LAB 2 Assignment:

1. Capturing a bulk TCP transfer from your computer to a remote server

Answering the following questions, by opening the Wireshark captured packet file Tcp- Ethereal-trace-1 in http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip

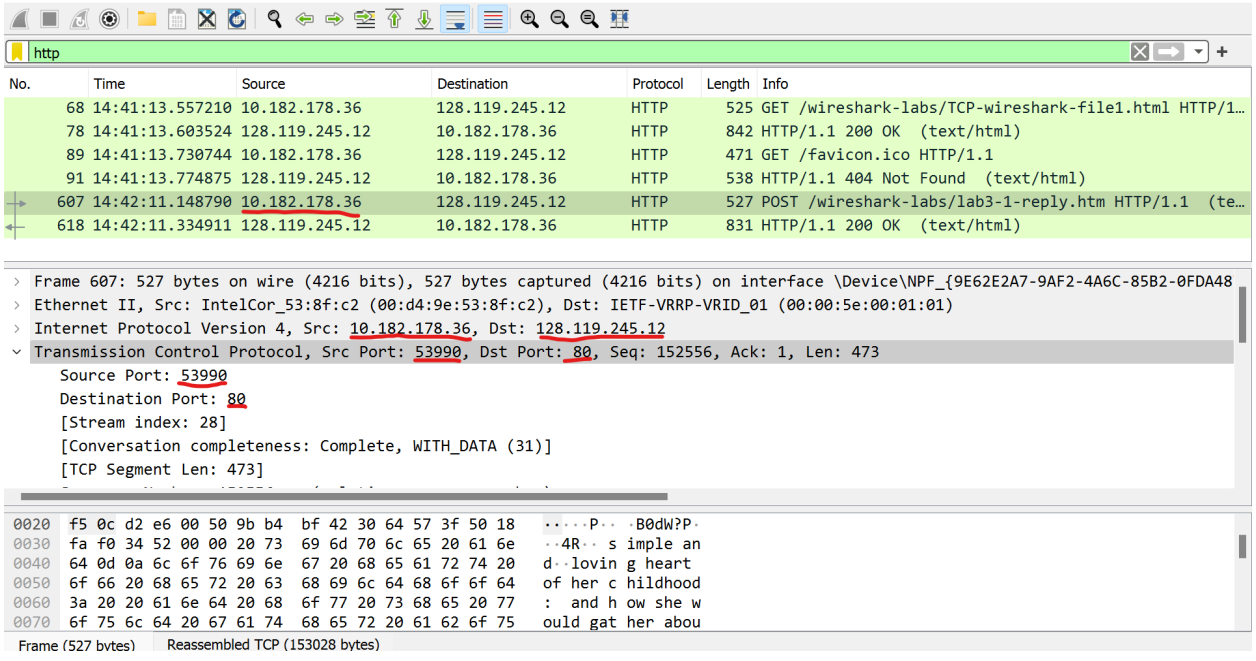
1. What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu?
→ The IP address of the client computer is 192.168.1.102 and the TCP port used is 1161

2. 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 the gaia.cs.umass.edu is 128.119.245.12 and the dest port used is 80.

Able to create my own trace:

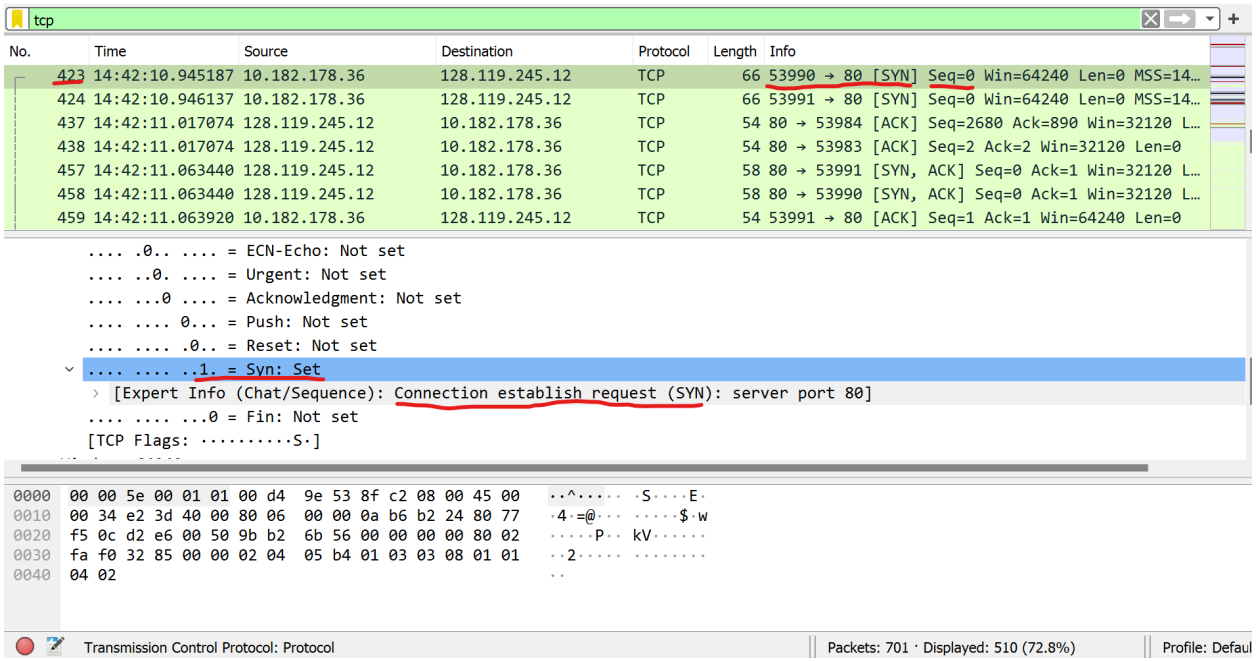
Using my own Trace:

3. What is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu?
The IP address of my client computer is 10.182.178.36 and the tcp port used is 53990.



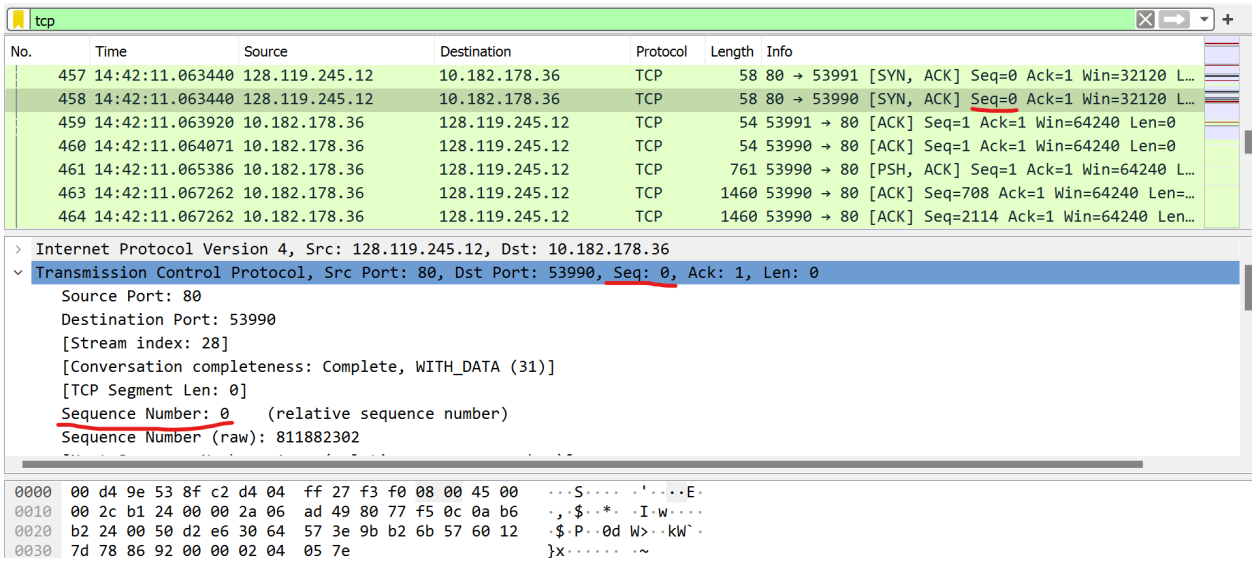
4. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in the segment that identifies the segment as a SYN segment?
→ The sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu to upload the file is 0.

