UDP TASKS| WIRESHARK

1. **Select one packet. From this packet, determine how many fields there are in the UDP header. (Do not look in the textbook! Answer these questions directly from what you observe in the packet trace.) Name these fields.**

Source Port, Destination Port, Length, Checksum

1. **From the packet content field, determine the length (in bytes) of each of the UDP header fields.**

Length = 50.

1. **The value in the Length field is the length of what? Verify your claim with your captured UDP packet.**

The value in the length field is the segment length of each packet. Since, the data is sent in chunks / small packets.

1. **What is the maximum number of bytes that can be included in a UDP payload?**

65,535 bytes

1. **What is the largest possible source port number?**

65,535

1. **What is the protocol number for UDP? Give your answer in both hexadecimal and decimal notation. (To answer this question, you’ll need to look into the IP header.)**

UDP protocol no. (17)2 , (11)16 , (RFC 768)

1. **Search “UDP” in Google and determine the fields over which the UDP checksum is calculated.**

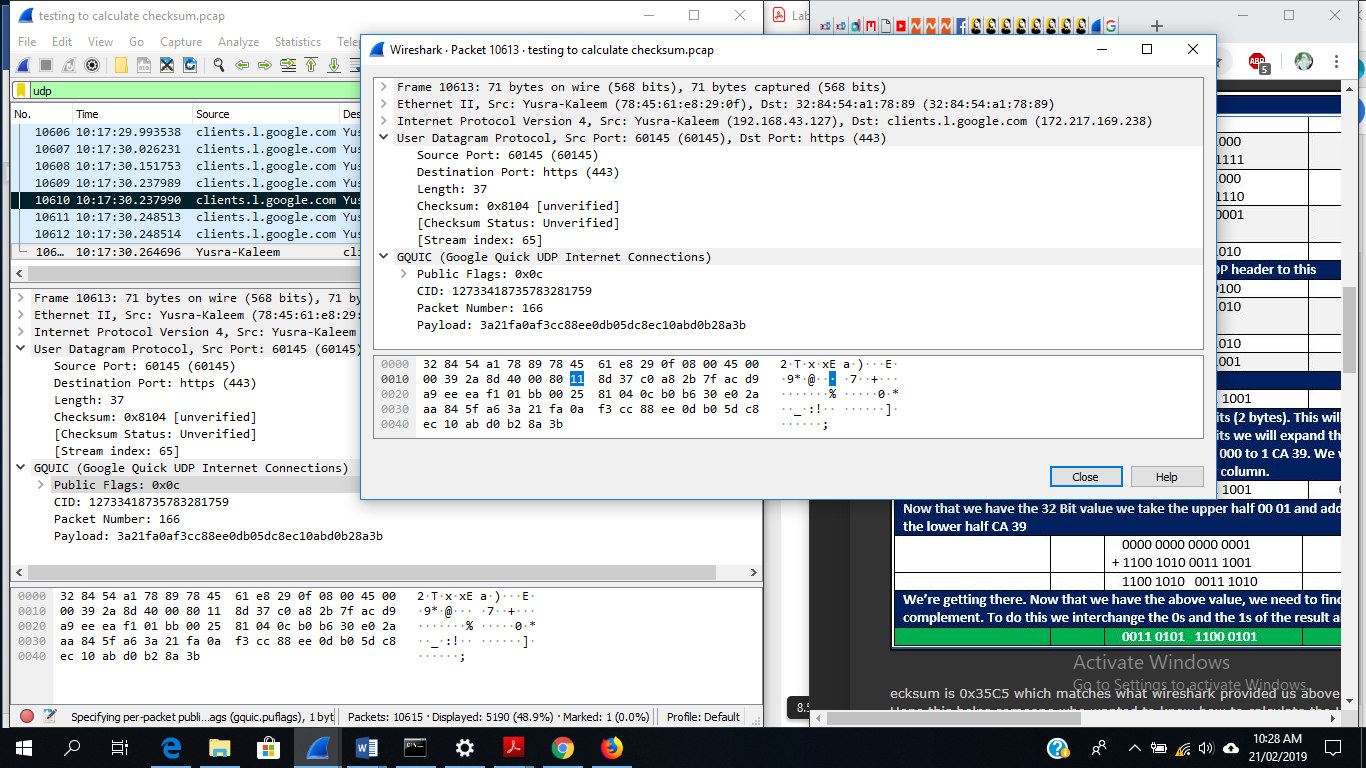
In UDP, checksum is calculated by adding source IP, destination IP, UDP protocol, padding length, UDP source port, UDP destination port, UDP length, UDP Data.

