Joel Castillo

CS6843 - Computer Networking

Prof. Portnoy

Wireshark Lab 3: TCP

Please Note: Screenshots are on the last page.

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](http://gaia.cs.umass.edu)?
   * IP Address - 192.168.1.102; Port - 1161
2. What is the IP address of [gaia.cs.umass.edu](http://gaia.cs.umass.edu)? On what port number is it sending and receiving TCP segments for this connection?
   * IP Address - 128.119.245.12; Port 80
3. What is the IP address and TCP port number used by your client computer (source) to transfer the file to [gaia.cs.umass.edu](http://gaia.cs.umass.edu)?
   * Wasn’t able to get the trace to work properly on my computer.

**Part 3 - TCP Basics**

1. 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](http://gaia.cs.umass.edu)? What is it in the segment that identifies the segment as a SYN segment?

The sequence number of the TCP SYN segment is 0 in the trace. Since the SYN flag is

set to 1, it indicates that the segment is a SYN segment.

1. What is the sequence number of the SYNACK segment sent by [gaia.cs.umass.edu](http://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](http://gaia.cs.umass.edu) determine that value? What is it in the segment that identifies the segment as a SYNACK segment?

The sequence number of the SYNACK segment is 0 in the trace. The value of the

ACKnowledgement field in the SYNACK segment is 1. This is determined by adding 1 to

the original sequence number of the SYN segment on the server.

Both the SYN and ACKnowledgment flags are set to 1, marking the segment as a

SYNACK segment.

1. 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.

Segment 4 contains the HTTP POST command. The sequence number for the segment

is 1.

1. Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection. What are the sequence numbers of the first 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 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 after the receipt of each ACK?

|  | Sequence Number | Sent Time | ACK Received Time | RTT (Seconds) | Estimated RTT |
| --- | --- | --- | --- | --- | --- |
| Segment 1 | 1 | 0.026477 | 0.053937 | 0.02746 | 0.02746 |
| Segment 2 | 566 | 0.041737 | 0.077294 | 0.035557 | 0.0285 |
| Segment 3 | 2026 | 0.054026 | 0.124085 | 0.070059 | 0.0337 |
| Segment 4 | 3486 | 0.054690 | 0.169118 | 0.11443 | 0.0438 |
| Segment 5 | 4946 | 0.077405 | 0.217299 | 0.13989 | 0.0558 |
| Segment 6 | 6406 | 0.078157 | 0.267802 | 0.18964 | 0.0725 |

EstimatedRTT = .875 \* EstimatedRTT + .125 \* SampleRTT

Segment 1: RTT for Segment 1 = .02746

Segment 2: .875 \* .02746 + .125 \* .035557 = .0285

Segment 3: .875 \* .0285 + .125 \* .070059 = .0337

Segment 4: .875 \* .0377 + .125 \* .11443 = .0438

Segment 5: .875 \* .0438 + .125 \* .13989 = .0558

Segment 6: .875 \* .0558 + .125 \* .19064 = .0725

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

The length of the first TCP segment is 565 bytes; The remaining 5 TCP segments are

1460 bytes.

1. 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?

The minimum amount of buffer space for the trace is 5840 bytes (shown in the first

ACKnowledgement). The receiver window reaches a maximum of 62780 bytes. The

sender is never throttled.

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

There are no retransmitted segments in the file. This can be determined by checking the sequence numbers of the segments in the file. If there were a retransmitted segment, the number of the transmitted segment would be smaller than the neighboring segments, instead of the numbers increasing in order.

1. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment?

|  | Sequence Number | 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 |

In segments 80, 87, and 88, the receiver is ACKnowledging every other segment. This is determined by checking the amount of data ACKnowledged by each ACK.

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

The throughput can be calculated by using the connection time as the average time period, which means average throughput is ratio of the total amount of data transferred and the time for the connection. In this case the total amount of data transmitted is 164090 (the ACKnowledged number of the last segment and the sequence number of the first segment). The time is the difference for the time instant of the first TCP segment and the last ACKnowledgment, 5.4294 seconds. Therefore the throughput is 164090/5.4294 = 30.222 KB/sec

**Part 4 - TCP Congestion Control In Action**

1. Use the Time-Sequence-Graph (Stevens) plotting tool to view the number versus time plot of segments being sent from the client to the [gaia.cs.umass.edu](http://gaia.cs.umass.edu) server. Can you identify where TCP’s slowstart phase begins and ends, and where congestion avoidance takes over? Comment on the ways in which the measured data differs from the idealized behavior of TCP that we’ve studied in the text.

TCP slowstart starts at the beginning of the transmission with the HTTP POST segment. The TCP window size lower-bound can be estimated by looking at the amount of unACKnowledged data, which is the amount of outstanding data. Based on the data obtained in the trace, the size of the window must be greater than 8192 bytes because the amount of outstanding data never exceeds 8192 bytes. This also means that we can’t determine the end of slowstart and the beginning of congestion control because the data is not being sent in a manner that would cause congestion control to happen, since the application is not sending any more than 8192 bytes at a time (it waits for a response).

In the text, ideal TCP behavior assumes that a sender is always sending data, and not necessarily waiting for an ACKnowledgement. In this case, the transfer is slowed down by the application waiting for acknowledgements before sending data, even if the sender can transmit more data (small file that is smaller than the transmission window). This means that transmissions are complete before the end of the slow start phase, meaning the transfer is slow.

1. Answer each of the two questions above for the trace that you have gathered when you transferred a file from your computer to [gaia.cs.umass.edu](http://gaia.cs.umass.edu).