→ In the flag section of the TCP segment, the SYN bit set to 1 identifies the segment as SYN segment. (marked red in the screenshot)



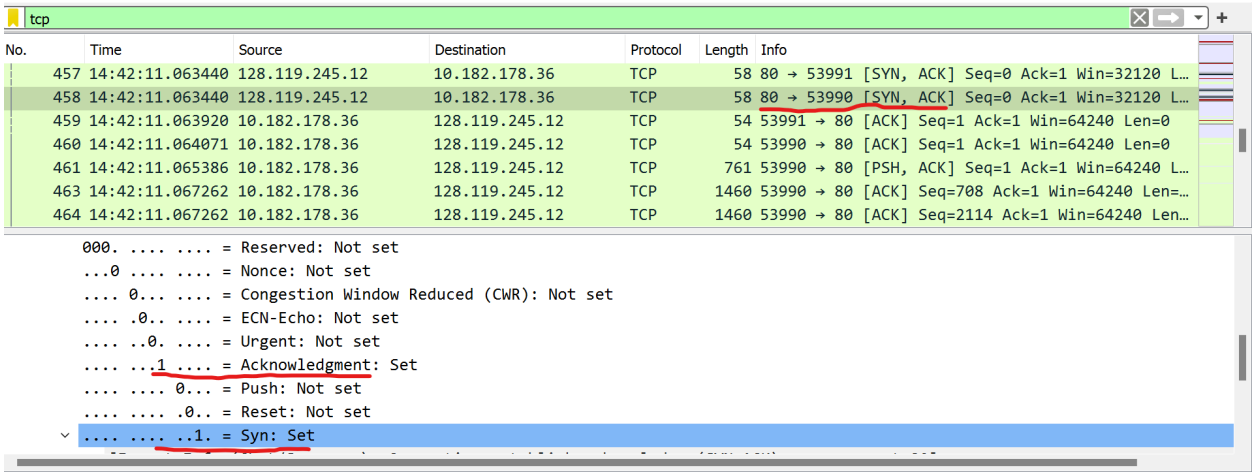
5. What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN?

→ The sequence number of SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN is 0



What is the value of the Acknowledgement field in the SYNACK segment?

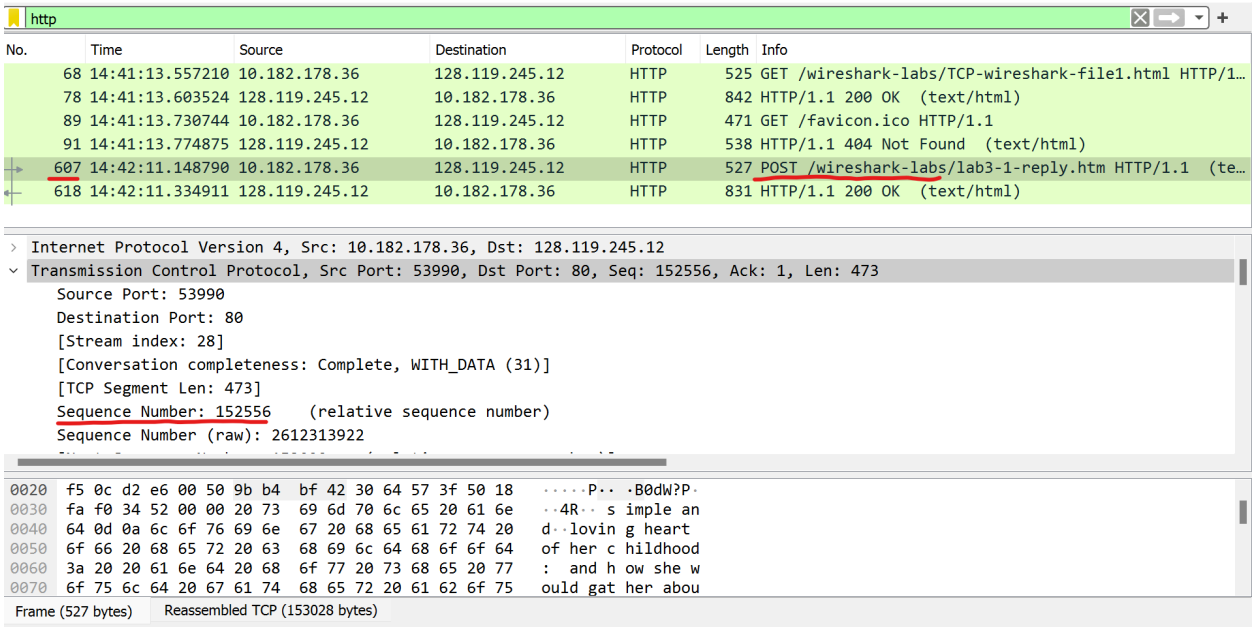
→ the value of the Acknowledgement field in the SYNACK segment id 1.



How did gaia.cs.umass.edu determine that value?
→ Since the sequence number from the client 10.182.178.36 had initial seq number as 0, hence the Acknowledgement from the gaia.cs.umass.edu server for the received segment from the client hence seq number is 0 and The server adds 1 to the sequence number of SYN tcp segments to give value 1 and SYN bit is 1.

What is it in the segment that identifies the segment as a SYNACK segment?
→ the Segment that identifies the segment as a SYNACK segment is the flag field in the TCP segment has Acknowledge and Syn bit set to 1 (shown in above screenshot)

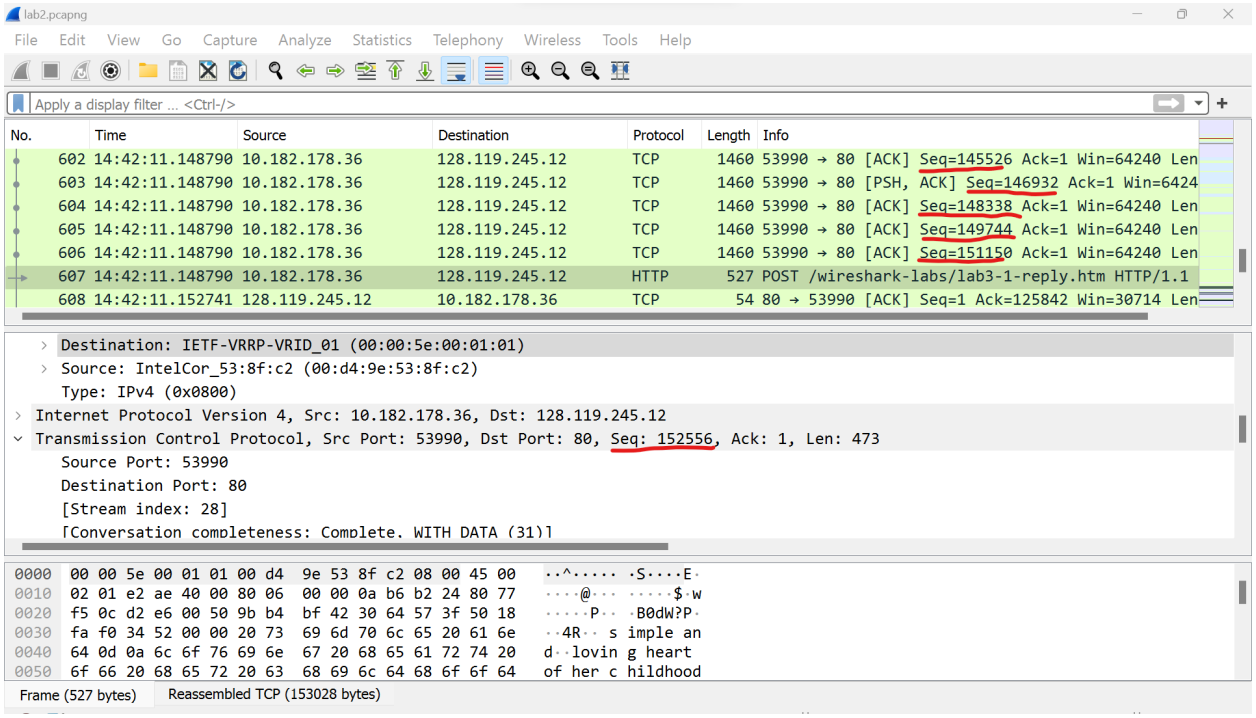
6. What is the sequence number of the TCP segment containing the HTTP POST command?
→ the sequence number of TCP segment containing the HTTP POST command is 152556



7. Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection. What are

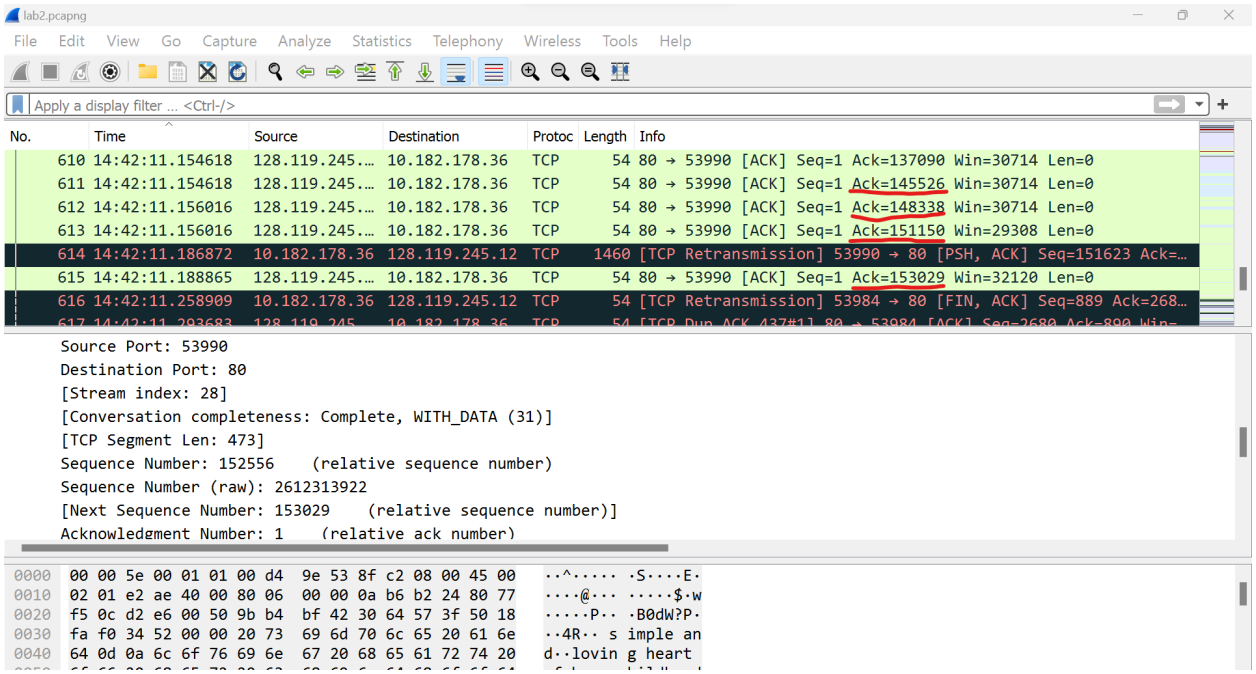
the sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST)?

→ The trace no of the first six segments are No: 602, 603, 604, 605,606,607
The sequence number for first six segments are 145526, 146932, 148338, 149744, 151150, 152556



At what time was each segment sent?
→ The segment with trace no 602, 603, 604, 605, 606, 607 was sent at 14:42:11.148790

When was the ACK for each segment received?
→ The Ack for segment with trace no 602 is trace no 611 with acknowledgement number as 145526.
The Ack for segment with trace no 603, 604 is trace no 612 with acknowledgement number as 148338.
The Ack for segment with trace no 605, 606 is trace no 613 with acknowledgement number as 151150.
The Ack for segment with trace no 607 is the trace no 615 with acknowledgement number as 153029.

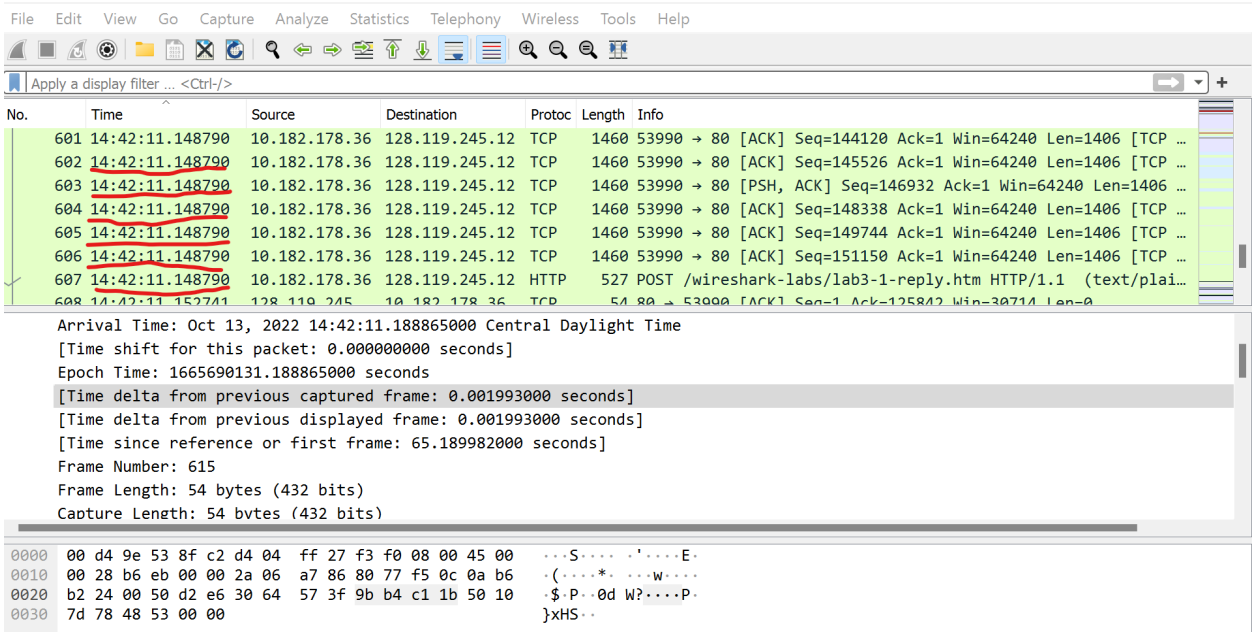


Given the difference between when each TCP segment was sent, and when its acknowledgement was received, What is the RTT value for each of the six segments?

