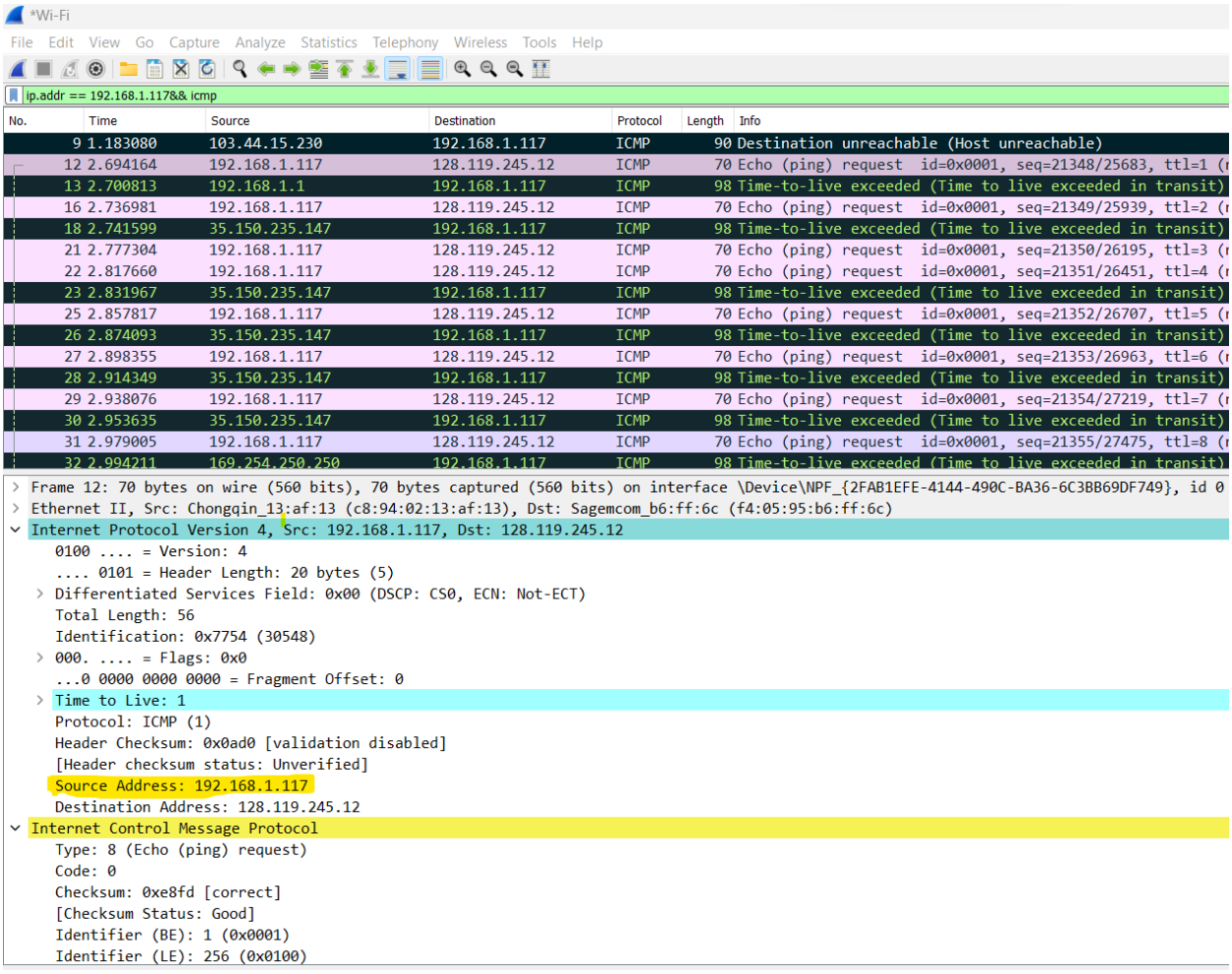
Computer Networks Assignment - 2

Submitted By: N V H Sowndarya
Email ID: sowndaryanoookala.vh@uga.edu
Submission Date: 26th March,23
UGA ID: 811594990

