Lab Exercise 4: Exploring TCP

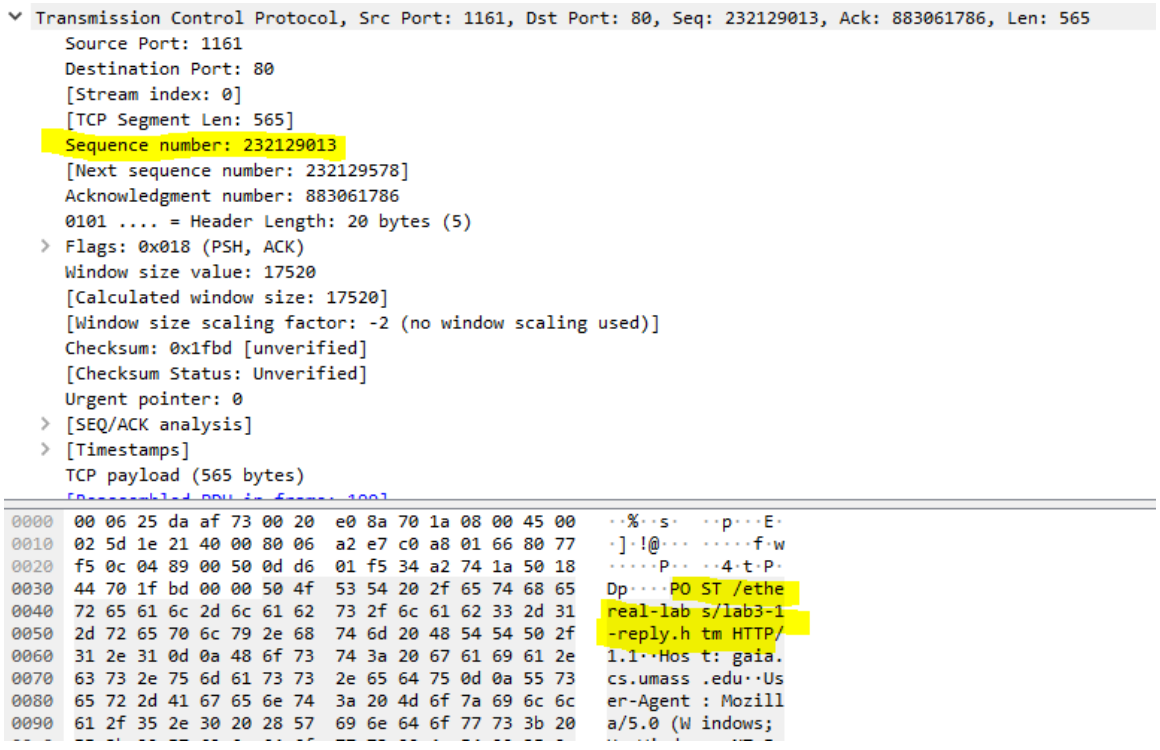
Exercise 1: Understanding TCP using Wireshark

Question 1. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection? 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?

- IP address of gaia.cs.umass.edu: 128.119.245.12, destination port: 80
- Source IP and port: 192.168.1.102, port: 1161

Question 2. What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you'll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a "POST" within its DATA field.

- 232129013



Question 3. 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) sent from the client to the web server (Do not consider the ACKs received from the server as part of these six segments)? At what time was each segment sent? When was the ACK for each segment received? 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? What is the *EstimatedRTT* value (see relevant parts of Section 3.5 or lecture slides) after the receipt of each ACK? Assume that the initial value of *EstimatedRTT* is equal to the measured RTT (*SampleRTT*) for the first segment, and then is computed using the *EstimatedRTT* equation for all subsequent segments. Set alpha to 0.125.

Sequence number:

232129013, 232129578, 232131038, 232132498 232133958 232135418

Time: time since the **first** frame in TCP stream

0.026477s, 0.041737s, 0.054026s, 0.05469s, 0.077405s, 0.078157s

When is ACK received: time since the **first** frame in TCP stream

0.053937s, 0.077294s, 0.124085s, 0.169118s, 0.217299s, 0.267802s

RTT:

0.02746s, 0.035557s, 0.070059s, 0.114428s,0.13894s, 0.189645s

EstimatedRTT:

$$\text{EstimatedRTT} = (1-\alpha)*\text{EstimatedRTT} + \alpha*\text{SampleRTT}$$

- sampleRTT refers to RTT above.

0.02746s, 0.0250975,0.0264049375,0.0318617s,0.042183s,0.054278s

Question 4. What is the length of each of the first six TCP segments?

With the assumption in question for the first TCP segment count from POST message.

Length of first six TCP are:

565, 1460, 1460, 1460,1460,1460

Question 5. What is the minimum amount of available buffer space advertised at the receiver for the entire trace? Does the lack of receiver buffer space ever throttle the sender?

Minimum buffer is 5840, which is shown in the first ack, and grow to a maximum 62780

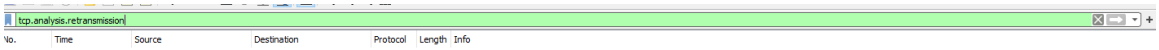
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.102	128.119.245.12	TCP	62	1161 → 80 [SYN] Seq=232129012 Win=16384 Len=0 MSS=1460 SACK_PERM=1
2	0.023172	128.119.245.12	192.168.1.102	TCP	62	80 → 1161 [SYN, ACK] Seq=883061785 Ack=232129013 Win=5840 Len=0 MSS=1460 SACK_PERM=1

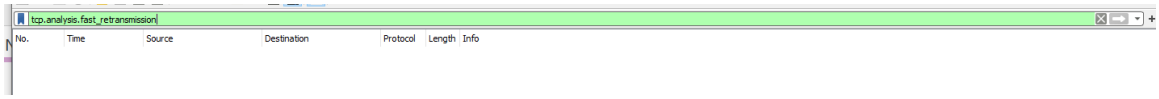
77	1.666151	192.168.1.102	128.119.245.12	TCP	946	1161 → 80 [PSH, ACK] Seq=232186285 Ack=883061786 Win=17520 Len=8
78	1.758227	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=883061786 Ack=232181905 Win=62780 Len=0
79	1.860063	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=883061786 Ack=232184825 Win=62780 Len=0
80	1.930880	128.119.245.12	192.168.1.102	TCP	60	80 → 1161 [ACK] Seq=883061786 Ack=232187177 Win=62780 Len=0

No throttle the sender.