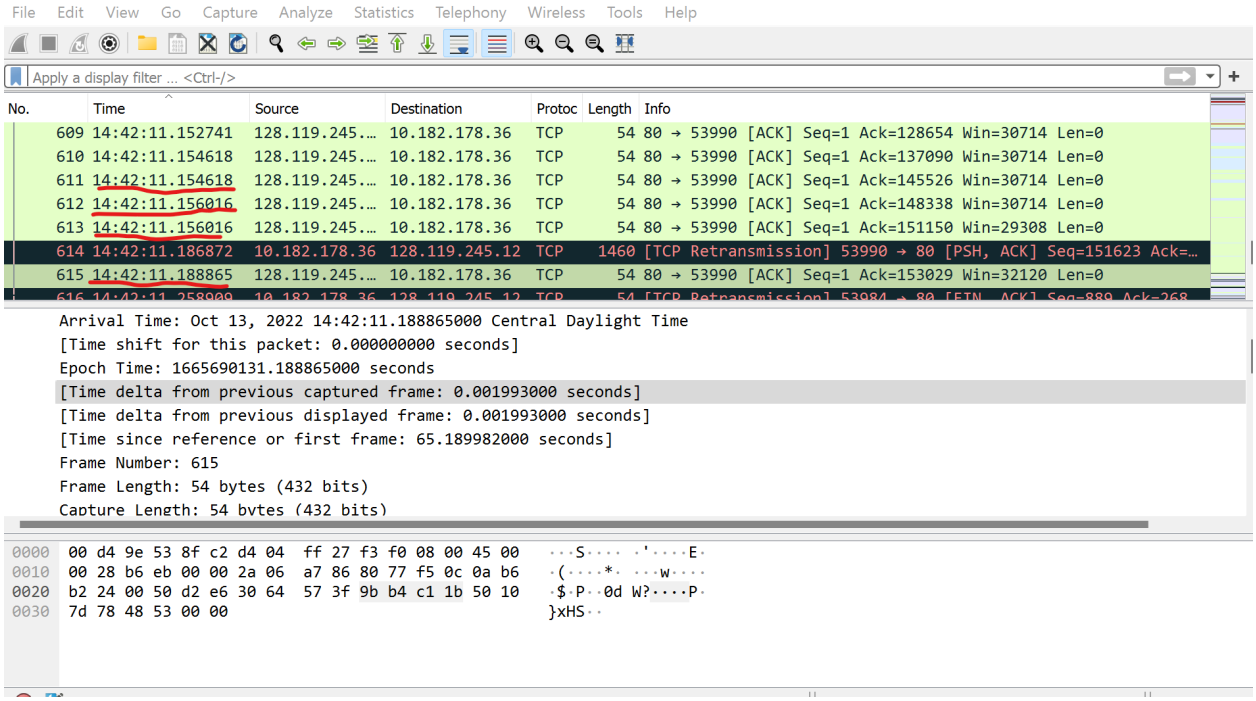
→

Segment Trace no	Sent time	Acknowledged time	RTT in sec
602	14:42:11.148790	14:42:11.154618	0.005828
603	14:42:11.148790	14:42:11.156016	0.007226
604	14:42:11.148790	14:42:11.156016	0.007226
605	14:42:11.148790	14:42:11.156016	0.007226
606	14:42:11.148790	14:42:11.156016	0.007226
607	14:42:11.148790	14:42:11.188865	0.040075

Sent time screenshot:



Acknowledgement time screenshot:



What is the EstimatedRTT value after the receipt of each ACK? Assume that the value of the EstimatedRTT is equal to the measured RTT for the first segment,

→

The formula for the EstimatedRTT is EstimatedRTT = 0.875 * EstimatedRTT + 0.125 * SampleRTT

EstimatedRTT after the receipt of the ACK of segment 1:

EstimatedRTT = 0.875 * 0.005828 + 0.125 * 0.005828
= 0.0050995 + 0.0007285
= 0.005828 sec

EstimatedRTT after the receipt of the ACK of segment 2:
EstimatedRTT = 0.875 * 0.005828 + 0.125 * 0.007226
= 0.00600275 sec

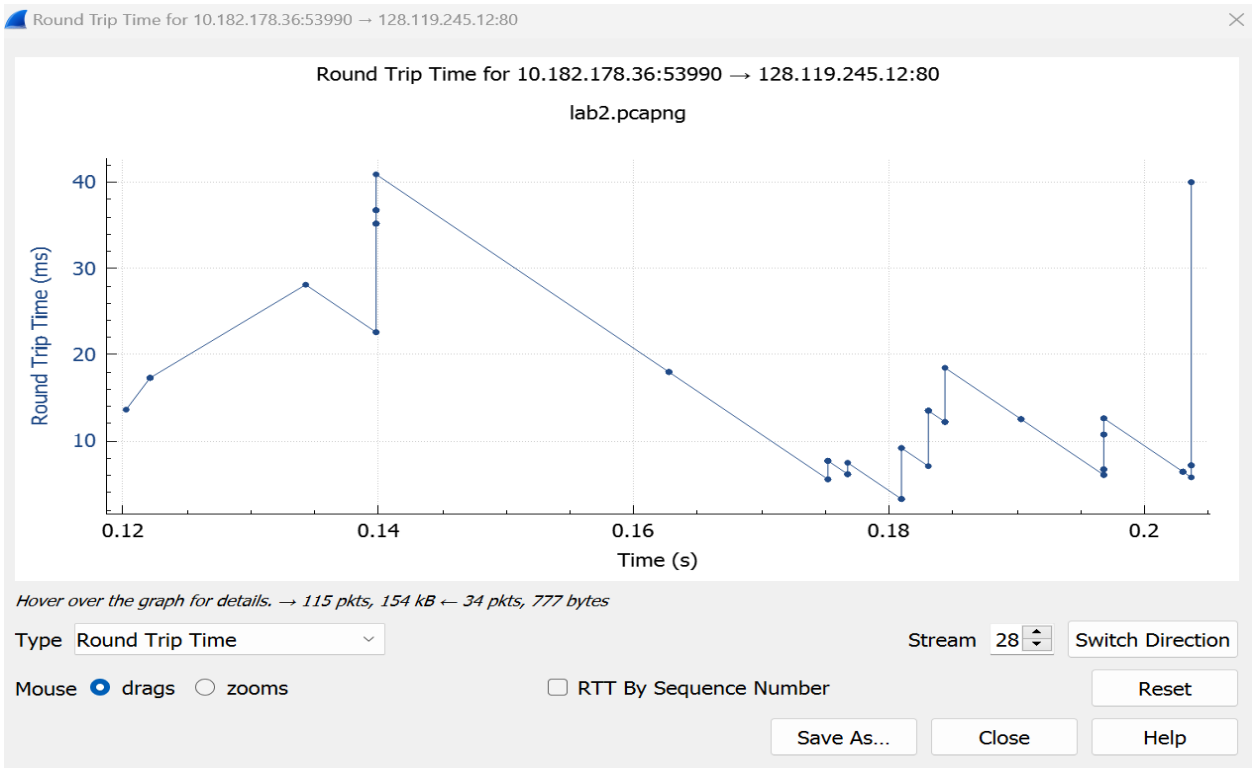
EstimatedRTT after the receipt of the ACK of segment 3:
EstimatedRTT = 0.875 * 0.00600275 + 0.125 * 0.007226
= 0.0061556563 sec

EstimatedRTT after the receipt of the ACK of segment 4:
EstimatedRTT = 0.875 * 0.0061556563 + 0.125 * 0.007226
= 0.0062894493 sec

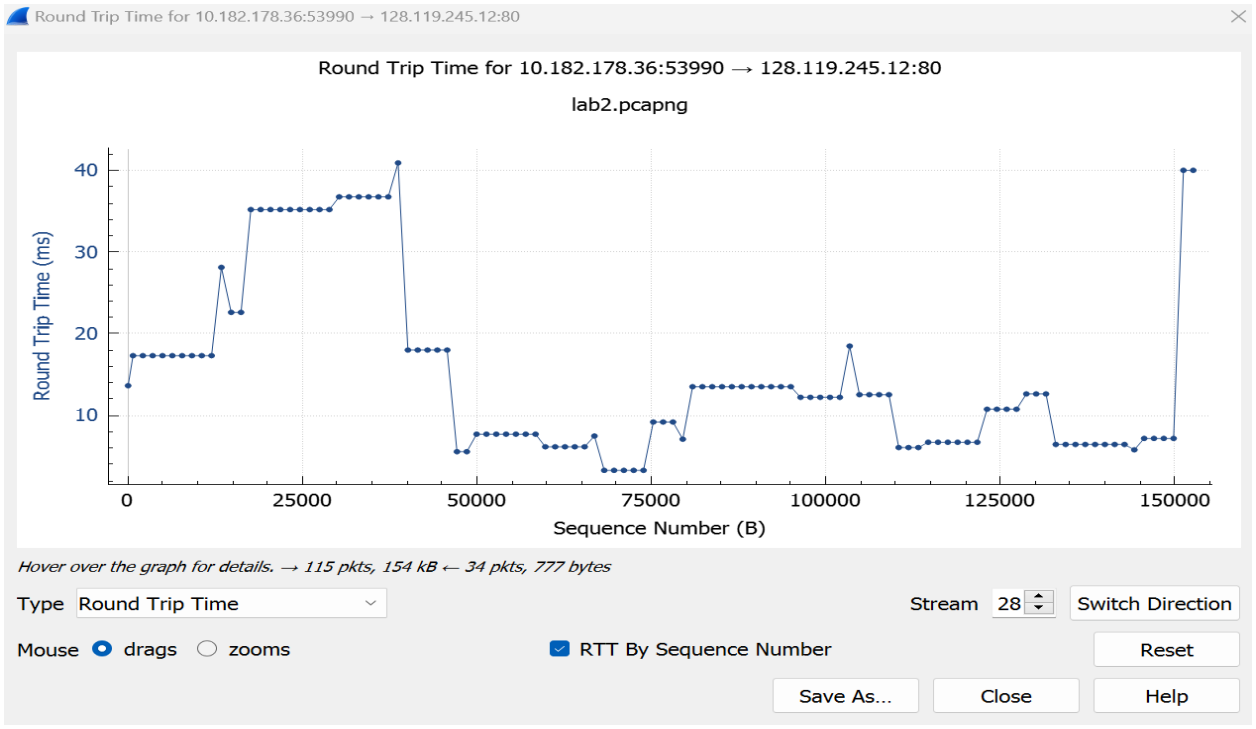
EstimatedRTT after the receipt of the ACK of segment 5:
EstimatedRTT = 0.875 * 0.0062894493 + 0.125 * 0.007226
= 0.0064065181 sec

