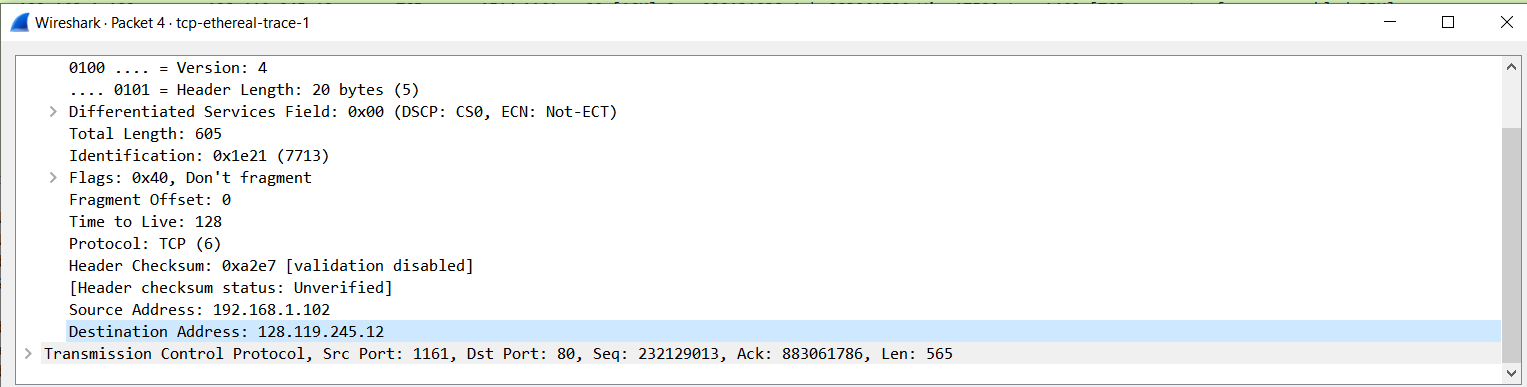
Lab04

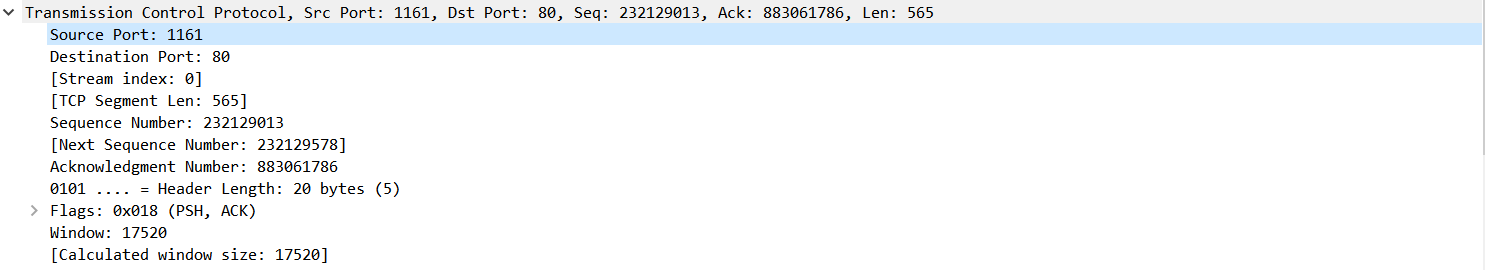
Exercises 1

1.

We can first obtain the ip address of both sender(client) and receiver(gaia.cs.umass.edu) from the IP segment



Then we can obtain the sender port number of receiver port number from the TCP segment