1. **Examine a pair of UDP packets in which the first packet is sent by your host and the second packet is a reply to the first packet. Describe the relationship between the port numbers in the two packets.**

When the packet is sent by our host, the port number / source port was 55865, and destination port was 53. Similarly, when we get the reply the source port was 53 and the destination port was 55866. The port number changes at our end due to thousands of users accessing a single server.

**Extra Credit**

1. Capture a small UDP packet. Manually verify the checksum in this packet. Show all work and explain all steps.



TCP TASKS | WIRESHARK

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? To answer this question, it’s probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the “details of the selected packet header window” (refer to Figure 2 in the “Getting Started with Wireshark” Lab if you’re uncertain about the Wireshark windows).

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?

If you have been able to create your own trace, answer the following question:

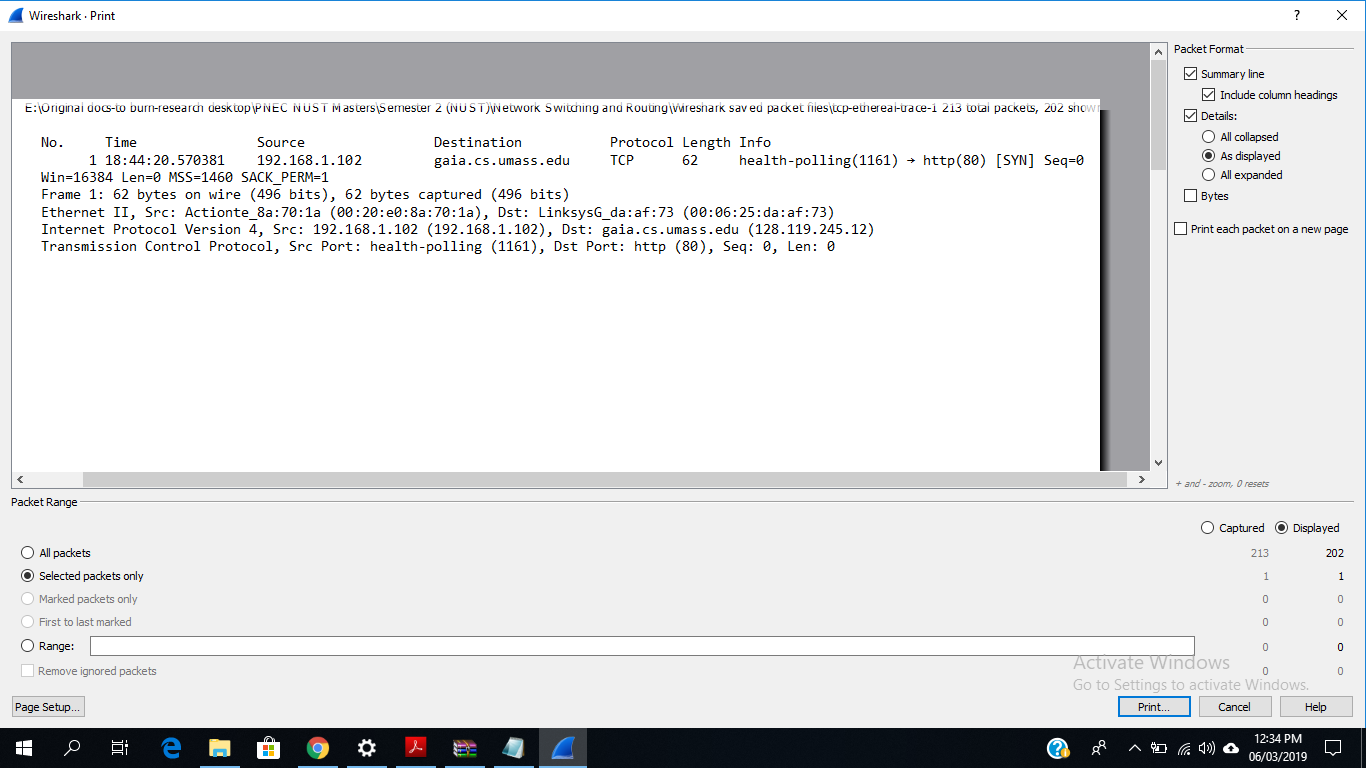
3. What is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu?