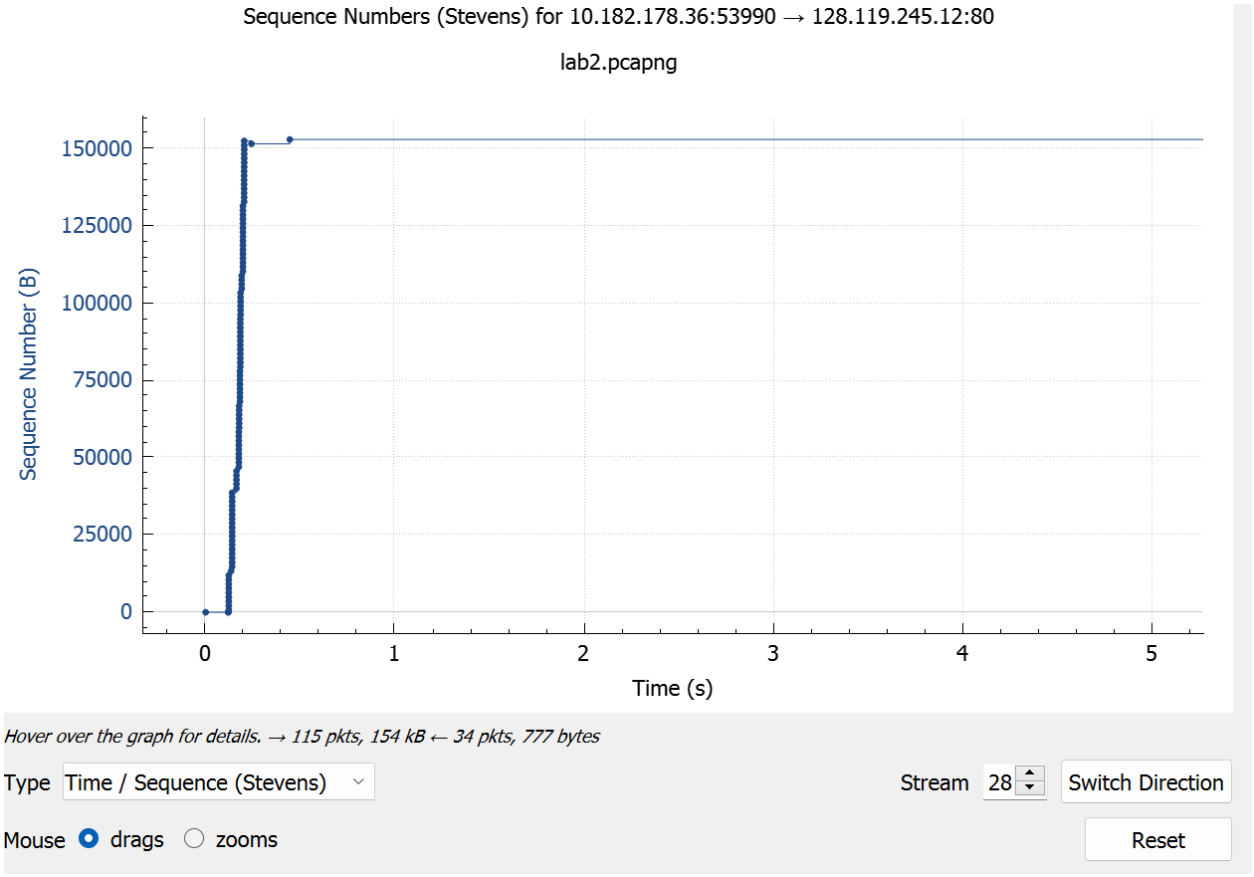
EstimatedRTT after the receipt of the ACK of segment 6:
EstimatedRTT = 0.875 * 0.0064065181 + 0.125 * 0.040075
= 0.0106150783 sec

Graph:
The Round Trip time 10.182.178.36:53990 to 128.119.245.12:80 where X-axis is Time in s and Y axis is RTT in ms



The Round Trip time 10.182.178.36:53990 to 128.119.245.12:80 where X-axis is Sequence number in B and Y axis is RTT in ms

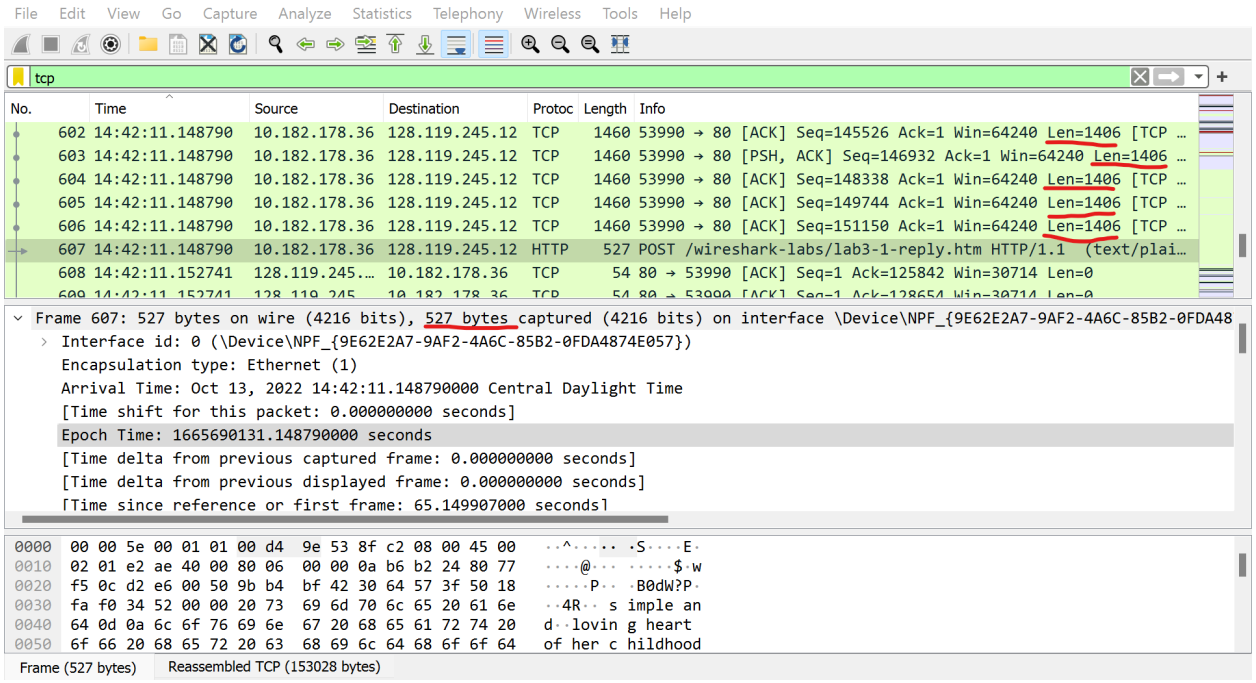




8.What is the length of each of the first six TCP segments?

