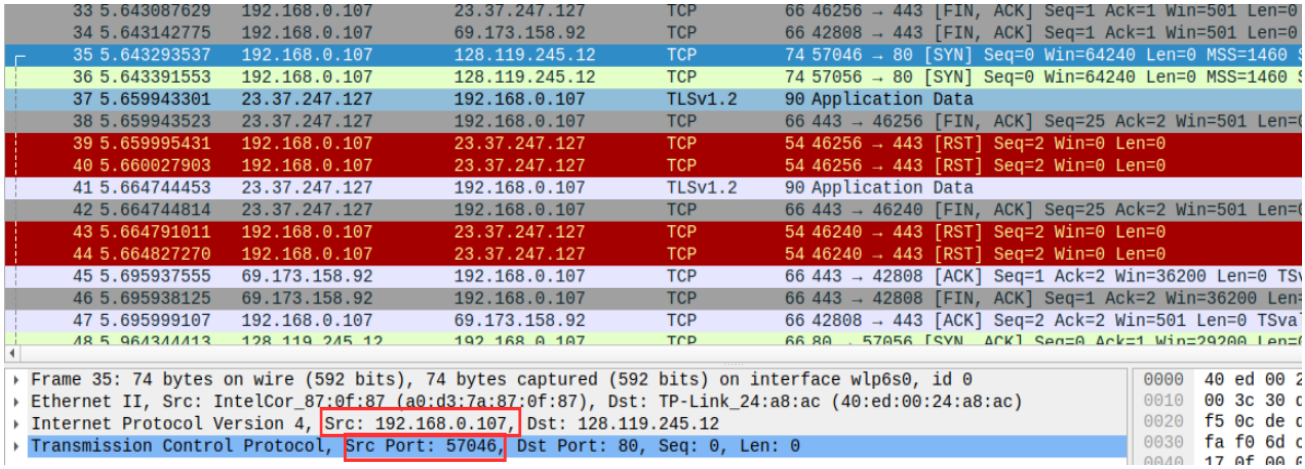
PART-A

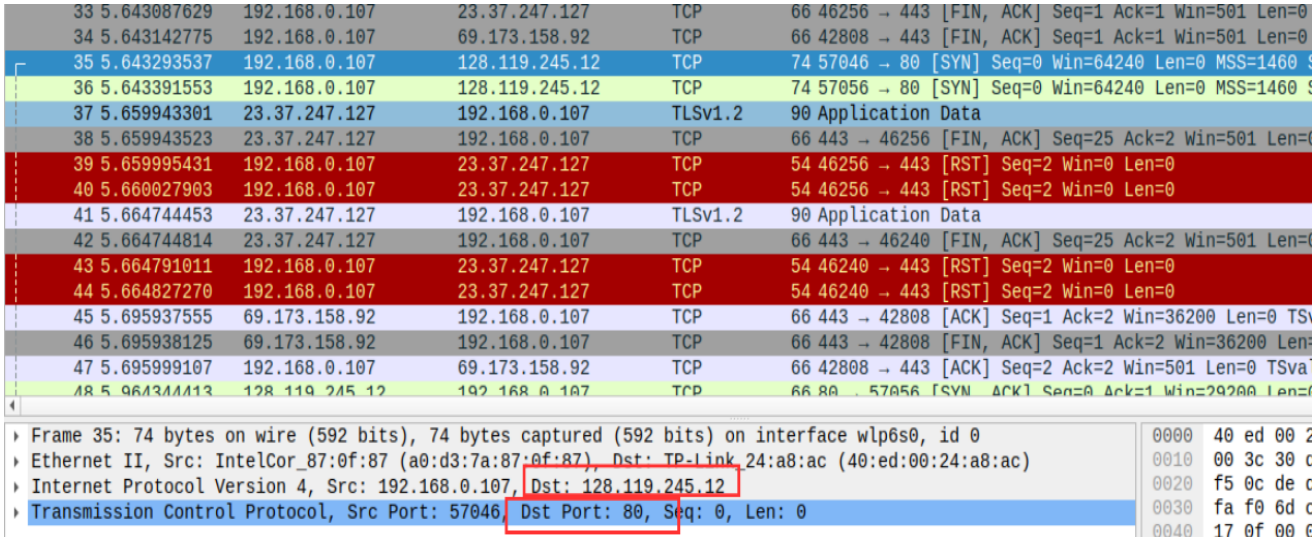
1.

The IP address and TCP port number used by the client computer (source) that is transferring the alice.txt file to gaia.cs.umass.edu is **192.168.0.107** and **57056** respectively. Given below the figure shows it.



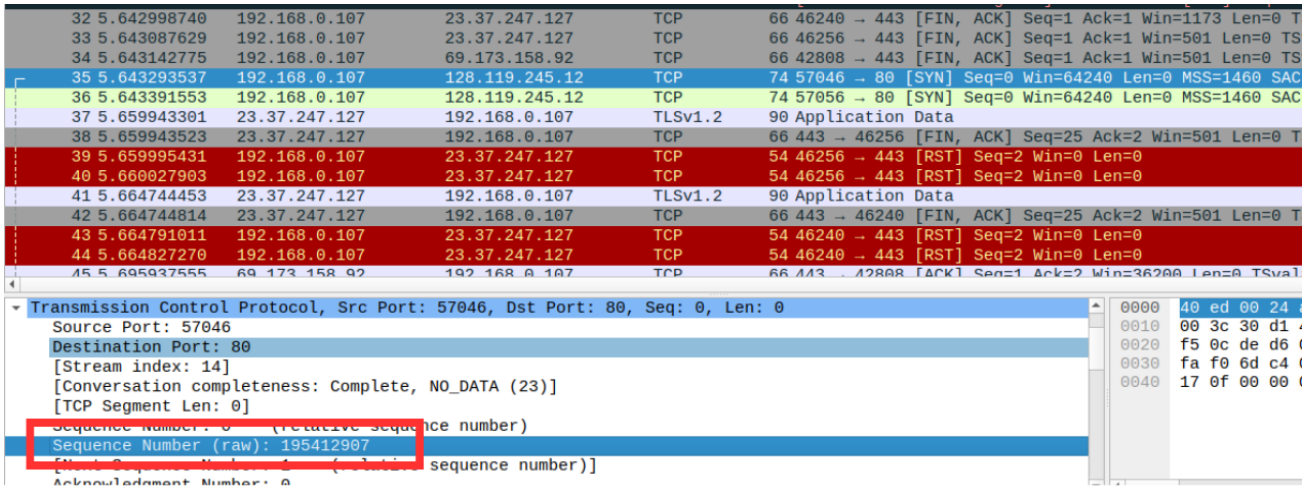
2.

The IP address of gaia.cs.umass.edu is **128.119.245.12** and the port number it is sending and receiving TCP segments for this connection is **80**. Given below the figure shows it.

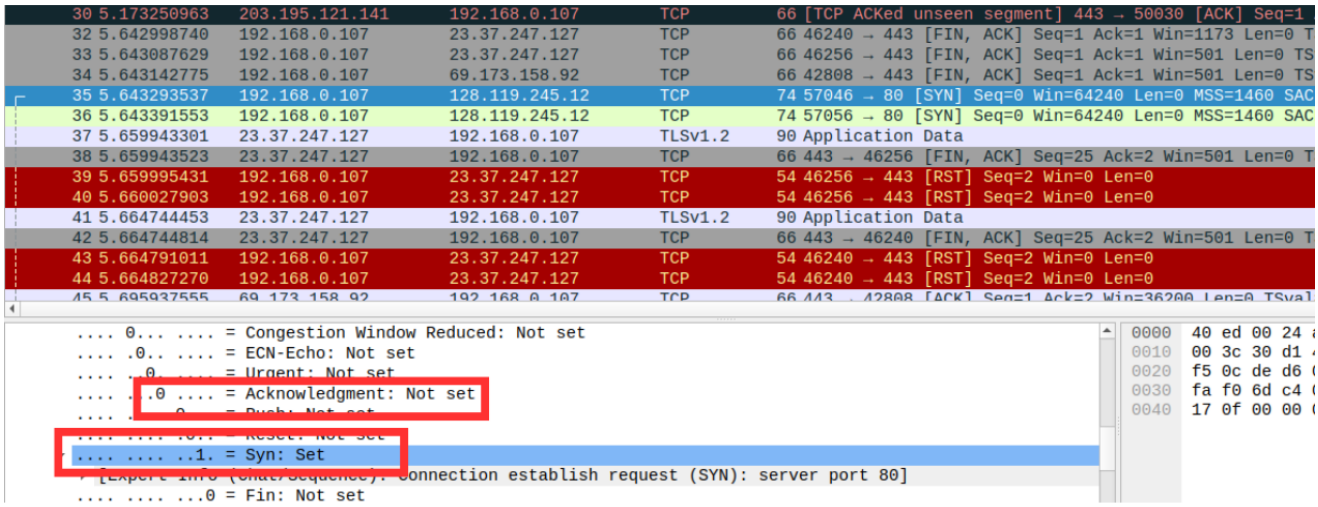


3.

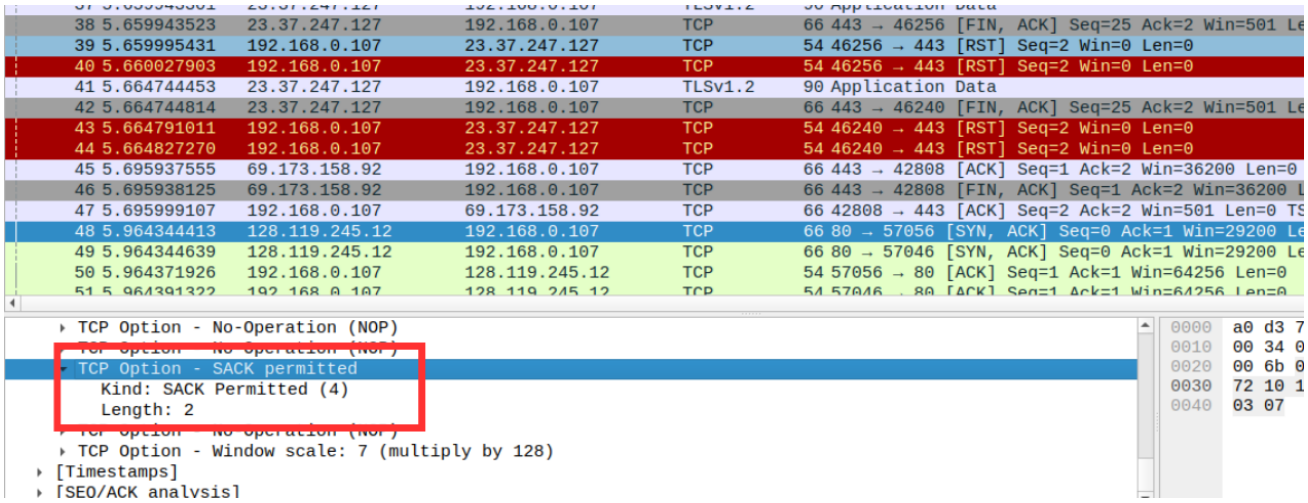
The sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu is **195412907**. Given below the figure shows it.



The **SYN bit is set to 1** inside the flags of the TCP header and tells us that the selected TCP segment identifies the SYN segment. Given below the figure shows it.



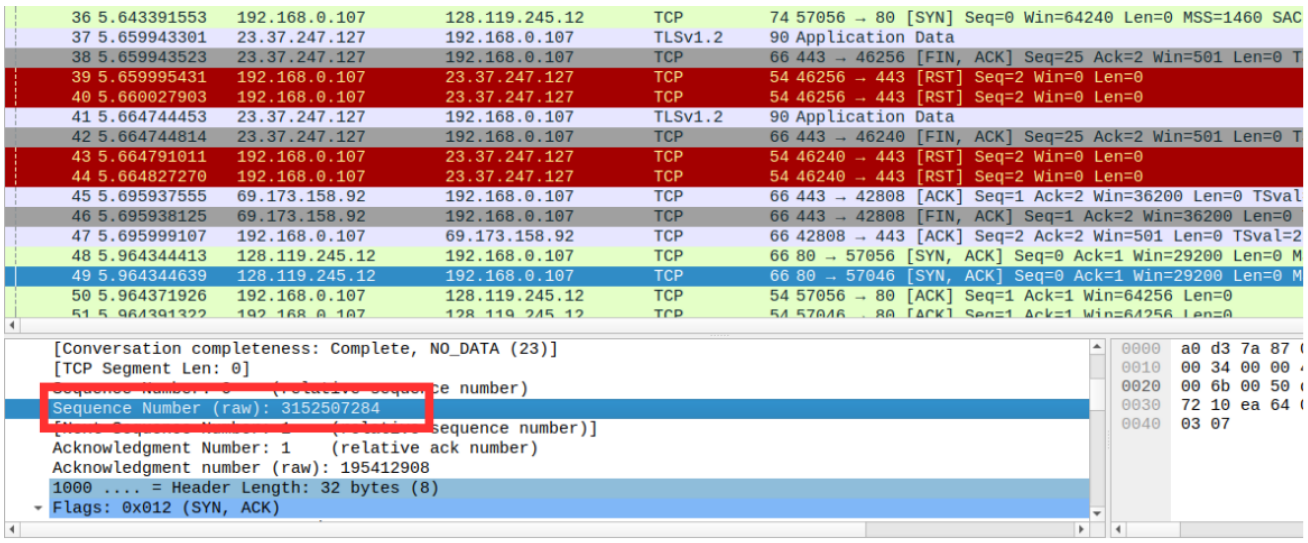
Yes, the TCP receiver in this session will be able to **use selective acknowledgements** as the SACK Permitted option is present in the TCP options section. Given below the figure shows it.



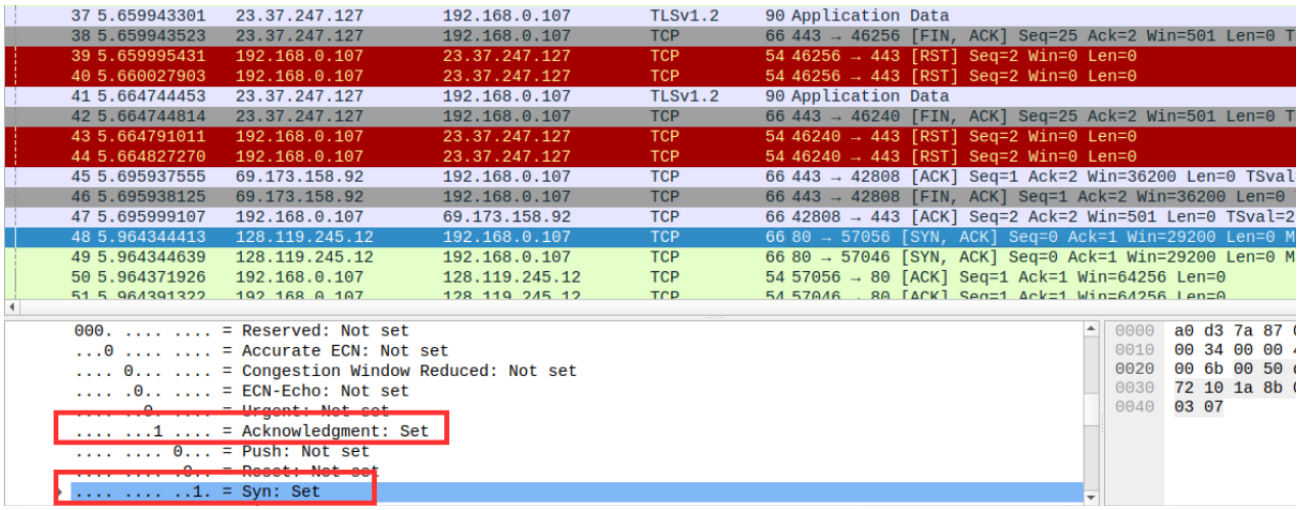
Note: My system sends two SYN requests, gets two SYNACK replies and then again sends two ACK to “gaia.cs.umass.edu”.

4.

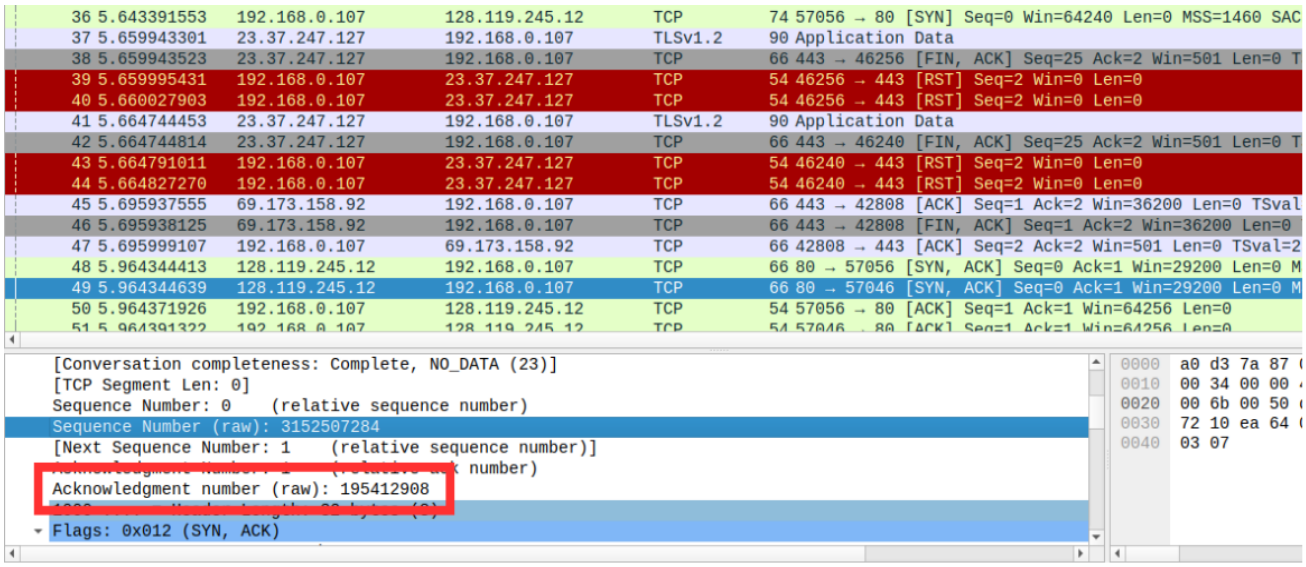
The sequence number of the TCP SYNACK segment that was sent by gaia.cs.umass.edu to the client computer in reply to the SYN is **3152507284**. Given below the figure shows it.



