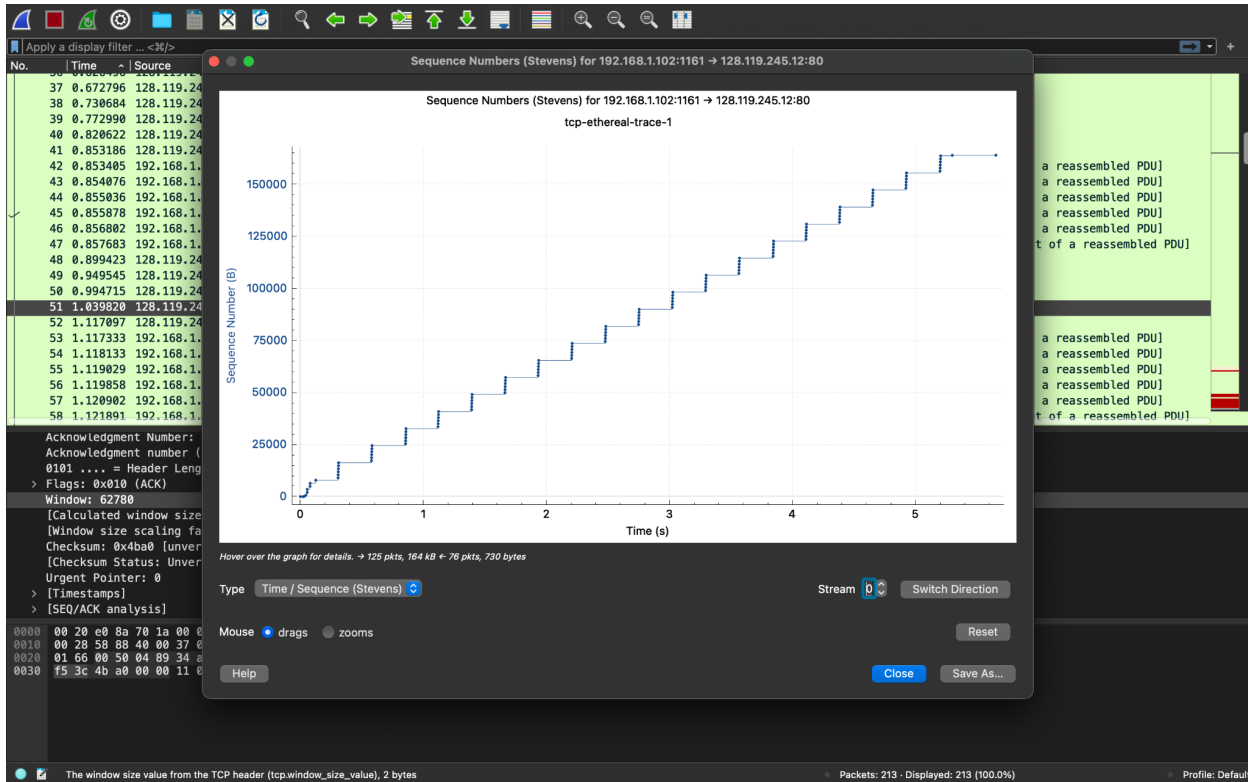
Packets: 213 - Displayed: 213 (100.0%)

Profile: Default

9). The minimum amount of buffer space is 5840 bytes.
The receiver window grows steadily, until the max buffer size of 62780 bytes is reached.
No, the sender is never throttled.

Foundation of Computer Networks - Wireshark Project

Bhavin Oza (bo2115)



10). As we can see in the graph above, the graph increases with sequence number. If any sequence number was retransmitted, then the graph would have been different as the sequence number for this retransmitted segment would be smaller than its consecutive sequences. Thus, we can say there was no retransmission.

11). We can check the total data acknowledged by the server, by calculating the difference of two consecutive ACK's. Thus, we find there are some instances where the server acknowledges 2920 bytes, while in some instances the server acknowledges 1460 bytes, and at times 566 bytes and 1467 bytes.

12). Throughput for the TCP Connection: 0.302 MBps

Total data transmitted = ACK for segment 202 (last) - SEQ for segment 4 (first)
= 164091 - 1 = 164090 bytes

Total transmission time = time for ACK for segment 202 (last) - time for segment 4 (first)
= 5.455830 - 0.026477
= 5.4294 seconds

Throughput = Total data transmitted / Total transmission time
= 164090 / 5.4294
= 0.302 MBps

13). TCP's slow start phase begins when the HTTP POST segment is sent out. Sender's congestion window size is used for identifying the congestion avoidance

Foundation of Computer Networks - Wireshark Project
Bhavin Oza (bo2115)

phase. To find the beginning of the congestion avoidance phase, we need to see how the congestion window size changes on arrival of ACK. On ACK arrival, if congestion window size increases by one maximum size of segment, the sender stays in the slow start phase. In congestion avoidance phase congestion window size is increasing by $1 /$ (current size of congestion window). Thus, these changes in the congestion window size, can help in pointing out the states of the client.

14). Here, if the sender can send data, sometimes there is not data available. Thus, we can say the behavior is dependent on the applications. However, in text the behavior of TCP makes an assumption that sender transmit more data. Thus, there is a possibility of traffic congestion. To tackle this problem, senders are expected to follow a feedback algorithm known as Additive Increase Multiplicative Decrease algorithm, so senders window size can be decreased.