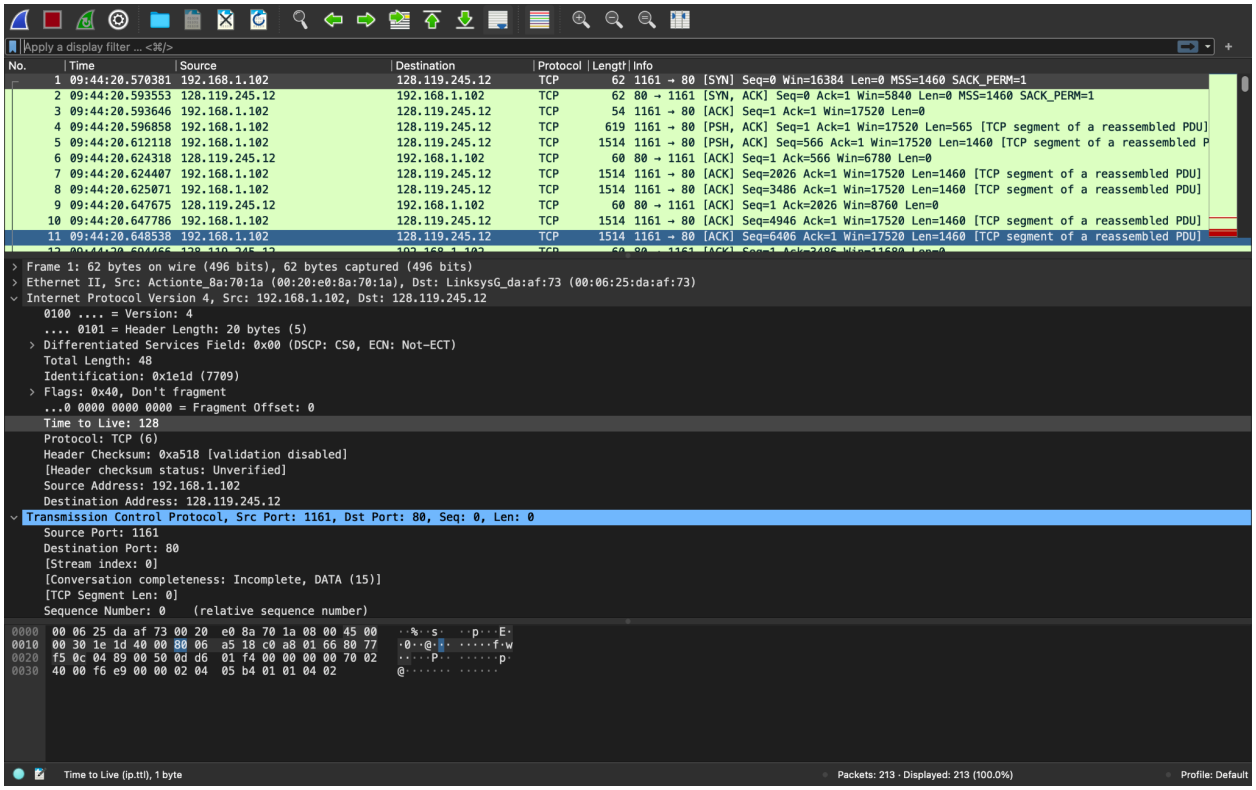
Foundation of Computer Networks - Wireshark Project

Bhavin Oza (bo2115)