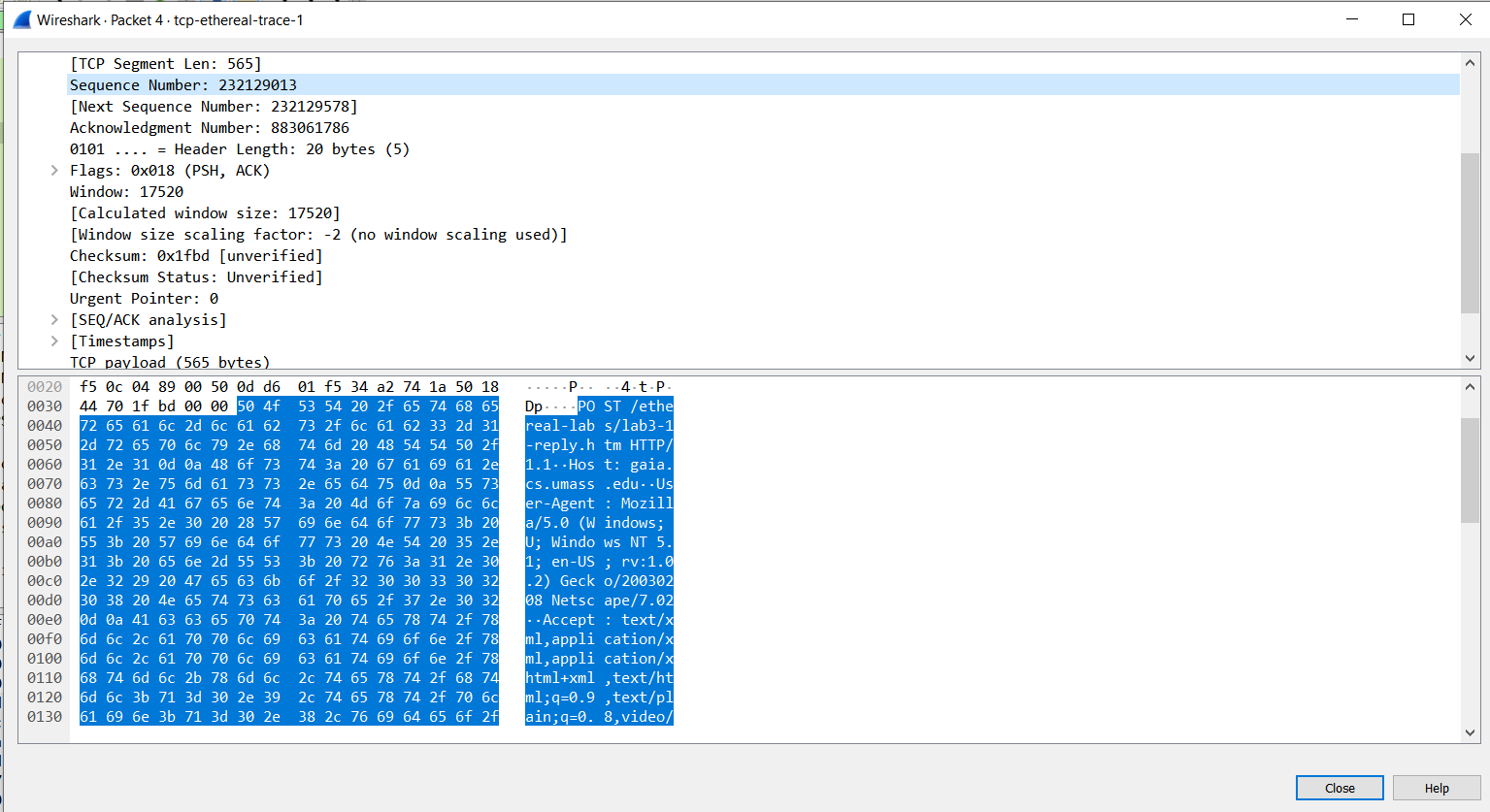
Therefore

client server is 192.168.1.102,port number is 1162

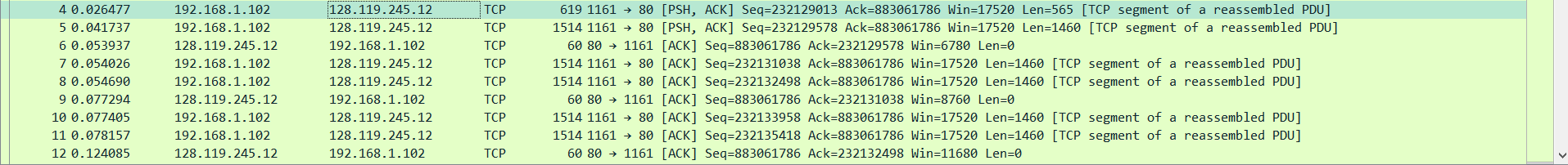
Receiver server is 129.119.245.12 port number is 80

2.

From my observation, packet 4 contains the word “POST” and “HTTP” in the data payload, it has the sequential number 232129013



3.



