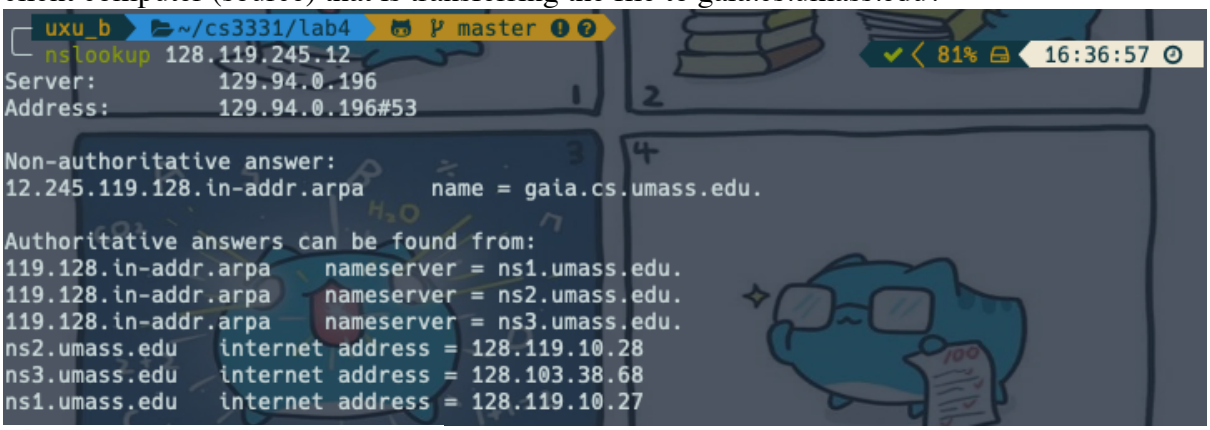
z5147986
Lab4

1.1
What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection? 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?



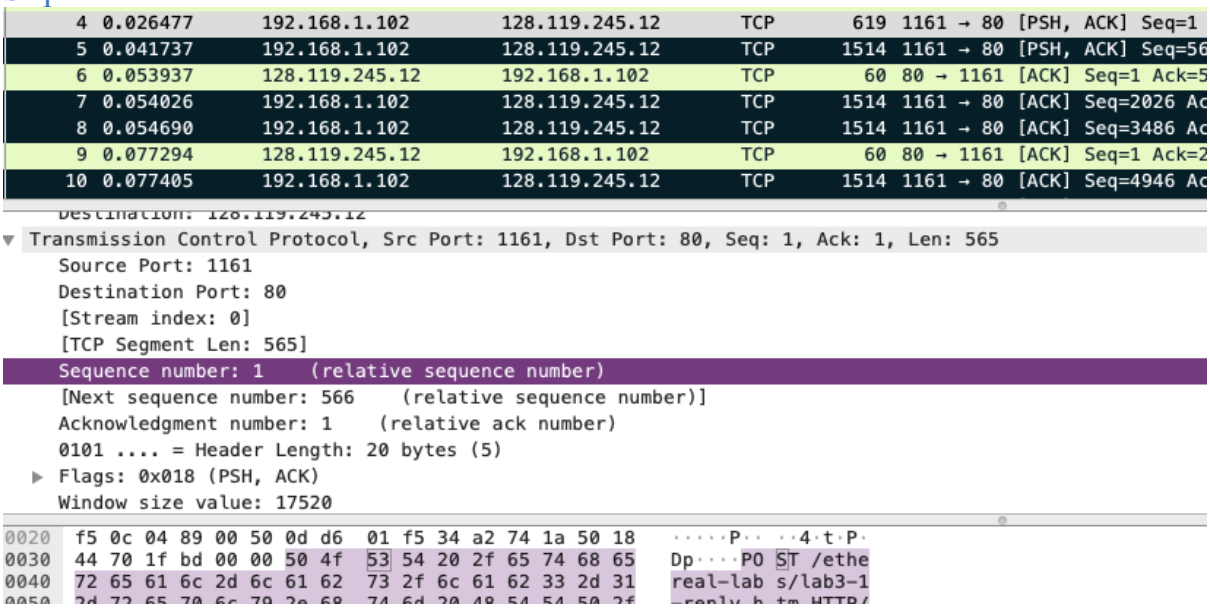
Source Port: 1161

Destination Port: 80

The Server IP is 128.119.245.12 and port is 1161
The Client IP is 192.168.1.102 and port is 80