→

Segment Trace No	**Length in bytes**
602	1460
603	1460
604	1460
605	1460
606	1460
607	527 – HTTP Post

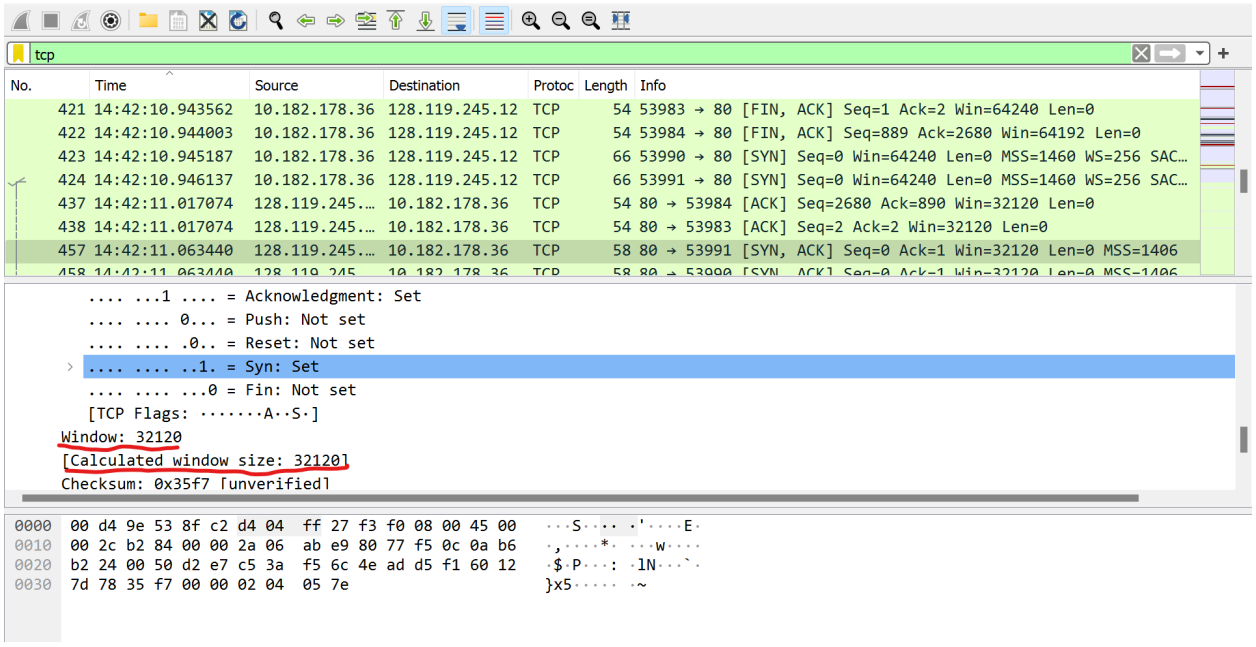


9.What is the minimum amount of available buffer space advertised at the received for the entire trace? Does the lack of receiver buffer space ever throttle the sender?

→ The minimum amount of available buffer space advertised at the received for the entire trace is seen in the first ACK Received from the server, Its value is 32120

This receiver window grows until it reaches the maximum receiver buffer size.

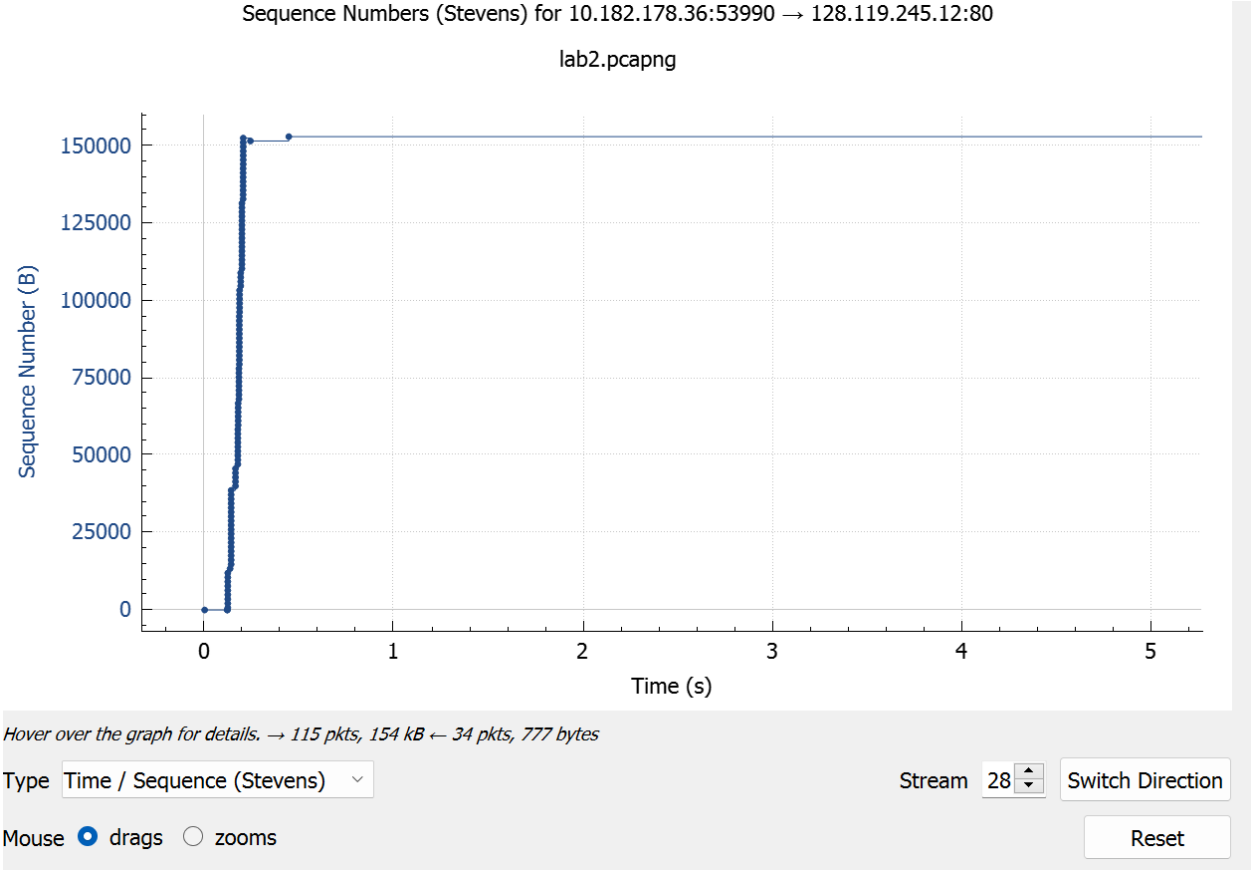
According to the trace, the sender is never throttled due to lacking of receiver buffer space.



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 segments in the trace file. Because in the time sequence graph (stevens), all sequence numbers are monotonically increasing

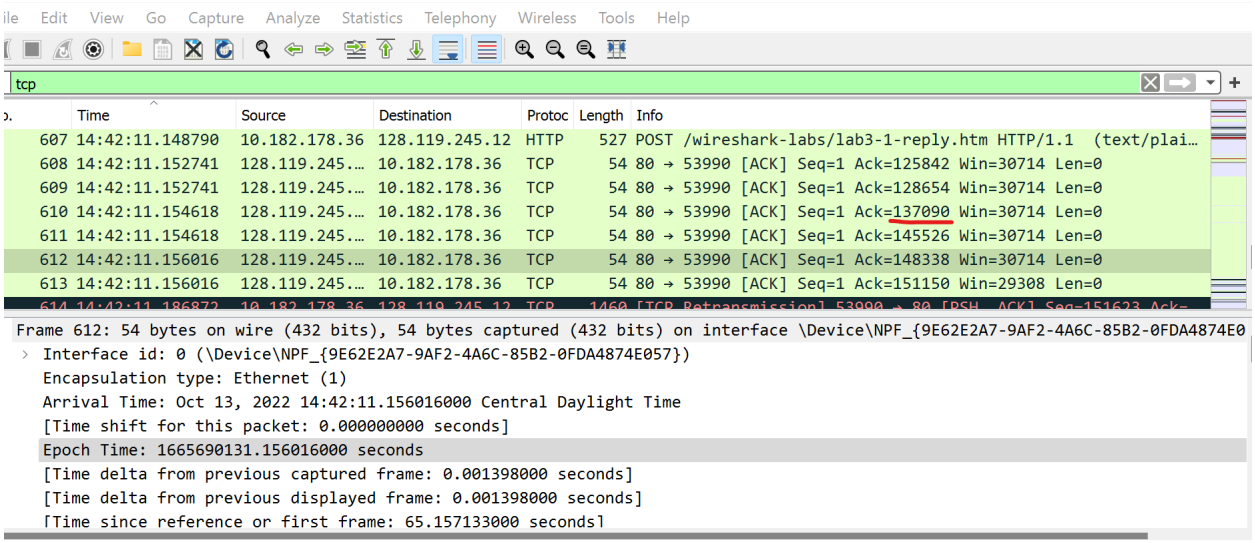


11. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment

Trace No	**Ack Sequence number**	**Acknowledged data in bytes**
477	708	708
479	4926	4218
480	9144	4218
481	13362	4218
504	17580	4218
510	21798	4218
511	26016	4218

