Wireshark for Transport Layer Protocols

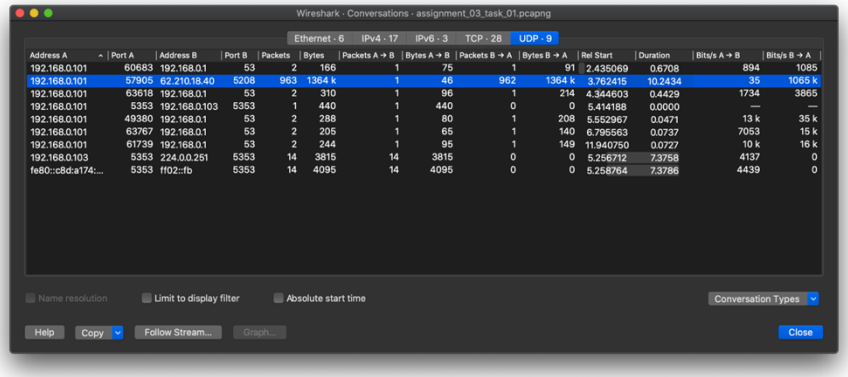
Advanced Computer Networks

Name: Bhavishya Sharma
Roll No: CS20MTECH12006

Task 01 - iperf3 to remote server in UDP

Question 01: How many UDP packets are exchanged in this communication between iperf3 client and remote server ?

Answer 01: Total UDP Packets = 963
Client to Server = 1
Server to Client = 962



Question 02: Who is sending bulk data to whom? What is the average size of the packet sent?

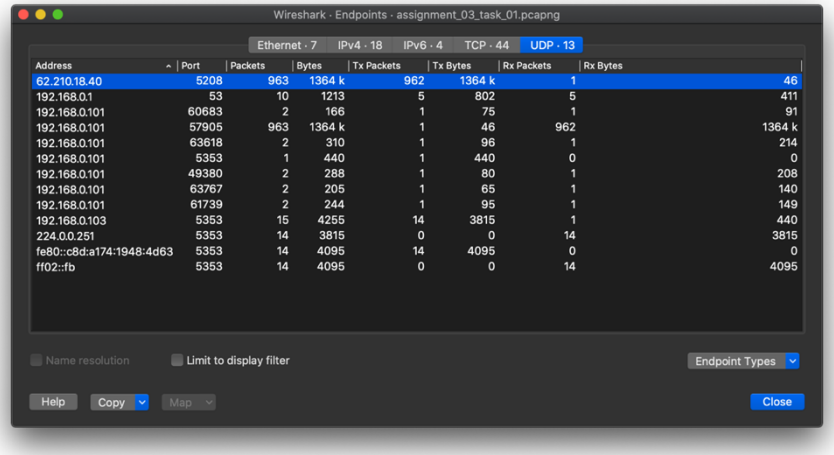
Answer 02: Server (62.210.18.40) is sending bulk data to client (192.168.0.101).

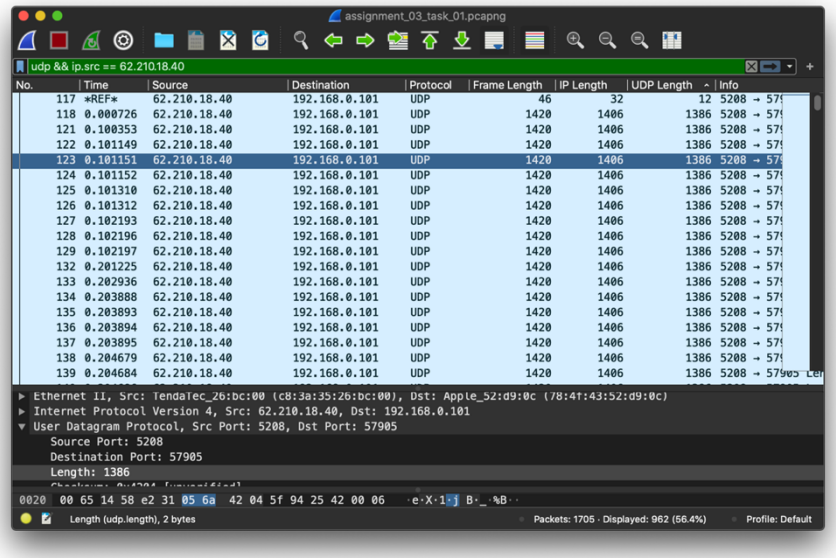
Average Frame Size = Total Bytes Received / Total Number of Packets
= 1364 kB / 962
= 1417.87942 bytes (Approx)

Average Ethernet Frame Size = ((1 x 46) + (961 x 1420)) / 962
= (46 + 1364620) / 962
= 1364666 / 962
= 1418.57173 bytes

Average IP Datagram Size = ((1 x 32) + (961 x 1406)) / 962
= (32 + 135166)) / 962
= 1351198 / 962
= 1404.57173 bytes

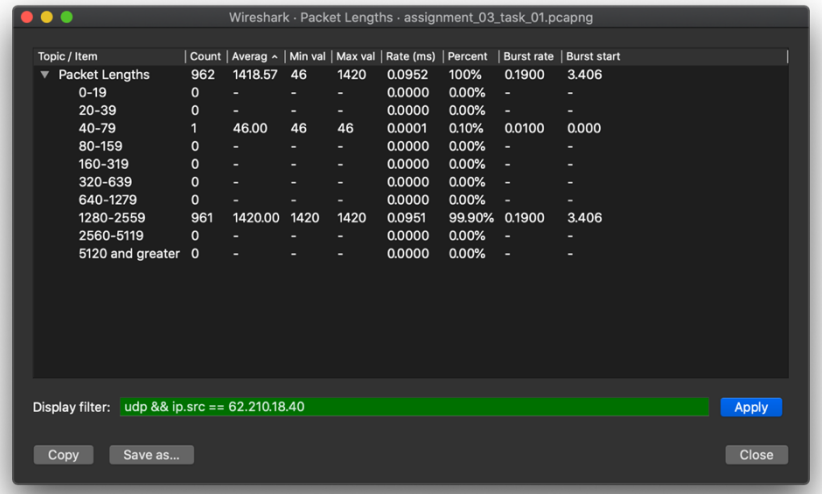
Average UDP Datagram Size = ((1 x 12) + (961 x 1386)) / 962
= (12 + 1331946) / 962
= 1331958 / 962
= 1384.57173 bytes

