

CS3331 Lab04

z5265106 Aniket Chavan

Exercise 1

Question 1

- What is the IP address of `gaia.cs.umass.edu`?
 - 128.119.245.12
- On what port number is it sending and receiving TCP segments for this connection?
 - Sending from 1161
 - Receiving from 80
- 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 is 192.168.1.102

Question 2

- What is the sequence number of the TCP segment containing the HTTP POST command?
 - 232129013

Question 3

- 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?

TCP Segment	Seq #	Sent (s)	ACK Received (s)	RRT (ms)	Estimated RTT (ms)
1	232129013	0.026477	0.053937	27.46	27.46
2	232129578	0.041737	0.077294	35.557	28.472
3	232131038	0.054026	0.124085	70.059	33.670
4	232132498	0.054690	0.169118	114.428	43.765
5	232133958	0.077405	0.217299	139.894	55.781
6	232135418	0.078157	0.267802	189.645	72.514

Question 4

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

TCP Segment	Seq #	Length (Bytes)
1	232129013	565
2	232129578	1460
3	232131038	1460
4	232132498	1460
5	232133958	1460
6	232135418	1460

Question 5

- What is the minimum amount of available buffer space advertised at the receiver for the entire trace?
 - The minimum window size advertised by the receiver (aria.cs.umass.edu) is **5840** bytes in the first acknowledgement from the server.
- Does the lack of receiver buffer space ever throttle the sender?
 - The sender is never throttled by a lack of buffer space.

Question 6

- Are there any retransmitted segments in the trace file?
 - No retransmitted segments.
- What did you check for (in the trace) in order to answer this question?
 - This can be checked by scanning the output for the duplicate ACK numbers sent from the receiver to the sender. Alternatively, Wireshark has a filter that shows only duplicate packets.

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).
 - The receiver typically ACKs each packet individually, this each ACK usually acknowledges 1460 bytes of data. Some packets, for instance packet 52 of the trace are a cumulative ACK of received packets, in this case packet 52 is an ACK of packet 46 and 47.

Question 8

- What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.
 - Looking at the sequence number of the last TCP packet and the time it was received we can calculate the throughput by dividing the amount of data transferred by the time taken.

From this
we can see that the throughput of the connection is

$$\frac{164901 \text{ bytes}}{5.297341} = \frac{30976 \text{ bytes}}{\text{second}} = 30.98 \text{ Kbps}$$

Exercise 2

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?
 - seq # of SYN for initiation of TCP connection: **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?
 - seq # of SYNACK from server to client: **1247095790**
- What is the value of the Acknowledgement field in the SYNACK segment?
 - seq # of ACK of SYNACK: **2818463619**
- How did the server determine that value?
 - To determine this value, it's an increment of 1 of the received SYN packet.

Question 3

- What is the sequence number of the ACK segment sent by the client computer in response to the SYNACK?
 - Seq # of ACK from client to server: **1247095791**
- What is the value of the Acknowledgment field in this ACK segment?
 - Seq # of ACK: **2818463619**
- Does this segment contain any data?
 - No it does not contain any data, we can conclude this by looking at the next packet sent [PSH, ACK] has the same sequence number. Hence there was not data being added or transfer between the previous packet and this one.

Question 4

- **Who has done the active close? client or the server?**
 - #304, client has done the active close.
- **how you have determined this?**
 - First FIN header sent
- **What type of closure has been performed? 3 Segment (FIN/FINACK/ACK), 4 Segment (FIN/ACK/FIN/ACK) or Simultaneous close?**
 - The server then responds with its own [FIN, ACK] and both sides subsequently ACK these packets. Thus it is a simultaneous close.

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?**
 - For the client: **2818463619 → 2818463652 = 33 bytes**
 - For the server: **1247095791 → 1247095831 = 40 bytes**
 - Total bytes (side A->B) = (final ACK from B) - (ISN of A) - 2
- **What relationship does this have with the Initial Sequence Number and the final ACK received from the other side?**
 - The final ack - the initial seq # is the number of bytes transferred by one side.