Since this lab is about TCP rather than HTTP, let’s change Wireshark’s “listing of captured packets” window so that it shows information about the TCP segments containing the HTTP messages, rather than about the HTTP messages. To have Wireshark do this, select *Analyze->Enabled Protocols.* Then uncheck the HTTP box and select *OK*. You should now see an Wireshark window that looks like as shown above:

This is what we’re looking for - a series of TCP segments sent between your computer and gaia.cs.umass.edu. We will use the packet trace that you have captured (and/or the packet trace *tcp-ethereal-trace-1* in http://gaia.cs.umass.edu/wireshark-labs/

**Solutions:**

*Packet trace of TCP ethereal downloaded file:*



*Answers of 1 & 2 of downloaded trace:*

Internet Protocol Version 4, Src: (192.168.1.102), Dst: gaia.cs.umass.edu (128.119.245.12)

Source Port: health-polling (1161), Destination Port: http (80)

*Packet of own trace:*

Graphical user interface, text, application

Description automatically generated

*Answers of 1 & 2 of own trace:*

Source: 192.168.1.102 (192.168.1.102) , Source Port: 51254 (51254)

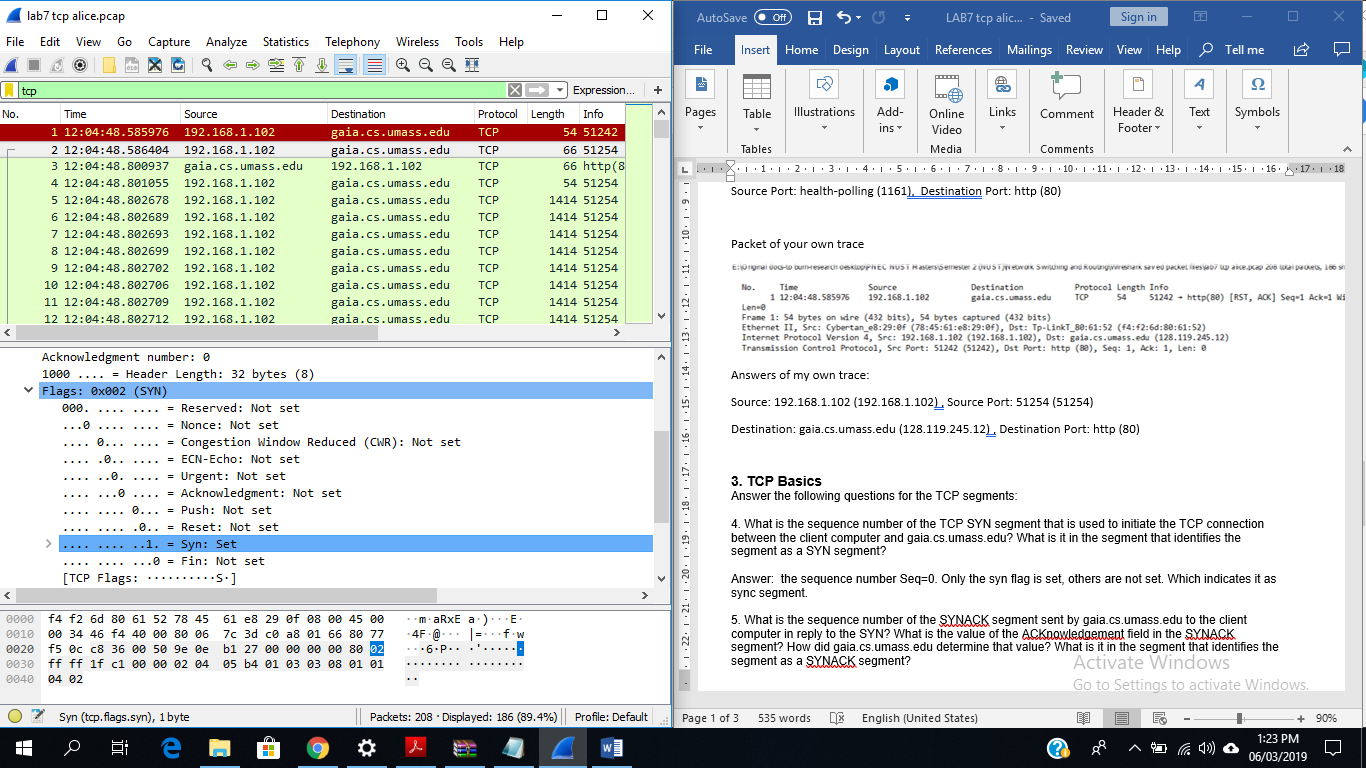
Destination: gaia.cs.umass.edu (128.119.245.12) , Destination Port: http (80)

**3. TCP Basics**

*Answer the following questions for the TCP segments:*

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?

***Answer***: the sequence number Seq=0. Only the syn flag is set, others are not set. Which indicates it as sync 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?