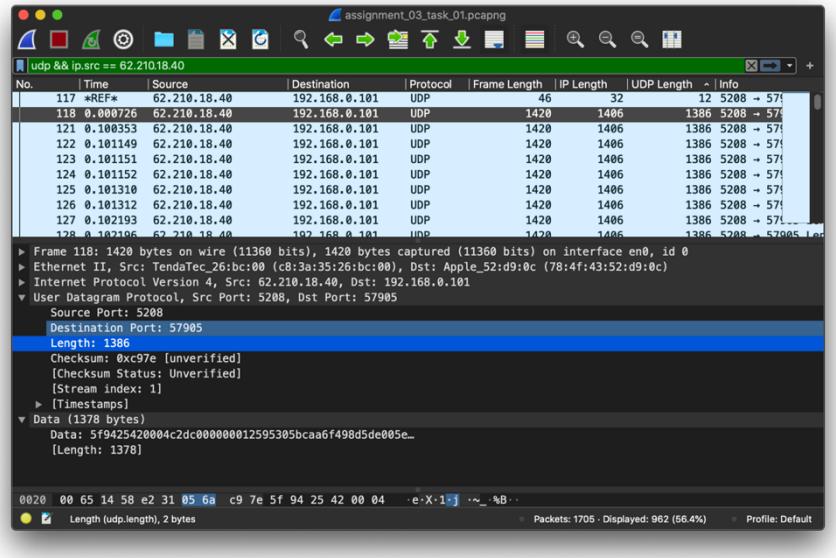




Question 03: Calculate the throughput (bytes transferred per unit time) for this UDP conversation using UDP's length field. Explain how you calculated this value using Wireshark capture in this experiment along with relevant screenshots. Verify your calculation with the one done by Wireshark using "Capture File properties" as well with the one displayed by iperf3 terminal. If you observe the major difference in your calculation and with the other two listed here, comment why and how?

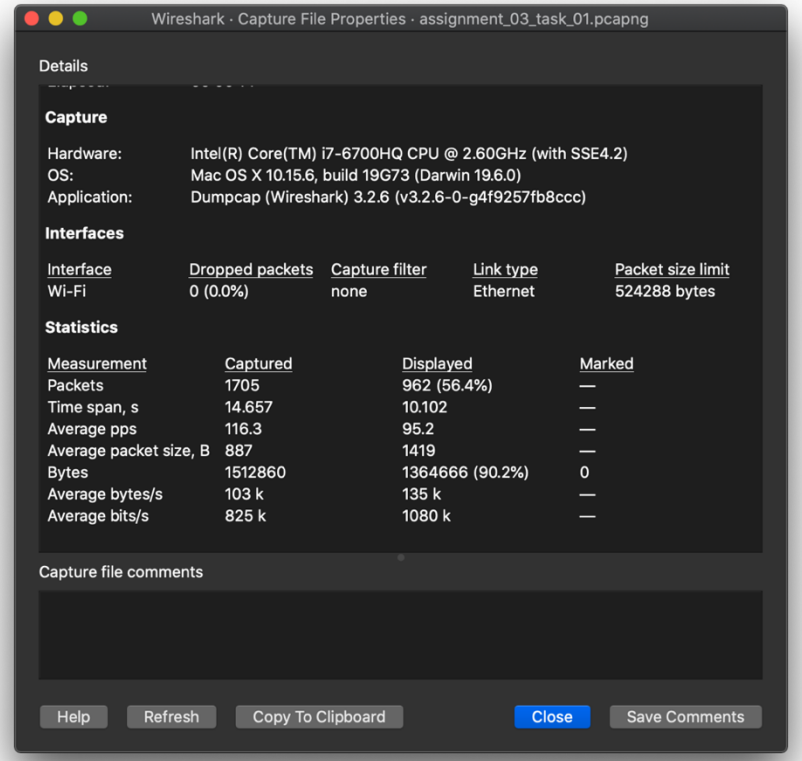
Answer 03: There is a fixed 14 Bytes Ethernet II header and 20 Bytes IP Header in each frame/packet.

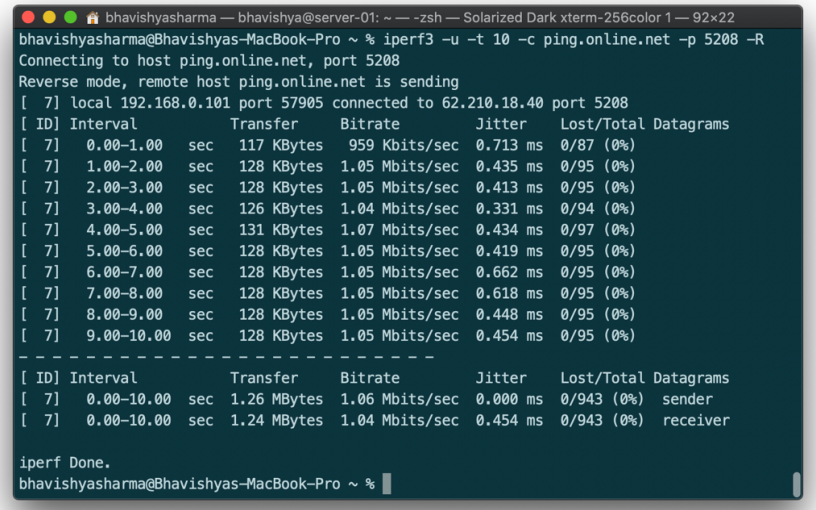




Total Packets Received	= 962
Total Bytes (UDP) Received	= (1 x (46-34)) + (961 x (1420-34))
	= (1 x 12) + (961 x 1386)
	= 12 + 1331946
	= 1331958 bytes
Total Time (Last Datagram)	= 10.102399 seconds
Throughput	= Total Bytes / Total Time
	= 1331958 / 10.102399
	= 131845.713 bytes/second
	= 131.845 kB/second
	= 1054 kb/second