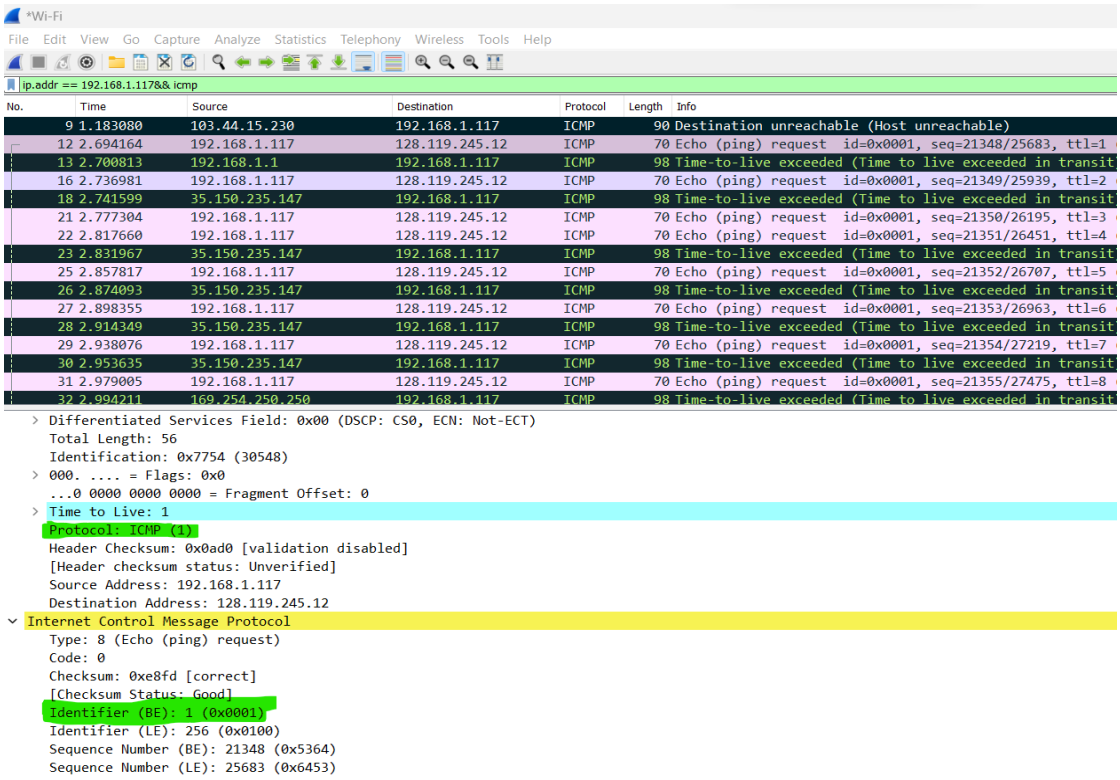
1. **Select the first ICMP Echo Request message sent by your computer and expand the Internet Protocol part of the packet in the packet details window. What is the IP address of your computer?**