***Answer***: The sequence number Seq=0, the ACK=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 and Acknowledgement flags are set to 1 which indicate that this segment is a 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.

***Answer***: Sequence number: 152321 (relative sequence number)

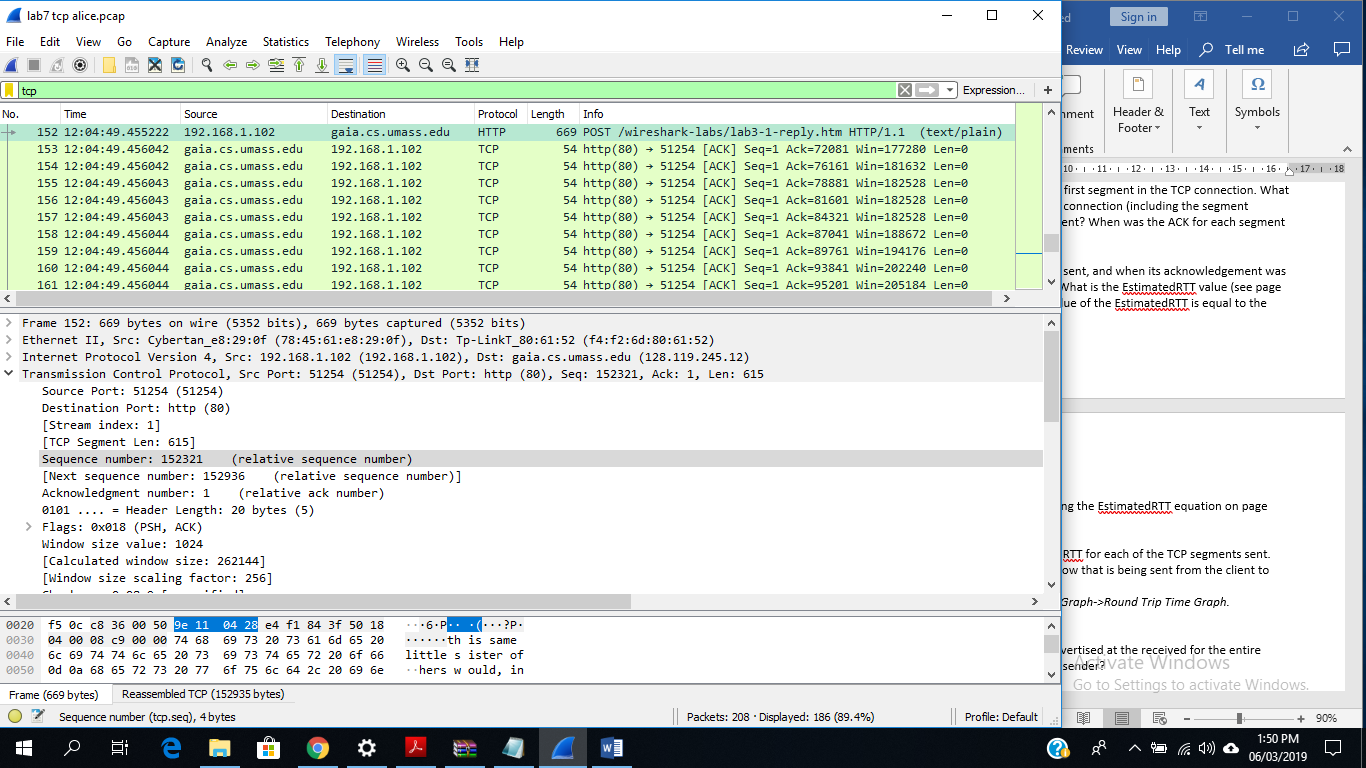
Graphical user interface, text, application

Description automatically generated

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?

