# Lab4

## Exercise 1

Q1.

The IP address of gaia.cs.umass.edu is 128.119.245.12;

The port number for it (server) sending and receiving TCP segments for this connection is 80;

The IP address of client computer (source) is 192.168.1.102;

The TCP port number for source is 1161;

Q2.

The sequence number of the TCP segment containing the HTTP POST command is 232129013

Q3.

The sequence number of the first six segments in the TCP connection:

| Sequence number | Send Time (s) | ACK Time (s) | RTT (s) | EstimateRTT (s) |
|---|---|---|---|---|
| 1. 232129013 | 0.026477 | 0.053937 | 0.027460 | 0.027460 |
| 2. 232129578 | 0.041737 | 0.077294 | 0.035557 | 0.028472 |
| 3. 232131038 | 0.054026 | 0.124085 | 0.070059 | 0.033670 |
| 4. 232132498 | 0.054690 | 0.169118 | 0.114428 | 0.043765 |
| 5. 232133958 | 0.077405 | 0.217299 | 0.139894 | 0.055781 |
| 6. 232135418 | 0.078157 | 0.267802 | 0.189645 | 0.072514 |

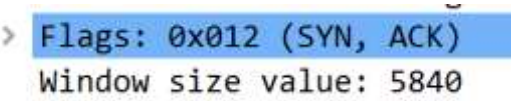
Q4.

Length of TCP segment (Bytes)

1. 565
2. 1460
3. 1460
4. 1460
5. 1460
6. 1460

Q5.

The minimum amount of available buffer space advertised at the receiver for the entire trace is 5840 which is from the first ACK form server.



No. This receiver window grows steadily until a maximum receiver buffer size of 62780 bytes. The sender is never throttled due to the lack of receiver buffer space by inspecting this trace.

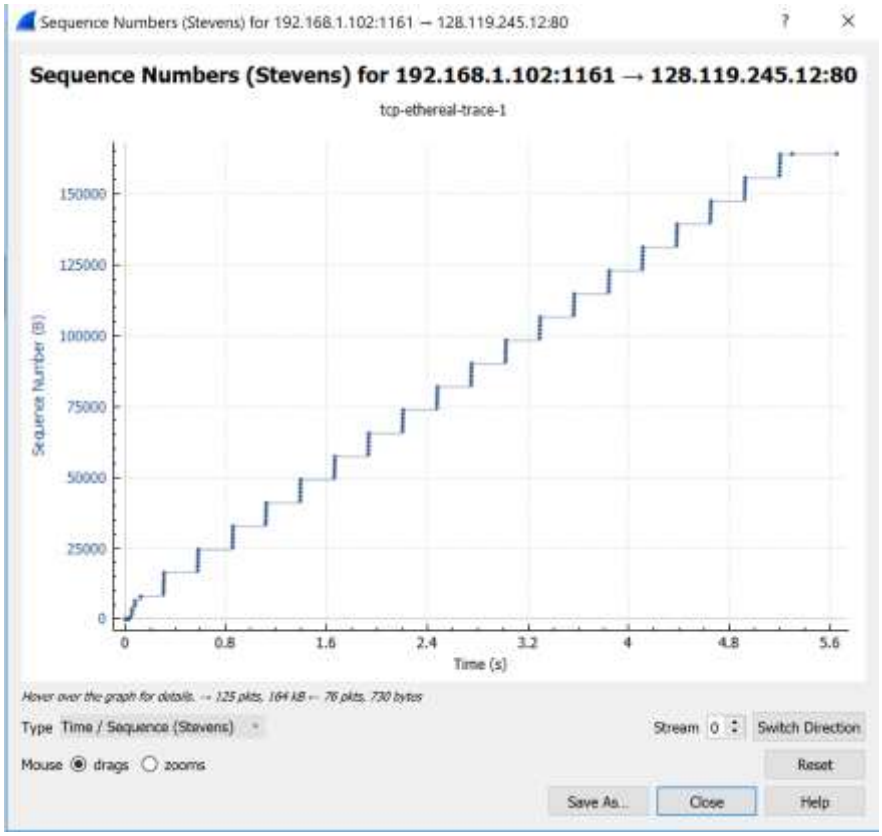
Q6.

No, there are not any retransmitted segments in the trace file.

I tried two ways to check this.

First one is to check send records one by one; to see whether there is a sequence number appears twice.

Second one is using the Time/Sequence (Stevens) in statistics->TCP Stream Graph.




From this graph, we can see that there is no retransmission in the trace file; if there is one retransmission, there should be one point that is obviously lower than its surrounding points.

Q7.

The receiver typically acknowledges 1460 bytes of data in an ACK.

```
54 1.118133     192.168.1.102     128.119.245.12     TCP     1514 1161 → 80 [ACK] Seq=232164061 Ack=883061786 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
55 1.119029     192.168.1.102     128.119.245.12     TCP     1514 1161 → 80 [ACK] Seq=232165521 Ack=883061786 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
56 1.119858     192.168.1.102     128.119.245.12     TCP     1514 1161 → 80 [ACK] Seq=232166981 Ack=883061786 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
```

```
59 1.200421     128.119.245.12     192.168.1.102     TCP      60 80 → 1161 [ACK] Seq=883061786 Ack=232164061 Win=62780 Len=0
60 1.265026     128.119.245.12     192.168.1.102     TCP      60 80 → 1161 [ACK] Seq=883061786 Ack=232166981 Win=62780 Len=0
```

from record 54,55 and 56, we can see that the host has sent 3 packets.

But it receives 2 ACKs which is a cumulative ACK which can be seen from record 59 and 60.

Q8.

The first sequence number is 232129013 (with the POST) at time 0.026477s

The last ACK number is 232293103 at 5.455830s.

Therefore, there are totally 164090 bytes of data been transferred.

And the time for this transfer is 5.429353 s.

The throughput = 164090/5.429353 = 30222 (Bytes per second)

## Exercise 2

Q1.

Sequence number of the TCP SYN is 2818463618

Q2.

Sequence number of SYNACK segment is 1247095790;

ACK in the SYNACK segment is 2818463619;

Server determines this by adding one byte to the sequence number sent from client. And this one byte is for the SYN bit from the client side.

Q3.

Sequence number of the ACK segment response to SYNACK is 2818463619

ACK number of this is 1247095791.

No, this segment does not contain data. Because No.297 segment is used to establish the connection; and the real data is sent by No.298 as they both use the same sequence number.

Q4.

Both client and server close the connection simultaneously.

Because from record number 304 and 305, we can see that both client and server send the

[FIN, ACK] at the same time.

It is the simultaneous close.

Q5.

The number of data that transferred from client to server is 33 bytes (2818463652 – 2818463619);

The number of data that transferred from server to client is 40 bytes (1247095831 – 1247095791);

The initial sequence number of each side plus its data bytes and plus 2 bytes (SYN and FIN) equals the final ACK received from the other side.