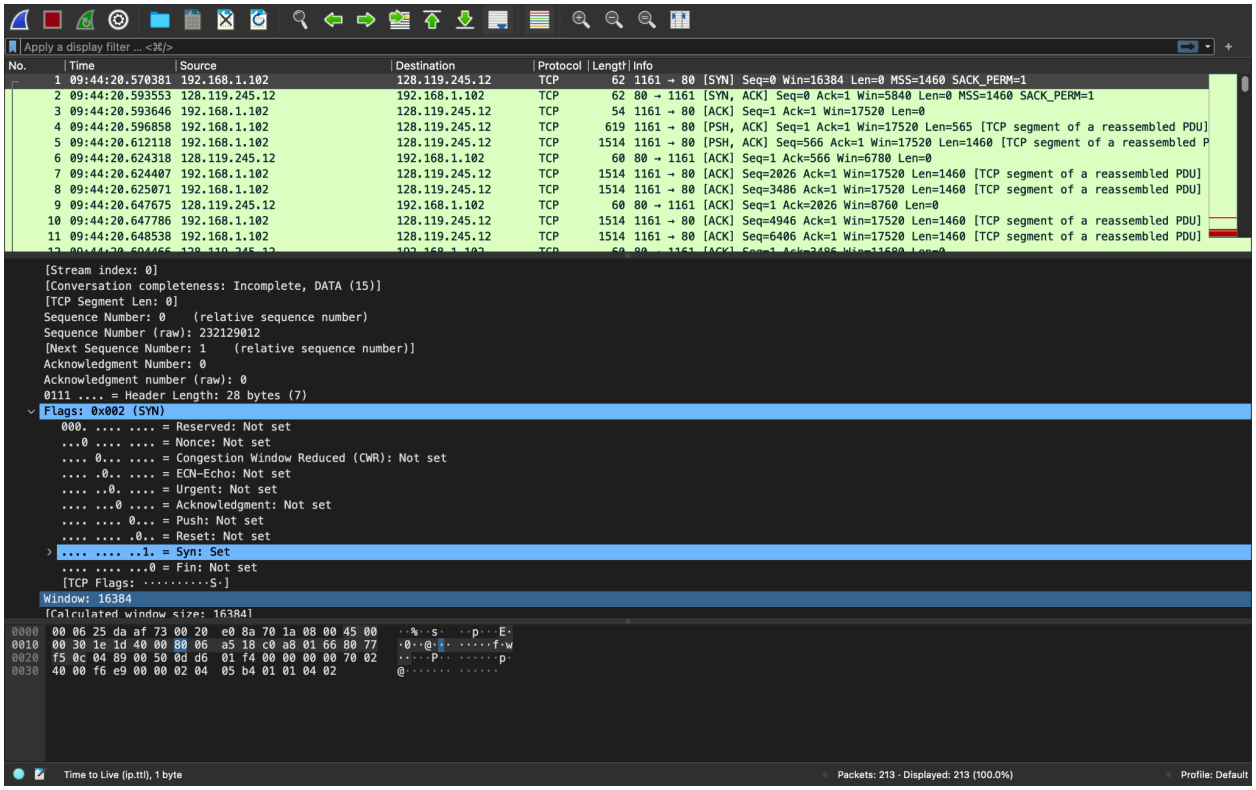


1). client computer (Source) IP address: 192.168.1.102, Port: 1161

2). gaia.cs.umass.edu (Destination) IP address: 128.119.245.12, Port: 80