***Answer***: Considering HTTP POST as first segment, the six segments sequence numbers are as follows:

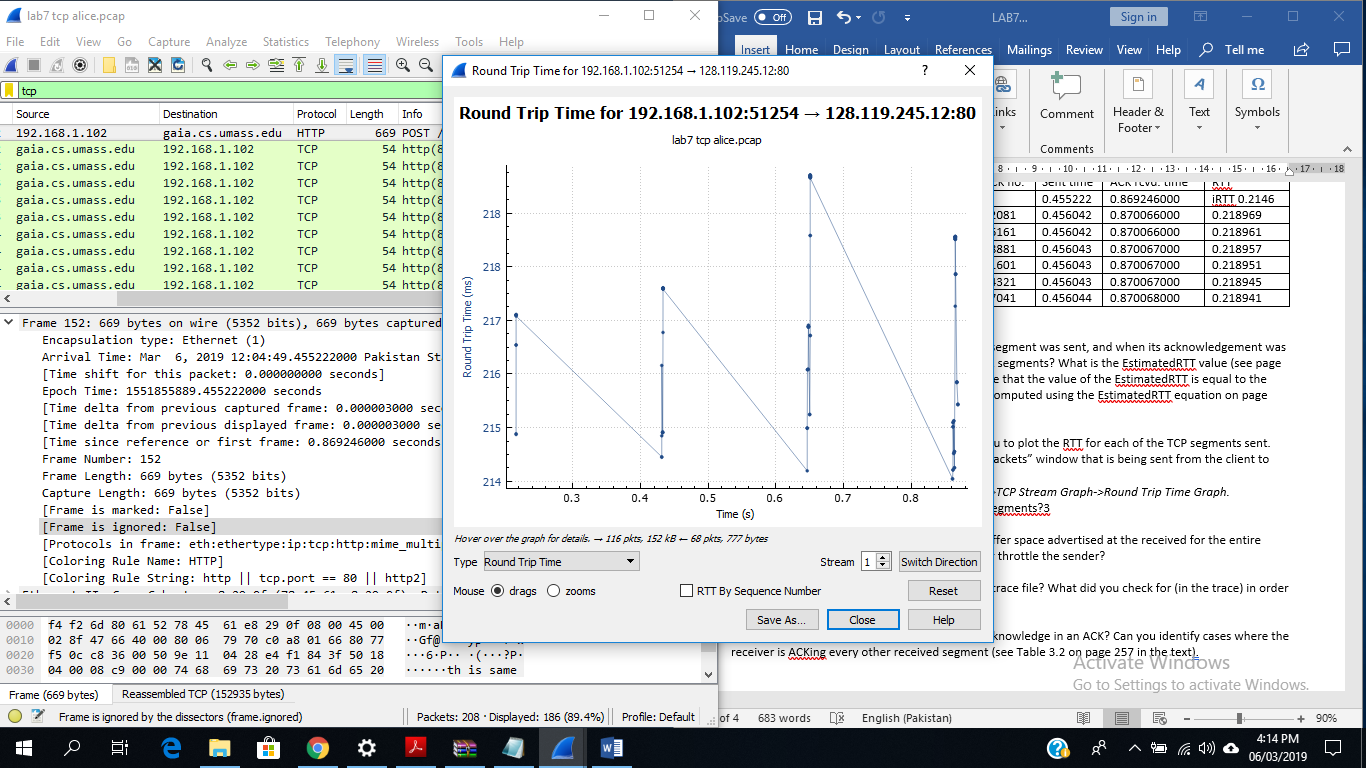
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Segment no. -> Serial no. | Sequence no. | ACK no. | Sent time | ACK rcvd. time | RTT |
| 1, | 152321 | 1 | 0.455222 | 0.869246000 | iRTT 0.2146 |
| 2 | 1 | 72081 | 0.456042 | 0.870066000 | 0.218969 |
| 3 | 1 | 76161 | 0.456042 | 0.870066000 | 0.218961 |
| 4 | 1 | 78881 | 0.456043 | 0.870067000 | 0.218957 |
| 5 | 1 | 81601 | 0.456043 | 0.870067000 | 0.218951 |
| 6 | 1 | 84321 | 0.456043 | 0.870067000 | 0.218945 |
| 7 | 1 | 87041 | 0.456044 | 0.870068000 | 0.218941 |