The **SYN and acknowledgment bit is set to 1** inside the flags of the TCP header and tells us that the selected TCP segment identifies the SYNACK segment. Given below the figure shows it.



The value of the Acknowledgement field in the SYNACK segment is **195412908.** Given below the figure shows it.



The value of the Acknowledgement field in SYNACK by gaia.cs.umass.edu is determined by the sequence number of the SYN segment received from my computer. It increments the value by 1 which indicates the next sequence number from my computer.

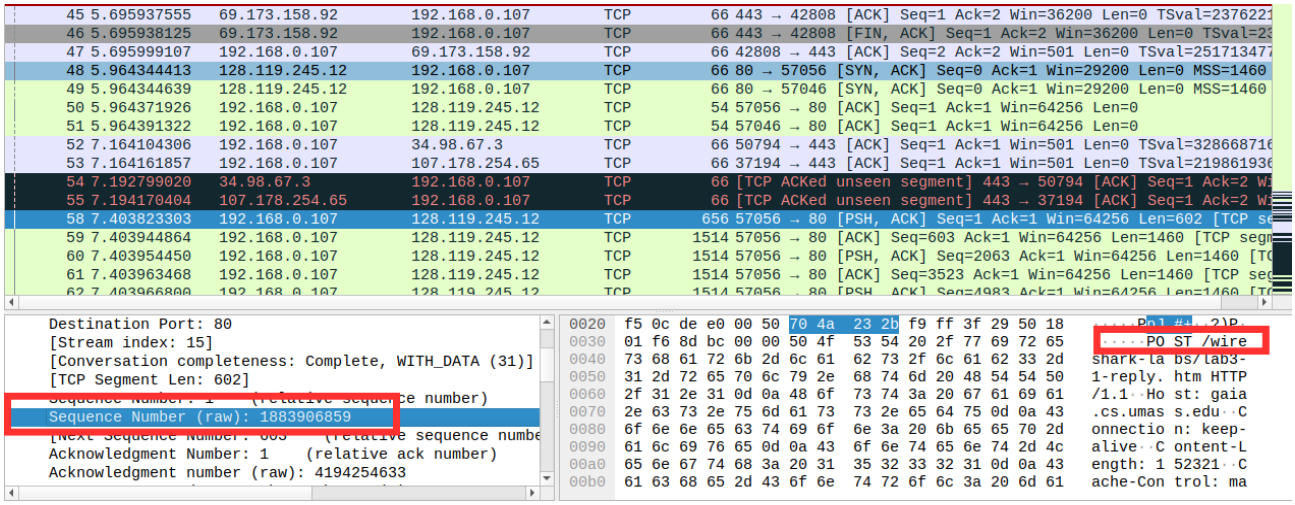
195412907 + 1 = 195412908

Note: My system sends two SYN requests, gets two SYNACK replies and then again sends two ACK to “gaia.cs.umass.edu”. The second set also follows the same approach

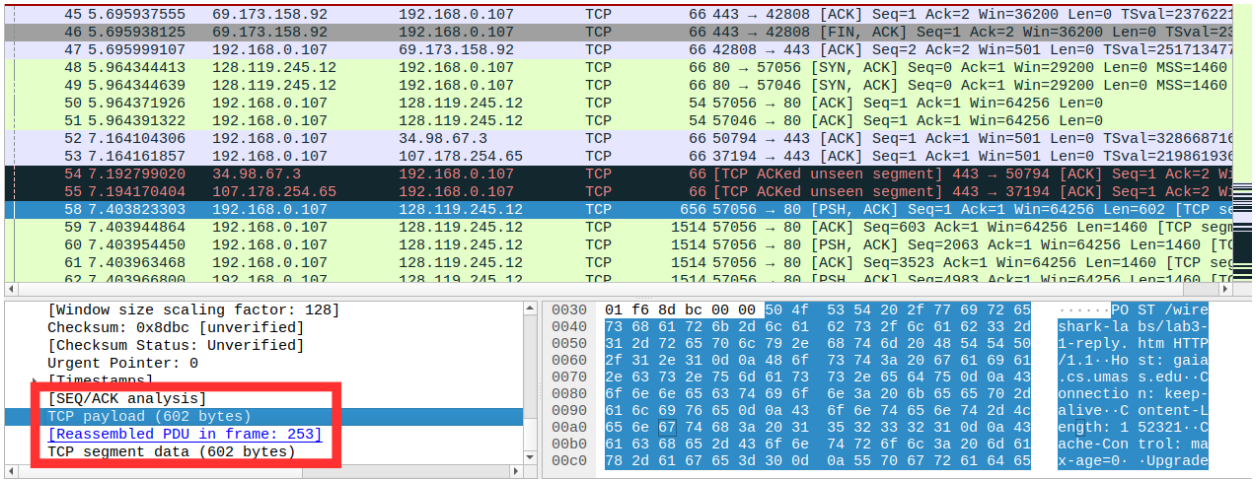
of increments the value by 1 of its sequence number to get the acknowledgement number.

5.

The sequence number of the TCP segment containing the header of the HTTP POST command is **1883906859**. Given below the figure shows it.



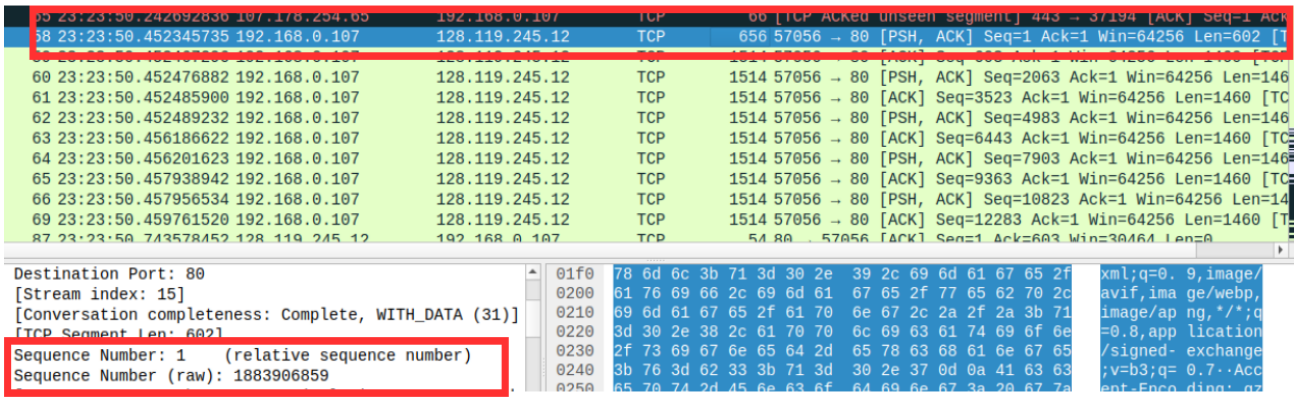
The payload (data) field of this TCP segment contains **602 Bytes**. Given below the figure shows it.



No, all of the data in the transferred file alice.txt fit into this single segment. The remaining bytes are transferred in next segments.

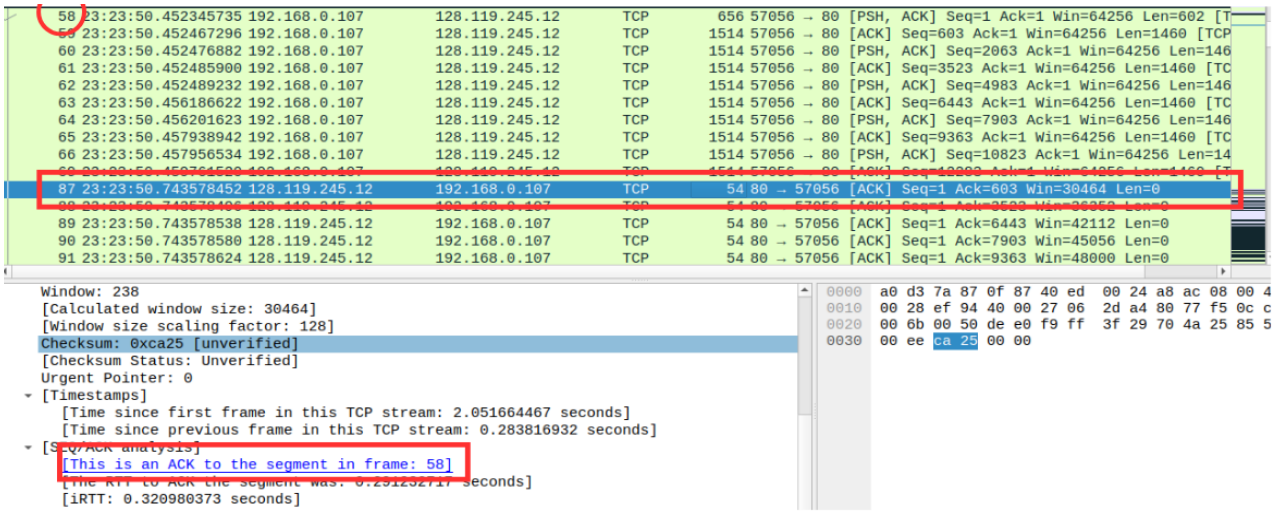
6.

The time at which the first segment was sent in the data-transfer part of the TCP connection is “**23:23:50.452345735**”.Given below the figure shows it.

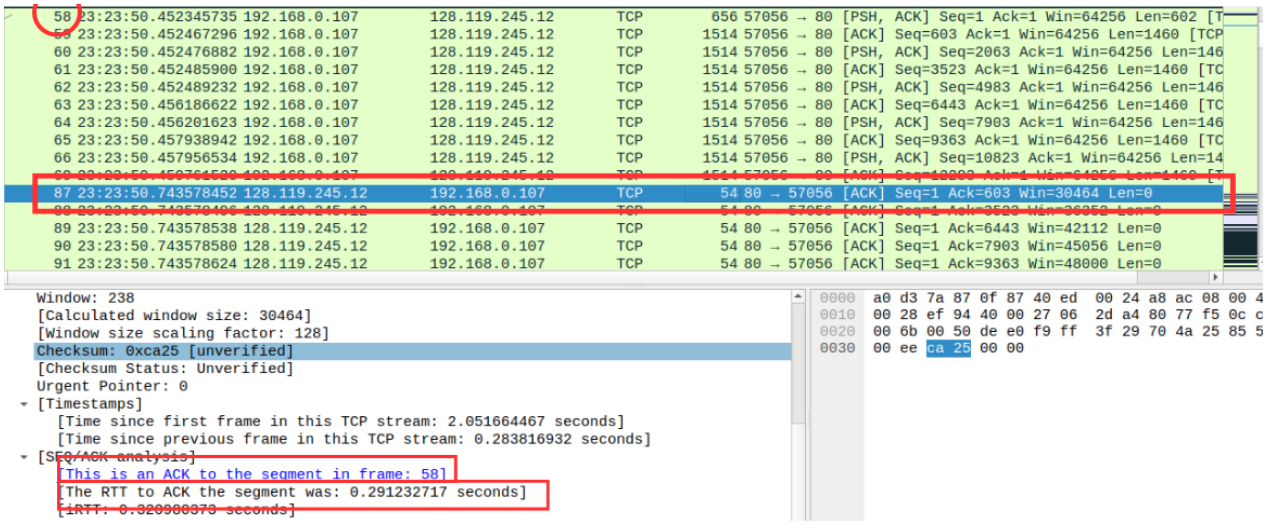


The first packet is of 602 bytes and the next expected relative sequence number is 602+1 = 603.

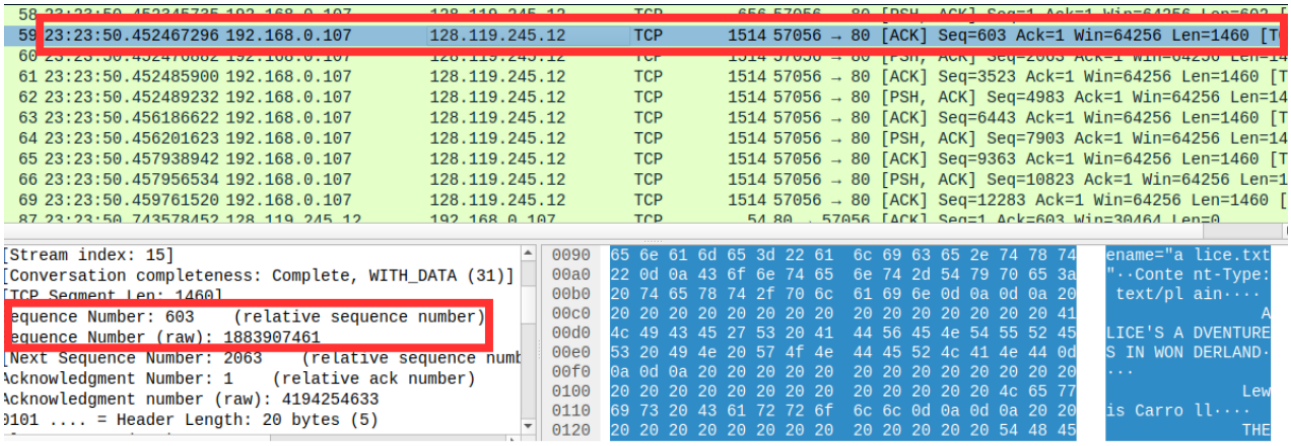
The time at which the ACK of the first segment was received in the data-transfer part of the TCP connection is “**23:23:50.743578452**”. Given below the figure shows it.



The RTT of the first data-containing segment is **0.291232717 seconds** . Given below the figure shows it.

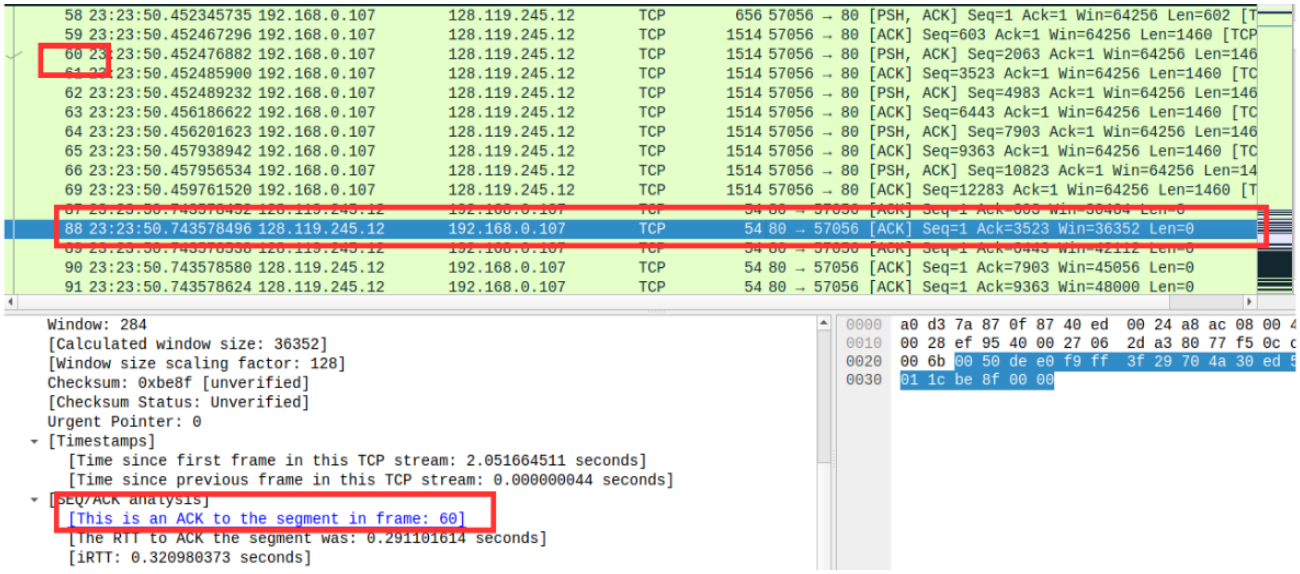


The time at which the second segment was sent in the data-transfer part of the TCP connection is "**23:23:50.452467296**". Given below the figure shows it.

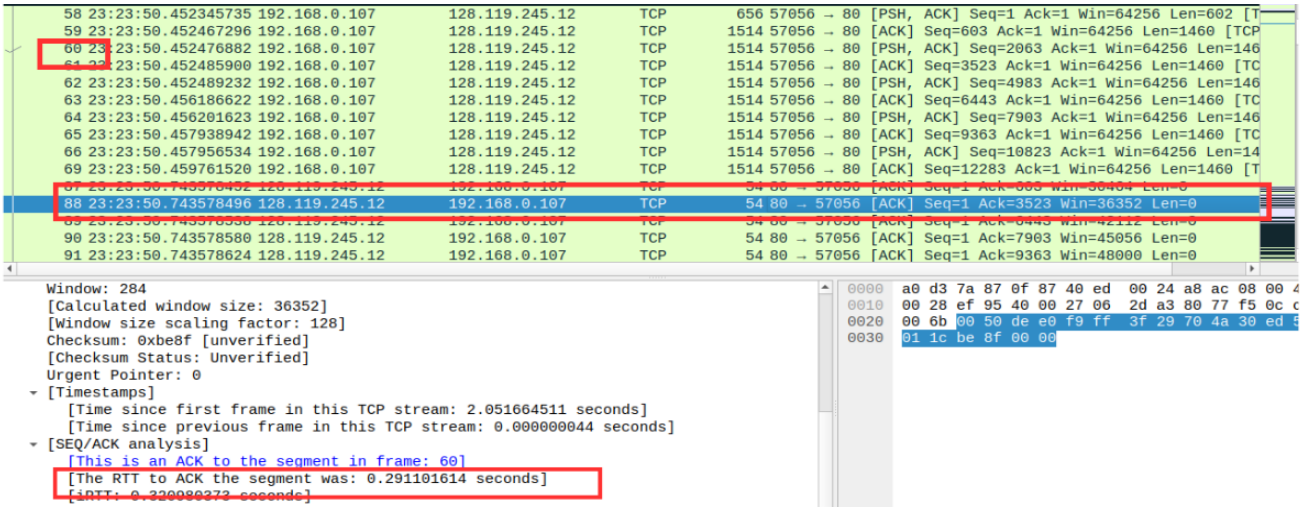


The time at which the ACK of the second segment was received in the data-transfer part of the TCP connection is "**23:23:50.743578496**".