   The IP address of my computer is 192.168.1.117



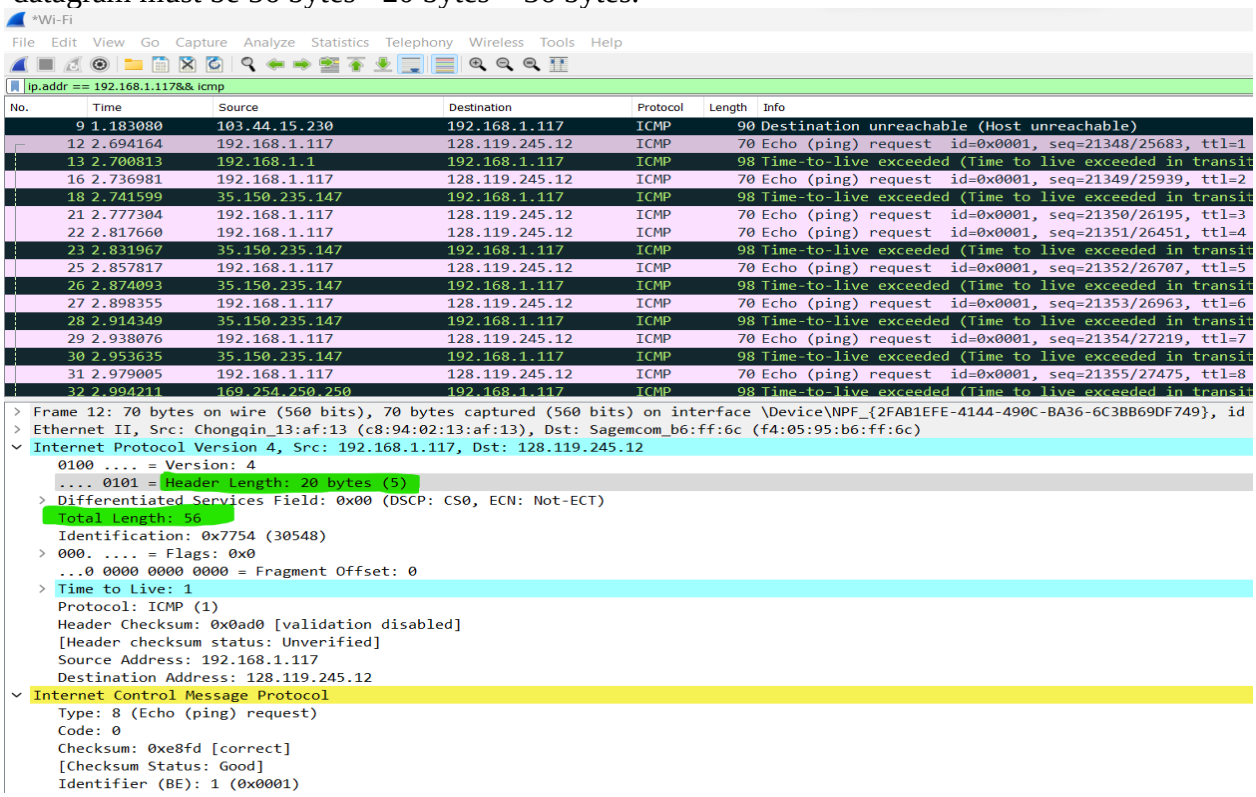
2. **Within the IP packet header, what is the value in the upper layer protocol field?**

   The value in the upper layer protocol field within the header, is ICMP 1(0x0001)