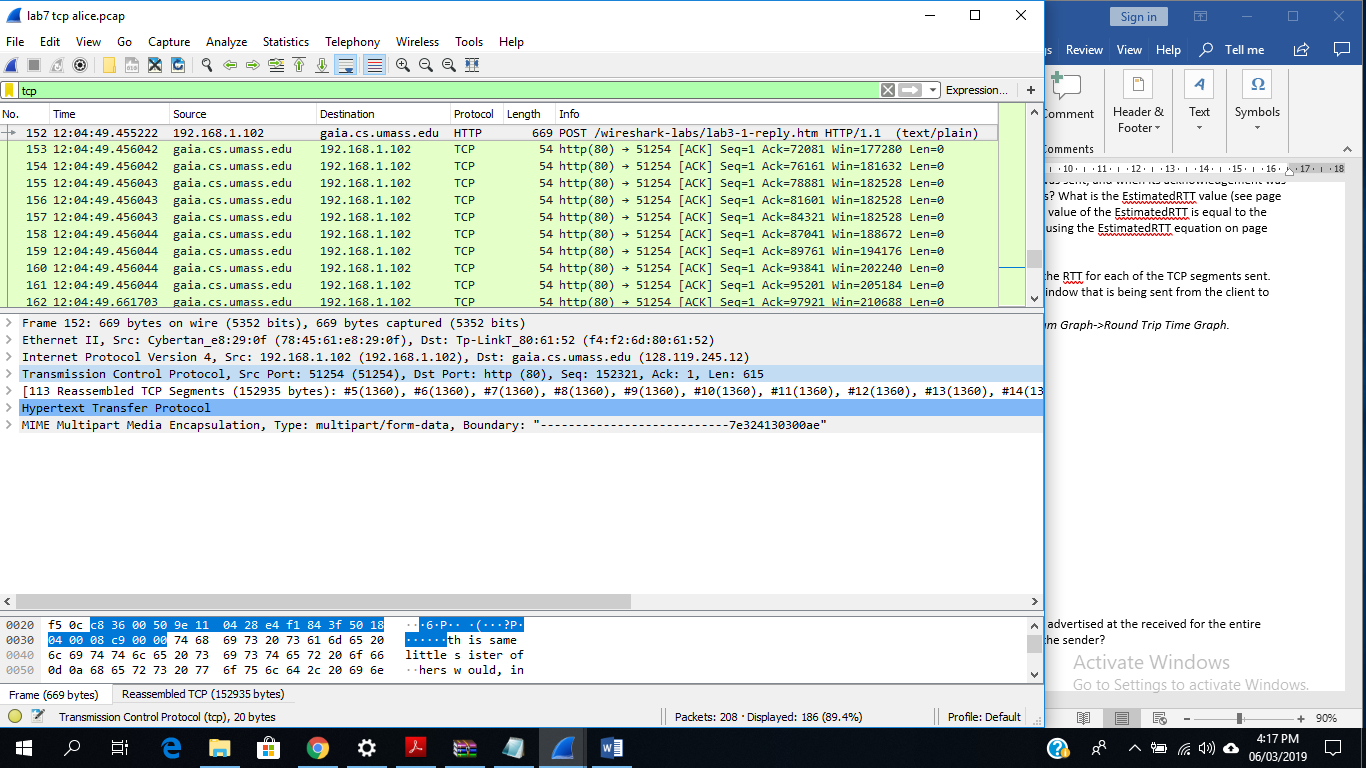
8. Given the difference between when each TCP segment was sent, and when its acknowledgement was received, what is the RTT value for each of the six segments? What is the EstimatedRTT value (see page 249 in text) after the receipt of each ACK? Assume that the value of the EstimatedRTT is equal to the measured RTT for the first segment, and then is computed using the EstimatedRTT equation on page 249 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

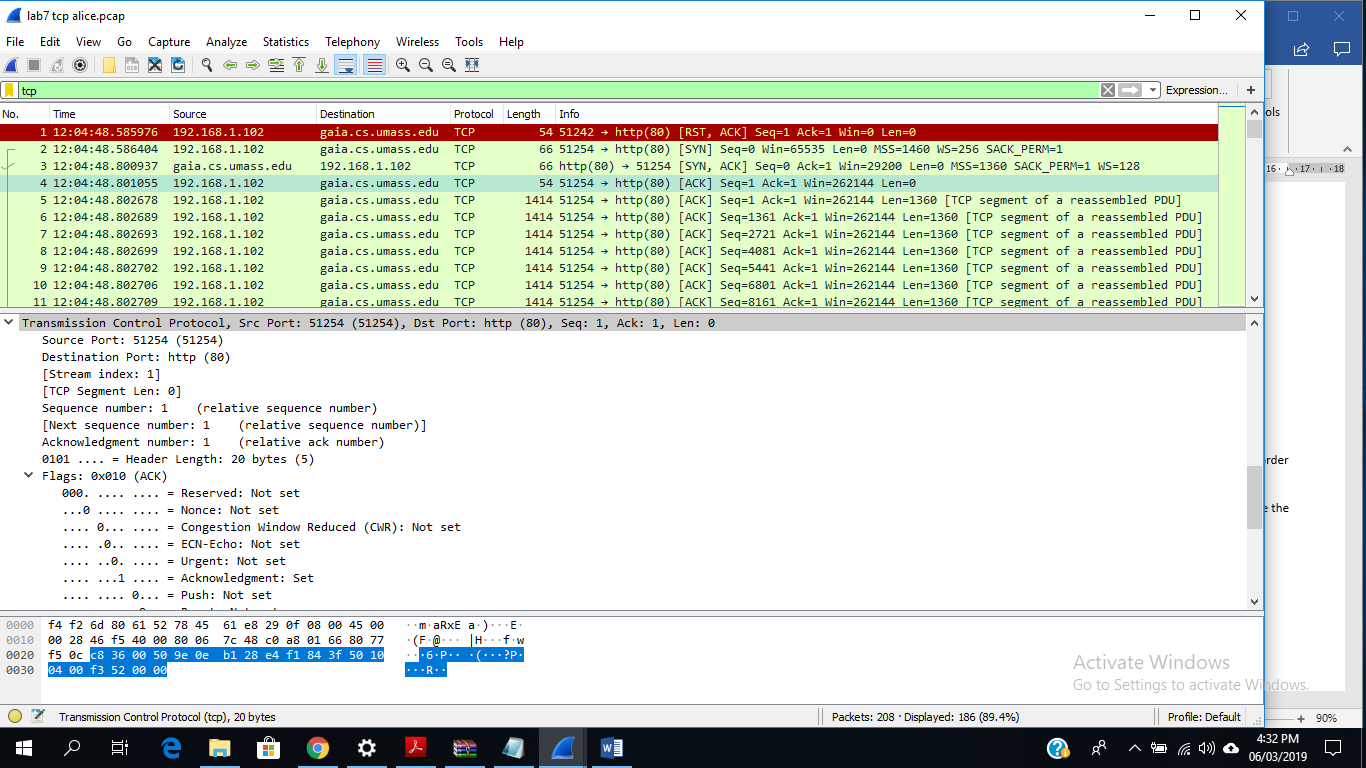
gaia.cs.umass.edu server. Then select: *Statistics->TCP Stream Graph->Round Trip Time Graph.*