Please see that the acknowledgement of the second TCP segment is not received. I tried to filter based on its acknowledgement number using "**tcp.ack==acknowledgment number**" but didn't find it. Hence, using the acknowledgment of the third segment considering cumulative acknowledgement. Given below the figure shows it.



The RTT of the second data-containing segment is **0.291101614 seconds**. Given below the figure shows it.

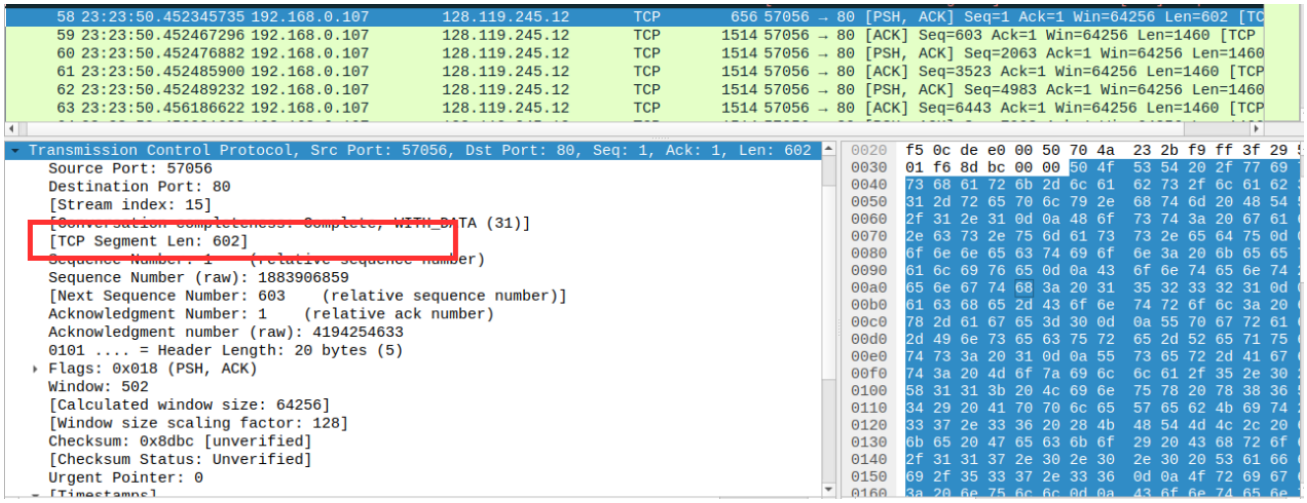


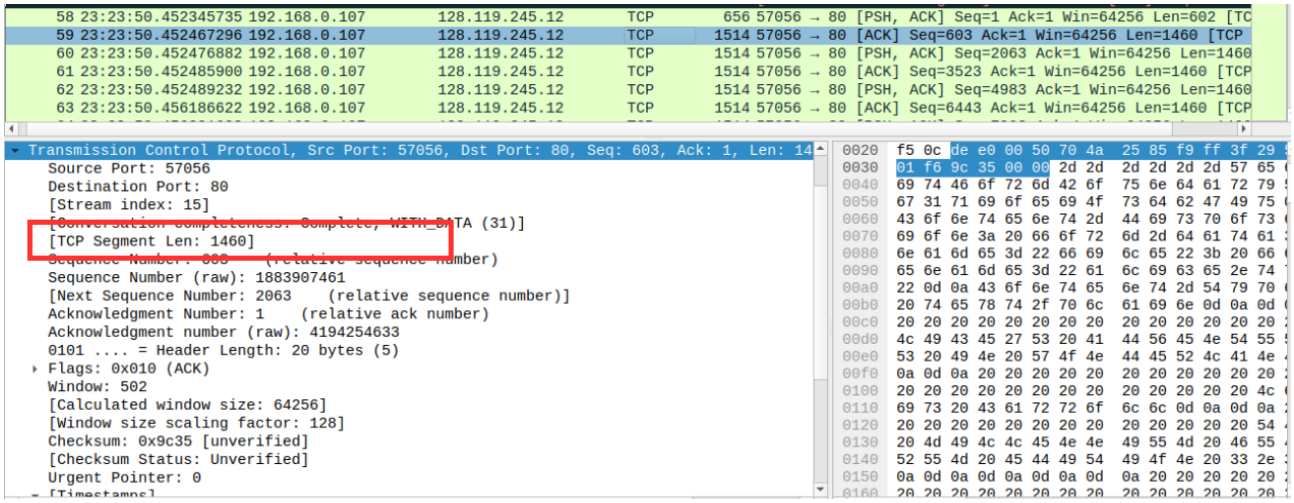
EstimatedRTT = (1 - α) * EstimatedRTT + α * SampleRTT
EstimatedRTT = RTT of first segment = 0.291232717 seconds
SampleRTT = RTT of current segment = 0.291101614 seconds
α = 0.125

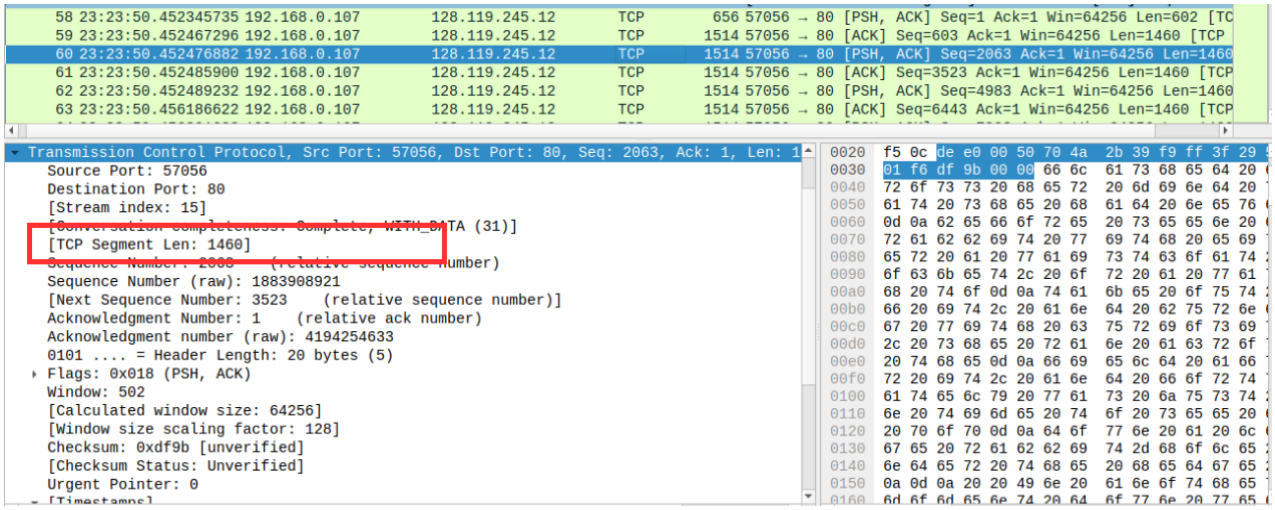
EstimatedRTT = (1 - α) * EstimatedRTT + α * SampleRTT
EstimatedRTT = (1 - 0.125) * 0.291232717 + 0.125. * 0.291101614
EstimatedRTT = 0.2912163292 seconds

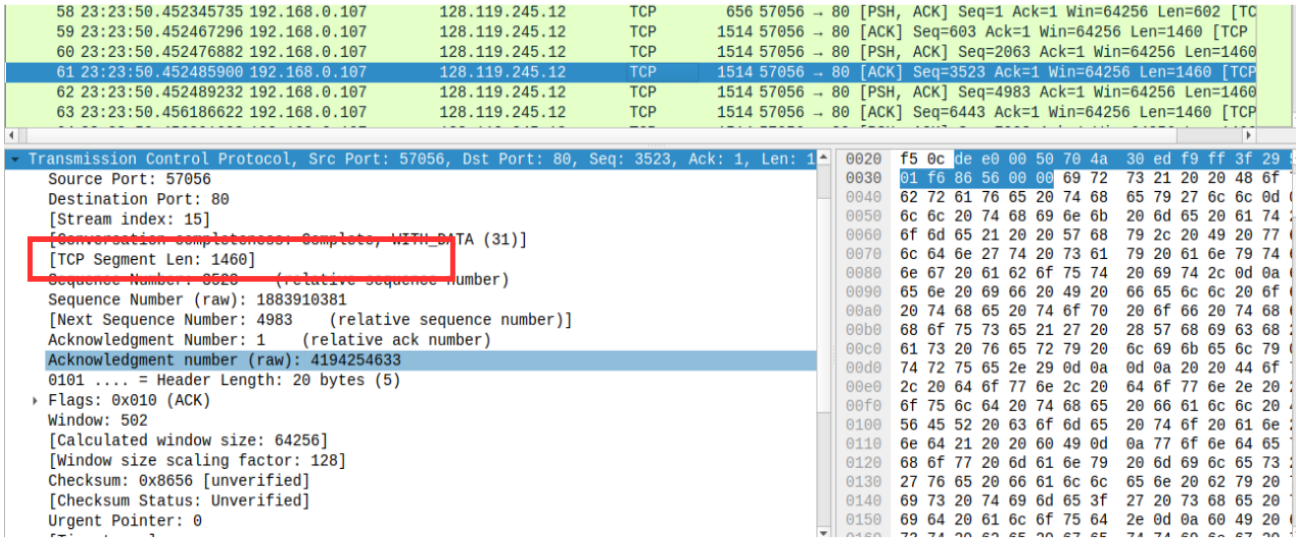
7.

The length (header plus payload) of each of the first four data-carrying TCP segments are **602, 1460, 1460, and 1460 Bytes.**Given below the figure shows it.





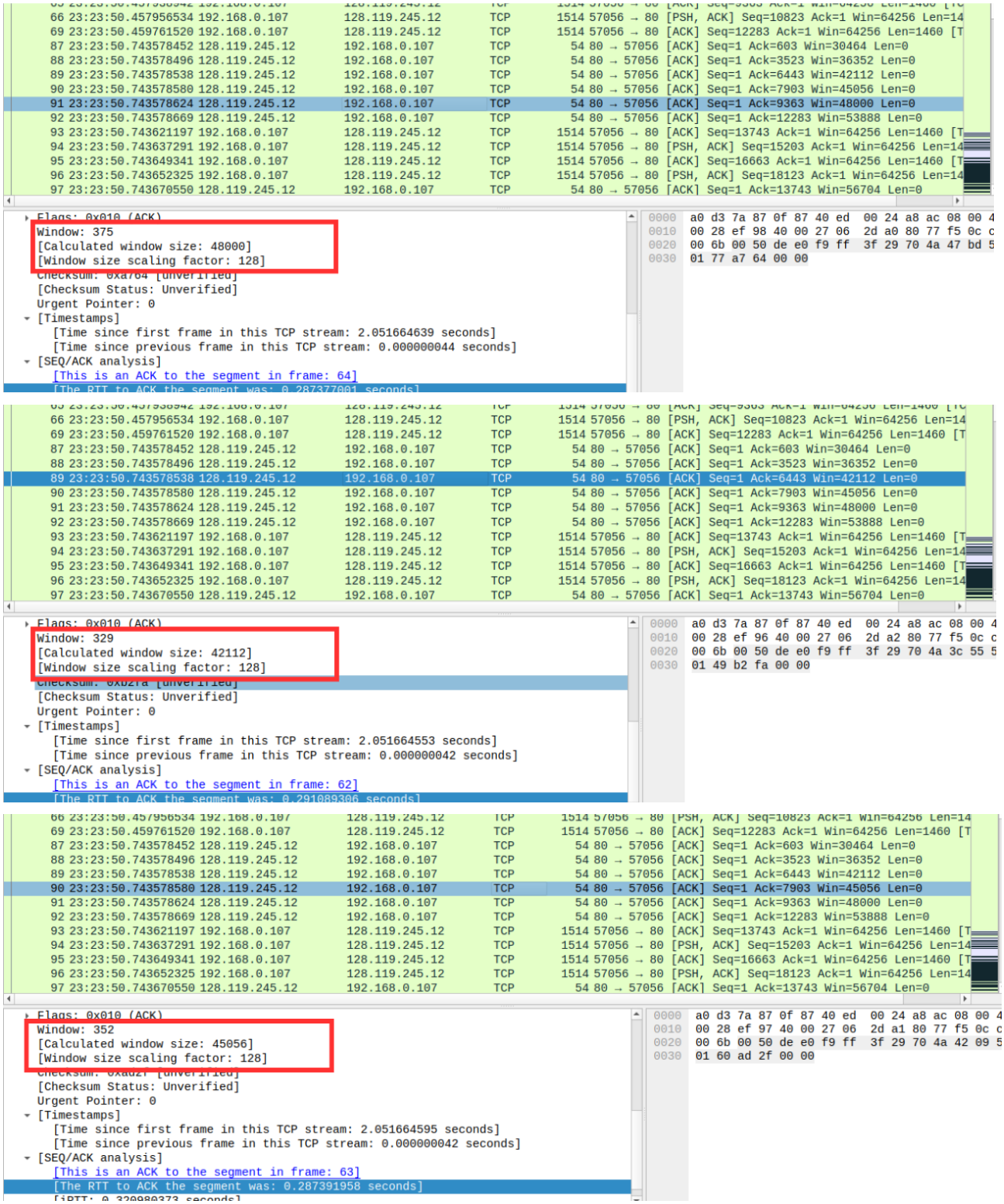


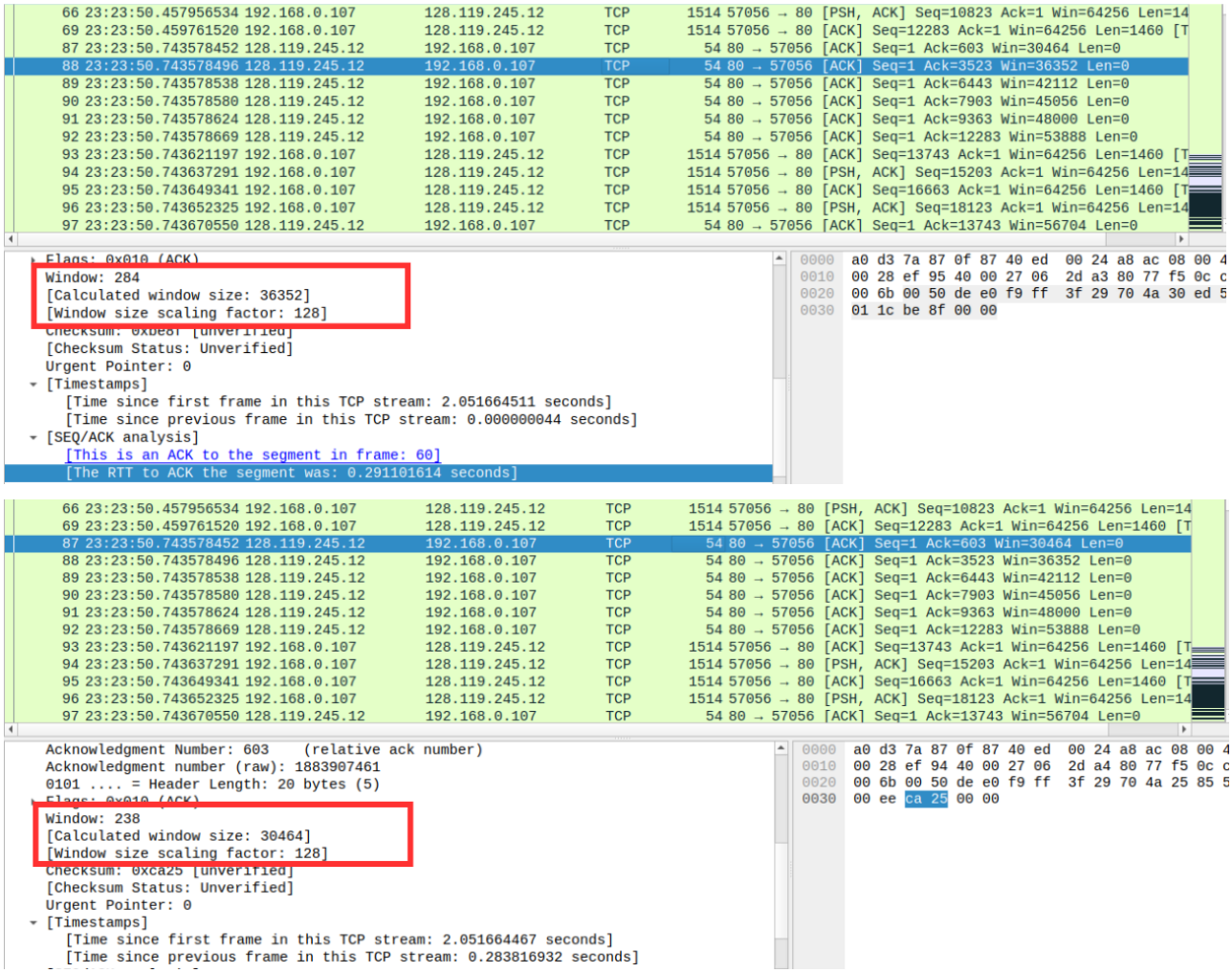


8.