1.2
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.

Sequence number is 1

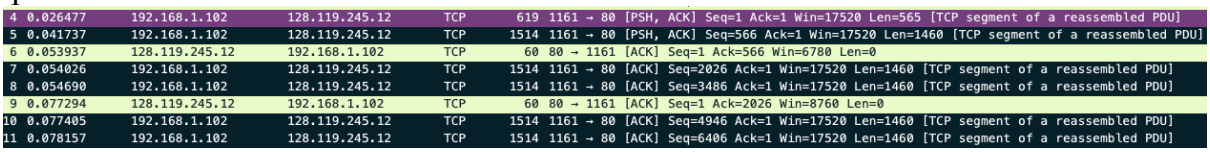


z5147986
Lab4

1.3

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) sent from the client to the web server (Do not consider the ACKs received from the server as part of these six segments)? 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 relevant parts of Section 3.5 or lecture slides) after the receipt of each ACK? Assume that the initial value of EstimatedRTT is equal to the measured RTT (SampleRTT) for the first segment, and then is computed using the EstimatedRTT equation for all subsequent segments. Set alpha to 0.125

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. However, do not use this graph to answer the above question.



Client → Web server

Segment	Sequence	Time(sec)	ACK	RTT(sec)
1	1	0.026477	1 + 565 = 566	0.053937 – 0.026477 = 0.02746
2	566	0.041737	566 + 1460 = 2026	0.077294 – 0.041737 = 0.035557
3	2026	0.054026	2026 + 1460 = 3486	0.124085 – 0.054026 = 0.070059
4	3486	0.054690	3486 + 1460 = 4946	0.169118 – 0.054690 = 0.114428
5	4946	0.077405	4946 + 1460 = 6406	0.217299 – 0.077405 = 0.139894
6	6406	0.078157	6406 + 1460 = 7866	0.267802 – 0.078157 = 0.189645

Estimated RTT(sec)

Segment 1 → (1-0.125) * 0.02746 + 0.125 * 0.02746 = 0.02746
Segment 2 → (1-0.125) * 0.02746 + 0.125 * 0.035557 = 0.02847
Segment 3 → (1-0.125) * 0.02847 + 0.125 * 0.070059 = 0.03367
Segment 4 → (1-0.125) * 0.03367 + 0.125 * 0.114428 = 0.04376
Segment 5 → (1-0.125) * 0.04376 + 0.125 * 0.139894 = 0.05578
Segment 6 → (1-0.125) * 0.05578 + 0.125 * 0.189645 = 0.07251

1.4

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

Segment 1 → 565
Segment 2 → 1460
Segment 3 → 1460
Segment 4 → 1460
Segment 5 → 1460

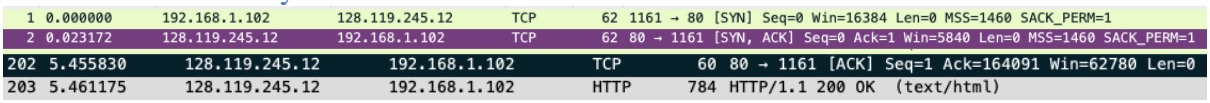
z5147986
Lab4

Segment 6 → 1460

1.5

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

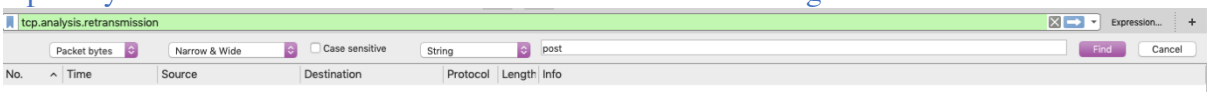
The minimum is 5840 bytes showed in the first ack from the server, then it grows till a maximum of 62780 bytes



