1. What is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu?

2. What is the IP address and port number used by gaia.cs.umass.edu to receive the file.

Solution: Client computer (source)

IP address: 192.168.1.102

TCP port number: 1161

Destination computer: gaia.cs.umass.edu

IP address: 128.119.245.12

TCP port number: 80

4. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in the segment that identifies the segment as a SYN segment?

Solution: Sequence number of the TCP SYN segment is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu. The value is 0 in this trace.

The SYN flag is set to 1 and it indicates that this segment is a SYN segment.

5. What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the ACKnowledgement field in the SYNACK segment? How did gaia.cs.umass.edu determine that value? What is it in the segment that identifies the segment as a SYNACK segment?

Solution: Sequence number of the SYNACK segment from gaia.cs.umass.edu to the client computer in reply to the SYN has the value of 0 in this trace.

The value of the ACKnowledgement field in the SYNACK segment is 1. The value of the

ACKnowledgement field in the SYNACK segment is determined by gaia.cs.umass.edu

by adding 1 to the initial sequence number of SYN segment from the client computer (i.e.

the sequence number of the SYN segment initiated by the client computer is 0.).

The SYN flag and Acknowledgement flag in the segment are set to 1 and they indicate

that this segment is a SYNACK segment.

4

Figure 3: Sequence number and Acknowledgement number of the SYNACK segment

6. What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you'll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a "POST" within its DATA field.

7. 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 in the TCP connection (including the segment containing the HTTP POST)? 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 page 237 in text) after the receipt of each ACK? Assume that the value of the EstimatedRTT is equal to the measured RTT for the first segment, and then is computed using the EstimatedRTT equation on page 237 for all subsequent segments.

Note: Wireshark has a nice feature that allows you to plot the RTT for each of the TCP segments sent. Select a TCP segment in the "listing of captured packets" window that is being sent from the client to the gaia.cs.umass.edu server. Then select: Statistics->TCP Stream Graph->Round Trip Time Graph.

Solution: The HTTP POST segment is considered as the first segment. Segments 1 – 6

are No. 4, 5, 7, 8, 10, and 11 in this trace respectively. The ACKs of segments 1 – 6 are

No. 6, 9, 12, 14, 15, and 16 in this trace.

Segment 1 sequence number: 1

Segment 2 sequence number: 566

Segment 3 sequence number: 2026

Segment 4 sequence number: 3486

6

Segment 5 sequence number: 4946

Segment 6 sequence number: 6406

The sending time and the received time of ACKs are tabulated in the following table.

Sent time ACK received time RTT (seconds)

Segment 1 0.026477 0.053937 0.02746

Segment 2 0.041737 0.077294 0.035557

Segment 3 0.054026 0.124085 0.070059

Segment 4 0.054690 0.169118 0.11443

Segment 5 0.077405 0.217299 0.13989

Segment 6 0.078157 0.267802 0.18964

EstimatedRTT = 0.875 * EstimatedRTT + 0.125 * SampleRTT

EstimatedRTT after the receipt of the ACK of segment 1:

EstimatedRTT = RTT for Segment 1 = 0.02746 second

EstimatedRTT after the receipt of the ACK of segment 2:

EstimatedRTT = 0.875 * 0.02746 + 0.125 * 0.035557 = 0.0285

EstimatedRTT after the receipt of the ACK of segment 3:

EstimatedRTT = 0.875 * 0.0285 + 0.125 * 0.070059 = 0.0337

EstimatedRTT after the receipt of the ACK of segment 4:

EstimatedRTT = 0.875 * 0.0337+ 0.125 * 0.11443 = 0.0438

EstimatedRTT after the receipt of the ACK of segment 5:

EstimatedRTT = 0.875 * 0.0438 + 0.125 * 0.13989 = 0.0558

EstimatedRTT after the receipt of the ACK of segment 6:

EstimatedRTT = 0.875 * 0.0558 + 0.125 * 0.18964 = 0.0725

second

7

Figure 5: Segments 1 – 6

Figure 6: ACKs of segments 1 - 6

Figure 7: Round Trip Time Graph

8. What is the length of each of the first six TCP segments?

Solution: Length of the first TCP segment (containing the HTTP POST): 565 bytes

Length of each of the other five TCP segments: 1460 bytes (MSS)

Figure 8: Lengths of segments 1 - 6

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

Solution: The minimum amount of buffer space (receiver window) advertised at gaia.cs.umass.edu for the entire trace is 5840 bytes, which shows in the first acknowledgement from the server. This receiver window grows steadily until a maximum receiver buffer size of 62780 bytes. The sender is never throttled due to lacking of receiver buffer space by inspecting this trace.

Figure 9: Minimum receive window advertised at gaia.cs.umass.edu (packet No. 2)

10. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?

Solution: There are no retransmitted segments in the trace file. We can verify this by checking the sequence numbers of the TCP segments in the trace file. In the TimeSequence-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 neighboring segments.

Figure 10: Sequence numbers of the segments from the source (192.168.1.102) to the destination

(128.119.245.12)

11. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (see Table 3.2 on page 247 in the text).

Solution: The acknowledged sequence numbers of the ACKs are listed as follows.

acknowledged sequence number acknowledged data

ACK 1 566 566

ACK 2 2026 1460

ACK 3 3486 1460

ACK 4 4946 1460

ACK 5 6406 1460

ACK 6 7866 1460

ACK 7 9013 1147

ACK 8 10473 1460

ACK 9 11933 1460

ACK 10 13393 1460

ACK 11 14853 1460

ACK 12 16313 1460

…

The difference between the acknowledged sequence numbers of two consecutive ACKs

indicates the data received by the server between these two ACKs. By inspecting the amount of acknowledged data by each ACK, there are cases where the receiver is

ACKing every other segment. For example, segment of No. 80 acknowledged data with

2920 bytes = 1460*2 bytes.

Figure 8: Cumulative ACKs (No. 80, 87, 88, etc) where the receiver is ACKing every other received

segment.

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

Solution: 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.