The minimum amount of available buffer space advertised to the client by gaia.cs.umass.edu among these first five data-carrying TCP segments is **238** which is equivalent to 238 * 128 = **30464 Bytes**

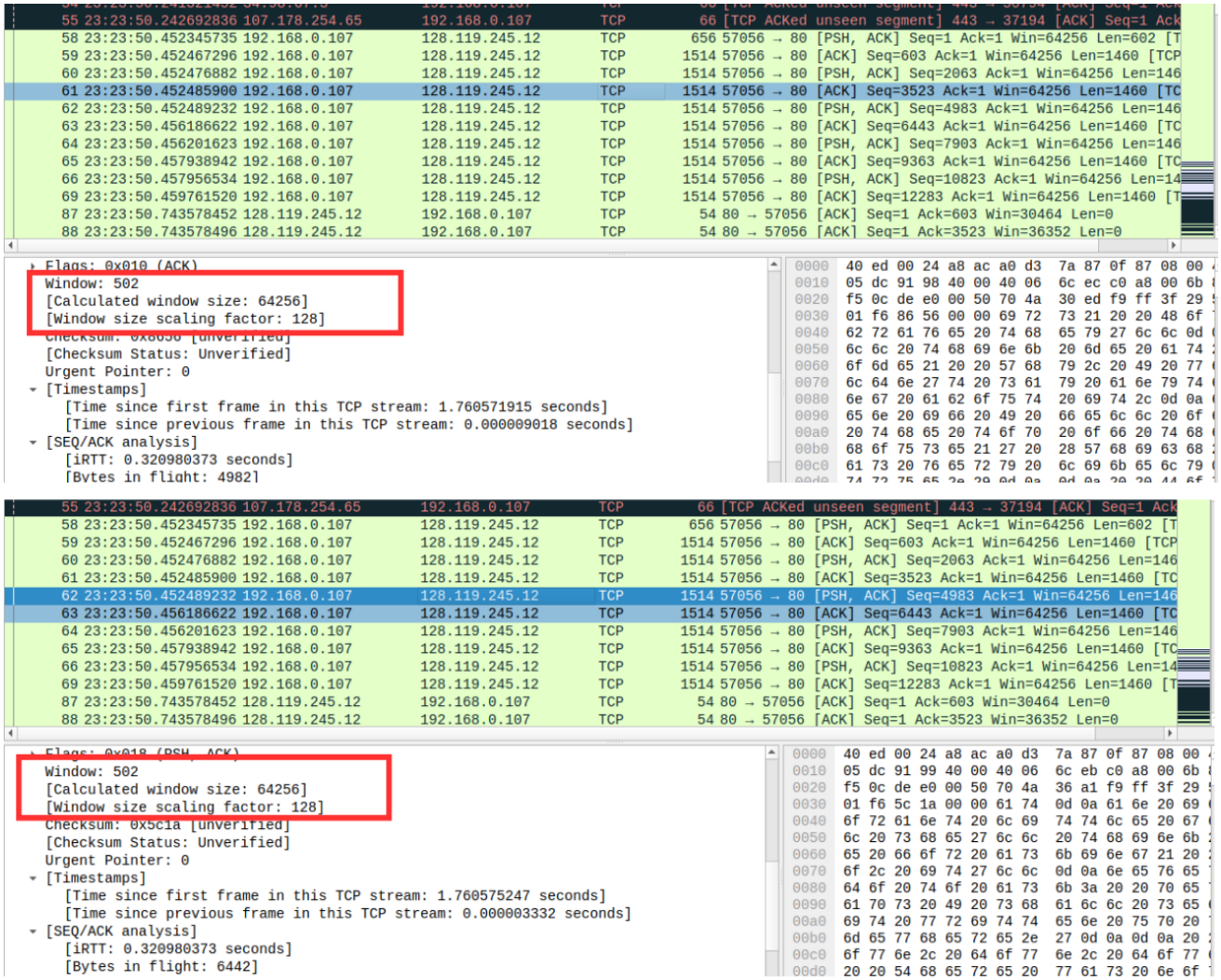
No, the lack of receiver buffer space ever throttles the sender for these first five data carrying segments because the advertised window size has ample buffer space for smooth communication between sender and receiver as the sender is sending packets of 1460 Bytes. Given below the figure shows it.


66 23:23:50.457956534 192.168.0.107 128.119.245.12 TCP 1514 57056 → 80 [PSH, ACK] Seq=10823 Ack=1 Win=64256 Len=14
69 23:23:50.459761520 192.168.0.107 128.119.245.12 TCP 1514 57056 → 80 [ACK] Seq=12283 Ack=1 Win=64256 Len=1460 [T
87 23:23:50.743578452 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=603 Win=30464 Len=0
88 23:23:50.743578496 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=3523 Win=36352 Len=0
89 23:23:50.743578538 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=6443 Win=42112 Len=0
90 23:23:50.743578580 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=7903 Win=45056 Len=0
91 23:23:50.743578624 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=9363 Win=48000 Len=0
92 23:23:50.743578669 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=12283 Win=53888 Len=0
93 23:23:50.743621197 192.168.0.107 128.119.245.12 TCP 1514 57056 → 80 [ACK] Seq=13743 Ack=1 Win=64256 Len=1460 [T
94 23:23:50.743637291 192.168.0.107 128.119.245.12 TCP 1514 57056 → 80 [PSH, ACK] Seq=15203 Ack=1 Win=64256 Len=14
95 23:23:50.743649341 192.168.0.107 128.119.245.12 TCP 1514 57056 → 80 [ACK] Seq=16663 Ack=1 Win=64256 Len=1460 [T
96 23:23:50.743652325 192.168.0.107 128.119.245.12 TCP 1514 57056 → 80 [PSH, ACK] Seq=18123 Ack=1 Win=64256 Len=14
97 23:23:50.743670550 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=13743 Win=56704 Len=0
Flags: 0x010 (ACK)
Window: 375
[Calculated window size: 48000]
[Window size scaling factor: 128]
[Checksum Status: Unverified]
Urgent Pointer: 0
[Timestamps]
[Time since first frame in this TCP stream: 2.051664639 seconds]
[Time since previous frame in this TCP stream: 0.000000044 seconds]
[SEQ/ACK analysis]
[This is an ACK to the segment in frame: 64]
[The RTT to ACK the segment was: 0.287377001 seconds]
0000 a0 d3 7a 87 0f 87 40 ed 00 24 a8 ac 08 00 4
0010 00 28 ef 98 40 00 27 06 2d a0 80 77 f5 0c c
0020 00 6b 00 50 de e0 f9 ff 3f 29 70 4a 47 bd 5
0030 01 77 a7 64 00 00
66 23:23:50.457956534 192.168.0.107 128.119.245.12 TCP 1514 57056 → 80 [PSH, ACK] Seq=10823 Ack=1 Win=64256 Len=14
69 23:23:50.459761520 192.168.0.107 128.119.245.12 TCP 1514 57056 → 80 [ACK] Seq=12283 Ack=1 Win=64256 Len=1460 [T
87 23:23:50.743578452 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=603 Win=30464 Len=0
88 23:23:50.743578496 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=3523 Win=36352 Len=0
89 23:23:50.743578538 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=6443 Win=42112 Len=0
90 23:23:50.743578580 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=7903 Win=45056 Len=0
91 23:23:50.743578624 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=9363 Win=48000 Len=0
92 23:23:50.743578669 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=12283 Win=53888 Len=0
93 23:23:50.743621197 192.168.0.107 128.119.245.12 TCP 1514 57056 → 80 [ACK] Seq=13743 Ack=1 Win=64256 Len=1460 [T
94 23:23:50.743637291 192.168.0.107 128.119.245.12 TCP 1514 57056 → 80 [PSH, ACK] Seq=15203 Ack=1 Win=64256 Len=14
95 23:23:50.743649341 192.168.0.107 128.119.245.12 TCP 1514 57056 → 80 [ACK] Seq=16663 Ack=1 Win=64256 Len=1460 [T
96 23:23:50.743652325 192.168.0.107 128.119.245.12 TCP 1514 57056 → 80 [PSH, ACK] Seq=18123 Ack=1 Win=64256 Len=14
97 23:23:50.743670550 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=13743 Win=56704 Len=0
Flags: 0x010 (ACK)
Window: 329
[Calculated window size: 42112]
[Window size scaling factor: 128]
[Checksum Status: Unverified]
Urgent Pointer: 0
[Timestamps]
[Time since first frame in this TCP stream: 2.051664553 seconds]
[Time since previous frame in this TCP stream: 0.000000042 seconds]
[SEQ/ACK analysis]
[This is an ACK to the segment in frame: 62]
[The RTT to ACK the segment was: 0.291089306 seconds]
0000 a0 d3 7a 87 0f 87 40 ed 00 24 a8 ac 08 00 4
0010 00 28 ef 96 40 00 27 06 2d a2 80 77 f5 0c c
0020 00 6b 00 50 de e0 f9 ff 3f 29 70 4a 3c 55 5
0030 01 49 b2 fa 00 00
66 23:23:50.457956534 192.168.0.107 128.119.245.12 TCP 1514 57056 → 80 [PSH, ACK] Seq=10823 Ack=1 Win=64256 Len=14
69 23:23:50.459761520 192.168.0.107 128.119.245.12 TCP 1514 57056 → 80 [ACK] Seq=12283 Ack=1 Win=64256 Len=1460 [T
87 23:23:50.743578452 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=603 Win=30464 Len=0
88 23:23:50.743578496 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=3523 Win=36352 Len=0
89 23:23:50.743578538 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=6443 Win=42112 Len=0
90 23:23:50.743578580 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=7903 Win=45056 Len=0
91 23:23:50.743578624 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=9363 Win=48000 Len=0
92 23:23:50.743578669 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=12283 Win=53888 Len=0
93 23:23:50.743621197 192.168.0.107 128.119.245.12 TCP 1514 57056 → 80 [ACK] Seq=13743 Ack=1 Win=64256 Len=1460 [T
94 23:23:50.743637291 192.168.0.107 128.119.245.12 TCP 1514 57056 → 80 [PSH, ACK] Seq=15203 Ack=1 Win=64256 Len=14
95 23:23:50.743649341 192.168.0.107 128.119.245.12 TCP 1514 57056 → 80 [ACK] Seq=16663 Ack=1 Win=64256 Len=1460 [T
96 23:23:50.743652325 192.168.0.107 128.119.245.12 TCP 1514 57056 → 80 [PSH, ACK] Seq=18123 Ack=1 Win=64256 Len=14
97 23:23:50.743670550 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=13743 Win=56704 Len=0
Flags: 0x010 (ACK)
Window: 352
[Calculated window size: 45056]
[Window size scaling factor: 128]
[Checksum Status: Unverified]
Urgent Pointer: 0
[Timestamps]
[Time since first frame in this TCP stream: 2.051664595 seconds]
[Time since previous frame in this TCP stream: 0.000000042 seconds]
[SEQ/ACK analysis]
[This is an ACK to the segment in frame: 63]
[The RTT to ACK the segment was: 0.287391958 seconds]
0000 a0 d3 7a 87 0f 87 40 ed 00 24 a8 ac 08 00 4
0010 00 28 ef 97 40 00 27 06 2d a1 80 77 f5 0c c
0020 00 6b 00 50 de e0 f9 ff 3f 29 70 4a 42 09 5
0030 01 60 ad 2f 00 00



9.

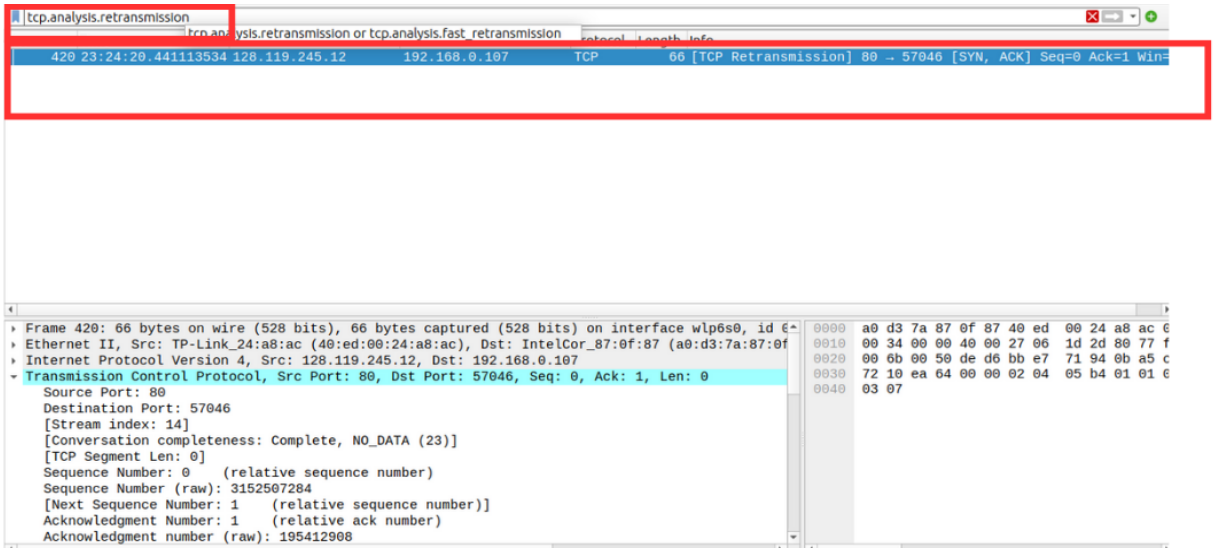
The minimum amount of available buffer space advertised by the client to gaia.cs.umass.edu is **502** which is equivalent to 502*128 = **64256 Bytes**. Given below the figure shows it.



The first observation is that each TCP segment from client to gaia.cs.umass.edu has the same window size whereas window size varies for the case of gaia.cs.umass.edu to client. One of the reasons for the given observation would be that data is being sent from client to gaia.cs.umass.edu which leads to a lesser buffer because of delay in processing. There is no data sent from gaia.cs.umass.edu to the client and hence the buffer is empty all the time.

10.

Yes, there are retransmitted segments in the trace file. The retransmitted segments are found using the filter “tcp.analysis.retransmission”.



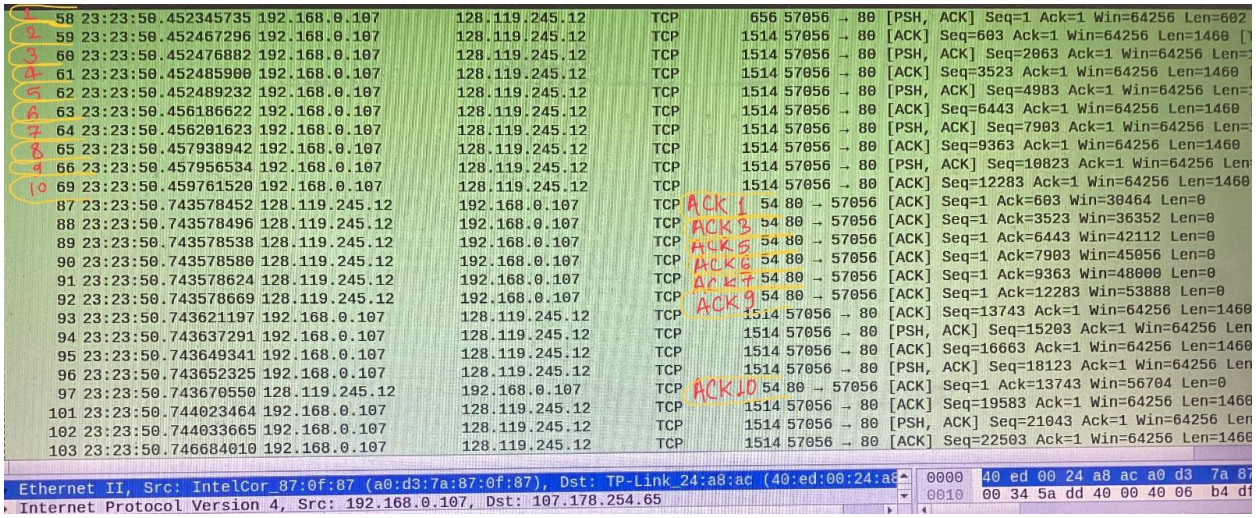
11.

Typically ACK acknowledges 1460 Bytes of data among the first ten data-carrying segments sent from the client to gaia.cs.umass.edu. Also, the segment size of each ACK segment is 0 Bytes.

There are certain instances where every packet is acknowledged and few acknowledgements are skipped as cumulative acknowledgment.

1st segment is acknowledged
2nd segment is not acknowledged
3rd segment is acknowledged
4th segment is not acknowledged
5th segment is acknowledged
6th segment is acknowledged
7th segment is acknowledged
8th segment is not acknowledged
9th segment is acknowledged
10th segment is acknowledged

Given below the figure showing details of the transmission:



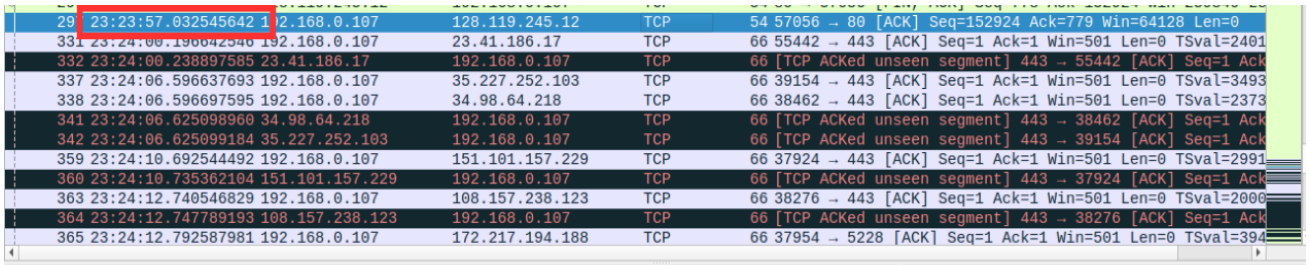
12.