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



10. 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 (receiver window) advertised at gaia.cs.umass.edu for the entire trace is 29200 bytes, which shows in the first acknowledgement from the server. No, the sender did not throttle in this trace.

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

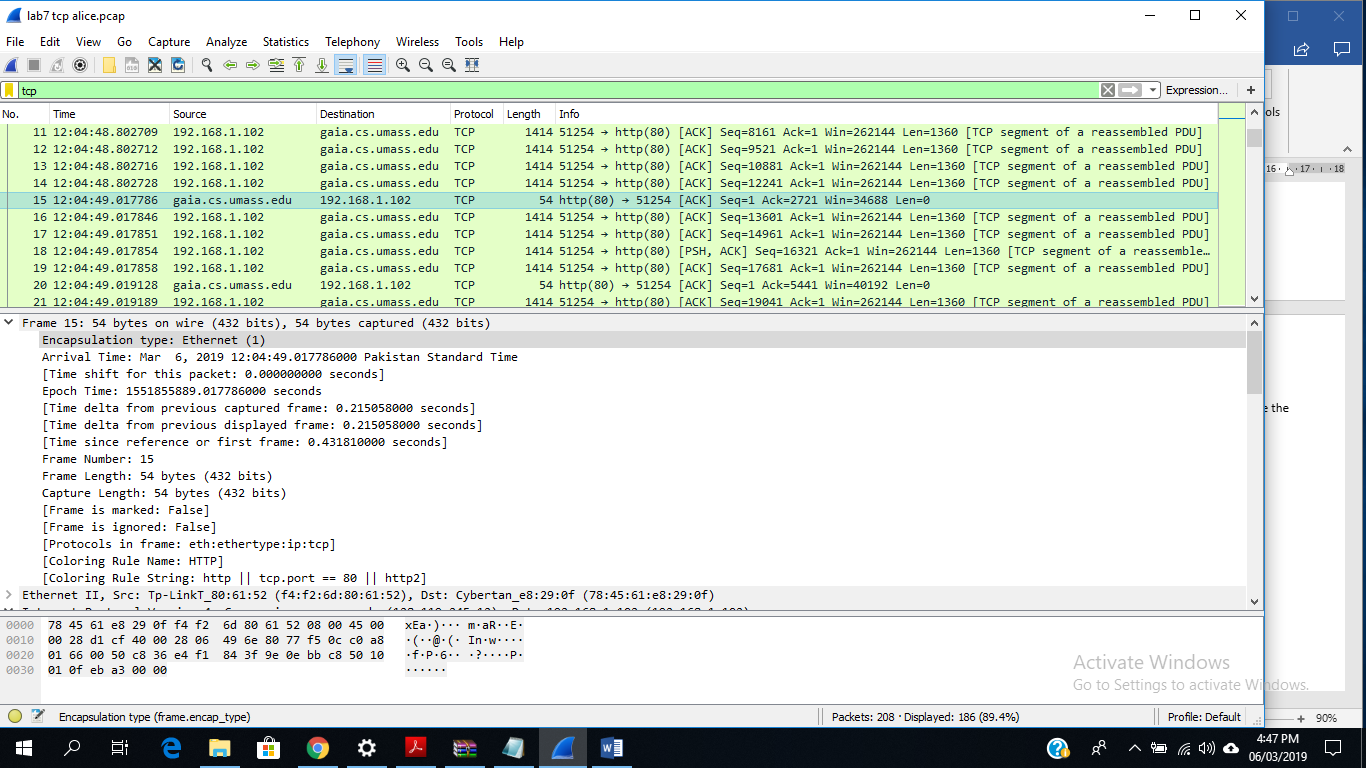
No, there are no transmitted segments in the file trace. In the TimeSequence-Graph (Stevens) of this trace, all sequence numbers from the source to the destination are increasing monotonically with respect to time