Throughput using Capture File Properties	= 1080 kb/second
Throughput as per iperf3	= 1.04 mb/second





The throughput calculated by “Capture File Properties” is higher than the one calculated above and the one reported by iperf3 because, capture file properties calculates the speed with the Ethernet II Frame size and it also includes upload packets, while the above calculation is based on “UDP Length” field for download only and iperf uses UDP data size.

Throughput using Ethernet Frame Size = (46 + 961 * 1420) / 10.102399
= 1364666 / 10.102399
= 135.083 kB/second

= 1080.6 kb/second

Throughput using Data Size = (4 + 961 * 1378) / 10.102399
= 1324262 / 10.102399
= 131.083 kB/second
= 1048 kb/second
= 1.048 mb/second

Task 02 - Bulk File Download with TCP

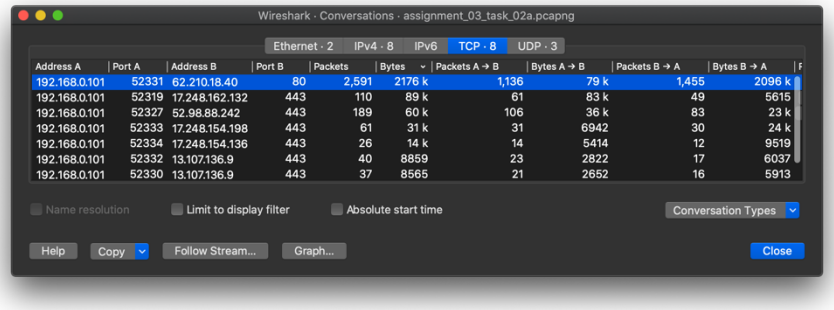
Start the Wireshark packet sniffer and start capturing. Visit http://ping.online.net/ on browser. From the bottom most list in the site “Download test files from this server :”.

a. Click on 2Mo.dat to download and save the respective file to your host machine successfully. Stop the wireshark capture and save the file for further analysis.
b. Click on 50Mo.dat to download and save the respective file to your host machine successfully. Stop the wireshark capture and save the file for further analysis.
c. Click on 200Mo.dat to download and save the respective file to your host machine. While this file gets downloaded cancel the download after 2-3 seconds. Stop the wireshark capture and save the file for further analysis.

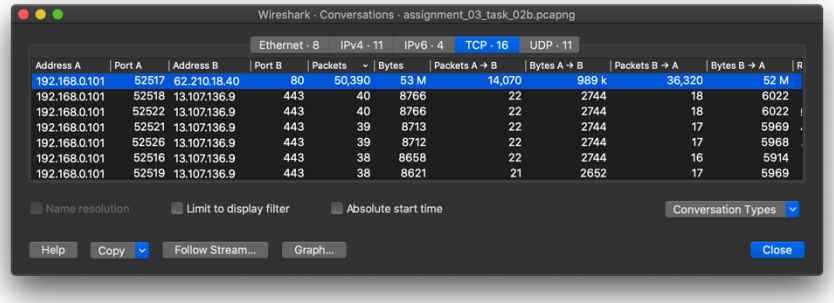
Question 01: How many TCP packets are exchanged in this communication client and remote server?

Answer 01: TCP Packets Exchanged:

a. Client = 192.168.0.101:52331, Server = 62.210.18.40:80
Client to Server = 1136
Server to Client = 1455
Total = 2591



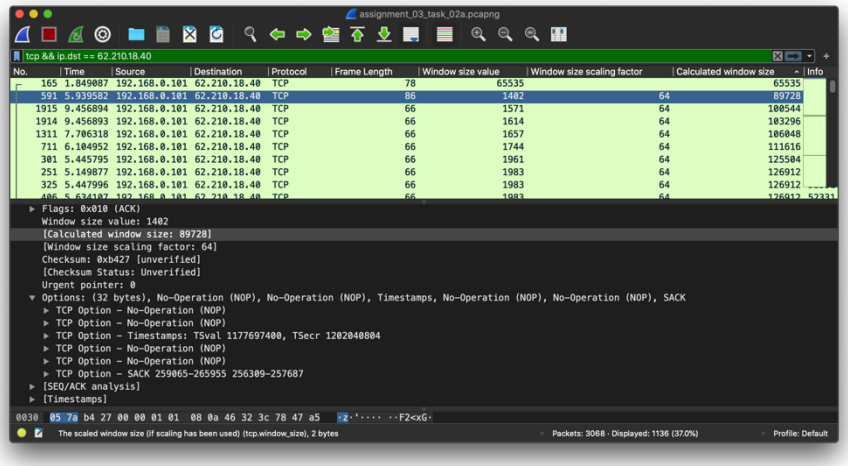
b. Client = 192.168.0.101:52517, Server = 62.210.18.40:80
Client to Server = 14070
Server to Client = 36320
Total = 50390



