Wireshark Lab: TCP

Brent Berghmans 1334252

### 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?

IP: 192.168.1.102

Port: 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?

IP: 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?

IP: 192.168.0.198

Port: 56736

### 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?

The sequence number is 0.

It is identified as a SYN segment because the Syn flag is set in the header of the TCP packet.

### 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?

The sequence number of the SYNACK is 0, the acknowledgement field is 1.

The acknowledgement field is 1 because the servers says “Ok, I received a packet with sequence number 0, I now expect a packet with sequence number 1.”.

It’s identified as a SYNACK because the Syn flag is set to 1 and the Acknowledgement flag is set to 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

The sequence number of this TCP segment is 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 the EstimatedRTT value (see Section 3.5.3, page 239 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 239 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.

See wireshark-tcp-7-brent-berghmans-1334252.xlsx for this question..

Also relevant:

wireshark-tcp-ping-brent-berghmans-1334252.png, wireshark-tcp-ping-brent-berghmans-1334252.txt

wireshark-tcp-7-RTT-brent-berghmans-13345252.png

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

|  |  |
| --- | --- |
| Sequence Number | Length (bytes) |
| 1 | 655 |
| 666 | 1460 |
| 2126 | 1460 |
| 3586 | 1460 |
| 5046 | 1460 |
| 6506 | 1460 |

### 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?

The lowest server window size of the whole stream was 14600.

Wireshark reported that the TCP window was full three times, so yes, the lack of buffer size did throttle the sender a couple of times.

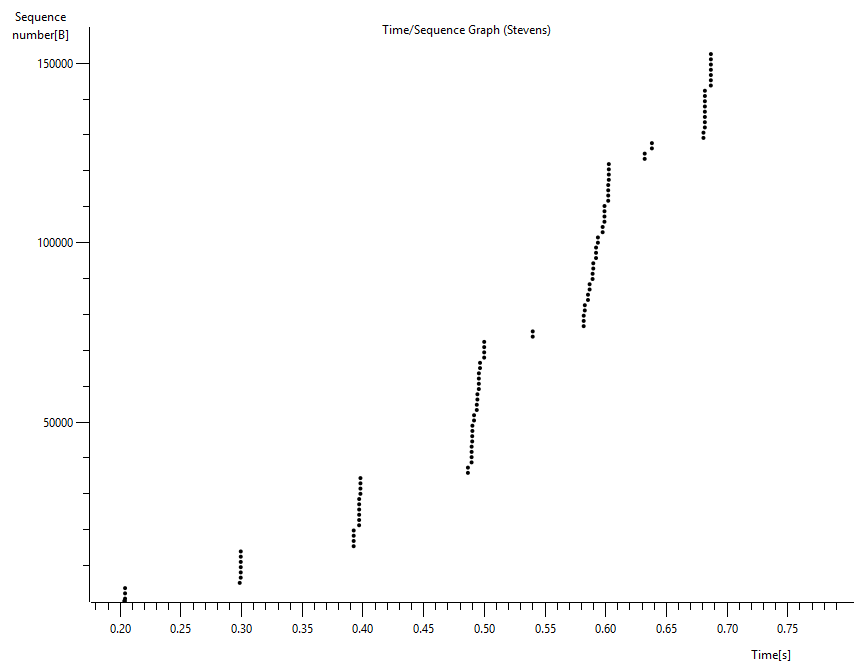
### 10. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?

Wireshark will report a segment retransmission in the package listing window by saying

[TCP Retransmission] in the info section.

I found 1 retransmission but it was on a different TCP stream than the file upload stream.

I then excluded other TCP streams and checked the graph of this stream.   
If there was a retransmission I should see 2 or more dots on a horizontal line but this does not seem to be the case.



### 

### 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).

The server acknowledges 1460 bytes on average. But there are a couple of cases where the server acknowledges 2920 bytes (2 \* 1460), for instance the segment with sequence number 2126 does not get acknowledged but the segment after that (with seq number 3586) does get acknowledged. So even though 3 segments were sent, only 2 ACK’s were received. A second example would be the jump in ACK’s from 12346 to 15266. The ACK of the segment with sequence number 13806 is missing there as well. I checked if maybe these ACK’s were delayed but that did not seem to be the case.

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

The timestamp of the first packet that is sent is 13:27:15.700675000.