512	30234	4218
522	38670	8436
530	47106	8436
531	49918	2812
540	61166	11248
541	66790	5624
554	72414	5624
555	75226	2812
562	78038	2812
563	80850	2812
569	89286	8436
570	97722	8436
571	103346	5624
589	106158	2812
590	114594	8436
599	120218	5624
600	123030	2812
608	125842	2812
609	128654	2812
610	137090	8436
611	145526	8436
612	148338	2812
613	151150	2812
615	153029	1879

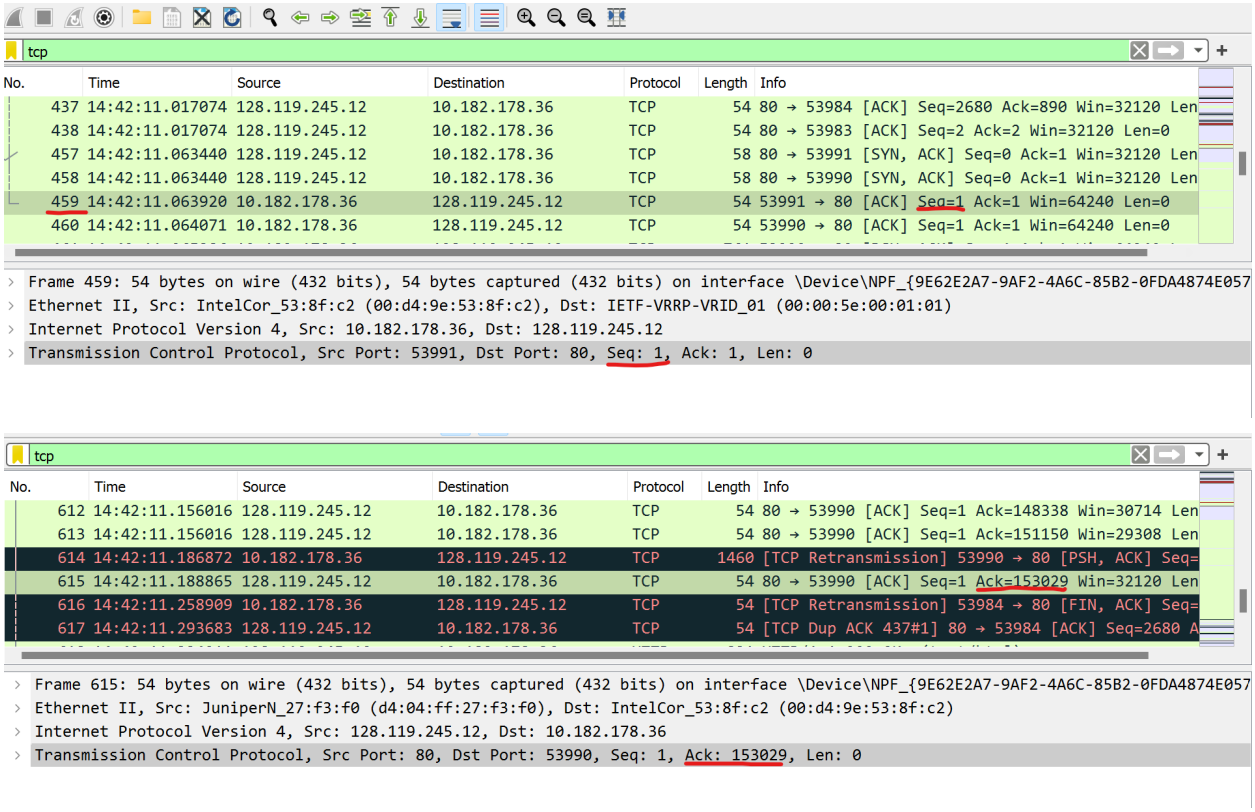
→ One of the case where the receiver is ACKing every other received segment is
In the Ack sequence number 137090 acknowledges the 3 times the 2812.



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

→ The Tcp throughput for the tcp connection is calculated by considering the Total bytes transferred per unit time. So the total amount of data transmitted can be computed by the difference between the sequence number of first tcp Segment (ie value is 1) and the last ACK acknowledged sequence number (value is 153029)

The value is shown in the screenshot below.



The time difference is calculated by the difference between the time of the first TCP segment (ie 14.42.11:063920 sec) And the last ACK segment(ie 14.42.11.188865 secs) as shown in the above screenshot.

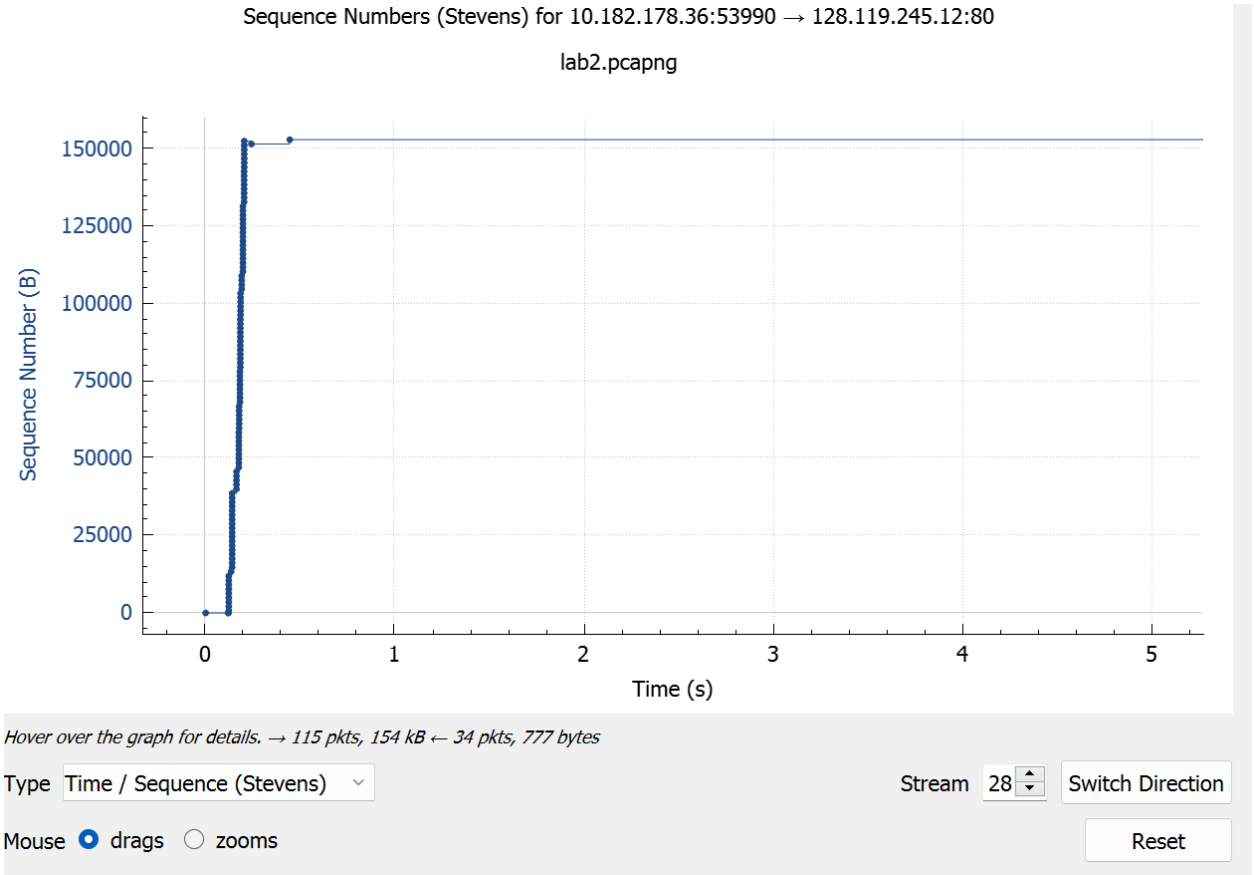
Throughput = $\frac{\text{total amount of data in bytes}}{\text{Total amount taken in seconds}} = \frac{153029 - 1}{0.188865-0.063920} = \frac{153028 \text{ bytes}}{0.124945 \text{ secs}}$ = 1224762.8956 bytes/sec