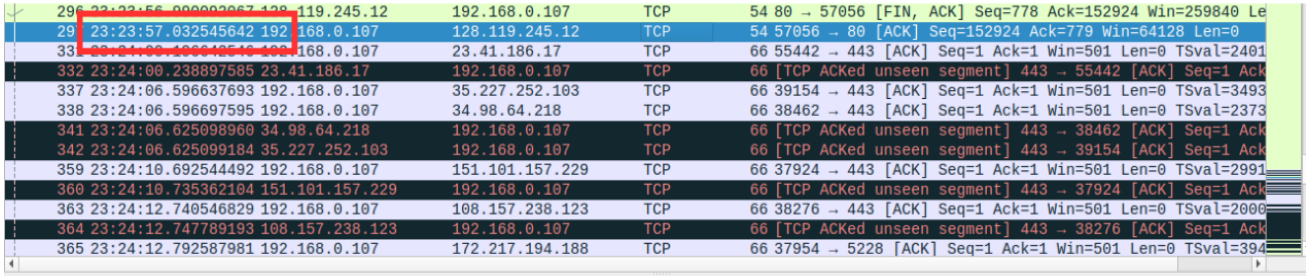
Initial time of establishing connection = 23:23:48.691815969
Time at which connection closed = 23:23:57.032545642
Total data sent = 152923 bytes

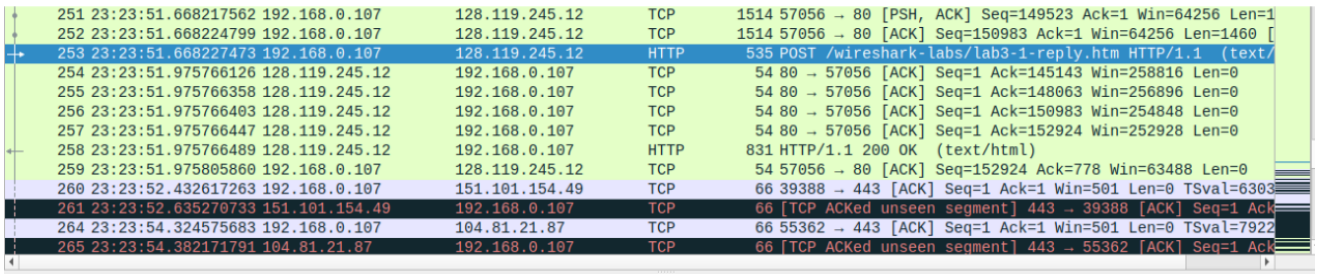
Total time taken = 23:23:57.032545642 - 23:23:48.691815969
= 6.340729673 seconds

Throughput = data sent/ time taken
= 152923 bytes / 6.340729673 seconds
= 24117.57 Bytes/second

Given below the figure showing details of the transmission:


29 23:23:57.032545642 1 2.168.0.107 128.119.245.12 TCP 54 57056 → 80 [ACK] Seq=152924 Ack=779 Win=64128 Len=0
331 23:24:00.196642546 192.168.0.107 23.41.186.17 TCP 66 55442 → 443 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=2401
332 23:24:00.238897585 23.41.186.17 192.168.0.107 TCP 66 [TCP ACKed unseen segment] 443 → 55442 [ACK] Seq=1 Ack
337 23:24:06.596637693 192.168.0.107 35.227.252.103 TCP 66 39154 → 443 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=3493
338 23:24:06.596697595 192.168.0.107 34.98.64.218 TCP 66 38462 → 443 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=2373
341 23:24:06.625098960 34.98.64.218 192.168.0.107 TCP 66 [TCP ACKed unseen segment] 443 → 38462 [ACK] Seq=1 Ack
342 23:24:06.625099184 35.227.252.103 192.168.0.107 TCP 66 [TCP ACKed unseen segment] 443 → 39154 [ACK] Seq=1 Ack
359 23:24:10.692544492 192.168.0.107 151.101.157.229 TCP 66 37924 → 443 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=2991
360 23:24:10.735362104 151.101.157.229 192.168.0.107 TCP 66 [TCP ACKed unseen segment] 443 → 37924 [ACK] Seq=1 Ack
363 23:24:12.740546829 192.168.0.107 108.157.238.123 TCP 66 38276 → 443 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=2000
364 23:24:12.747789193 108.157.238.123 192.168.0.107 TCP 66 [TCP ACKed unseen segment] 443 → 38276 [ACK] Seq=1 Ack
365 23:24:12.792587981 192.168.0.107 172.217.194.188 TCP 66 37954 → 5228 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=394
Source Port: 57056
Destination Port: 80
[Stream index: 15]
[Conversation completeness: Complete, WITH_DATA (31)]
[TCP Segment Len: 0]
Sequence Number: 152924 (relative sequence number)
Sequence Number (raw): 1884059782
[Next Sequence Number: 152924 (relative sequence number)]
Acknowledgment Number: 779 (relative ack number)
Acknowledgment number (raw): 4194255411
0101 .... = Header Length: 20 bytes (5)
Flags: 0x010 (ACK)
Window: 501
[Calculated window size: 64128]
0000 40 ed 00 24 a8 ac a0 d3 7a 87 0f 87
0010 00 28 92 00 40 00 40 06 72 38 c0 a8
0020 f5 0c de e0 00 50 70 4c 78 86 f9 ff
0030 01 f5 73 11 00 00

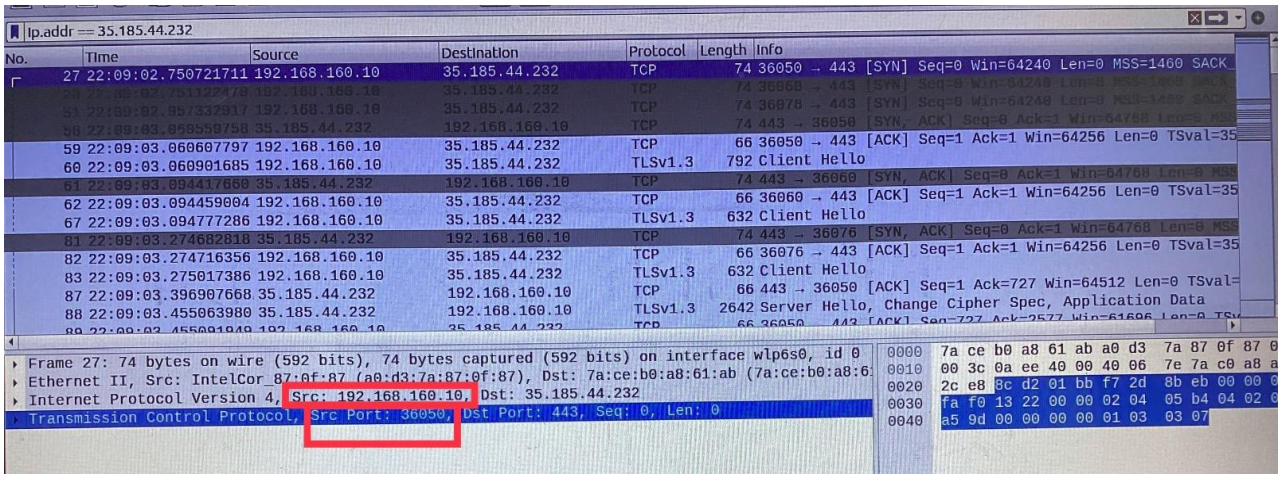

296 23:23:56.000002067 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [FIN, ACK] Seq=778 Ack=152924 Win=259840 Le
29 23:23:57.032545642 192.168.0.107 128.119.245.12 TCP 54 57056 → 80 [ACK] Seq=152924 Ack=779 Win=64128 Len=0
33 23:24:00.196642546 192.168.0.107 23.41.186.17 TCP 66 55442 → 443 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=2401
332 23:24:00.238897585 23.41.186.17 192.168.0.107 TCP 66 [TCP ACKed unseen segment] 443 → 55442 [ACK] Seq=1 Ack
337 23:24:06.596637693 192.168.0.107 35.227.252.103 TCP 66 39154 → 443 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=3493
338 23:24:06.596697595 192.168.0.107 34.98.64.218 TCP 66 38462 → 443 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=2373
341 23:24:06.625098960 34.98.64.218 192.168.0.107 TCP 66 [TCP ACKed unseen segment] 443 → 38462 [ACK] Seq=1 Ack
342 23:24:06.625099184 35.227.252.103 192.168.0.107 TCP 66 [TCP ACKed unseen segment] 443 → 39154 [ACK] Seq=1 Ack
359 23:24:10.692544492 192.168.0.107 151.101.157.229 TCP 66 37924 → 443 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=2991
360 23:24:10.735362104 151.101.157.229 192.168.0.107 TCP 66 [TCP ACKed unseen segment] 443 → 37924 [ACK] Seq=1 Ack
363 23:24:12.740546829 192.168.0.107 108.157.238.123 TCP 66 38276 → 443 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=2000
364 23:24:12.747789193 108.157.238.123 192.168.0.107 TCP 66 [TCP ACKed unseen segment] 443 → 38276 [ACK] Seq=1 Ack
365 23:24:12.792587981 192.168.0.107 172.217.194.188 TCP 66 37954 → 5228 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=394
Source Port: 57056
Destination Port: 80
[Stream index: 15]
[Conversation completeness: Complete, WITH_DATA (31)]
[TCP Segment Len: 0]
Sequence Number: 152924 (relative sequence number)
Sequence Number (raw): 1884059782
[Next Sequence Number: 152924 (relative sequence number)]
Acknowledgment Number: 779 (relative ack number)
Acknowledgment number (raw): 4194255411
0101 .... = Header Length: 20 bytes (5)
Flags: 0x010 (ACK)
Window: 501
0000 40 ed 00 24 a8 ac a0 d3 7a 87 0f 87
0010 00 28 92 00 40 00 40 06 72 38 c0 a8
0020 f5 0c de e0 00 50 70 4c 78 86 f9 ff
0030 01 f5 73 11 00 00


251 23:23:51.668217562 192.168.0.107 128.119.245.12 TCP 1514 57056 → 80 [PSH, ACK] Seq=149523 Ack=1 Win=64256 Len=1
252 23:23:51.668224799 192.168.0.107 128.119.245.12 TCP 1514 57056 → 80 [ACK] Seq=150983 Ack=1 Win=64256 Len=1460 [
253 23:23:51.668227473 192.168.0.107 128.119.245.12 HTTP 535 POST /wireshark-labs/lab3-1-reply.htm HTTP/1.1 (text/
254 23:23:51.975766126 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=145143 Win=258816 Len=0
255 23:23:51.975766358 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=148063 Win=256896 Len=0
256 23:23:51.975766403 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=150983 Win=254848 Len=0
257 23:23:51.975766447 128.119.245.12 192.168.0.107 TCP 54 80 → 57056 [ACK] Seq=1 Ack=152924 Win=252928 Len=0
258 23:23:51.975766489 128.119.245.12 192.168.0.107 HTTP 831 HTTP/1.1 200 OK (text/html)
259 23:23:51.975805860 192.168.0.107 128.119.245.12 TCP 54 57056 → 80 [ACK] Seq=152924 Ack=778 Win=63488 Len=0
260 23:23:52.432617263 192.168.0.107 151.101.154.49 TCP 66 39388 → 443 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=6303
261 23:23:52.635270733 151.101.154.49 192.168.0.107 TCP 66 [TCP ACKed unseen segment] 443 → 39388 [ACK] Seq=1 Ack
264 23:23:54.324575683 192.168.0.107 104.81.21.87 TCP 66 55362 → 443 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=7922
265 23:23:54.382171791 104.81.21.87 192.168.0.107 TCP 66 [TCP ACKed unseen segment] 443 → 55362 [ACK] Seq=1 Ack
Urgent Pointer: 0
[Timestamps]
[Time since first frame in this TCP stream: 2.976313488 seconds]
[Time since previous frame in this TCP stream: 0.000002674 seconds]
[SEQ/ACK analysis]
[iRTT: 0.320980373 seconds]
[Bytes in flight: 10701]
[Bytes sent since last PSH flag: 1941]
TCP payload (481 bytes)
TCP segment data (481 bytes)
[106 Reassembled TCP Segments (152923 bytes): #5 (602), #59(1460), #60(1460), #61(1460), #62(1
Hypertext Transfer Protocol
MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "----WebKitFormBounda
00000000 50 4f 53 54 20 2f 77 69 72 65 73
00000010 6c 61 62 73 2f 6c 61 62 33 2d 31
00000020 79 2e 68 74 6d 20 48 54 54 50 2f
00000030 48 6f 73 74 3a 20 67 61 69 61 2e
00000040 61 73 73 2e 65 64 75 0d 0a 43 6f
00000050 69 6f 6e 3a 20 6b 65 65 70 2d 61
00000060 0a 43 6f 6e 74 65 6e 74 2d 4c 65
00000070 20 31 35 32 33 32 31 0d 0a 43 61
00000080 6f 6e 74 72 6f 6c 3a 20 6d 61 78
00000090 30 0d 0a 55 70 67 72 61 64 65 2d
000000a0 75 72 65 2d 52 65 71 75 65 73 74
000000b0 0a 55 73 65 72 2d 41 67 65 6e 74
000000c0 69 6c 6c 61 2f 35 2e 30 20 28 58

PART-B

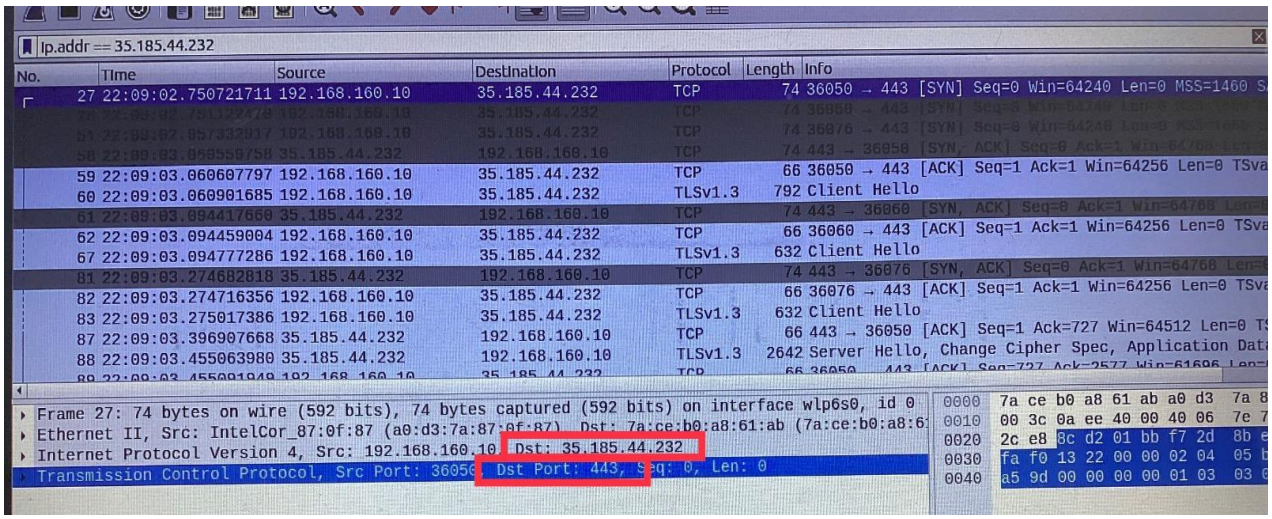
1.

The IP address and TCP port number used by the client computer (source) to get the homepage of cse.iith.ac.in is **192.168.160.10** and **36050** respectively. Given below the figure shows it.



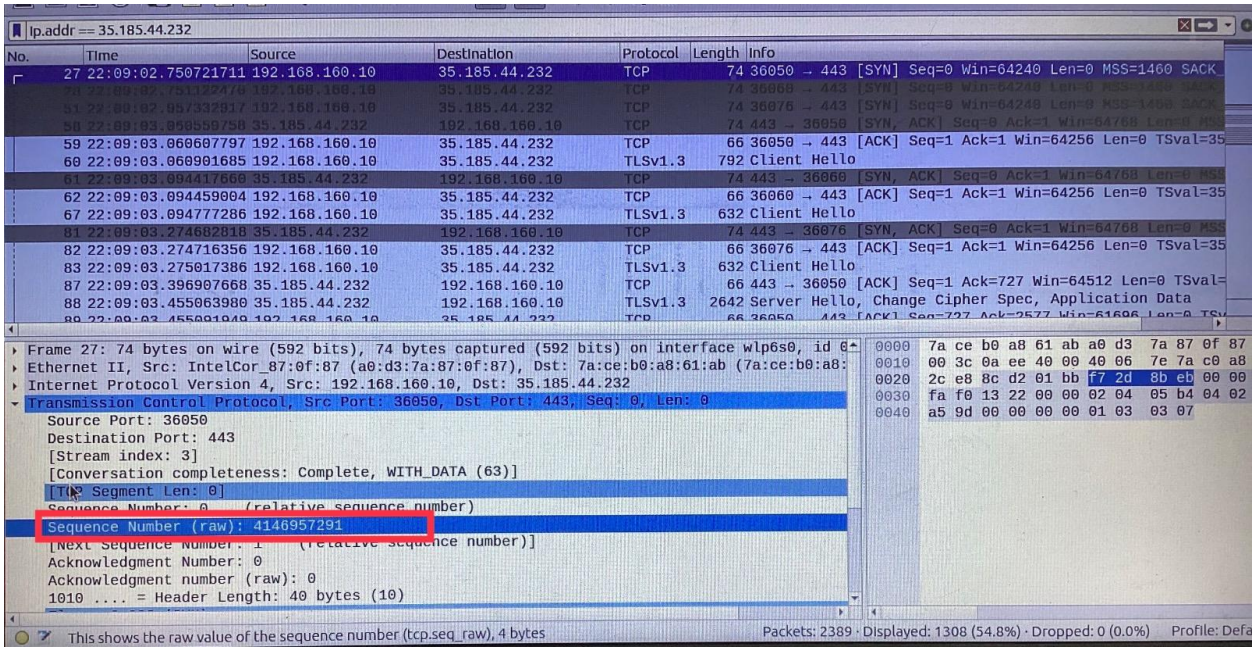
2.

The IP address of cse.iith.ac.in is **35.185.44.232** and the port number it is sending and receiving TCP segments for this connection is **443**. Given below the figure shows it.

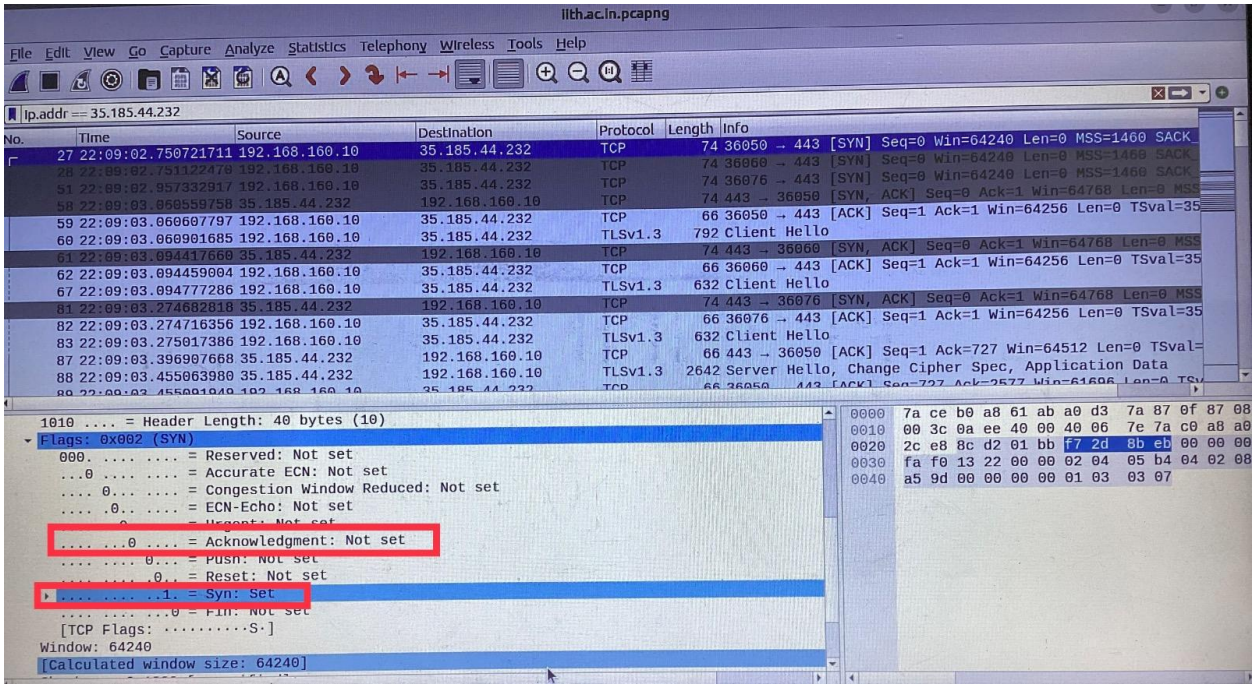


3.

