Daniel Kushner

Vanessa Ulloa

CST 311

02 June 2015

Lab 5

1. 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?
   1. The address is 192.168.1.102 and the port number is 1161
2. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?
   1. The IP address is 128.119.245.12 and the port number is 80
3. If you have been able to create your own trace, answer the following question: What is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu?
   1. I used the Wireshark trace this week as I am traveling abroad.
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?
   1. It has a SYN flag that is set to 1 that indicates it’s a SYN segment. The sequence number has a value of zero
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?
   1. The sequence number of the segment from umass.edu to the client has a value of 0 because it is responding to the initial message, which started with a value of 0. It is identified as a SYNACK segment because the ACK message has added a 1 to it settings its value to 1. If I understand correctly packets going back and fourth here each add 1 so the first packet from the client was 0 and the first ack is 1.
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.
   1. The 4th segment contains the HTTP POST command. The sequence number has a value of 1.
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 thEstimatedRTTvalue (see Section 3.5.3, page 239 in text) after the receipt of each ACK? Assume that the value of the EstimatedRTTis equal to the measured RTT for the first segment, and then is computed using the EstimatedRTT equation on page 239 for all subsequent segments.
   1. Segments 1-6 are found in packet numbers 4, 5, 7, 8, 10 and 11 and the ACK for them are 6, 9, 12, 14, 15 and 16

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Sent Time | ACK Received | Time RTT | Sequence Number |
| Segment 1 | 0.026477 | 0.053937 | 0.02746 | 1 |
| Segment 2 | 0.041737 | 0.077294 | 0.035557 | 566 |
| Segment 3 | 0.054026 | 0.124085 | 0.070059 | 2026 |
| Segment 4 | 0.05469 | 0.169118 | 0.11443 | 3486 |
| Segment 5 | 0.077405 | 0.217299 | 0.13989 | 4946 |
| Segment 6 | 0.078157 | 0.267802 | 0.18 | 6406 |

|  |  |
| --- | --- |
| Estimated RTT After ACK of: (in seconds) | |
| Segment 1: | 0.02746 |
| Segment 2: | 0.0285 |
| Segment 3: | 0.0337 |
| Segment 4: | 0.0438 |
| Segment 5: | 0.0558 |
| Segment 6: | 0.0725 |

1. What is the length of each of the first six TCP segments?
   1. The first segment is 565 Bytes and the others are 1460 Bytes each
2. 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?
   1. The minimum amount of buffer space is 5840 Bytes. The sender is never throttled due to lack of buffer space.
3. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?
   1. No there are no retransmitted traces. I used the time vs sequence graph provided in the lab to determine this. All of the sizes are similar, if it had lost a packet one section would be smaller than the others.
4. 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).
   1. The difference between ACKs is the amount of data received by the server between each. In some cases it is acknowledging every two packets this is indicated in instances where the acknowledged data is twice its normal size so 1460 \* 2
5. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.
   1. You take the total time period and compute the ratio between the total mount of data and the transmission time. In this case the total data is 164090 and the time is 5.4294 seconds.

So our total would be 30.222 kiloBytes per second. I think if you take a shorter time period you can use this to get a throughput for a given instant.

1. Use the Time-Sequence-Graph(Stevens) plotting tool to view the sequence number versus time plot of segments being sent from the client to the gaia.cs.umass.edu server. Can you identify where TCP’s slow start phase begins and ends, and where congestion avoidance takes over? Comment on ways in which the measured data differs from the idealized behavior of TCP that we’ve studied in the text
   1. I don’t see how you can answer this question it doesn’t appear congested. I think the TCP sender is not sending enough data to make it congested.
2. Answer each of two questions above for the trace that you have gathered when you transferred a file from your computer to gaia.cs.umass.edu