

Lab4

Yiting Liu (z5211008)

Exercise 1: Understanding TCP using Wireshark

Q1.

- The IP address of gaia.cs.umass.edu is **128.119.245.12**.
- The port number is **80**.
- The IP address and TCP port number of the client is **192.168.1.102** and **1161**.

Q2.

- The sequence number is **232129013**.

4	0.026477	192.168.1.102	128.119.245.12	TCP	619 [TCP segment of a reassembled PDU]
5	0.041737	192.168.1.102	128.119.245.12	TCP	1514 [TCP segment of a reassembled PDU]
6	0.053937	128.119.245.12	192.168.1.102	TCP	60 80->1161 [ACK] Seq=883061786 Ack=232129578 Win=6780 Len=0
7	0.054026	192.168.1.102	128.119.245.12	TCP	1514 [TCP segment of a reassembled PDU]
▶ Internet Protocol Version 4, Src: 192.168.1.102 (192.168.1.102), Dst: 128.119.245.12 (128.119.245.12)					
▼ Transmission Control Protocol, Src Port: 1161 (1161), Dst Port: 80 (80), Seq: 232129013, Ack: 883061786, Len: 565					
Source Port: 1161 (1161)					
Destination Port: 80 (80)					
[Stream index: 0]					
[TCP Segment Len: 565]					
Sequence number: 232129013					
[Next sequence number: 232129578]					

Q3.

No.	Sequence Number	Send Time (s)	Receive Time (s)	RTT (s)	Estimated RTT (s)	Length (bytes)
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	0.033699	1460
4	232132498	0.054690	0.169118	0.114428	0.043790	1460
5	232133958	0.077405	0.217299	0.139894	0.055803	1460
6	232135418	0.078157	0.267802	0.189645	0.072533	1460

Q4.

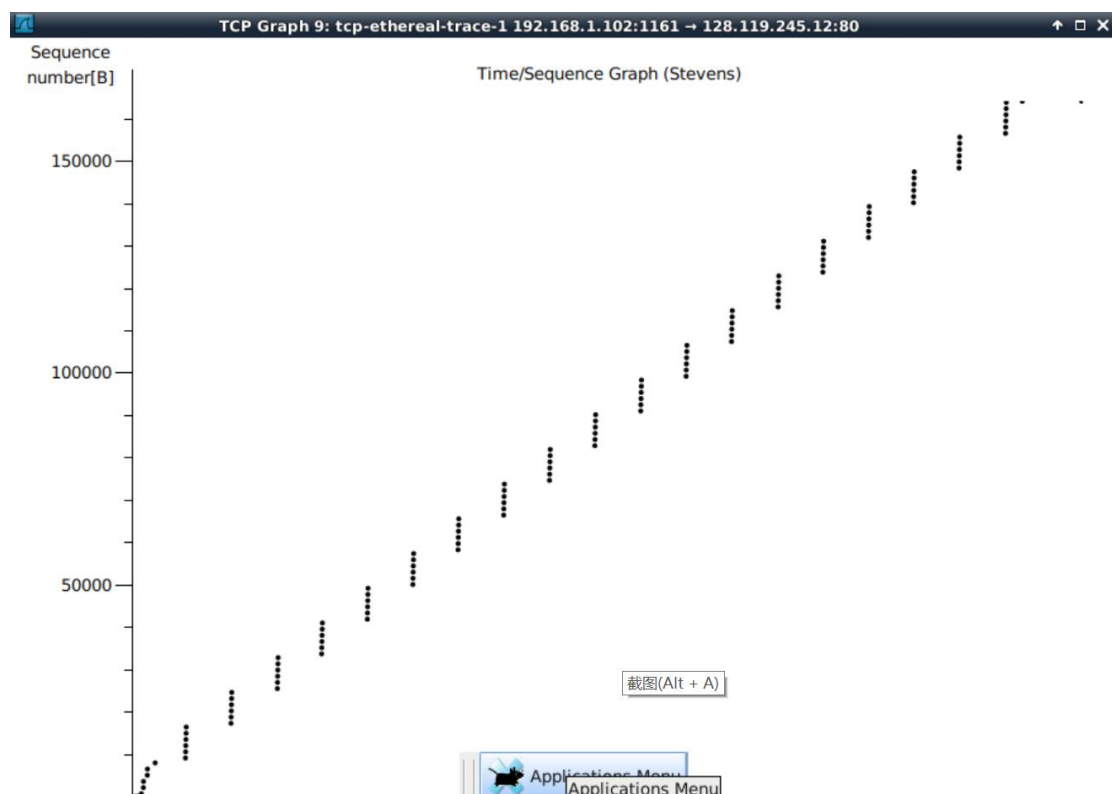
The answer is shown in the last column (in **Blue**).

Q5.

- The minimum amount of available buffer space is **5840**.
- **No, it doesn't throttle the sender.** Although the minimum amount of available buffer space advertised at the receiver is just 5840, the used amount of buffer space is very large, which is always be 62780. Therefore, it doesn't throttle the sender.

Q6.

- **No retransmitted segments in the trace file.**
- I check it by **draw a Time/Sequence Graph** (Statistics -> TCP StreamGraph -> Time-Sequence Graph). In this graph, it shows the relations between Sequence numbers and Time. As we can see, there is no repeat sequence number in the graph. Therefore, there is no retransmitted segments.



Q7.

At the beginning, when the window sizes of receiver growing up to 62780, the receiver only acknowledge **one packet** in one ACK. But when the window sizes of receiver arrive 62780, it acknowledges **several packets**

in one ACK. That's because the receiver uses **Delayed ACK**, so that one ACK packet can acknowledge several in-order packets.

Q8.

Total time (Except connection establishment and termination):

$$5.45883 - 0.026477 = 5.432353 \text{ sec}$$

Total data (Last sequence number – first sequence number)

$$232293103 - 232129678 = 163,425 \text{ bytes}$$

Therefore, the throughput = $163,425 / 5.432353 = \mathbf{30083.65 \text{ byte/sec}}$

Exercise 2: TCP Connection Management

Q1.

The sequence number of the TCP SYN is **2818463618**.

Q2.

- The sequence number of the SYNACK is **1247095790**.
- The value of Acknowledgement field is **2818463619**.
- The number in Acknowledgement field is equal to the sequence number of SYN plus one.

Q3.

- The sequence number to reply SYNACK is **2818463619**.
- The value of Acknowledgment field is **1247095791**.
- This segment doesn't contain any data.

Q4.

This is a **simultaneous closure**, which is made by client and server simultaneously. When we look at No. 304 – 308, we can find client send a FINACK with sequence number x (2818463652) and then server send another FINACK with sequence number y (1247095831), then both of client and server send an ACK with ACK number y+1 (x+1). This process is same with simultaneous closure. Therefore, this is simultaneous closure.

Q5.

- From client to server: $2818463653 - 2818463619 - 2 = \mathbf{32 \text{ bytes}}$
- From server to client: $1247095832 - 1247095790 - 2 = \mathbf{40 \text{ bytes}}$

Data transferred = the final ACK received from the other side – ISN