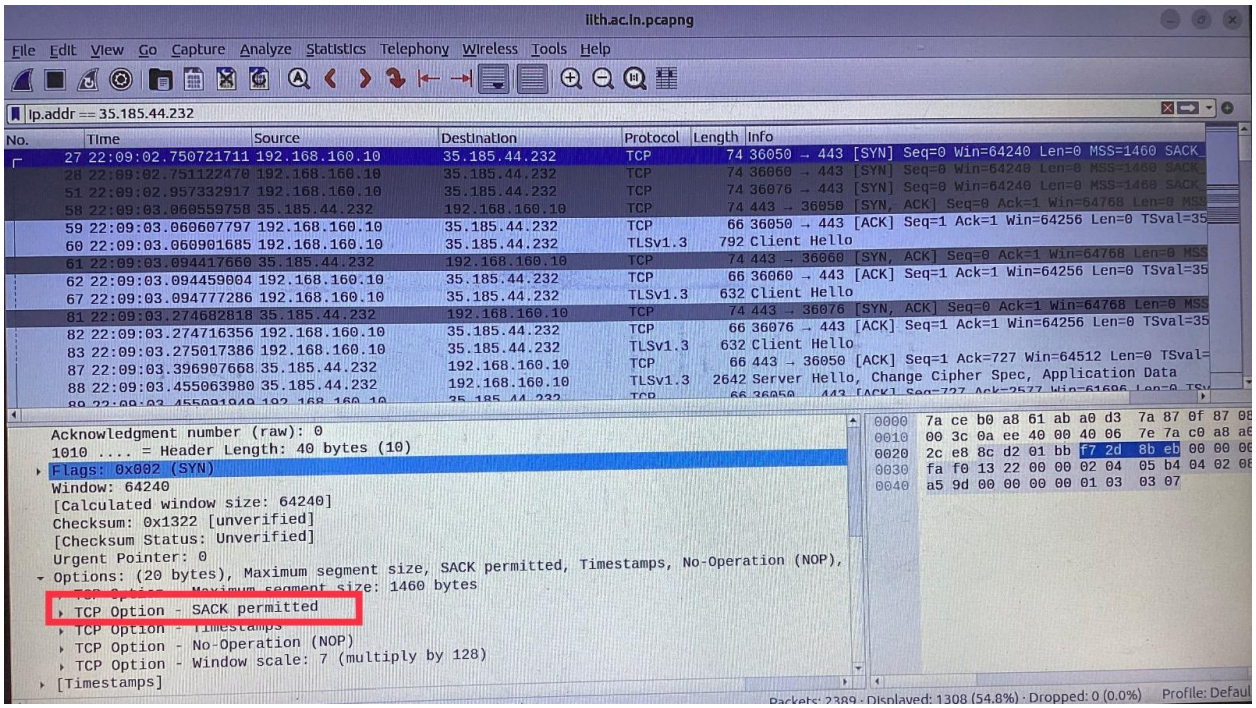
The sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and cse.iith.ac.in is **4146957291**. Given below the figure shows it.



The **SYN bit is set to 1** inside the flags of the TCP header and tells us that the selected TCP segment identifies the SYN segment. Given below the figure shows it.



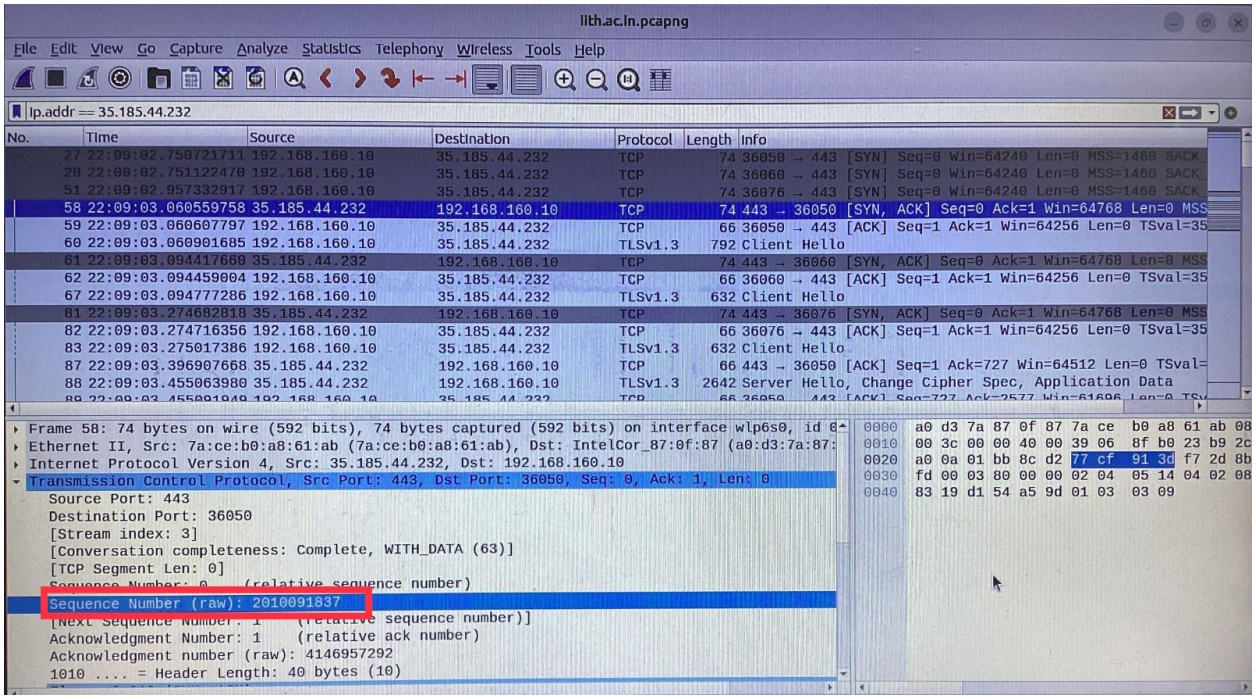
Yes, the TCP receiver in this session will be able to **use selective acknowledgements** as the SACK Permitted option is present in the TCP options section. Given below the figure shows it.



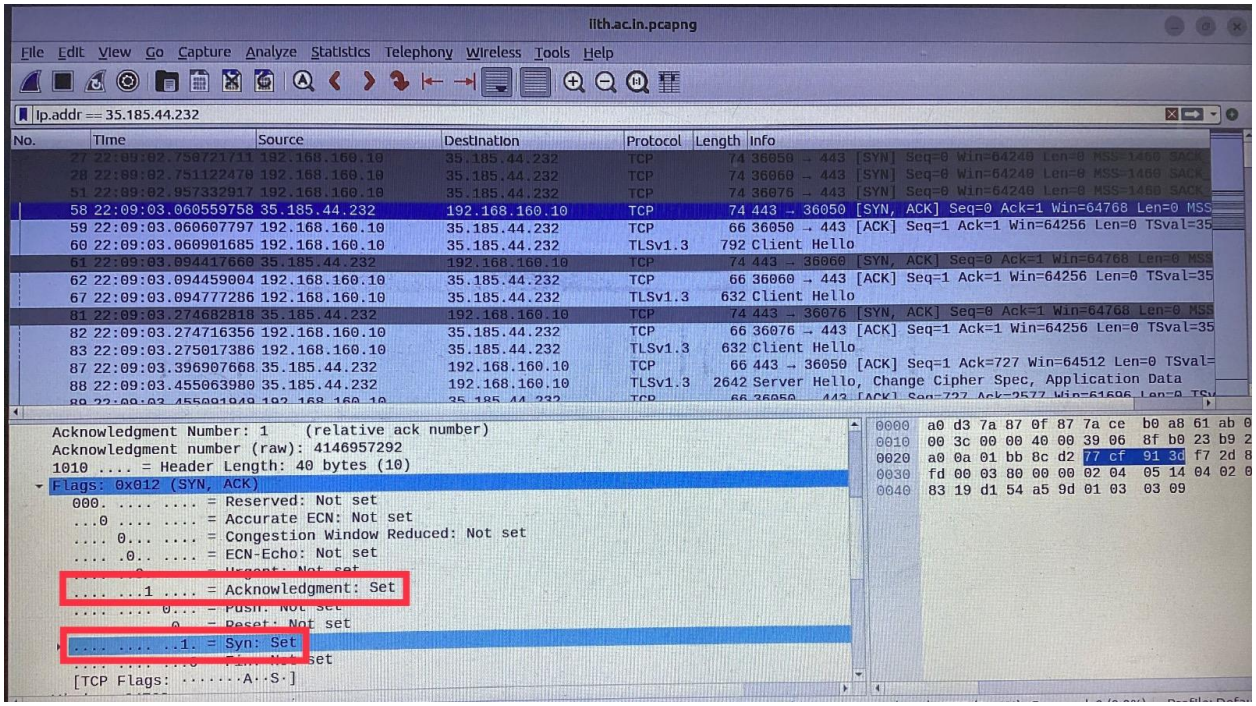
Note: My system sends three SYN requests, gets three SYNACK replies and then again sends three ACK to “cse.iith.ac.in”.

4.

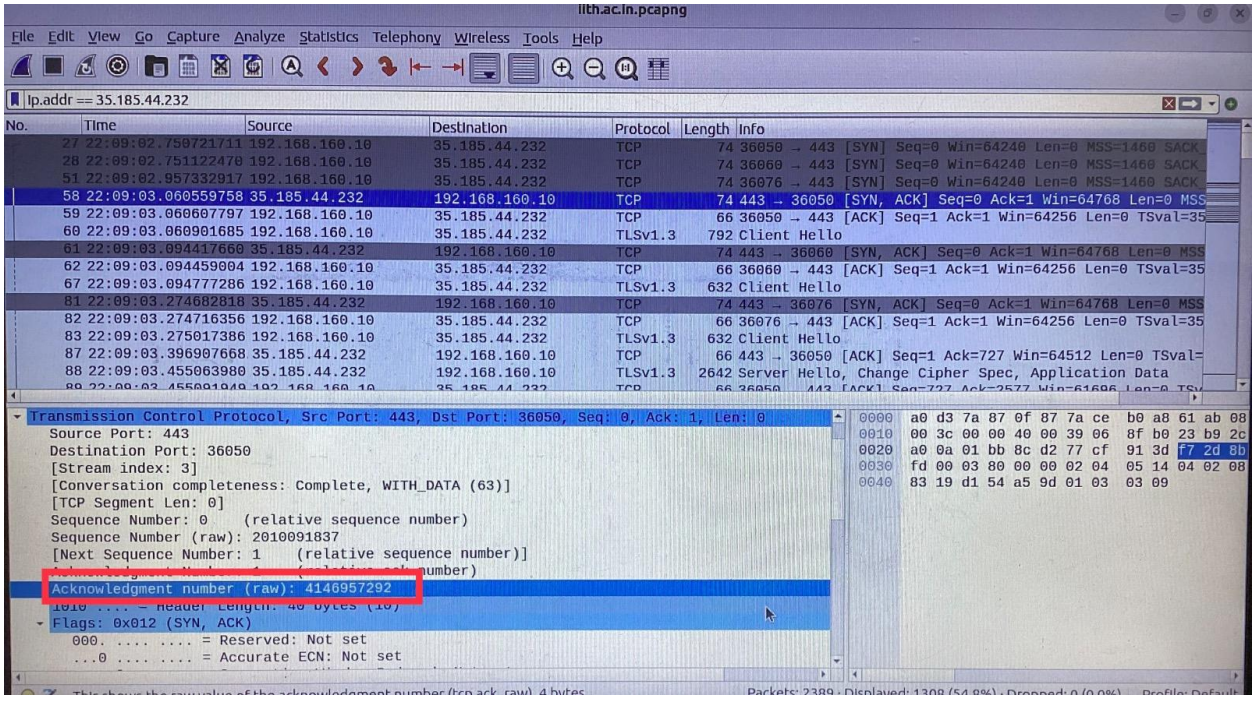
The sequence number of the TCP SYNACK segment that was sent by cse.iith.ac.in to the client computer in reply to the SYN is **2010091837**. Given below the figure shows it.



The **SYN and acknowledgment bit is set to 1** inside the flags of the TCP header and tells us that the selected TCP segment identifies the SYNACK segment. Given below the figure shows it.



The value of the Acknowledgement field in the SYNACK segment is **4146957292.** Given below the figure shows it.



The value of the Acknowledgement field in SYNACK by gaia.cs.umass.edu is determined by the sequence number of the SYN segment received from my computer. It increments the value by 1 which indicates the next sequence number from my computer.

4146957291 + 1 = **4146957292**

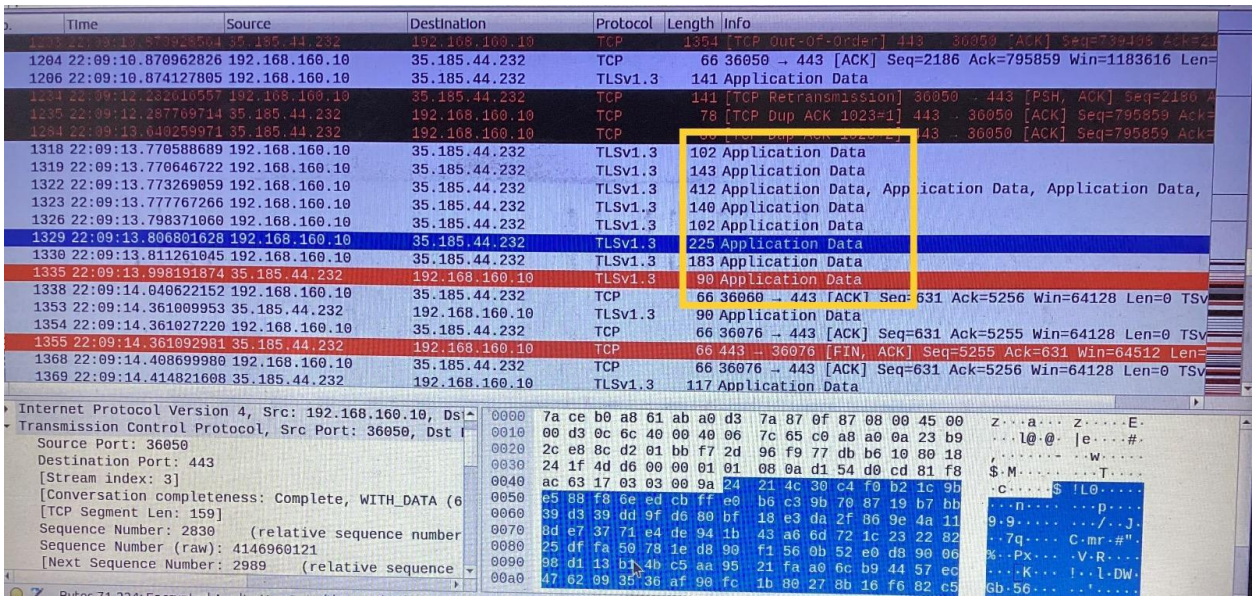
Note: My system sends three SYN requests, gets two SYNACK replies and then again sends three ACK to “cse.iith.ac.in”. The second and third set also follow the same approach of incrementing the value by 1 of its sequence number to get the acknowledgement number.

5.

I am not able to see the GET request or any other details of the encrypted communication. To decrypt HTTPS traffic in Wireshark, It is required to have access to the private key associated with the SSL/TLS certificate used during the communication.

The answer below is mentioned based on the TLS communication sequence diagram.

- It is visible that first TCP connection is made and then the TLS hello and its reply is done.
- Also, the key exchange and data transfer take place, but the place where data transfer takes place i.e., “GET” message in data content is not visible.
- **No, all of the data in the transfer from the IITH web server to client does not fit into a single segment.** Multiple segments can be seen in the wireshark to get the content of the web page from the server.



6.

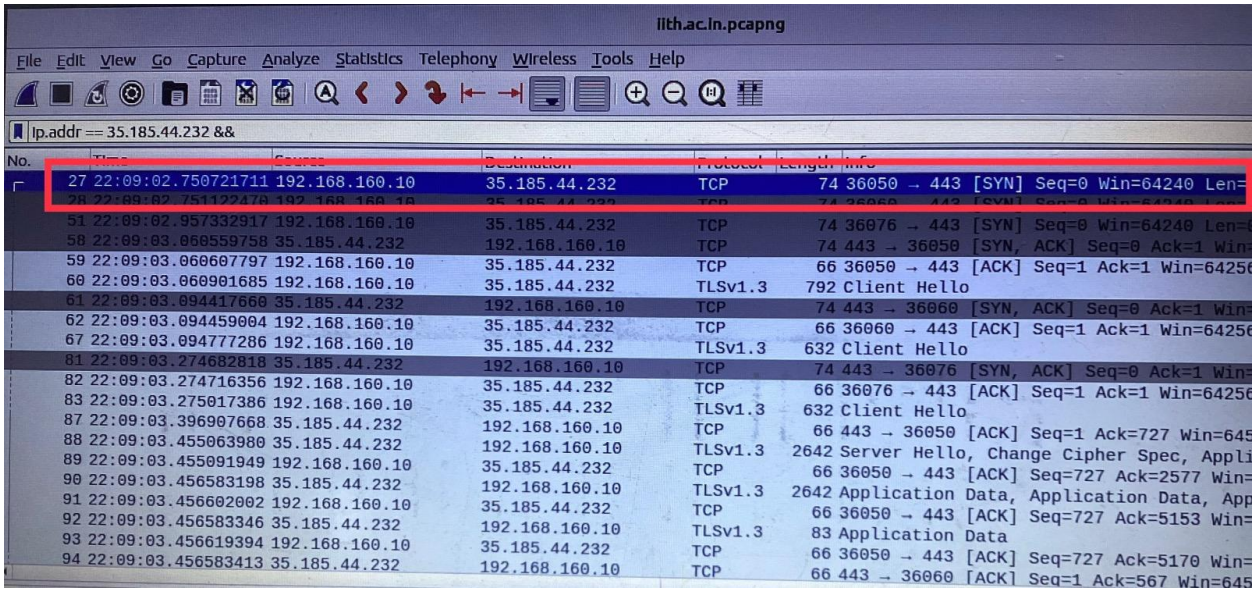
I am not able to see the GET request or any other details of the encrypted communication. To decrypt HTTPS traffic in Wireshark, It is required to have access to the private key associated with the SSL/TLS certificate used during the communication.