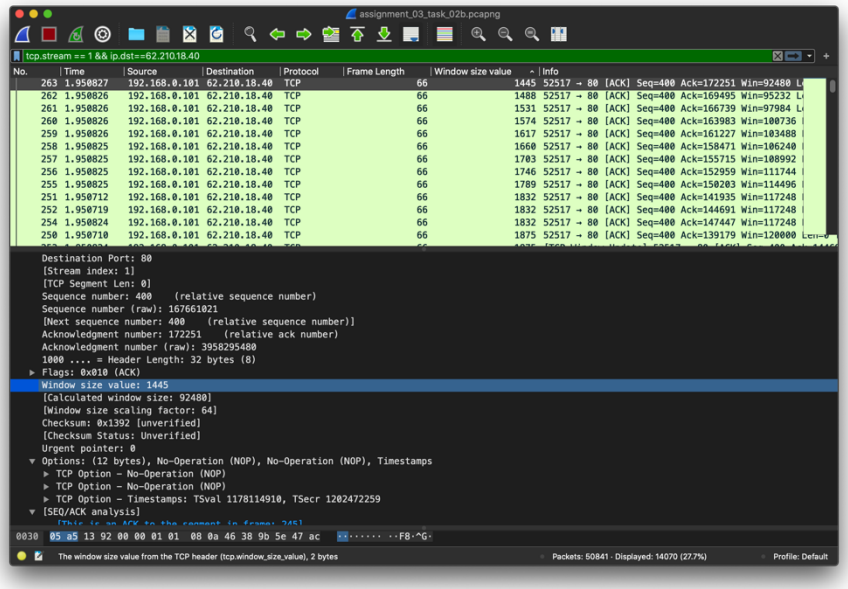
Question 02: What is the minimum amount of available buffer space advertised at the client/receiver for the entire trace?

Answer 02: Client Available Buffer Size = WindowSize * 2 ^ WindowScale. Window Scale is set via TCP Options in the SYN packet.

a. Minimum Window Size = 1402
   Window Scale = 6
   Minimum Buffer Size = 1402 * 2^6
   = 1402 * 64
   = 89728 bytes



b. Minimum Window Size = 1445
   Window Scale = 6
   Minimum Buffer Size = 1445 * 2^6
   = 1445 * 64
   = 92480



Question 03: Pick any 5 TCP segments from server to client which are not part of initial TCP connection establishment and final connection termination.

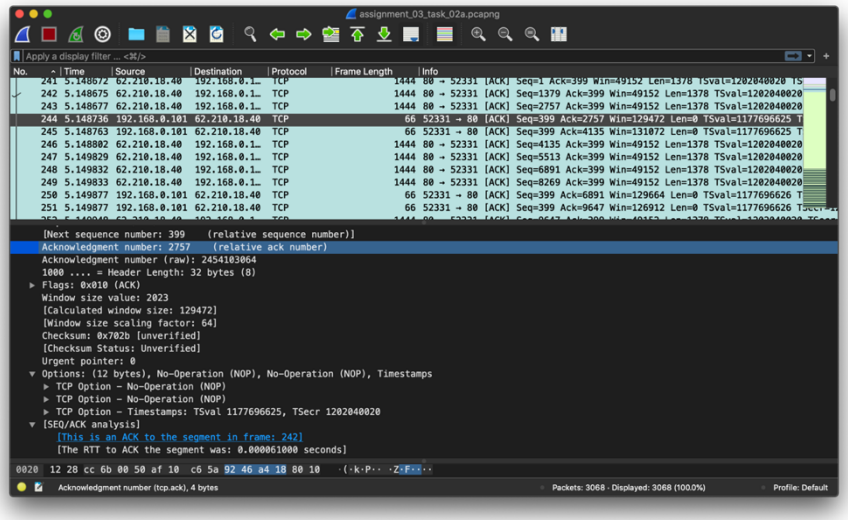
1. Make a table listing for each of these segments, the length of each of these TCP segments, the sequence number, time when the segment was sent, time when the respective ACK for each segment was received, length of the respective ACK segment. Place the screenshot of Wireshark of at least one such segment with respective ACK as a proof of observation and calculation. What is the maximum length out of all?
2. Given the difference between when each TCP segment was sent, and when its acknowledgement was received, what is the RTT value for each of these segments? 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, and then is computed using the EstimatedRTT equation (From chapter 3 of the referred text book in the class) for all subsequent segments. Place these calculated values appropriately in the table formed in #3.1 above.
3. Plot the RTT Graph for any TCP segment out of these using the graph feature of Wireshark. Plot another graph manually from the table above for Sample RTT and estimated RTT (Similar to “RTT samples and RTT estimates” graph from section “Round-Trip Time Estimation and Timeout” of the referred textbook in the class).
4. Comment on your understanding of Estimated RTT calculation and plotted RTT graphs.