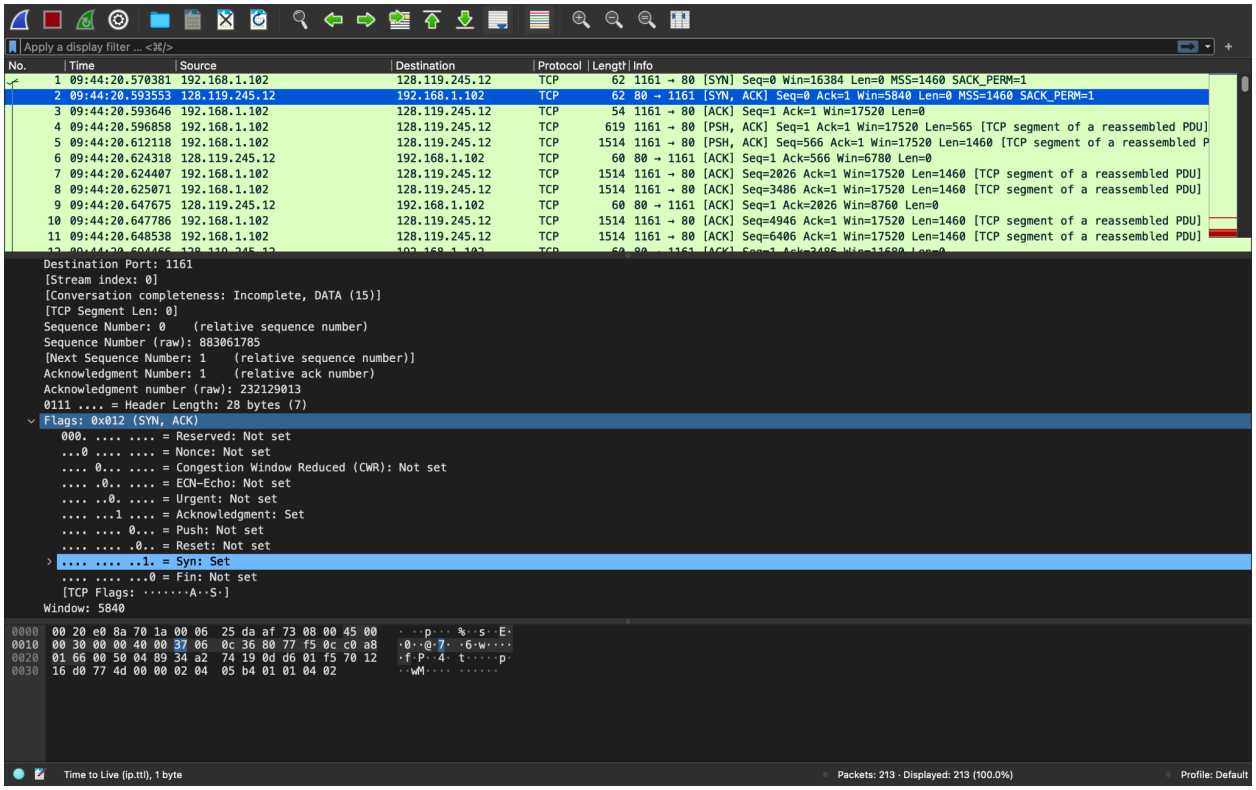
3). Using trace file provided in wireshark labs

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)



