

# Lab 4

## Exercise 1

---

### 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 are the IP address and TCP port numbers used by the client computer (source) that is transferring the file to gaia.cs.umass.edu?*

The IP address of gaia.cs.umass.edu is **128.119.245.12** and it's using port **80** for this connection.

Source address is **192.168.1.102** and port number is **1161**.

### Question 2

*What is the sequence number of the TCP segment containing the HTTP POST command?*

The sequence number is: **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? what time was each segment sent? When was the ACK received? Given the difference between them, what is the RTT value for each of the six segments?*

The data is shown below:

order	Seq number	Sent time	ACK received	RTT	Estimated RTT
1	232129013	0.026477	0.053937	0.02746	0.02746
2	232129578	0.041737	0.077294	0.03556	0.02847
3	232131038	0.054026	0.124085	0.07006	0.03367
4	232132498	0.054690	0.169118	0.11443	0.04377
5	232133958	0.077405	0.217299	0.13989	0.05578
6	232135418	0.078157	0.267802	0.18965	0.07251

The EstimatedRTT is computed according to the following formula with  $\alpha = 0.125$ :

$$EstimatedRTT_t = (1 - \alpha) EstimatedRTT_{t-1} + \alpha SampleRTT$$

## Question 4

*What is the length of each of the first six TCP segments?*

The lengths of these segments are:

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

## Question 5

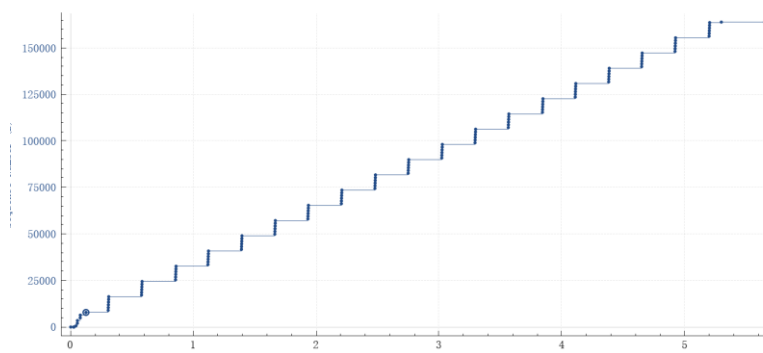
*What is the minimum amount of available buffer space advertised at the receiver for the entire trace? Does the lack of receiver buffer space ever throttle the sender?*

The minimum amount of available buffer is **5840** according to the first SYNACK package. And the sender is not ever throttled, even the smallest buffer size is four times as the segment size above.

## 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?*

There is no retransmission because there is no duplicated entry recorded. This can also be proved by the time/series diagram because for each sequence number, there at most one corresponding time.



## 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?*

Typically, **1460 bytes** of data is acknowledged in an ACK. In the beginning, the receiver is ACKing each TCP segment separately. It can also be found at 59<sup>th</sup> and 60<sup>th</sup> segment that the receiver is ACKing two TCP segments at once because the ACK number indicates that 2920 bytes of data is received. It is because the receiver may wait for another segment in-order for a while. If another segment is received, the receiver may send ACK two these two back together.

## Question 8

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

The throughput is 30.22 Kbytes/sec.

The throughput is calculated dividing the total data transferred by time elapsed. The sequence number at establishment is 232129013 and the last ACK number is 232293103. The difference between them 164090 is the amount of data transferred.

The time elapsed is then  $5.455830 - 0.026477 = 5.429353$

Then we have:

$$throughput = 164090 / 5.429353 = 30222.75398 \text{ bytes/sec} = 30.22 \text{ Kbytes/sec}$$

## Exercise 2

---

### 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?*

The sequence number is **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?*

The sequence number is **1247095790**, the value is **2818463619**, and the value is just the sequence number plus one. The ack number should be the received seq number plus amount data received. Because there is no extra data sent when establishing the connection, the number is just incremented by one.

## 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?*

The sequence number is **2818463619**, the value in acknowledgement field is **1247095791**. This segment doesn't contain any data because the next segment sent has the same sequence number as this one.

## 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?*

**Both** the client and the server did the active close, because the client sent FIN to server first. As a result, **simultaneous close** was performed.

## 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?*

Data transferred (excluding data sent in connection establishment and termination):

Client → Server: **33** bytes

Server → Client: **40** bytes

The difference between Initial sequence number and final CK received is just these values plus 2. Because the sequence number and ACK number are used to record amount of data transferred. And sequence number is incremented by 1 in establishment and termination.