The answer below is mentioned based on the TLS communication sequence diagram.

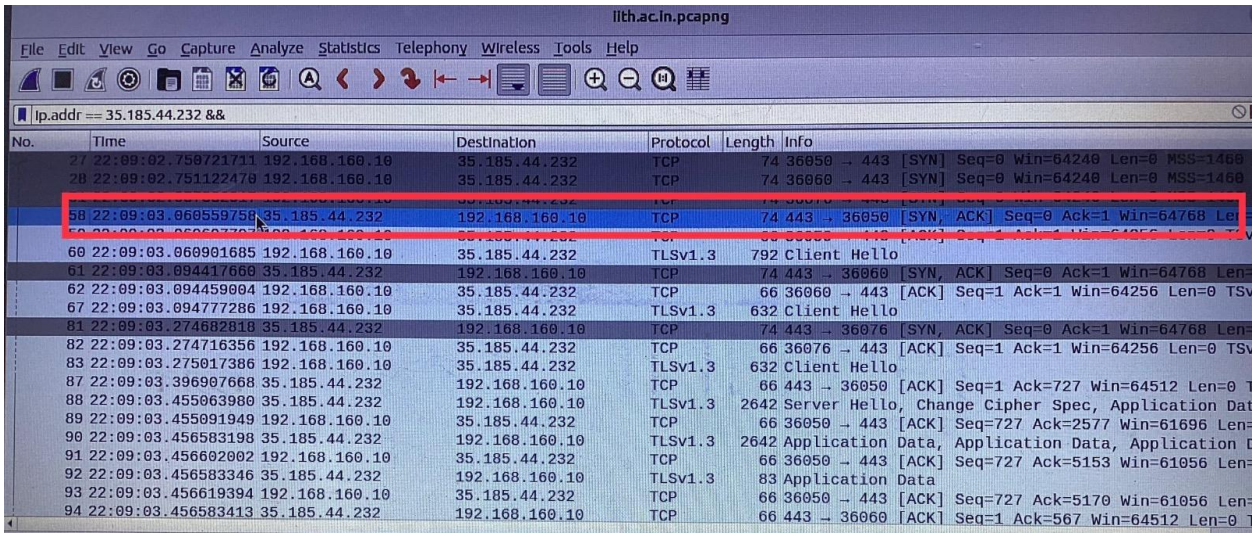
- It is visible that first TCP connection is made and then the TLS hello and its reply is done.
- Also, the key exchange and data transfer take place, but the place where data transfer takes place i.e., “GET” message in data content is not visible.

The RTT calculation below is based on the the first two request sent from client to “cse.iith.ac.in”

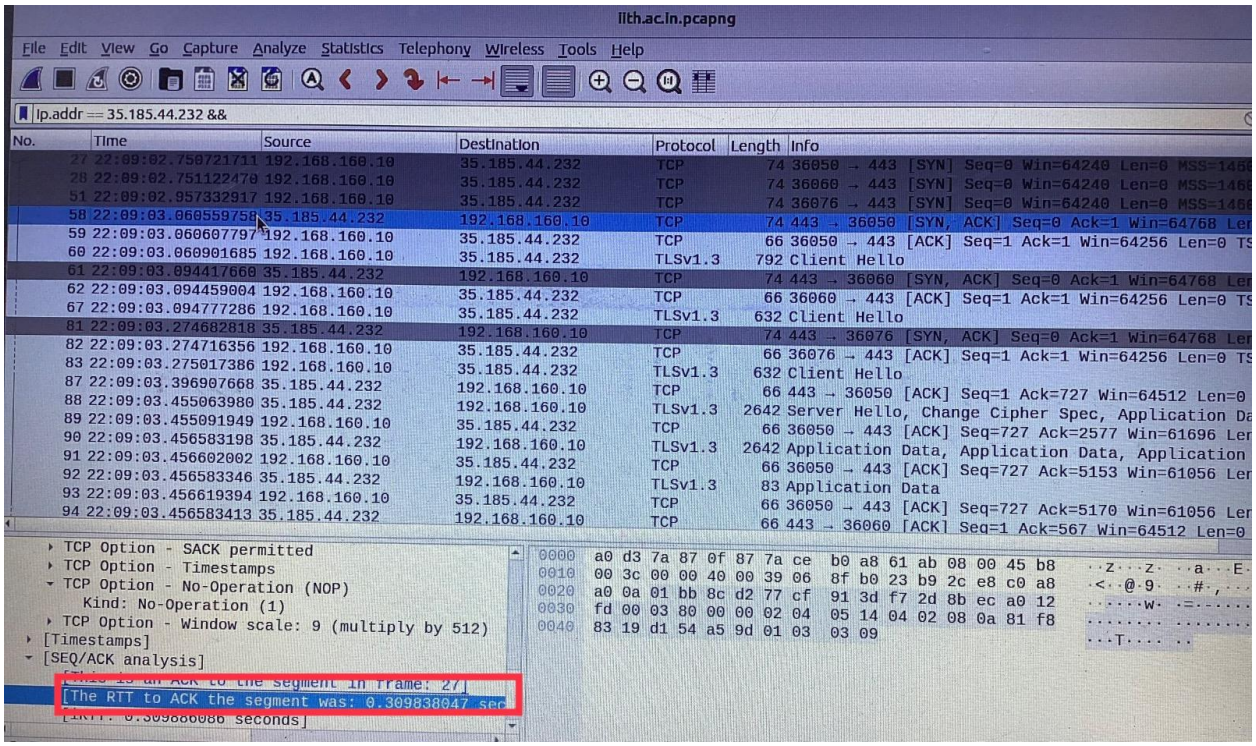
The time at which the first segment was sent is “**22:09:02.750721711**”.Given below the figure shows it.



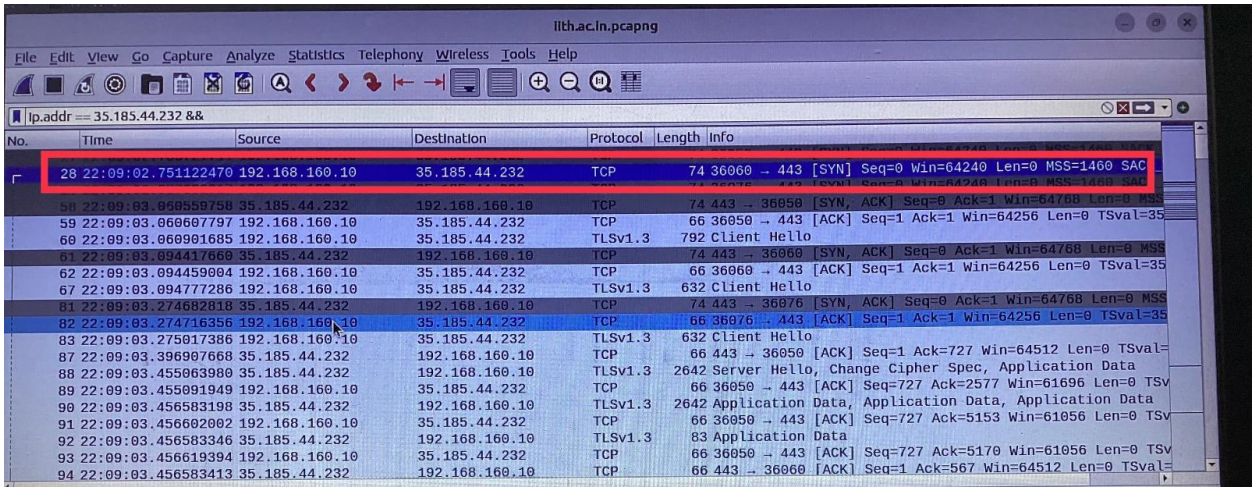
The time at which the ACK of the first segment was received is "**22:09:03.060559758**". Given below the figure shows it.



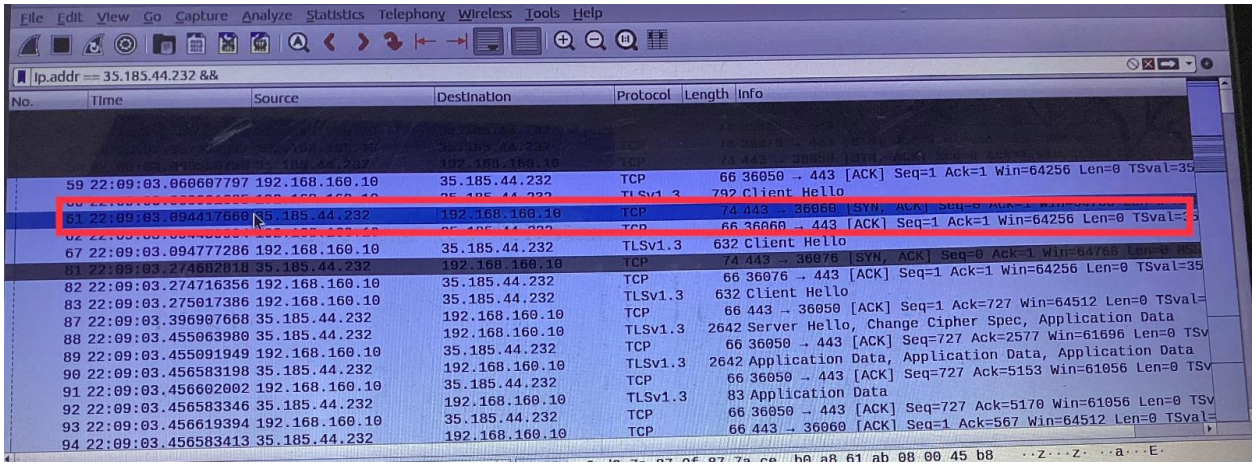
The RTT of the first data-containing segment is **0.309838047 seconds** . Given below the figure shows it.



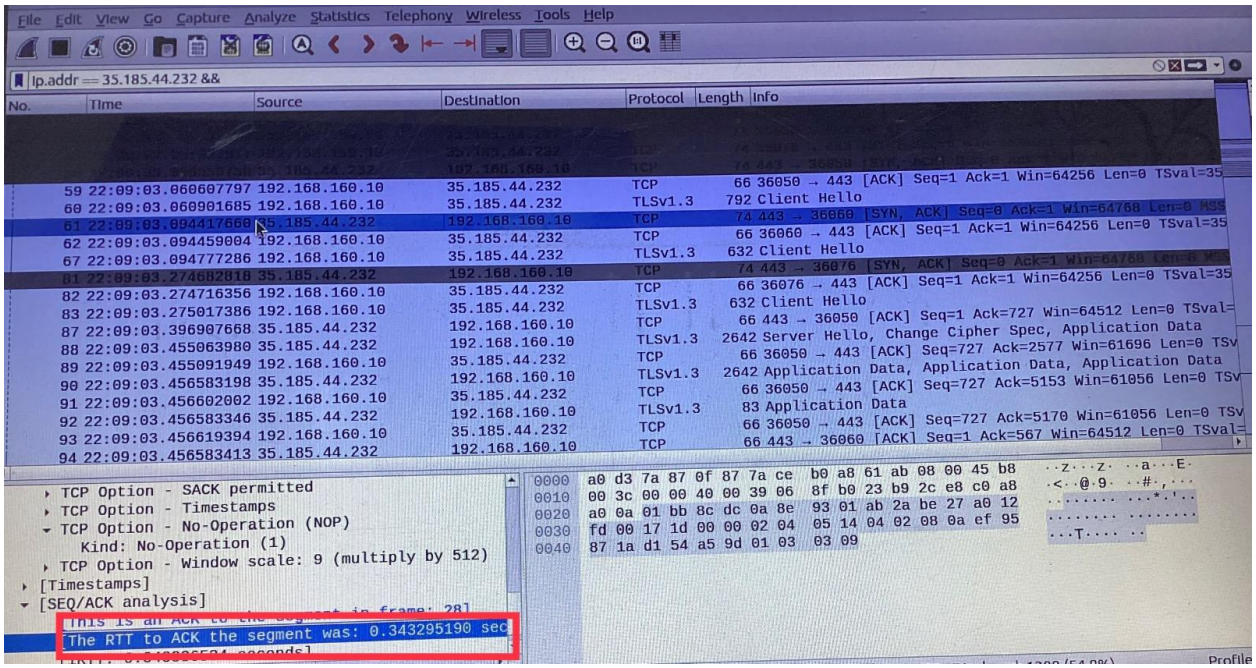
The time at which the second segment was sent is "**22:09:02.751122470**". Given below the figure shows it.



The time at which the ACK of the second segment was received is "**22:09:03.094417660**". Given below the figure shows it.



The RTT of the second data-containing segment is **0.343295190 seconds**. Given below the figure shows it.

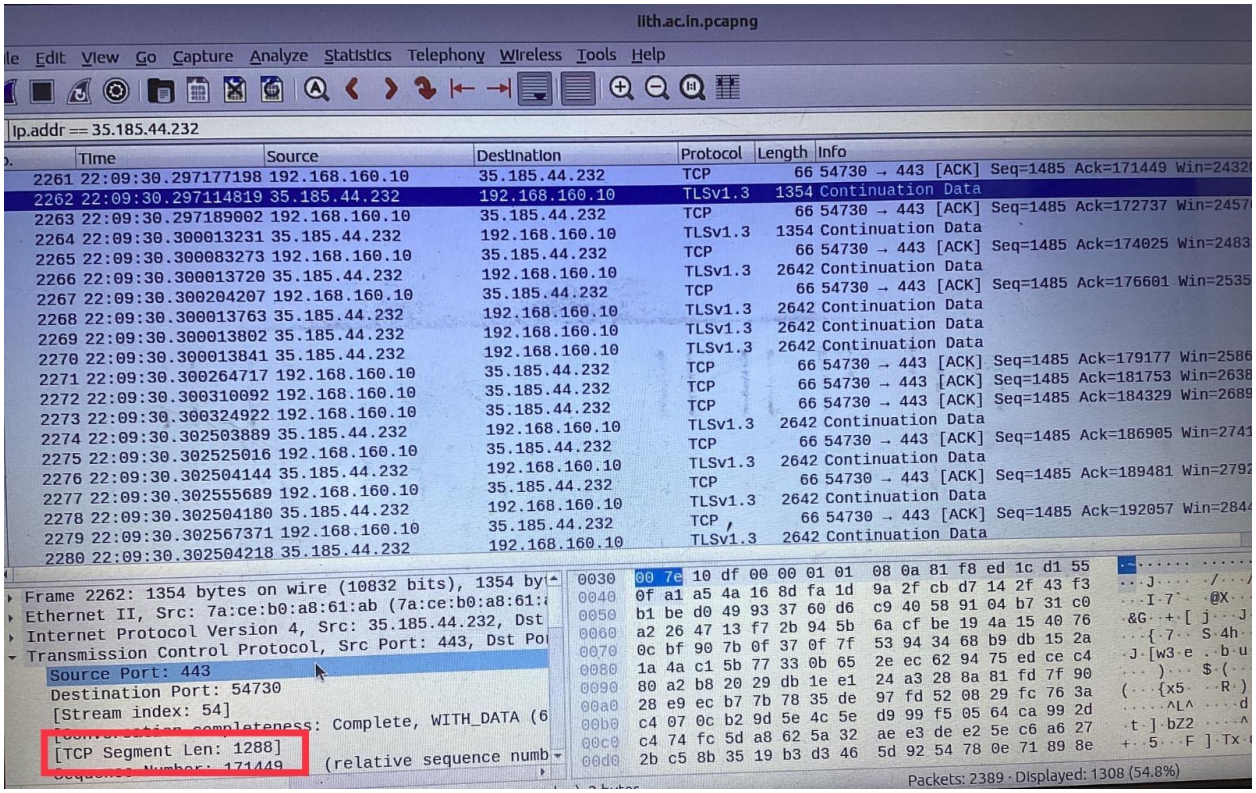


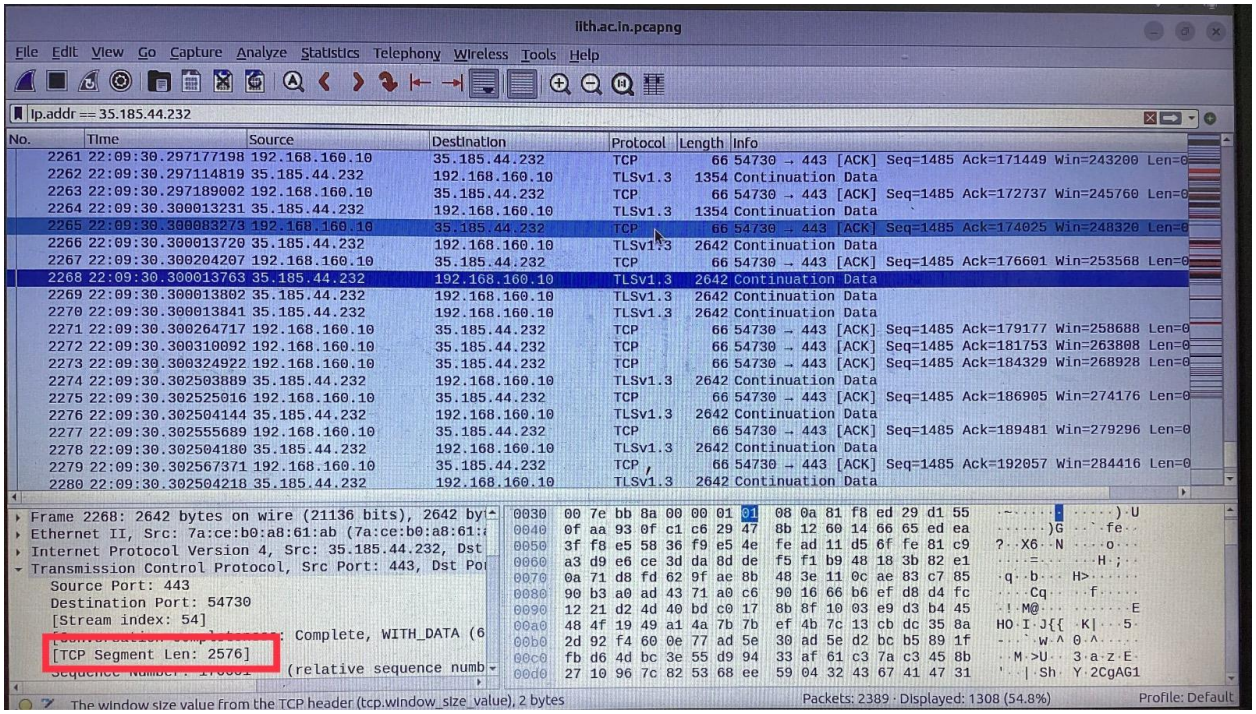
EstimatedRTT = (1 - α) * EstimatedRTT + α * SampleRTT
EstimatedRTT = RTT of first segment = 0.309838047 seconds
SampleRTT = RTT of current segment = 0.343295190 seconds
α = 0.125

EstimatedRTT = (1 - α) * EstimatedRTT + α * SampleRTT
EstimatedRTT = (1 - 0.125) * 0.309838047 + 0.125. * 0.343295190
EstimatedRTT = 0.3140205601 seconds

7.

