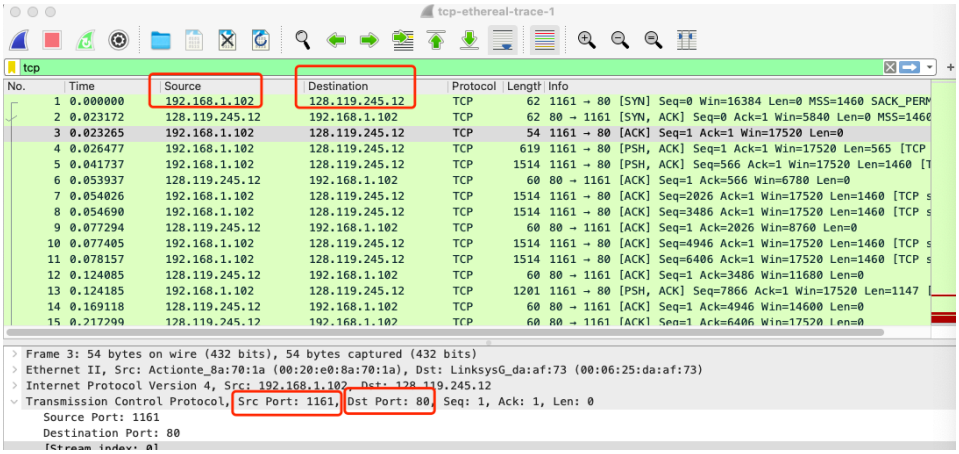
Lab4

Z5216214

Haojin Guo

Exercise 1: Understanding TCP using Wireshark

Question 1.



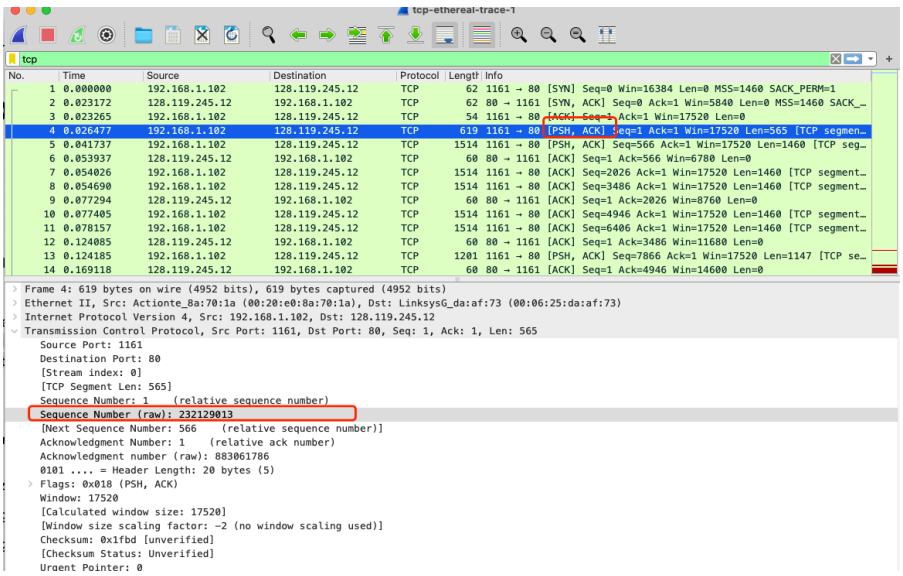
1) the IP address of gaia.cs.umass.edu = destination IP address = 128.119.245.12.

2) the post number of gaia.cs.umass.edu = destination port number = 80

3) the client IP address = source IP address = 192.168.1.102

4) the client computer port number = source port number = 1161

Question 2.



The sequence number of the TCP segment containing the POST method is 232129013.

Question 3.

The details of the first six segments containing the HTTP POST are provided below:

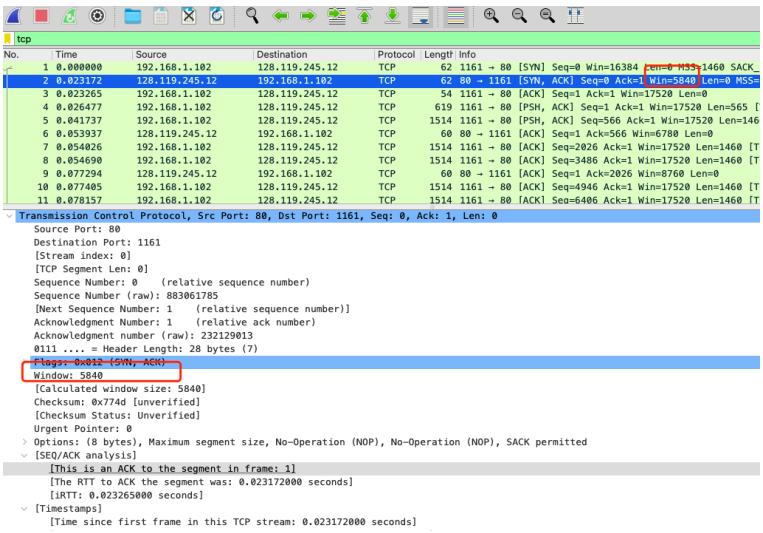
No	Seq #	Length(in Bytes)	Send time (seconds)	ACK received (seconds)	SampleRTT	EstimatedRTT
1	232129013	565	0.026477	0.053937	0.02746	0.02746
2	232129578	1460	0.041737000	0.077294000	0.035557000	0.028472
3	232131038	1460	0.054026000	0.124085000	0.070059000	0.03367
4	232132498	1460	0.054690000	0.169118000	0.114428000	0.043765
5	232133958	1460	0.077405000	0.217299000	0.139894000	0.055781
6	232135418	1460	0.078157000	0.267802000	0.189645000	0.072514

EstimatedRTT = (1 – 0.125)* EstimatedRTT + 0.125 * SampleRTT

Question 4.

No	Seq #	Length(in Bytes)
1	232129013	565
2	232129578	1460
3	232131038	1460
4	232132498	1460
5	232133958	1460
6	232135418	1460

Question 5.

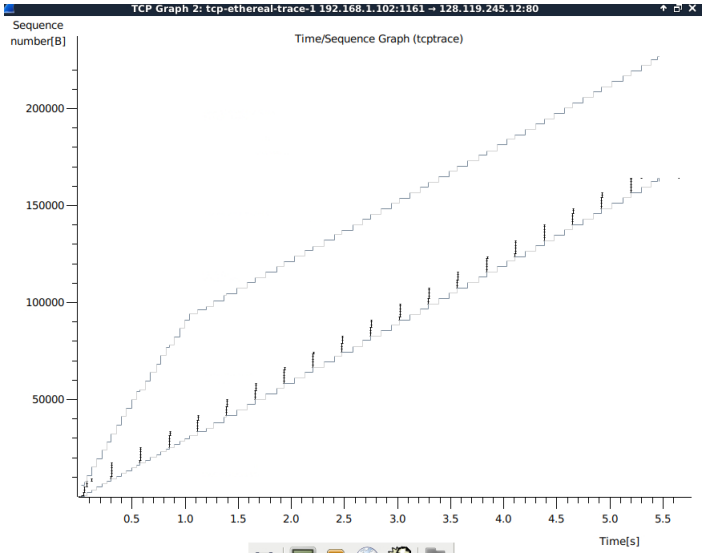


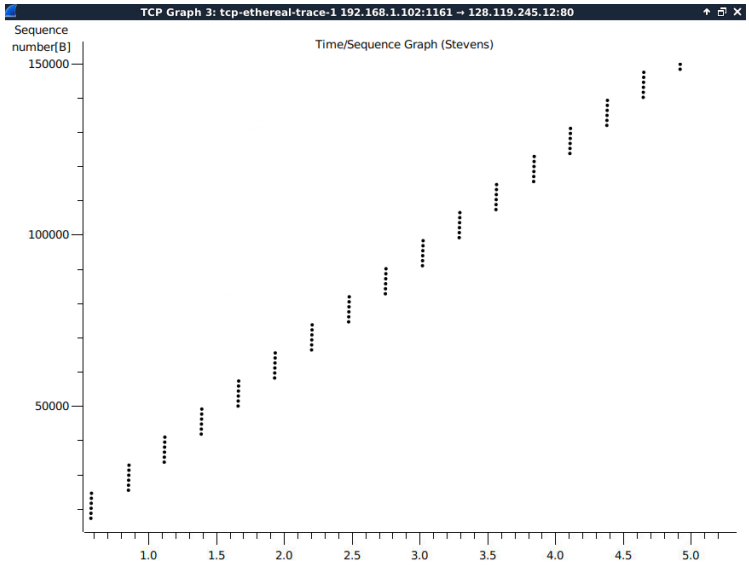
The minimum available buff space is 5840 in the entire trace.

And receiver window does not seem to throttle the sender because the receiver window size is always more than 1460 bytes. Even when the receiver window is at its lowest (5840 = 1460*4), the sender is constrained by congestion window rather than the flow control window.