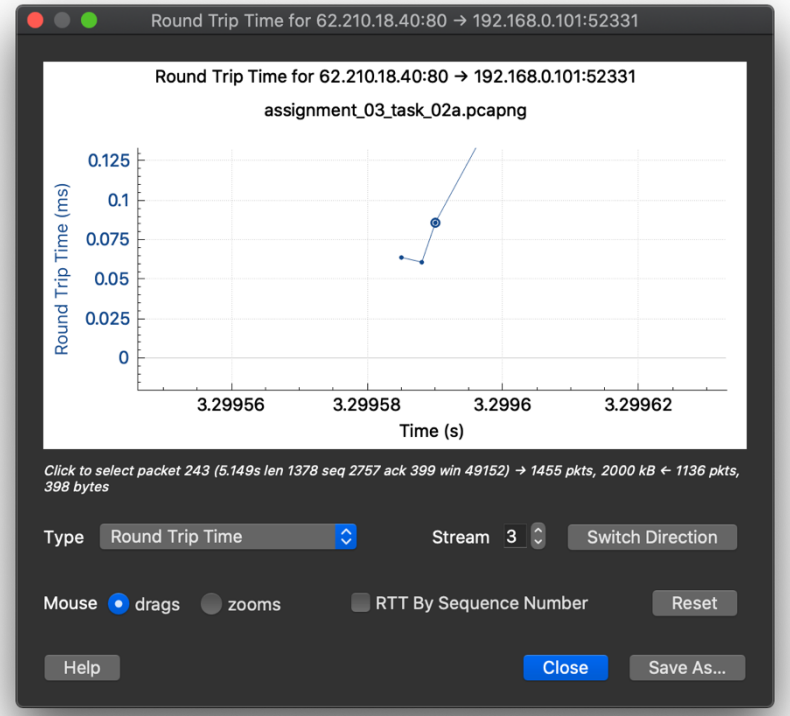
Answer 03: Part – A

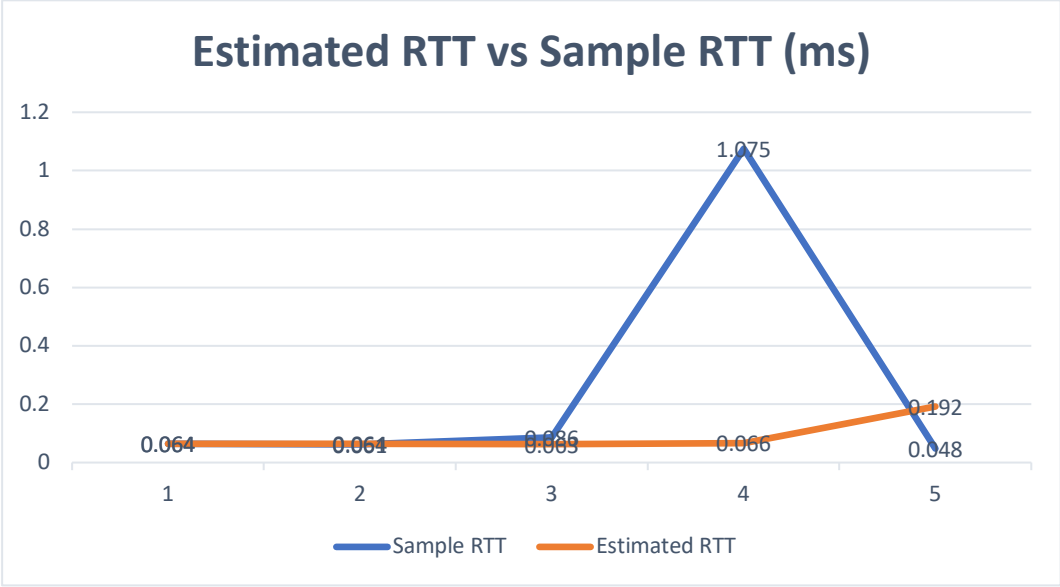
As we are capturing on client-side, for server to client communication, we have the timestamp at which packet was received at client and at which ack was sent by client. This RTT time does not have network distance delay. If we monitored client to server communication, this rtt will be significantly higher.

S.No.	Sent Time	SEQ	Length	ACK Time	ACK Len	RTT	Estimated RTT
1	5.148672	1 (2454100308)	1378	5.148736	0	0.000064	0.000064
2	5.148675	1379 (2454101686)	1378			0.000061	0.000064
3	5.148677	2757 (2454103064)	1378	5.148763	0	0.000086	0.000063
4	5.148802	4135 (2454104442)	1378	5.149877	0	0.001075	0.000066
5	5.149829	5513 (2454105820)	1378			0.000048	0.000192

Maximum length is the 1378 for data from server to client. For ACK from client to server, length is 0 i.e. no data, header only.





