Resources / Lab Exercises (/COMP3331/18s2/resources/17340) / Lab Exercise 4: Solutions

Lab Exercise 4: Solutions

Exercise 1: TCP

Question 1. (**0.5 mark**)

The IP address of gaia.cs.umass.edu is 128.119.245.12 and the port number used for this connection is 80 (indicating a web server). The IP address for the host (in MIT) is 192.168.1.102 and the port number is 1161.

Question 2. (**0.5 mark**)

The sequence number of the TCP segment containing the POST method is 232129013. Notice that the sequence number is same as the sequence number for the ACK segment that was sent prior to this segment. The reason for this is that the ACK segment contained no data and hence the sequence number was not incremented.

Question 3. (**1.5 mark**)

The details for the first six segments and the corresponding ACKs are provided below along with the calculation of the *EstimatedRTT* :

No.	Sequence #	Length (in bytes)	Time sent	Time ACK received	Sample RTT	Estimated RTT
1	232129013	565	0.026477	0.053937	0.02746	0.02746
2	232129578	1460	0.041737	0.077294	0.035557	0.028472
3	232131038	1460	0.054026	0.124085	0.070059	0.03367
4	232132498	1460	0.054690	0.169118	0.114428	0.043765
5	232133958	1460	0.077405	0.217299	0.139894	0.055781
6	232135418	1460	0.078157	0.267802	0.189645	0.072514

Recall that EstimatedRTT = 0.875 * EstimatedRTT + 0.125 * Sample RTT.

Be careful when you note down the times. Note that the ACK always contains the next expected sequence number in the acknowledgement field.

Question 4. (**0.5 mark**)

See the above table for the TCP segment lengths. Note that the length in the table indicates the payload contained in the TCP header. It does not include the 20 byte TCP header.

Question 5. (**0.5 mark**)

To answer this question examine all the ACK (and SYNACK) packets sent by the server to the client. The minimum advertised window is 5840 and this is advertised in the SYNACK segment. The receiver window does not seem to throttle the sender at all. Even when the advertised receiver window is at its lowest value (5840: equivalent to 4 MSS packets of 1460 bytes), the sender seems to be constrained by congestion window as opposed to the flow control window. Notice that in the later part of the trace, when the congestion window has grown to a reasonable size, the receiver advertised window is very large (mostly 62780 bytes: 43 MSS segments). Hence the lack of receiver buffer space does not seem to be an issue with this connection.

Question 6. (**0.5 mark**)

There are no retransmitted segments in the trace file. If at all a packet is retransmitted there would be a repeat entry for that segment which is retransmitted with the same sequence number. However there is no such occurrence in the entire trace file. You can also verify this by checking the sequence numbers of the TCP segments in the trace file. In the *Time- Sequence-Graph (Stevens*) of this trace, all sequence numbers from the source (192.168.1.102) to the destination (128.119.245.12) are increasing monotonically with respect to time. If there is a retransmitted segment, the sequence number of this retransmitted segment should be smaller than those of its neighbouring segments.

Question 7. (**1 mark**)

In the early part of the trace file (probably slow-start phase), we notice that each packet is individually being ACKed by the receiver. We observe a behaviour where the sender transmits a burst of packets and the receiver then sends back ACKs for each of them. However later on in the trace, in particular at segment # 60, you will notice that the ACK with the acknowledgement field as 232166981 is actually acknowledging two segments with sequence #232164061 and # 232165521. From this point on there are several instances when the receiver sends an ACK for every other received segment. The receiver is typically sending a cumulative ACK for two TCP segments that it receives. As explained earlier this is due to the fact that TCP uses Delayed ACKs where the receiver waits for up to 500 msec for the arrival of another in-order segment and then sends a cumulative ACK for both of the received segments. (Refer to the Receiver event and action table in Section 3.5.4 of the text or Week 5 Slide No 65 Lecture notes).

Question 8. (**1 mark**)

The computation of TCP throughput largely depends on the selection of averaging time period. As a common throughput computation, in this question, we select the average time period as the whole connection time. Then, the average throughput for this TCP connection is computed as the ratio between the total amount data and the total transmission time. The total amount data transmitted can be computed by the difference between the sequence number of the first TCP segment (i.e. 1 byte for No. 4 segment) and the acknowledged sequence number of the last ACK (164091 bytes for No. 202 segment). Therefore, the total data are 164091 - 1 = 164090 bytes. The whole transmission time is the difference of the time instant of the first TCP segment (i.e., 0.026477 second for No.4 segment) and the time instant of the last ACK (i.e., 5.455830 second for No. 202 segment). Therefore, the total transmission time is 5.455830 - 0.026477 = 5.4294 seconds. Hence, the throughput for the TCP connection is computed as 164090/5.4294 = 30.222 KByte/sec.

Exercise 2: TCP

Question 1. (**0.5 mark**)

Seq = 2818463618. This is the Initial Sequence Number (ISN) chosen by the client.

Question 2. (**0.5 mark**)

Sequence Number in SYNACK Segment from the server = 1247095790. The ACK No is 2818463619. Server has added 1 in the ISN from the client to arrive at the ACK number.

Question 3. (**0.5 mark**)

The Client has sent Segment with Seq No = 2818463619 and ACK = 1247095791. There is no data included in this last segment of the three way handshake. This can be confirmed by looking at the later segment 298 as it is using the same Seq No.

Question 4. (**1.5 marks**)

The conversation is terminated by a Simultaneous close. This means that both Client and Server have initiated the FIN without receiving FIN from the other side (their FINs No 304 and 305 have crossed over in the network). This can be verified by looking at the Seq and ACK numbers in these FIN segments. ACK has not been incremented by 1 in the FIN that has been generated. Thus both Client and Server have initiated active close simultaneously.

Question 5. (**1 mark**)

Client: ISN = 2818463618, last ACK received by the Client has ACK No = 2818463653

Data sent = final ACK received - ISN - 2 (1 for SYN and 1 for FIN) = 2818463653 - 2818463618 - 2 = 33 Bytes

Server: ISN = 1247095790, last ACK received by the Server = 1247095832

Data sent = 1247095832 - 1247095790 - 2 = 40 Bytes

Resource created 4 months ago (Monday 16 July 2018, 02:50:39 PM), last modified 2 months ago (Tuesday 11 September 2018, 02:16:57 PM).

Comments

🔖   🔍 (/COMP3331/18s2/forums/search?forum_choice=resource/17345)   💬 (/COMP3331/18s2/forums/resource/17345)

💬 Add a comment

There are no comments yet.