COMP9331 LAB4 – Exploring TCP

Changfeng LI(z5137858)

Exercise 1: Understanding TCP Using Wireshark

Qs1-1: 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?

Qs1-2: 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 Ethereal window, looking for a segment with a "POST" within its DATA field.

Qs1-3: 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.

Qs1-4. What is the length of each of the first six TCP segments?

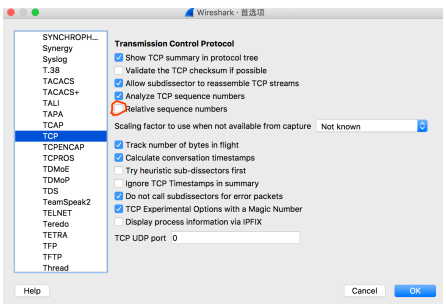
Qs1-5. What is the minimum amount of available buffer space advertised at the receiver for the entire trace? Does the lack of receiver buffer space ever throttle the sender?

Qs1-6. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?

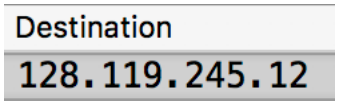
Qs1-7. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (recall the discussion about delayed acks from the lecture notes or Section 3.5 of the text).

Qs1-8. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

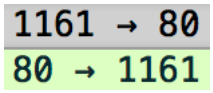
Before:



As1-1:
The IP address is 128.119.245.12

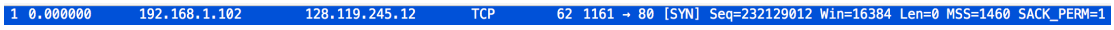


Port number: 1161 & 80



Client IP and Its TCP port:
IP: 192.168.1.102
Port: 1161

As1-2:

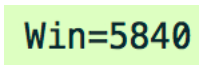


SeqNum = 232129012

As1-3,1-4:

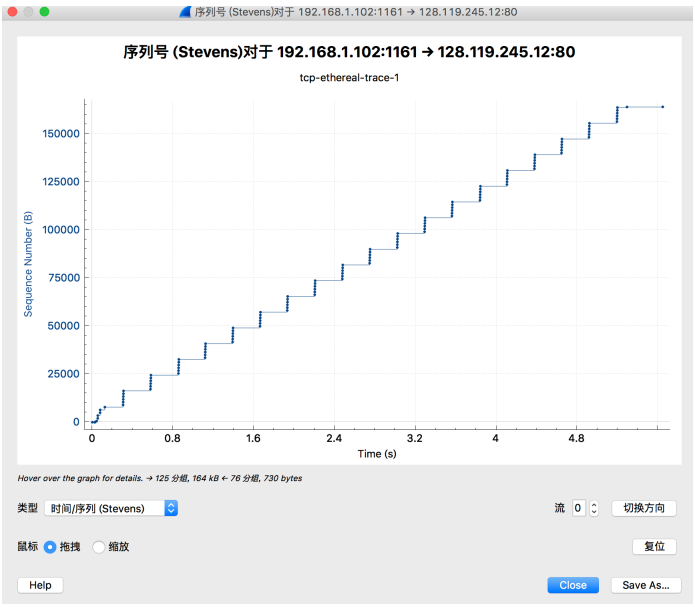
	seqNum(c->s)	T_segsent	T_ackrecv	sampleRTT	estRTT	len(seg)
1	232129013	0.026477	0.053937	0.02746	0.02746	565
2	232129578	0.041737	0.077294	0.035557	0.028472	1460
3	232131038	0.054026	0.124085	0.070059	0.033670	1460
4	232132498	0.054690	0.169118	0.114428	0.043765	1460
5	232133958	0.077405	0.217299	0.139894	0.055781	1460
6	232135418	0.078157	0.267802	0.189645	0.072514	1460

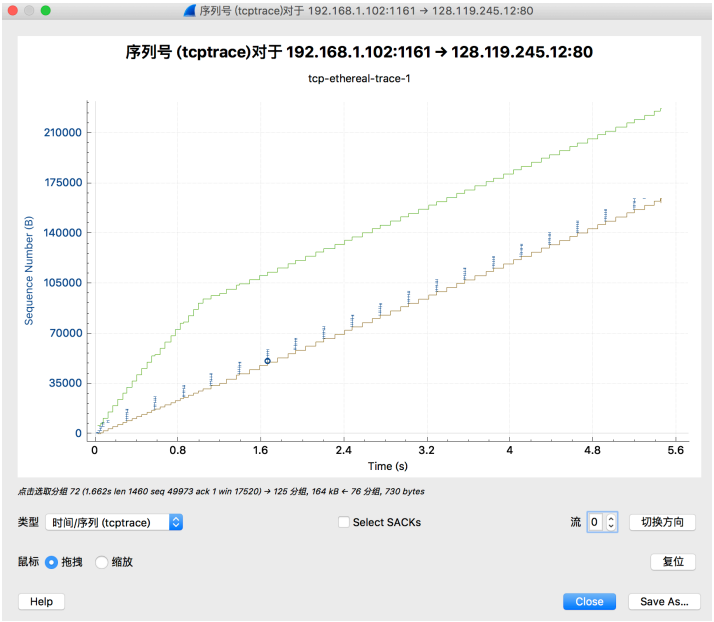
As1-5:



minWinSize = 5840
explaining: No, Because Buffer size also larger than segment size, it doesn't throttle the sender.

As1-6:
No, Because of my checking, I get all seqNum increasing when time grows.





As1-7: Most of them are of 1460 bytes in ACK segment inspection.

As1-8:

```
4  0.026477     192.168.1.102        128.119.245.12       TCP      619  1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565

202  5.455830     128.119.245.12       192.168.1.102        TCP       60  80 → 1161 [ACK] Seq=1 Ack=164091 Win=62780 Len=0
```

Throughput = Data/Time = (164091-1)/(5.456-0.026) = 30219.153bps

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

Qs2-1. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and server?
Qs2-2. 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?

Qs2-3 . 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?
Qs2-4 . 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?
Qs2-5 . 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?

As2-1:
In default, In block 1 SYN 0 is the initial case, In block 2 SYN 1 is the first response SYNACK. block 3 is ACK, so for every 3 block we update SEQ number by adding 1, nb_iter = (295-1)/3 = 98
So, Initial SYN SEQ = 2818463618 – 98 = 2818463520
(I am not sure if the progress starts at block 295 or I assume it with block 1 and all procedure is the same from block 1 to block 294, If it starts at block 295, The answer is 2818463618 rather than 2818463520)

As2-2:
In block 296, SYN, SEQ=1247095790, ACK=2818463619
Determination method: The system compute by adding 1 with SEQ number in last block.

As2-3:
In block 297, response, SEQ=2818463619, ACK=1247095791, It contains data.

As2-4:
Closing of type “3 segments FIN/FINACK/ACK” are done. Active close is performed by the side that issues the FIN first . So It is done by client because in block 304, client 10.9.16.201 send final ACK.
We can conclude this by observing block 304, 305 and 306. 304 just start sending FIN packets, and 305 makes response, 306 to finalise by sending the last ACK. And waiting 2 MSL after deal with final ACK. And block 308 tells the final ACK after 2 MSL.

As2-5:
nb_bytes = 2818463653-2818463618 = 35
The relationship is bytes equals final ACK minus initial SEQ