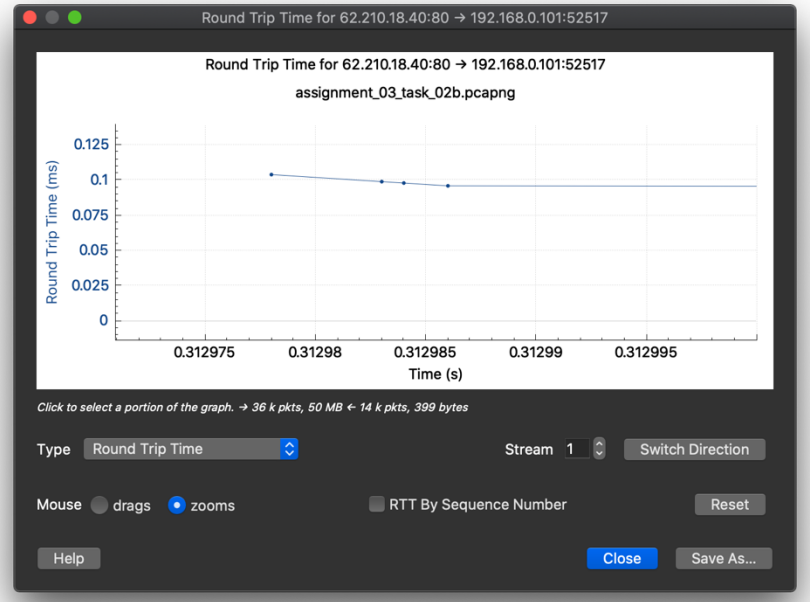


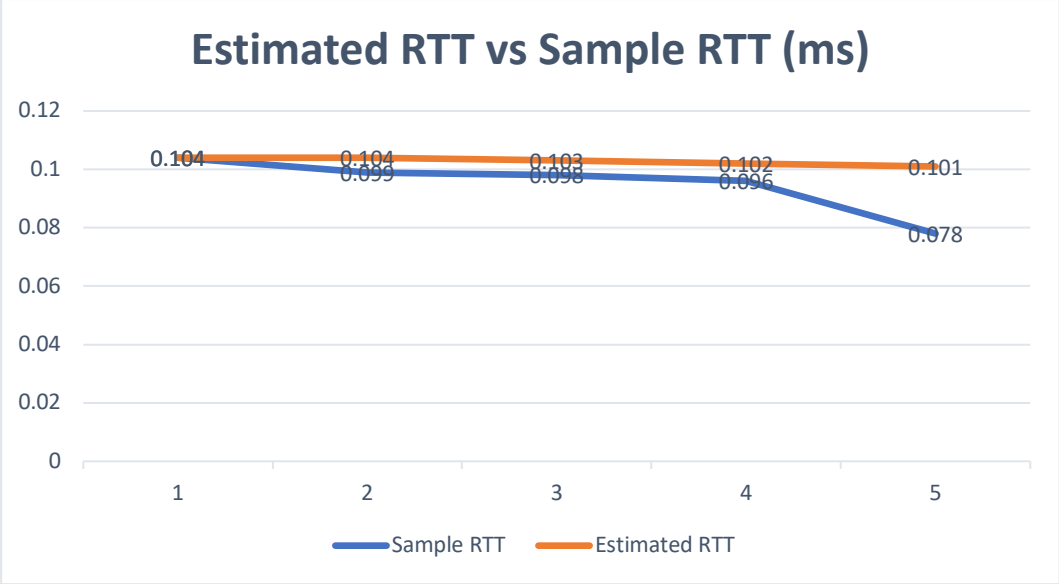
TCP estimates the RTT for next packet using RTT from previous packets. It takes the Estimated RTT and Sample/Actual RTT of previous packet and divides the difference between them in a ration of $(1-\alpha) : \alpha$. This is the Estimated RTT for next packet. So the new RTT is dependent on Sample RTT by α and on previous Estimated RTT by $1-\alpha$.

Part – B

S.No.	Sent Time	SEQ	Length	ACK Time	ACK Len	RTT	Estimated RTT
1	1.478419	1 (3958124608)	1378	1.478523	0	0.000104	0.000104
2	1.478424	1379 (3958125986)	1378			0.000099	0.000104
3	1.478425	2757 (3958127364)	1378	1.478523	0	0.000098	0.000103
4	1.478427	4135 (3958128742)	1378		0	0.000096	0.000102
5	1.479362	5513 (3958130120)	1378	1.479440		0.000078	0.000101

Maximum length is the 1378 for data from server to client. For ACK from client to server, length is 0 i.e. no data, header only.

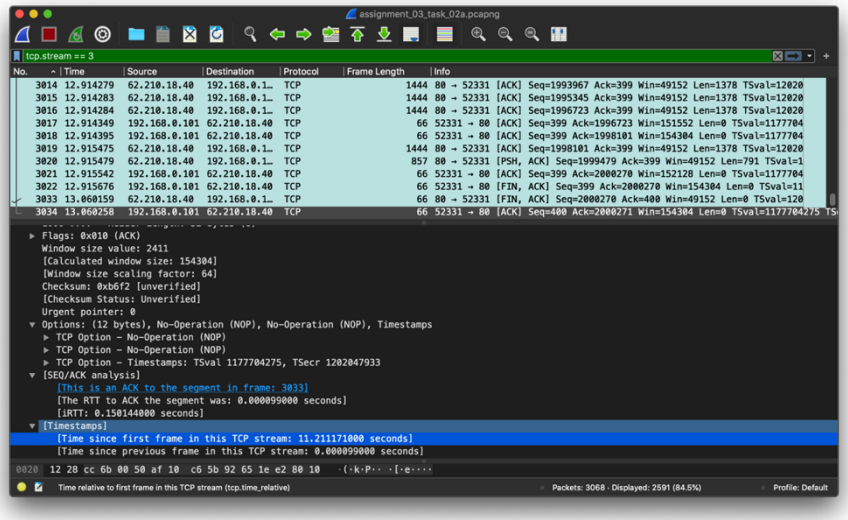


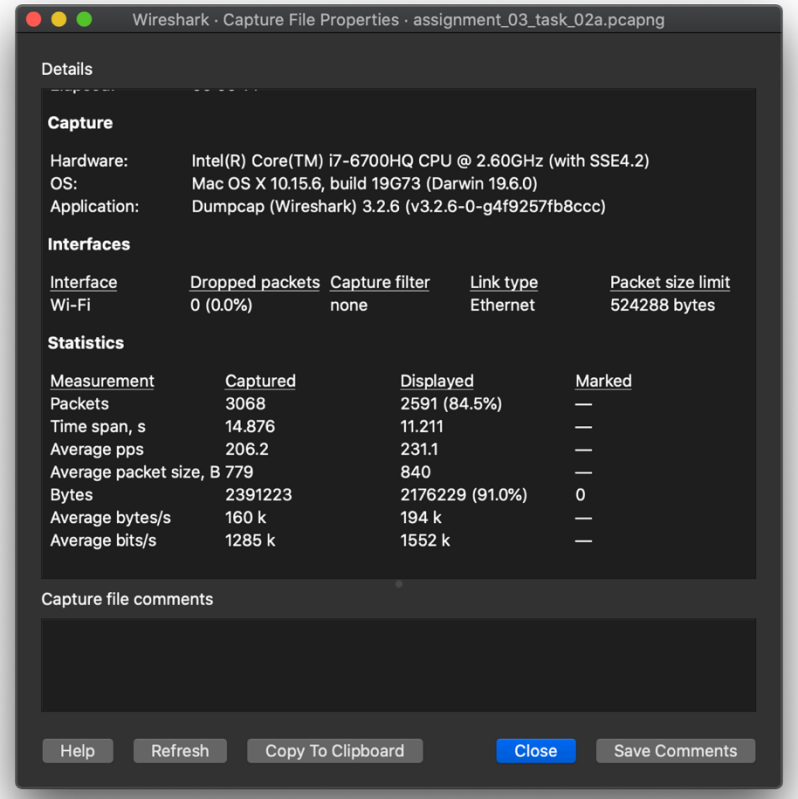


Question 04: Calculate the overall throughput (bytes transferred per unit time) for this TCP conversation using different fields of TCP from the captured file. Explain how you calculated this value using Wireshark capture in this experiment along with relevant screenshots. Verify your calculation with the one done by Wireshark using "Capture File properties". If you observe the major difference in your calculation and one calculated by Wireshark, comment why and how?