= 1.224 Megabytes/secs

TCP congestion control in action:

Time-Sequence-Graph(Stevens):

Using my own Trace:

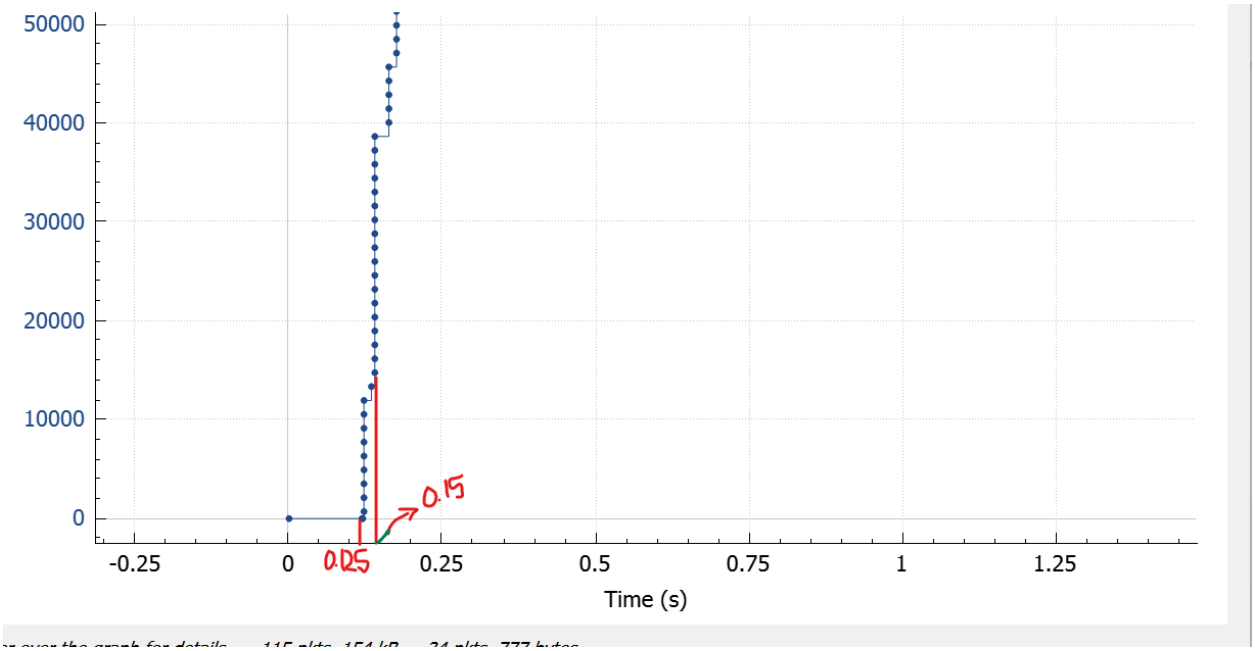


13. Use the Time-Sequence-Graph(Stevens) plotting tool to view the sequence number versus time plot of segments being sent from the client to the gaia.cs.umass.edu server. Can you identify where TCP's slow start phase begins and ends, and where congestion avoidance takes over? Comment on ways in which the measured data differs from the idealized behavior of TCP that we've studied in the text.

→ The slow start of the TCP seems to begin at about 0.125 seconds and then ends at about 0.15 seconds. The congestion avoidance takes place for 0.025 (0.15-0.125) secs because it cuts the amount of data to be sent.

The measured data is only using a fraction of the window size instead of the idealized 1/3 to a half.

The zoomed screenshot of the graph is attached below:



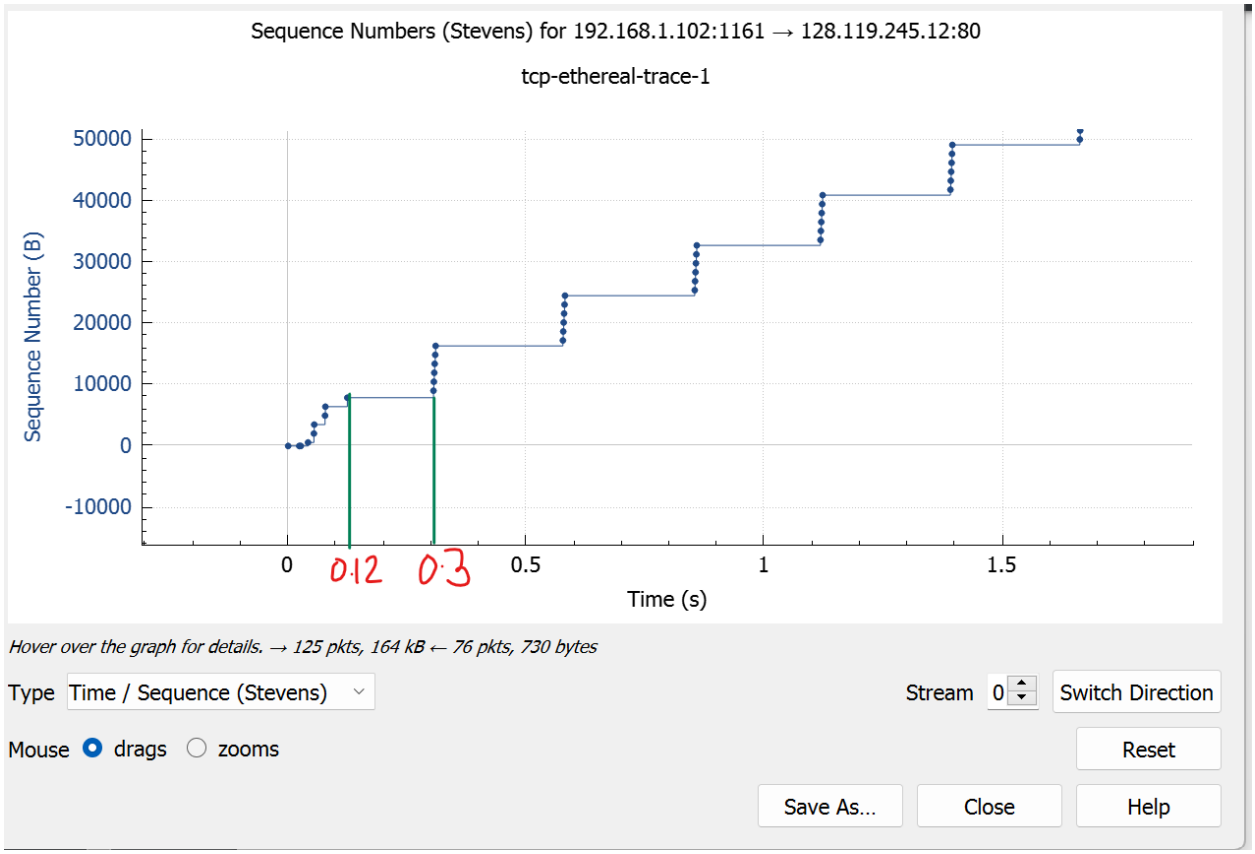
14. Answer each of two questions above for the trace that you have gathered when you transferred a file from your computer to gaia.cs.umass.edu
→ the questions have been answered.

From the Given Trace tcp-ethereal-trace-1

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 where TCP's slow start phase begins and ends, and where congestion avoidance takes over? Comment on ways in which the measured data differs from the idealized behavior of TCP that we've studied in the text.

→ The slow start of the TCP seems to begin at about 0.12 seconds and then ends at about 0.3 seconds.
The congestion avoidance takes place for 0.18 (0.3-0.12) secs because it cuts the amount of data to be sent.
The measured data is only using a fraction of the window size instead of the idealized 1/3 to a half.

The screenshot is attached below:



14. Answer each of two questions above for the trace that you have gathered when you transferred a file from your computer to gaia.cs.umass.edu

→ the questions have been answered.