Question 6. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?

No. I check for the sequence number of each segment, and use it compare to the previous sent segment. So, the sequence number of current segment must be greater than previous one.





Question 7. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (recall the discussion about delayed acks from the lecture notes or Section 3.5 of the text).

For the first segment, receiver receive 565 bytes data, as the return ACK – sequence number sent by sender = 232129578 – 232129013 = 565.

For the subsequent segment, we can calculate the how much data by the returning ACK – sequence number sent by sender. Typically acknowledge data is 1460 bytes.

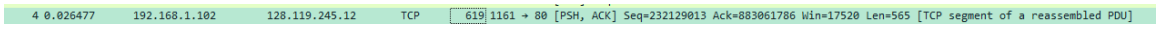
There is no delay Acking, as check through all segment, the server ack for each packet received only.

Question 8. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value

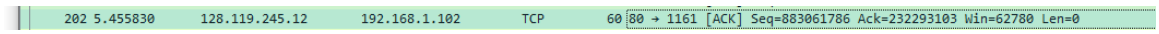
Get the initial sequence number and final ack number sent by receiver to calculate the total length of data sent by sender.

232293103 - 232129013 = 164090 bytes

Initial seq:

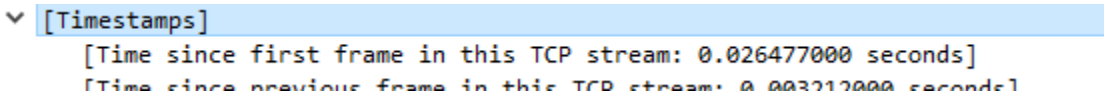


Final ack:

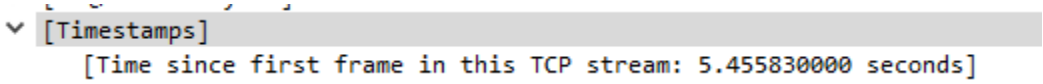


Time = 5.4583s – 0.026477 = 5.431823

Time for initial seq:



Time for final ack:



Throughput = 164090/ 5.431823 = 30209.011 bytes/s

Exercise 2: TCP Connection Management

No	Source IP	Destination IP	Protocol	Info
295	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [SYN] Seq=2818463618 win=8192 MSS=1460
296	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [SYN, ACK] Seq=1247095790 Ack=2818463619 win=262144 MSS=1460
297	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [ACK] Seq=2818463619 Ack=1247095791 win=65535
298	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [PSH, ACK] Seq=2818463619 Ack=1247095791 win=65535
301	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [ACK] Seq=1247095791 Ack=2818463652 win=262096
302	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [PSH, ACK] Seq=1247095791 Ack=2818463652 win=262144
303	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [ACK] Seq=2818463652 Ack=1247095831 win=65535
304	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [FIN, ACK] Seq=2818463652 Ack=1247095831 win=65535
305	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [FIN, ACK] Seq=1247095831 Ack=2818463652 win=262144
306	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [ACK] Seq=2818463652 Ack=1247095832 win=65535
308	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [ACK] Seq=1247095831 Ack=2818463653 win=262144

Question 1 . What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and server?

2818463618

Question 2. What is the sequence number of the SYNACK segment sent by the server to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did the server determine that value?

- Seq: 1247095790
- Ack: 2818463619, determine the sequence number sent by sender + SYN bit = 1 bit, which the returning ACK is 2818463618 + 1 = 2918463619
-

Question 3 . What is the sequence number of the ACK segment sent by the client computer in response to the SYNACK? What is the value of the Acknowledgment field in this ACK segment? Does this segment contain any data?

- Seq: 2818463619
- 1247095791
- No, this segment doesn't contain data. As the next TCP segment from sender has the sequence number 2818463619, which is same as this one. therefore no data in this segment

Question 4 . Who has done the active close? client or the server? how you have determined this? What type of closure has been performed? 3 Segment (FIN/FINACK/ACK), 4 Segment (FIN/ACK/FIN/ACK) or Simultaneous close?

The both done the activate close. By checking the sequence number in first two segments + the ack number in last two segment in below figure. We can know both client and server done the activate close.

And it is a simultaneous close.

304	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [FIN, ACK] Seq=2818463652 Ack=1247095831 win=65535
305	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [FIN, ACK] Seq=1247095831 Ack=2818463652 win=262144
306	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [ACK] Seq=2818463652 Ack=1247095832 win=65535
308	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [ACK] Seq=1247095831 Ack=2818463653 win=262144

Question 5 . How many data bytes have been transferred from the client to the server and from the server to the client during the whole duration of the connection? What relationship does this have with the Initial Sequence Number and the final ACK received from the other side?

SYN and FIN is 1 byte

Data from client: final_ACK – initial_seq = 2818463653 – 2818463618 – (SYN+FIN) = 33 bytes

Data from Server: 1247095832 – 1247095790 – 2 = 40 bytes

Realationship:

ACK = initial_sequence + total_data_length