Answer 04: a. TCP Connection Duration (SYN to FIN) = 11.211171 seconds
Bytes Downloaded = 2 x 10^6 bytes
Throughput = Bytes / Time
= 2000000 / 11.211171
= 178393 bytes/second
= 178.4 kb/second

Throughput as per Capture File Properties = 194 kb/second
This throughput contains all uplink and downlink packets including all headers (Ethernet, IP, TCP).
Total Bytes (Up+Down+Headers) = 2176230.1
Throughput = 2176230.1 / 11.211171
= 194112 bytes/second
= 194 kb/second





b. TCP Connection Duration (SYN to FIN) = 88.170177 seconds
Bytes Downloaded = 50 x 10^6 bytes
Throughput = Bytes / Time
= 50000000 / 88.170177
= 567085 bytes/second
= 567 kb/second

Throughput as per Capture File Properties = 605 kb/second

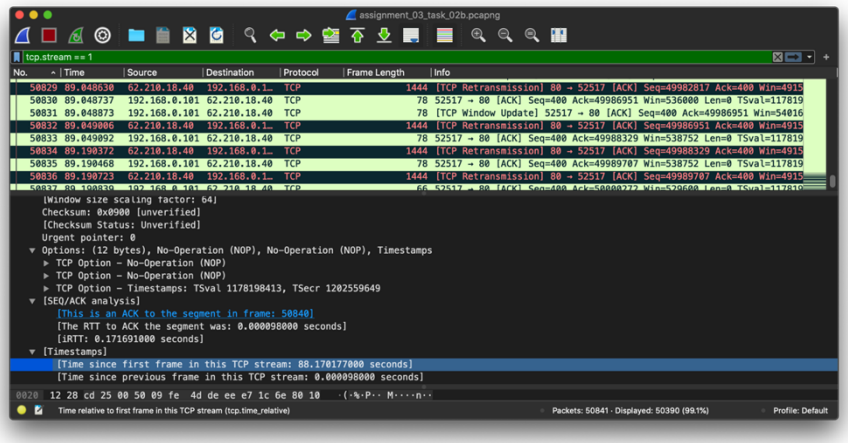
This throughput contains all uplink and downlink packets including all headers (Ethernet, IP, TCP).

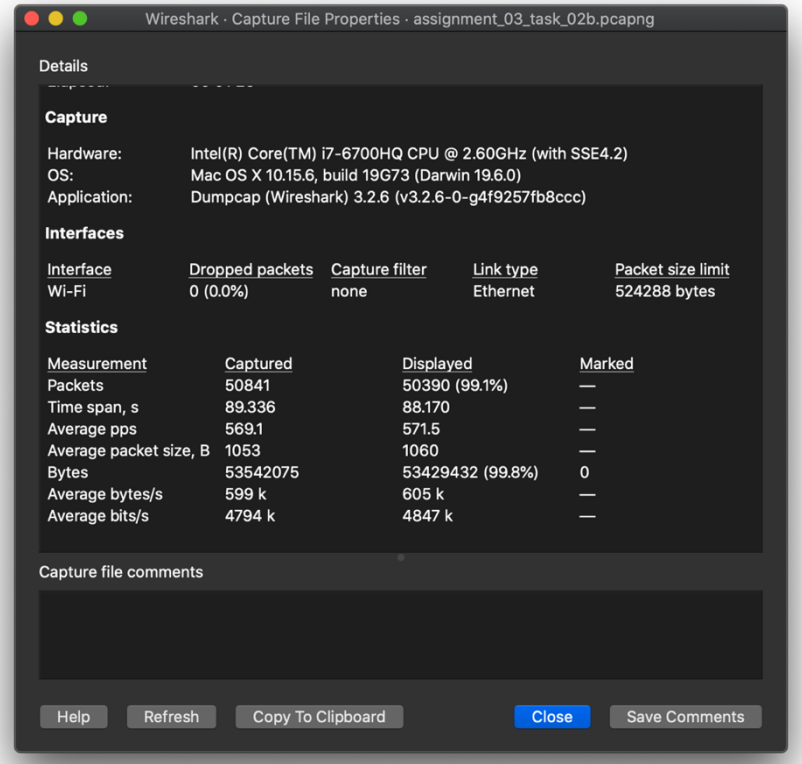
Total Bytes (Up+Down+Headers) = 53429470.81