Graphical user interface, chart

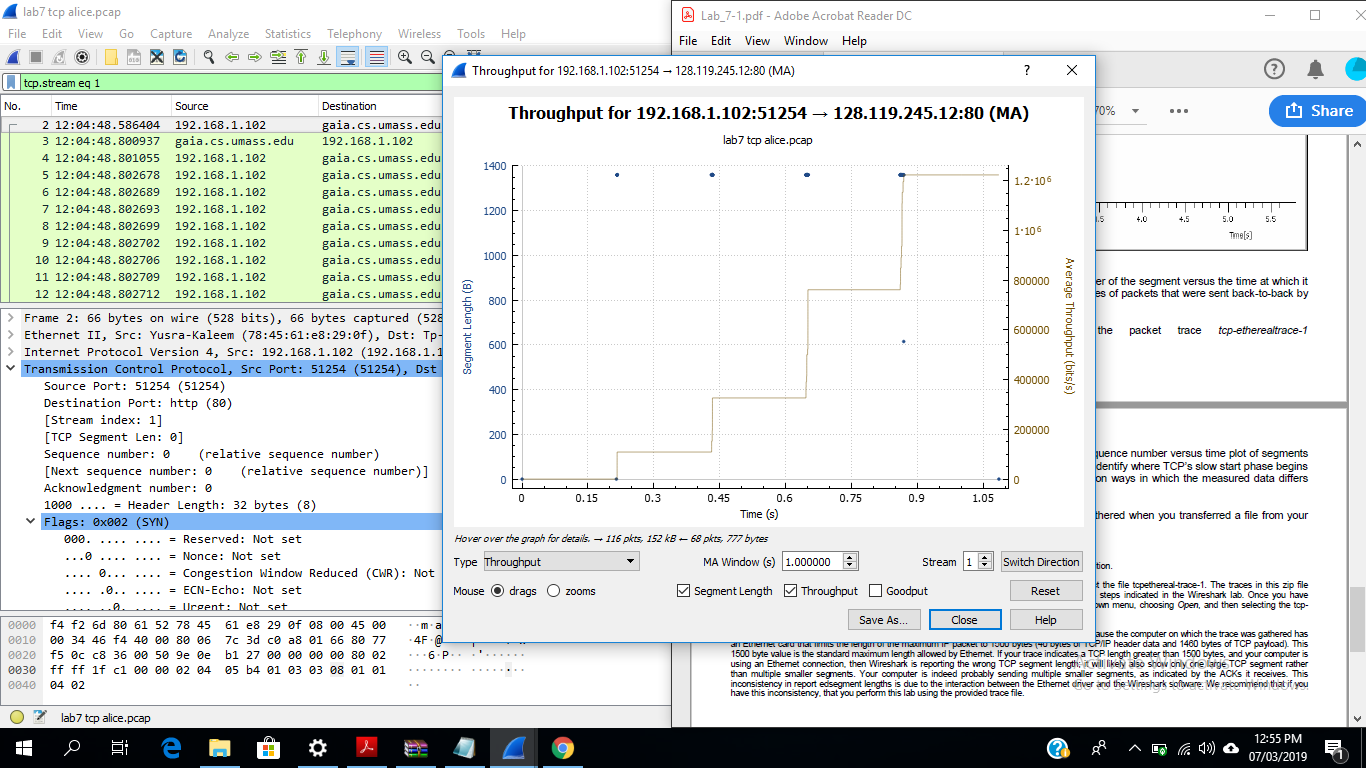
Description automatically generated

12. 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 257 in the text).

The amount of data acknowledged is varied in the trace. Example cases are shown below.



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



***Find out FIN, FIN ACK in a TCP stream.***