Started from the packet after handshaking

Table of packet sent by client

NUM| SEQ| TIME|

4,232129013,0.026477

5,232129578,0.041737

7,232131038,0.054026

8,232132498,0.054690

10,232133958,0.077405

11,232135418,0.078157

Table of ACK sent by receiver

NUM|ACK|PACKET RECEIVED| time

6,232129013 + 565,4,0.052937

9,232129578 + 1460,5,0.077294

12,232131038 + 1460,7,0.124085

14,232132498 + 1460,0.169118

15,232133958 + 1460,0.217299

16,,232135418 + 1460, 0.267802

RTT = send time - ACK received time

RRT1 = 0.052937 - 0.026477 = 0.02746

RTT2 = 0.077294 - 0.041737 = 0.035557

RTT3 = 0.124085 – 0.054026 = 0.070059

RTT4 = 0.169118 – 0.054690 = 0.114428

RTT5 = 0.217299 – 0.077405 = 0.139894

RTT6 = 0.267802 – 0.078157 = 0.189645

Estimated RTT

0.875 \* 0.02746 + 0.125 \* 0.02746 = 0.02746

0.875 \* 0.02746 + 0.125 \* 0.035557 = 0.0284721

0.875\* 0.02746 + 0.125 \*0.070059 = 0.03278488

0.875 \* 0.02746 + 0.125 \*0.114428 = 0.038331

0.875 \* 0.02746 + 0.125 \*0.139894 = 0.04151425

0.875 \* 0.02746 + 0.125 \*0.189645 = 0.04773313

4.

We can obtain the size of the tcp packet from the variable Len from the info field of the packet, therefore

Packet4: 565 bytes

Packet5: 1460 bytes

Packet7: 1460 bytes

Packet8: 1460 bytes

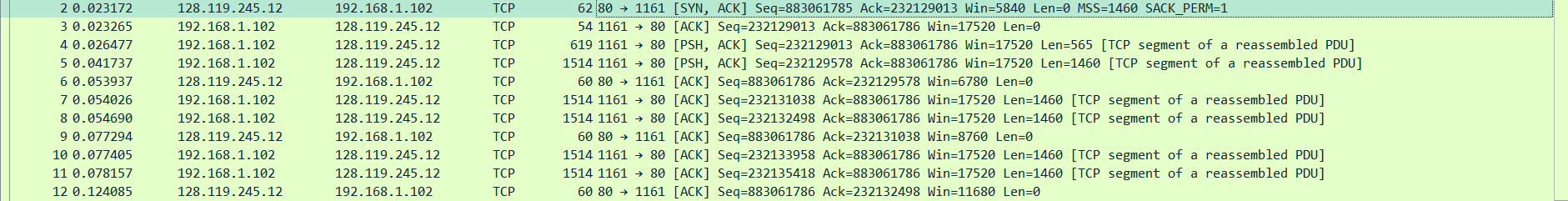
Packet10: 1460 bytes

Packet11: 1460 bytes

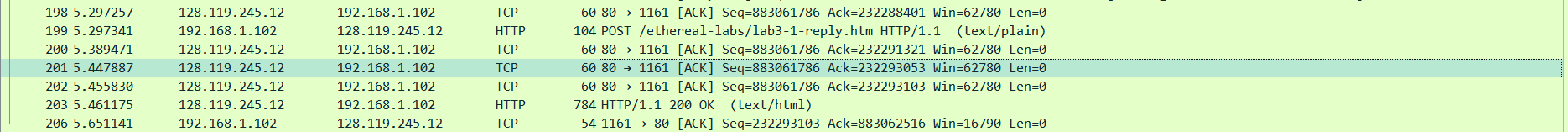
5.