The length (header plus payload) of each of the first four data-carrying TCP segments are **1288, 1288, 2576, and 2576 Bytes.**Given below the figure shows it.

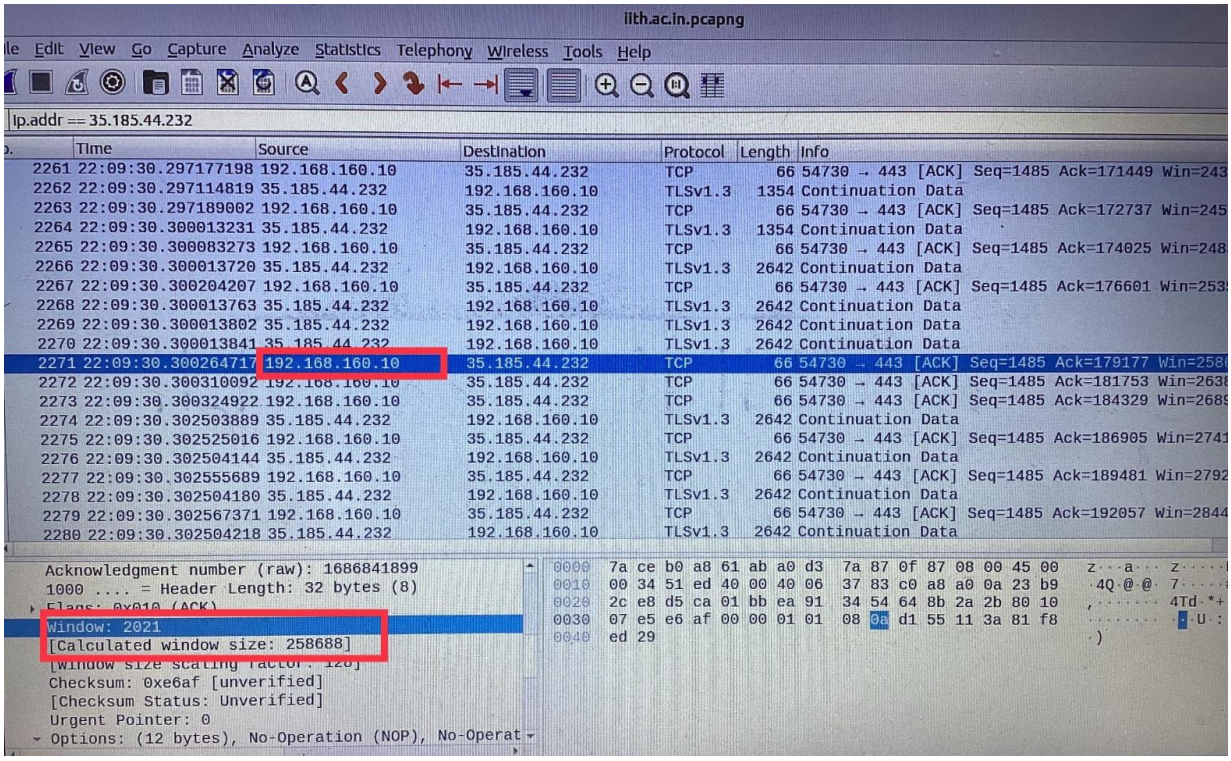




8.

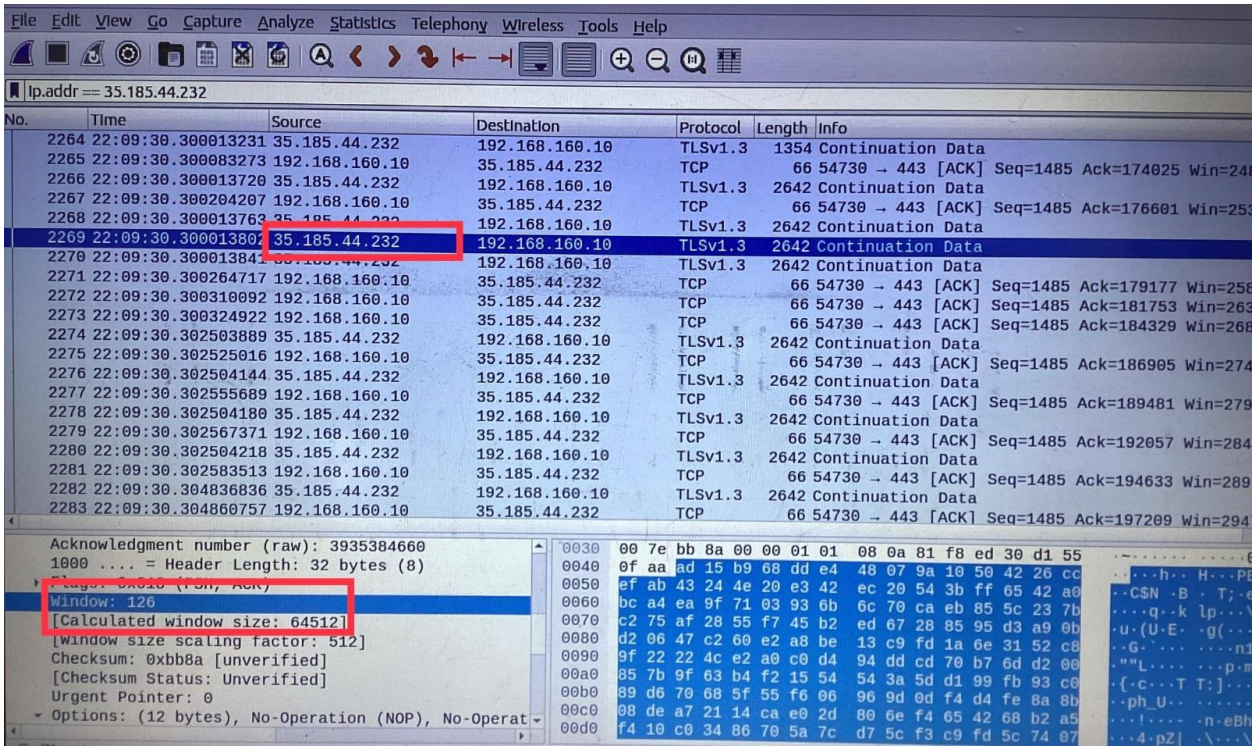
The minimum amount of available buffer space advertised by the client to ‘cse.iith.ac.in’ among these first five data-carrying TCP segments is **2021** which is equivalent to 2021 * 128 = **258688 Bytes**

No, the lack of receiver buffer space ever throttles the sender for these first five data carrying segments because the advertised window size has ample buffer space for smooth communication between sender and receiver as the sender is sending packets of 2576 Bytes. Given below the figure shows it.



9.

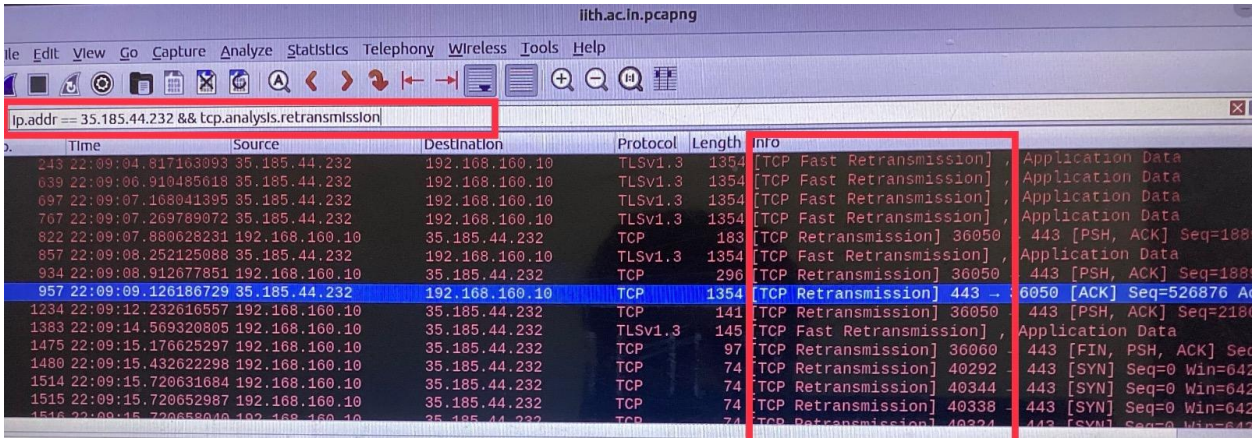
The minimum amount of available buffer space advertised by the ‘cse.iith.ac.in’ to client is **126** which is equivalent to 126*512 = **64512 Bytes**. Given below the figure shows it.



The first observation is that each TCP segment by ‘cse.iith.ac.in’ to client has the same window size whereas window size varies for the case of client to ‘cse.iith.ac.in’. One of the reasons for the given observation would be that data is being sent from ‘cse.iith.ac.in’ to the client which leads to a lesser buffer because of delay in processing. There is no data sent from client to ‘cse.iith.ac.in’ and hence the buffer is empty all the time.

10.

Yes, there are retransmitted segments in the trace file. The retransmitted segments are found using the filter “ip.addr == 35.185.44.232 && tcp.analysis.retransmission”.



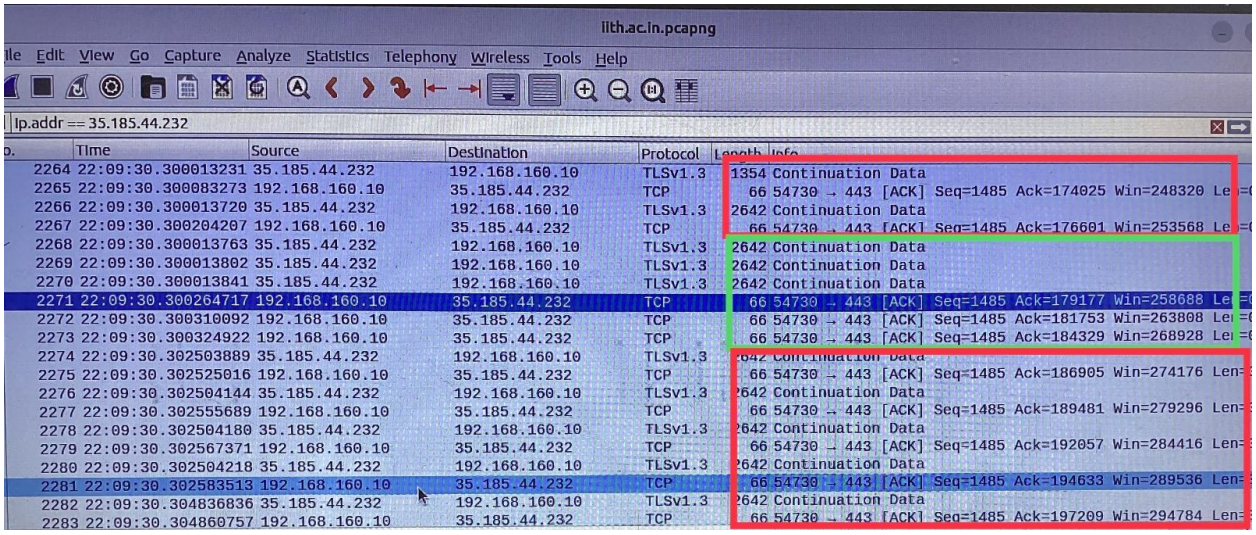
11.

Typically ACK acknowledges 1460 Bytes of data among the first ten data-carrying segments sent from the cse.iith.ac.in to client. Also, the segment size of each ACK segment is 0 Bytes.

There are certain instances where every packet is acknowledged and few acknowledgements are cumulative acknowledgments.

1st segment is acknowledged
2nd segment is acknowledged
3rd 4th and 5th are cumulative acknowledged
6th segment is acknowledged
7th segment is acknowledged
8th segment is acknowledged
9th segment is acknowledged
10th segment is acknowledged

Given below the figure showing details of the transmission:



OBJ

12.

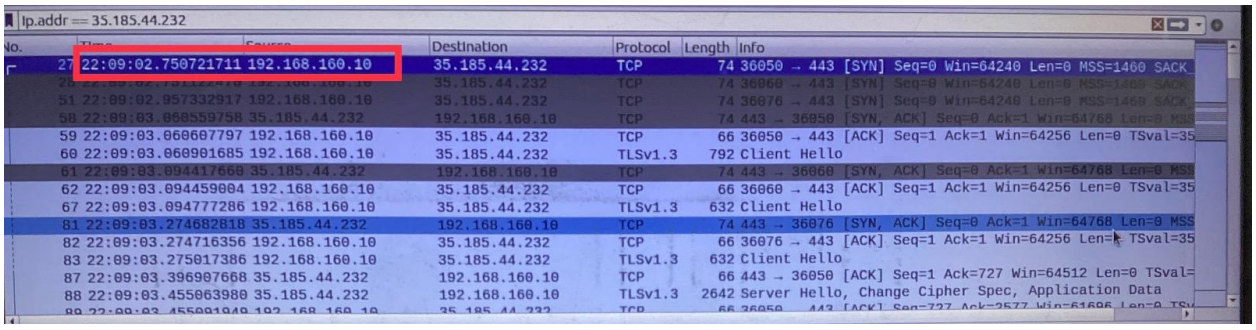
Initial time of establishing connection = 22:09:02.750721711
Time at which connection closed = 22:09:36.723582483
Total data sent = 1424 Kbytes

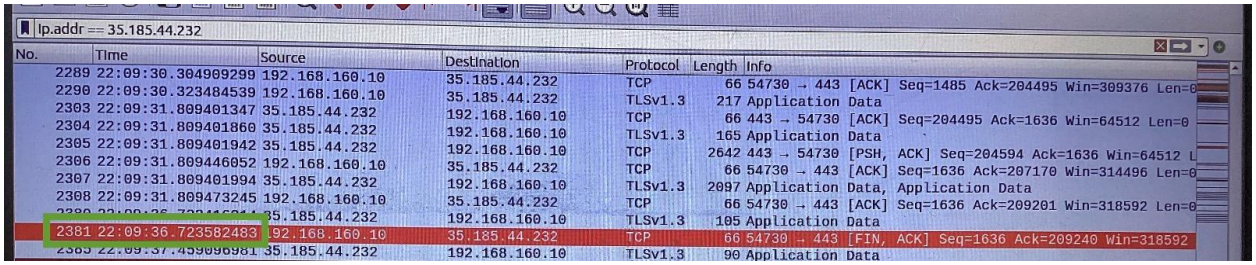
(approx page size of home page of ‘cse.ith.ac.in’ taken from analytics of third party website as the actual content is not found in wireshark may be due to TLS security)

Total time taken = 22:09:36.723582483 - 22:09:02.750721711
= 33.973510772 seconds

Throughput = data sent/ time taken
= 1424000 bytes / 33.973510772 seconds
= 41919.34 Bytes/second

Given below the figure showing details of the transmission:




ip.addr == 35.185.44.232
No. Time Source Destination Protocol Length Info
2289 22:09:30.304909299 192.168.160.10 35.185.44.232 TCP 66 54730 → 443 [ACK] Seq=1485 Ack=204495 Win=309376 Len=0
2290 22:09:30.323484539 192.168.160.10 35.185.44.232 TLSv1.3 217 Application Data
2303 22:09:31.809401347 35.185.44.232 192.168.160.10 TCP 66 443 → 54730 [ACK] Seq=204495 Ack=1636 Win=64512 Len=0
2304 22:09:31.809401860 35.185.44.232 192.168.160.10 TLSv1.3 165 Application Data
2305 22:09:31.809401942 35.185.44.232 192.168.160.10 TCP 2642 443 → 54730 [PSH, ACK] Seq=204594 Ack=1636 Win=64512 L
2306 22:09:31.809446052 192.168.160.10 35.185.44.232 TCP 66 54730 → 443 [ACK] Seq=1636 Ack=207170 Win=314496 Len=0
2307 22:09:31.809401994 35.185.44.232 192.168.160.10 TLSv1.3 2097 Application Data, Application Data
2308 22:09:31.809473245 192.168.160.10 35.185.44.232 TCP 66 54730 → 443 [ACK] Seq=1636 Ack=209201 Win=318592 Len=0
2381 22:09:36.723582483 192.168.160.10 35.185.44.232 TCP 66 54730 → 443 [FIN, ACK] Seq=1636 Ack=209240 Win=318592
35.185.44.232 192.168.160.10 TLSv1.3 90 Application Data