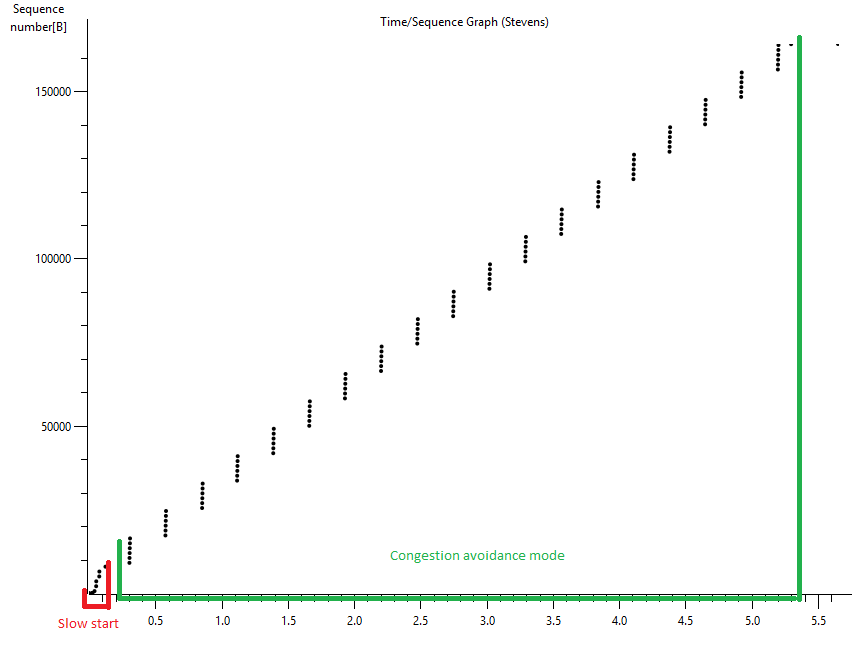
The timestamp of the last ACK received is 13:27:16.282666000.

Now we can calculate the time difference which is 0.581991s.

The file size is 150KB.

150KB / 0.581991s is roughly 257KB/s which is 257 \* 1000 Bytes/s.

### 13. 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 slowstart 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.

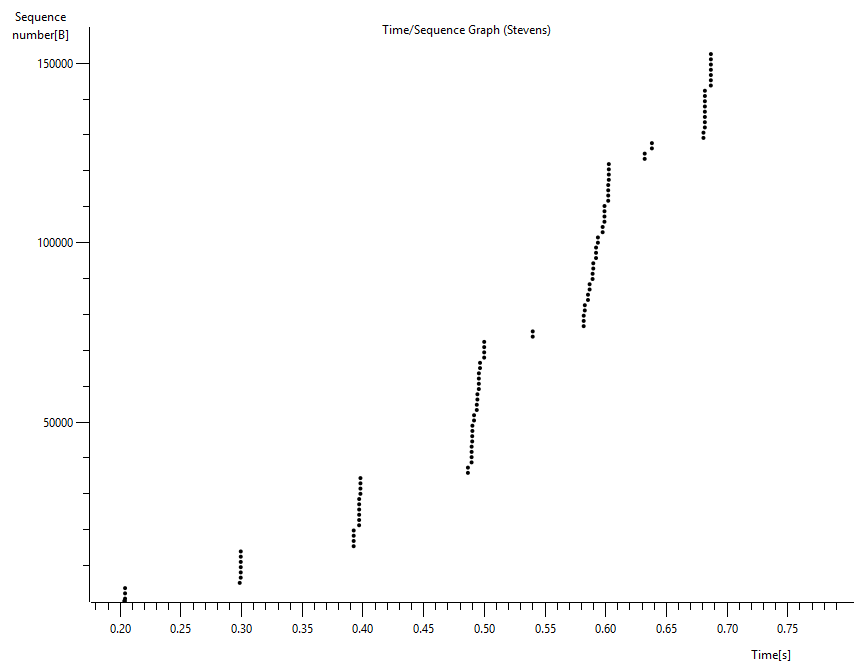


We see that segments are sent in groups of 5, the transmit window does not seem to increase nor decrease which is not the normal for congestion avoidance mode, it should be a constant loop of increasing the transmit window with 1 until congestion, we would then divide the transmit window by 2. We see neither of those things happening.

### 

### 

### 14. 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



I’m not seeing any linear activity here so I assume we are in slow start mode the whole time, we might be in congestion avoidance at the end but I do not have enough data to determine if that is the case. We can also see that only a few segments are sent at certain times, for instance at 0.55 and 0.65. I think the slowdown at 0.65 is due to the server tcp window being full and we having to wait for ACK’s. I do not know why the anomaly at 0.55 happened.