We can find the buffer size value from the Win field of the receiver’s packet

The minimum value is 5840 bytes coming from the 4th packet

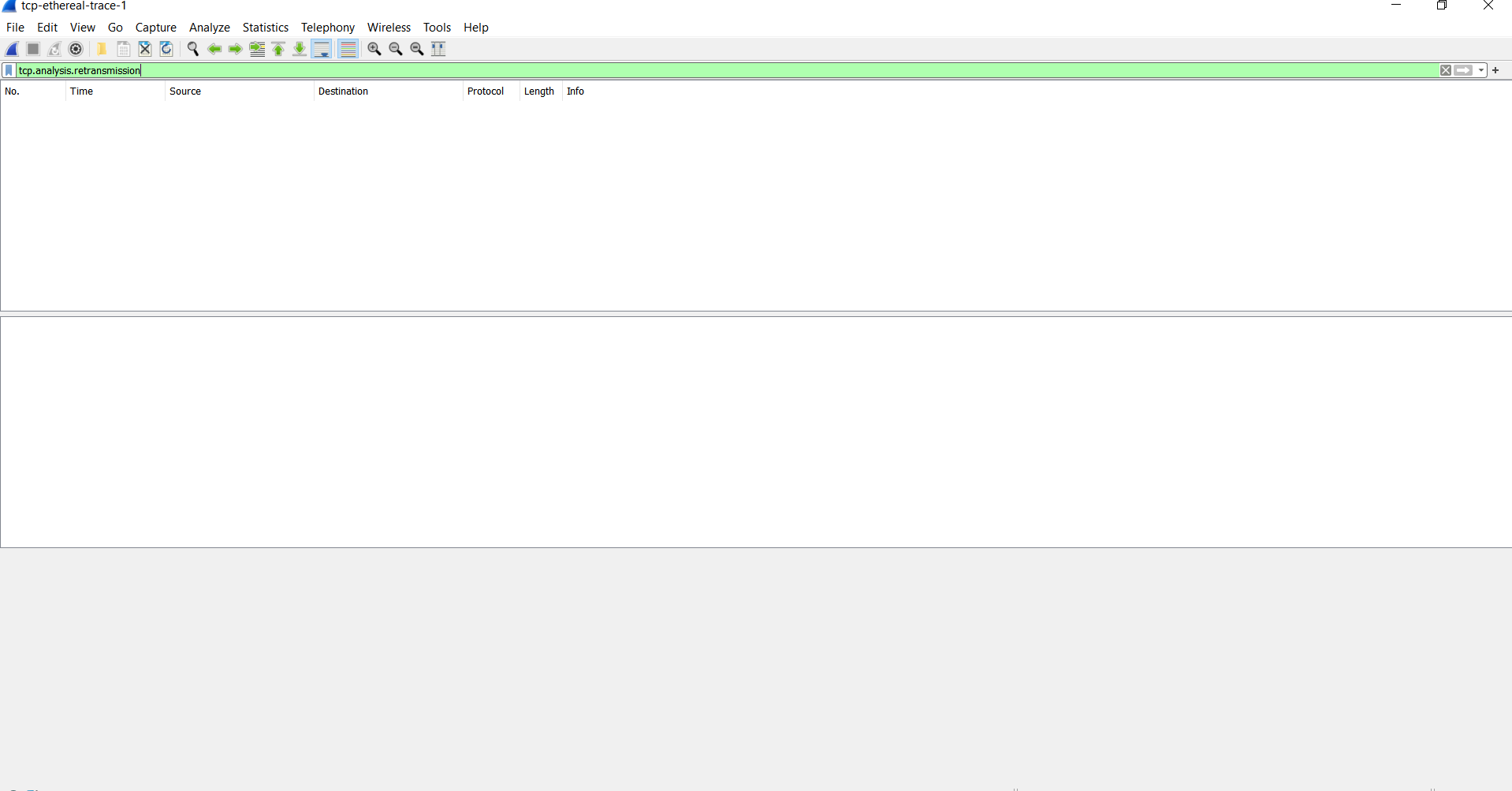


The Maximum value is 62780 bytes which can be found from the info of several packets



6.

We can use the command tcp.analysis.retransmission to obtain the retransmitted packets

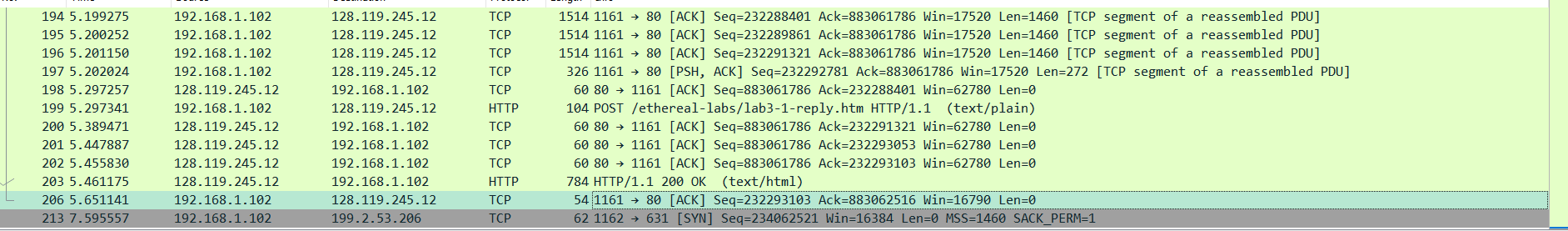


As we can see from the picture that no result has returned by the command, therefore there are no retransmitted packets.

7.

The receiver always received the data segment of 1460 bytes. Not in this case, because the server ack for each packet received,so there is no delay in acking.

8.