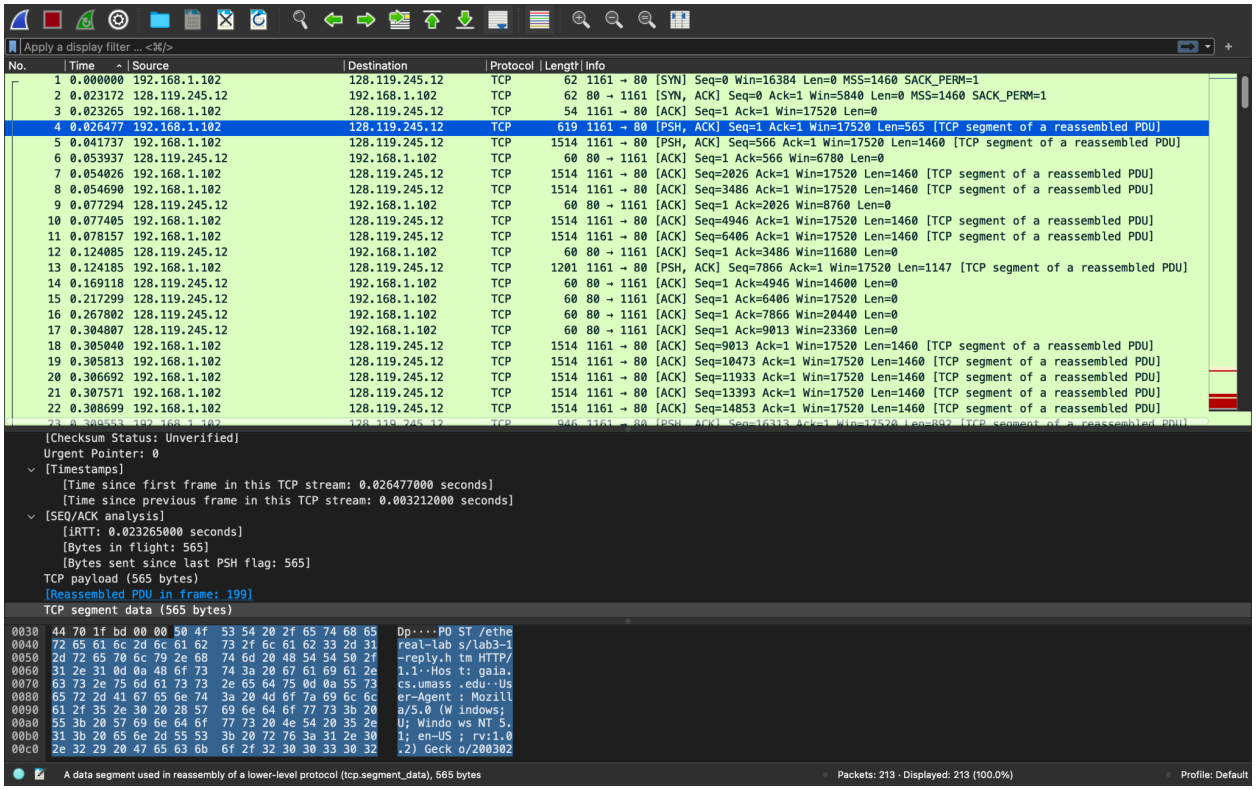
5).

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)



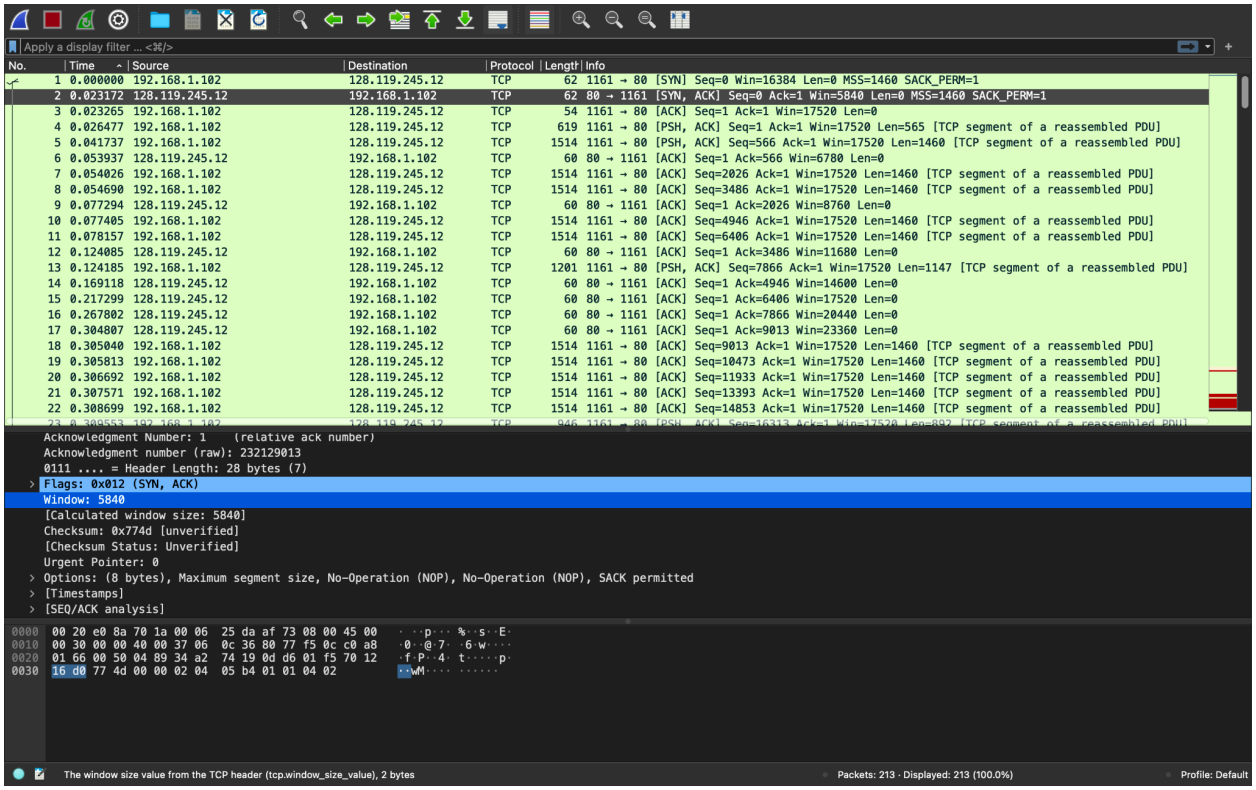
6). SEQ number of TCP segment (Segment no. 4) containing POST command is 1