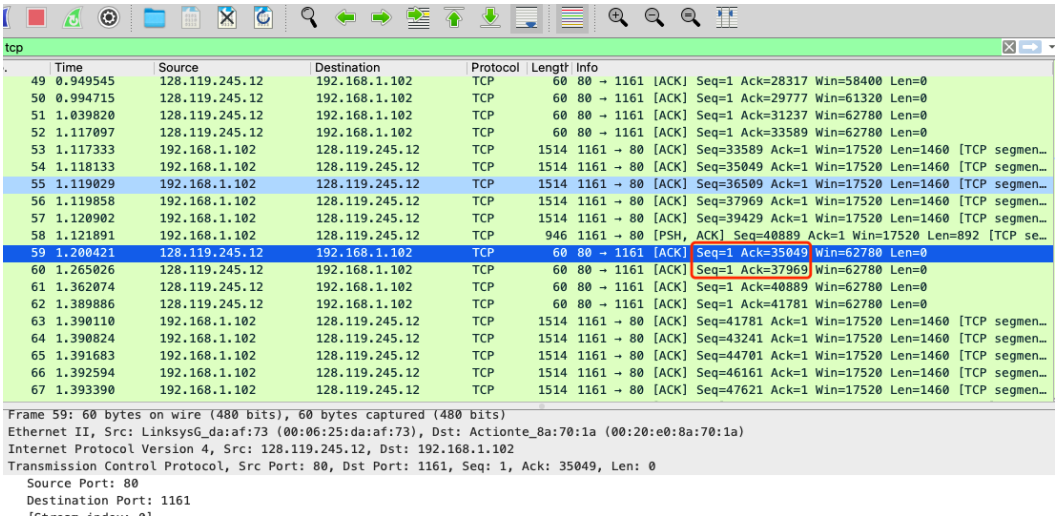
Question 6.

There is no retransmitted segments in the trace file. First, in the entire trace file there haven't the repeat entry for the segment which is retransmitted with the same sequence number. Meanwhile, according to the Time-Sequence-Graph (Stevens and tcptrace) as below, all the sequence numbers are increasing monotonically with respect to time. If there is a retransmitted segment, the sequence number of this retransmitted segment should be smaller than those of its neighbouring segments. Therefore, there is no retransmitted segments in the trace file.





Question 7.



Typically, the receiver acknowledge in an ACK is 1460 bytes.

However, in segment #60, the receiver ACK number difference between # 59 and #60 is 2920. (2920 = 1460*2). This is because the time between the two packets sent by the sender is too short, and TCP uses Delayed ACKs where the receiver waits for up to 500 msec for the arrival of another in-order segment, and then sends a cumulative ACK for both of the received segments。

Question 8.

1) The total amount data transmitted = lackACK (based on No 202 segment) – firstSeqNum (based on No. 4 segment)

Specially, lackACK (raw) = 232293103, firstSeqNum = 232129013

2) Time = the time instant of the last ACK (No. 202 segment, 5.45583 seconds]) - the time instant of the first TCP segment (No.4 segment, 0.026477 seconds])

3) Therefore,

$$Throughout = \frac{transmitted\ data}{time} = \frac{lastACK - firstSeqNum}{time} = \frac{232293103 - 232129013}{5.45583 - 0.026477} Byte/\sec =$$

$30222.904\ byte/sec$

Exercise 2: TCP Connection Management

Question 1.

The sequence number is 2818463618, which is used to initiate the TCP connection between the client computer and server.

Question 2.

1)The sequence number in SYNACK sent by the server to the client is 1247095790.

2)The ACK is 2818463619.

3)The ACK number sent by server is the result of adding 1 to the Sequence number. (ACK = SeqNum + 1, SYN occupies one byte.)

Question 3.

1) Sequence number = 2818463619 is sent by the client computer in response to the SYNACK,
2) and ACK= 1247095791.
3) There is no data included. Because the last segment of the three way handshake dose not contain any data transmission. Also, from the later segment No. 298, the seq number sent by the server to the client has the same Seq No 2818463619.

Question 4.

1) The close process is Simultaneous close.

From the last 4 segments (304, 305, 306, 308), it is easy to that their FINACK is sent by the client and server together (No 304 and 305 have crossed over in the network) through piggybacking, also the sequence numbers have no change.

Question 5.

1) Client,

the initial sequence number sent by client = 2818463618,

the lack ACK received by the client has ACK = 2818463653.

Data transformed form the client to server = lackACK - ISN – (1SYN + 1FIN) = 2818463653 – 2818463618 -2 = 33Bytes

2) Sever:

the initial Seq No = 1247095790,

the lack ACK received by server = 1247095832.

Data transformed form the server to client = lackACK - ISN – (1SYN + 1FIN) = 1247095832– 1247095790-2 = 40Bytes