The throughput can be calculated by

total amount of data / time of transfer

The first packet sent(exclude the 3 way handshaking)

4 0.026477 92.168.1.102 128.119.245.12 TCP 619 1161 → 80 [PSH, ACK] Seq=232129013 Ack=883061786 Win=17520 Len=565 [TCP segment of a reassembled PDU]

The last packet sent(ACK)

206 5.651141 192.168.1.102 128.119.245.12 TCP 54 1161 → 80 [ACK] Seq=232293103 Ack=883062516 Win=16790 Len=0

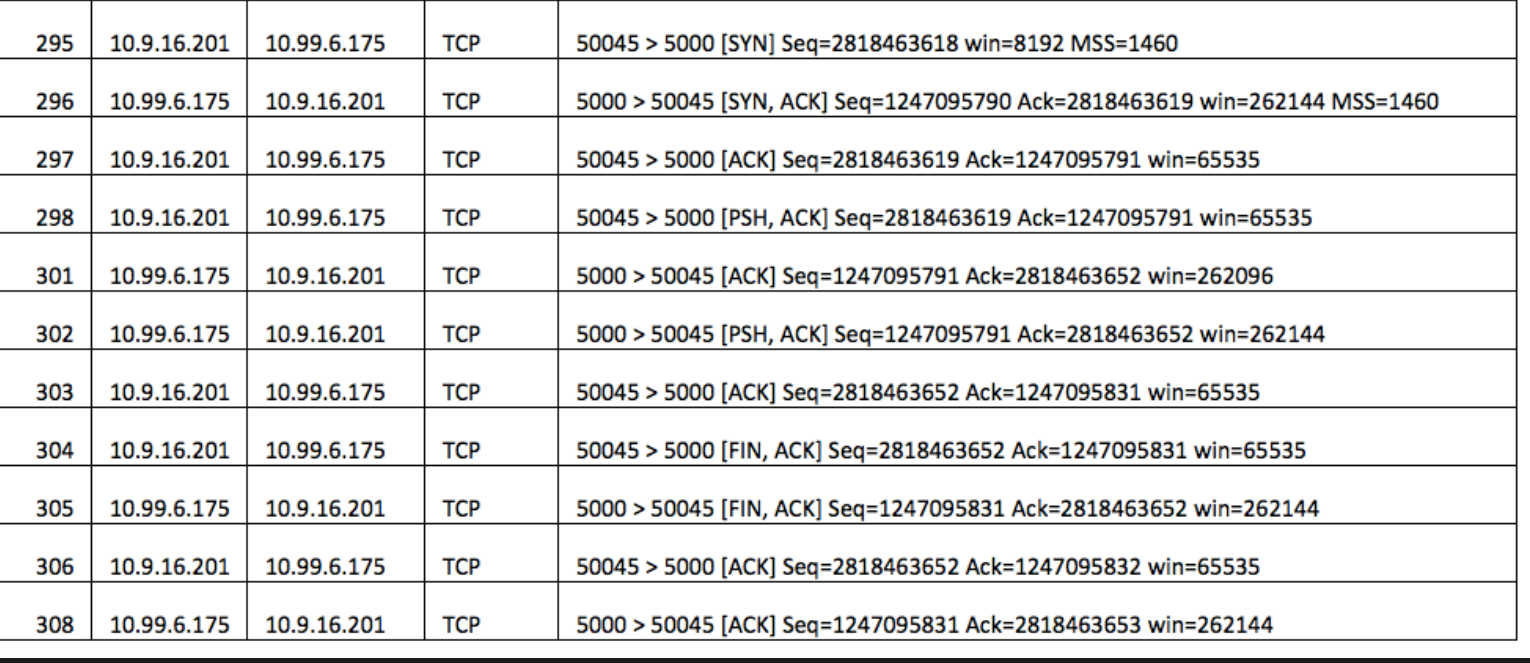
Total amount of data = 232293103 - 232129013

= 164091 bytes = 164090 bytes

Time of transfer = 5.651141 - 0.026477 = 5.624664s

Throughput = 164090 bytes/5.624664 s = 291732.2982 bytes/s

Exercise 2



1.The sequential number is 2818463618

2.The sequential number is 1247095790

The acknowledgement is 2818463619, this acknowledgement number is determined by the sequential number from the sender(seq num in q1) plus

The size of data( 1 byte)

3. The sequential number is 28184636190, the corresponding ack number is 1247095791, the segment contains 1 byte of data

4.both client and server send the FINACK at the almost the same time(N304,N305) which indicates the type of closure is Simultaneous close

5.

From Num 298 to Num 303

Data from client to server =

seq(before FIN) - seq(after the handshaking)

= 2818463652 - 2818463619 = 33 bytes

Data from server to client =

seq(before FIN) - seq(after the handshaking)

= 1247095831 - 1247095791 = 40 bytes

The relationship is final ack = initial seq + data sent