Throughput = 53429470.81 / 88.170177

= 605981 bytes/second

= 605 kb/second

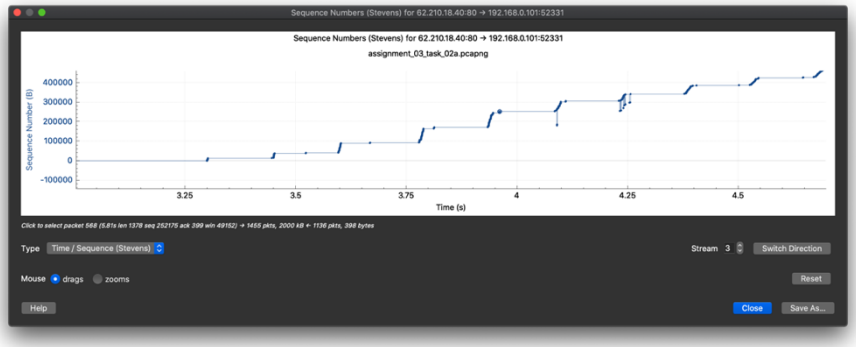


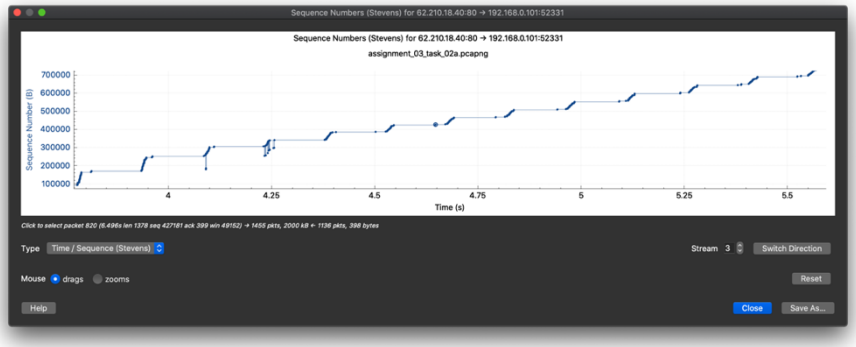


Question 05: Using any active TCP segment (pick the packet of bulk data length, e.g: 5668) involved in the download process from server to client, capture the TCP's functioning using the Time-Sequence-Graph (Stevens) plotting tool to view the sequence number versus time plot of segments being sent from the

server to the client. Can you identify where TCP's slow start phase begins and ends, and where congestion avoidance takes over? If not possible, why?

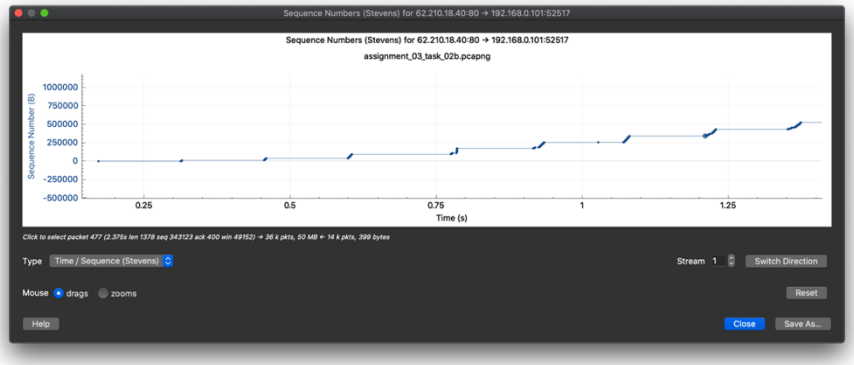
Answer 05:





The server sends first cluster at 3.3 seconds and the cluster size doubles till 3.8 seconds. At 3.95, server goes into congestion avoidance phase as the cluster size is almost similar to previous cluster. At 4.1 seconds, congestion occurs and packets are retransmitted and the server goes into congestion detection phase. Congestion occurs till 4.25 seconds and after that server shifts back to congestion avoidance phase as the cluster size in increasing slowly.

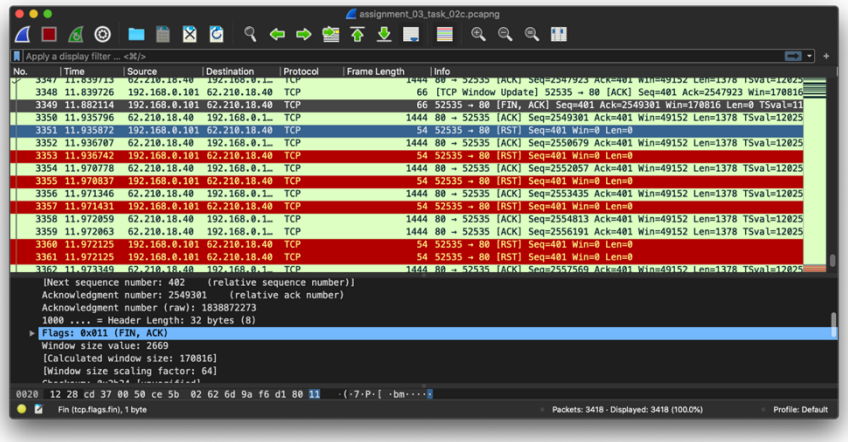
Cluster	Time	Size	Phase
1	3.3	10	Slow Start
2	3.45	20	Slow Start
3	3.6	38	Slow Start
4	3.8	57	Slow Start Threshold – Congestion Avoidance
5	3.95	59	Congestion Avoidance
6	4.1	39	Congestion Detection - Retransmission



Cluster	Time	Size	Phase
1	0.3	10	Slow Start
2	0.45	20	Slow Start
3	0.6	40	Slow Start
4	0.8	57	Slow Start Threshold – Congestion Avoidance
5	0.9	68	Congestion Avoidance

Question 06: Observe and clearly explain with screenshots, how TCP connection gets terminated in this case, as well as which fields of TCP influence this, due to cancelling of the download in between.

Answer 06:



When client the download is cancelled, the client send FIN (packet 3349) to the server indicating it does not want to send anymore data to server. In normal close, the client waits for ACK from server while the client can still, but here the client has stopped listening to the server. So for every segment received after sending FIN, the server will get a RST (red rows) response indicating that the client has closed the connection. All the packets on the way at the time of sending FIN are received at the client size and for each packet, a RST signal is sent.

PLAGIARISM STATEMENT

I certify that this assignment/report is my own work, based on my personal study and/or research and that I have acknowledged all material and sources used in its preparation, whether they be books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. I also certify that this assignment/report has not previously been submitted for assessment in any other course, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that I have not copied in part or whole or otherwise plagiarised the work of other students and/or persons. I pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, I understand my responsibility to report honour violations by other students if I become aware of it.

Name: Bhavishya Sharma
Roll No: CS20MTECH12006

Signature