1.6

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

There are no retransmitted segments in the trace file. I sorted the source in ascending order and check the sequence number, then I found a more efficient way is to use tcp.analysis.retransmission in wireshark which returned nothing.



1.7

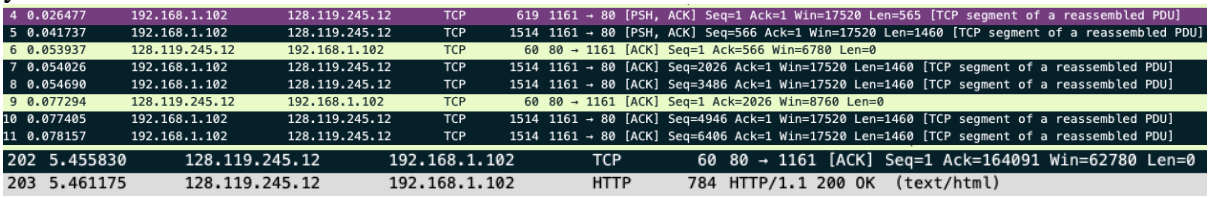
How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (recall the discussion about delayed acks from the lecture notes or Section 3.5 of the text).

Data the receiver typically acknowledge in an ACK is 1460 bytes.

I can't seem to identify cases where receiver is acking every other received segment. The server ack for each packet received, thus there is no delay acking.

1.8

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



The total amount of data is the difference between the sequence number of the last ACK and the first segment.

164091 – 1 = 164090 bytes

The total transmission time is the difference between the time of last ACK and the time of first segment.

5.455830 – 0.026477 = 5.429353 seconds

The throughput is total amount of data divide by total transmission time

164090/5.429353 = 30222.7539819 bytes/second

z5147986
Lab4

2.1
What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and server?
The sequence number is 2818463618

2.2
What is the sequence number of the SYNACK segment sent by the server to the client computer in reply to the SYN? What is the value of the acknowledgment field in the SYNACK segment? How did the server determine that value?
The sequence number is 1247095790
The acknowledgment is 2818463619, determined by seq from client → server + 1

2.3
What is the sequence number of the ACK segment sent by the client computer in response to the SYNACK? What is the value of the Acknowledgment field in this ACK segment? Does this segment contain any data?
The sequence number is 2818463619
The acknowledgment is 1247095791
This segment doesn't contain any data

2.4
Who has done the active close? client or the server? how you have determined this? What type of closure has been performed? 3 Segment (FIN/FINACK/ACK), 4 Segment (FIN/ACK/FIN/ACK) or Simultaneous close?
10.9.16.201, the client done the active close, since the client sent the FINACK first.
The type of closure used is simultaneous close.

304	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [FIN, ACK] Seq=2818463652 Ack=1247095831 win=65535
305	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [FIN, ACK] Seq=1247095831 Ack=2818463652 win=262144
306	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [ACK] Seq=2818463652 Ack=1247095832 win=65535
308	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [ACK] Seq=1247095831 Ack=2818463653 win=262144

2.5
How many data bytes have been transferred from the client to the server and from the server to the client during the whole duration of the connection? What relationship does this have with the Initial Sequence Number and the final ACK received from the other side?
Data bytes from client → server is 2818463653 – 2818463618 – 2(SYN + FIN) = 33 bytes
Data bytes from server → client is 1247095832 – 1247095790 – 2(SYN + FIN) = 40 bytes
The relationship is the bytes sent