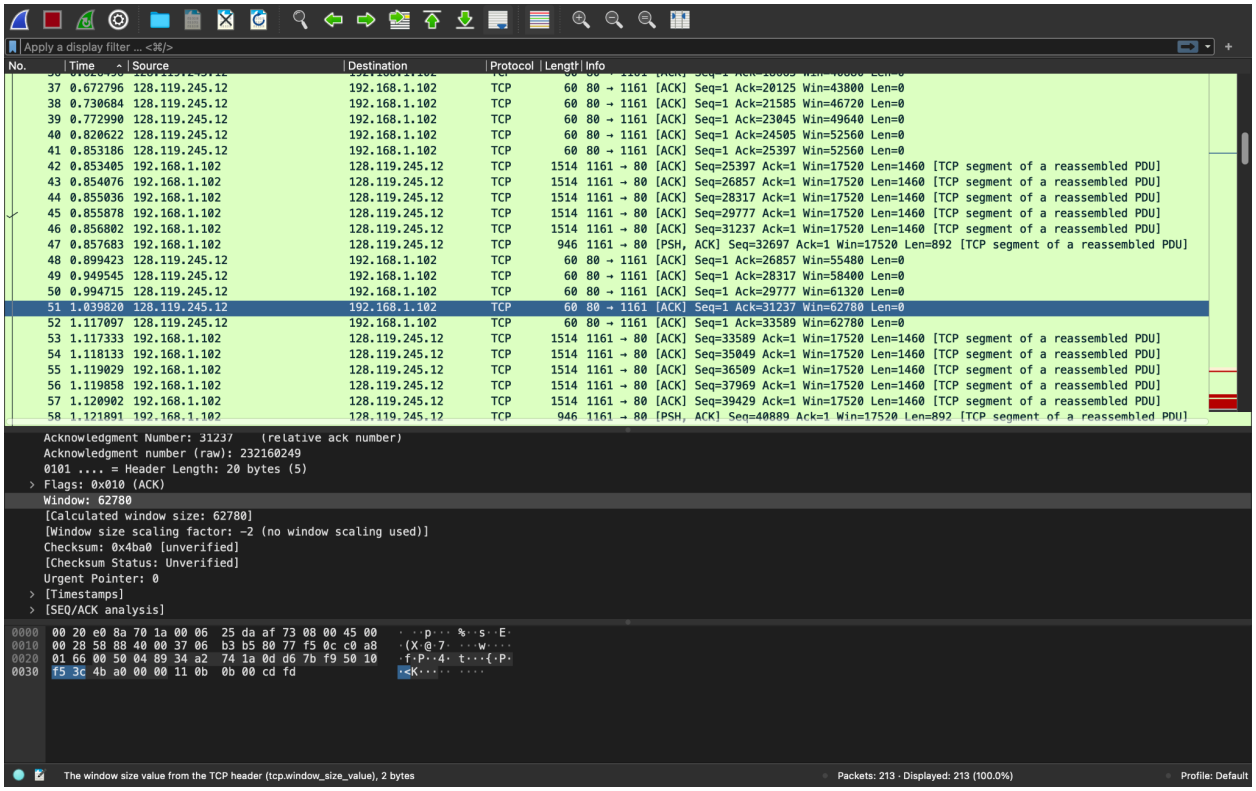
7). The 6 segments starting Segment no.4 with their SEQ and ACK are as follows:

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

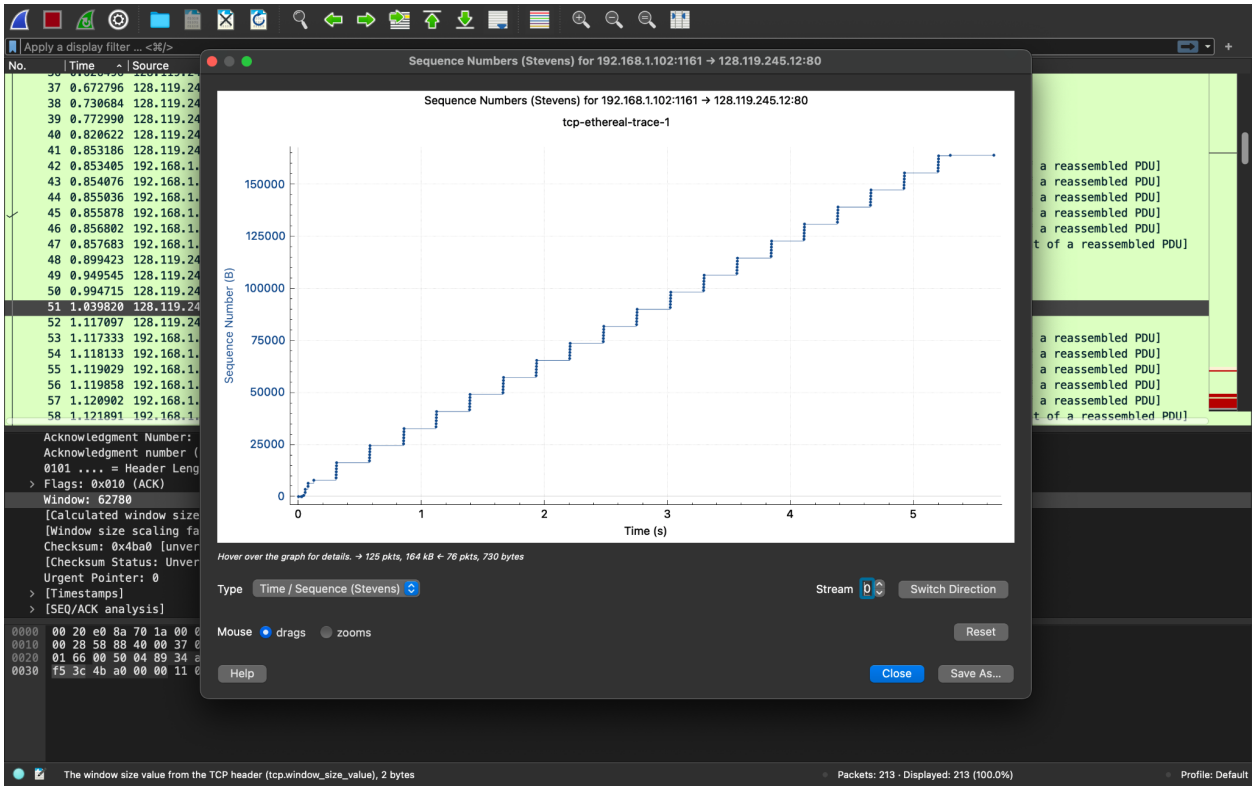
Foundation of Computer Networks - Wireshark Project
Bhavin Oza (bo2115)





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

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.