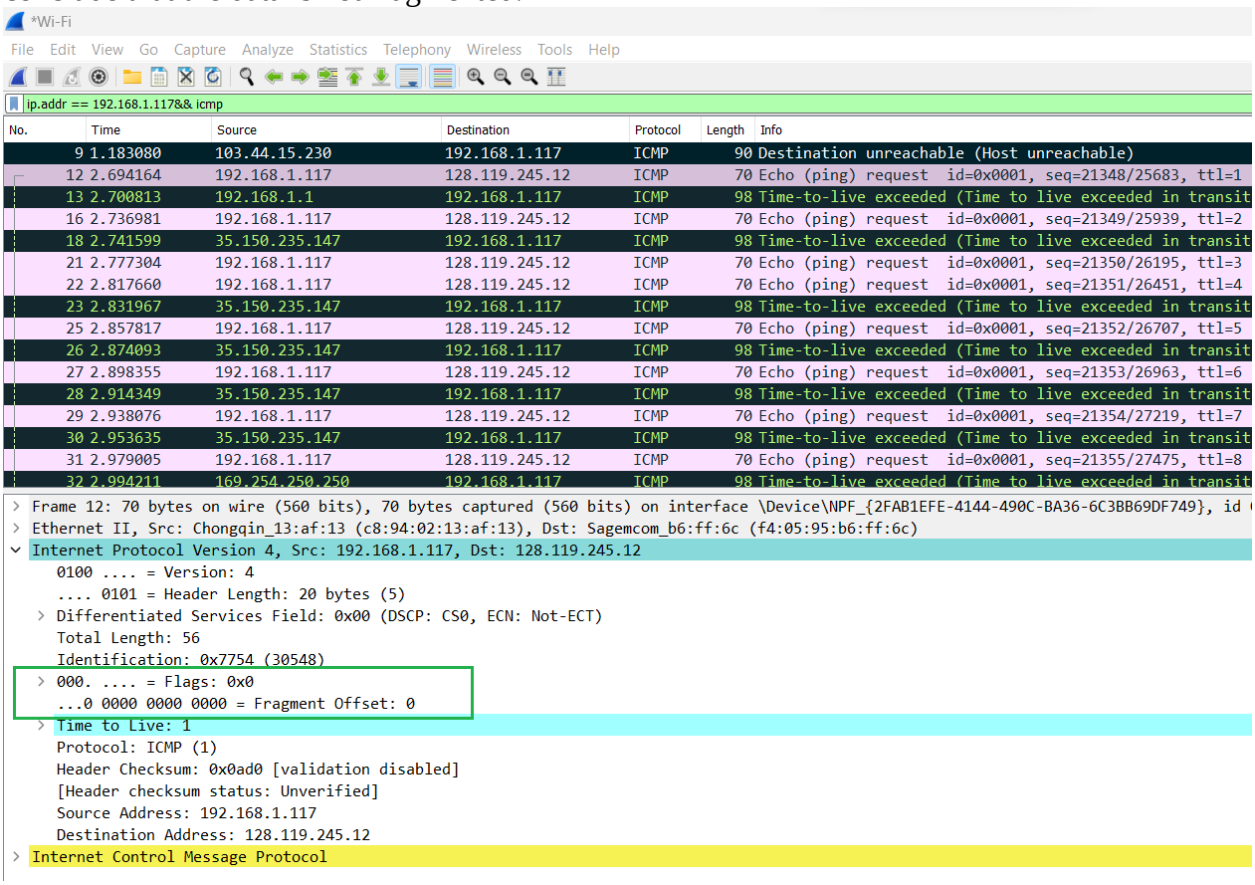
3. How many bytes are in the IP header? How many bytes are in the payload of the IP datagram? Explain how you determined the number of payload bytes.

The header length is 20 bytes, and the total length is 56 bytes. Therefore, the payload of the IP datagram must be 56 bytes - 20 bytes = 36 bytes.



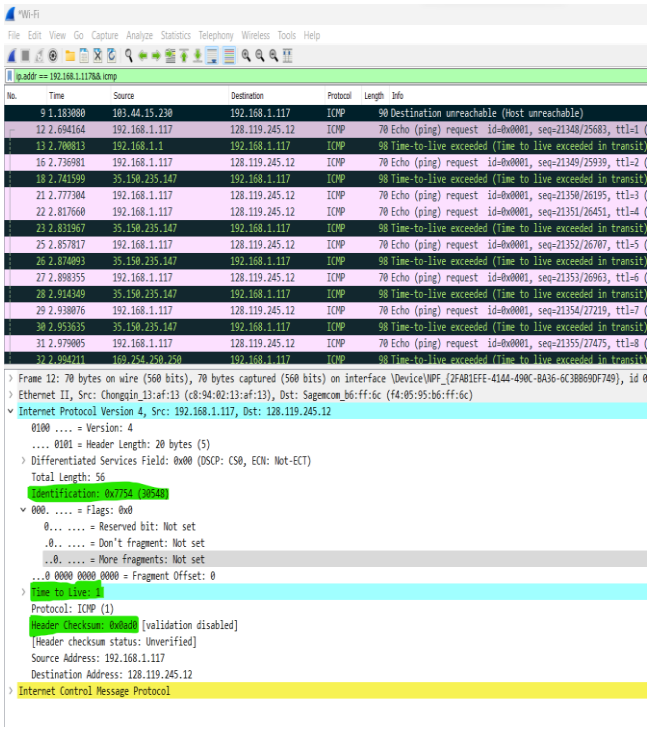
4. **Has this IP datagram been fragmented? Explain how you determined whether or not the datagram has been fragmented.**

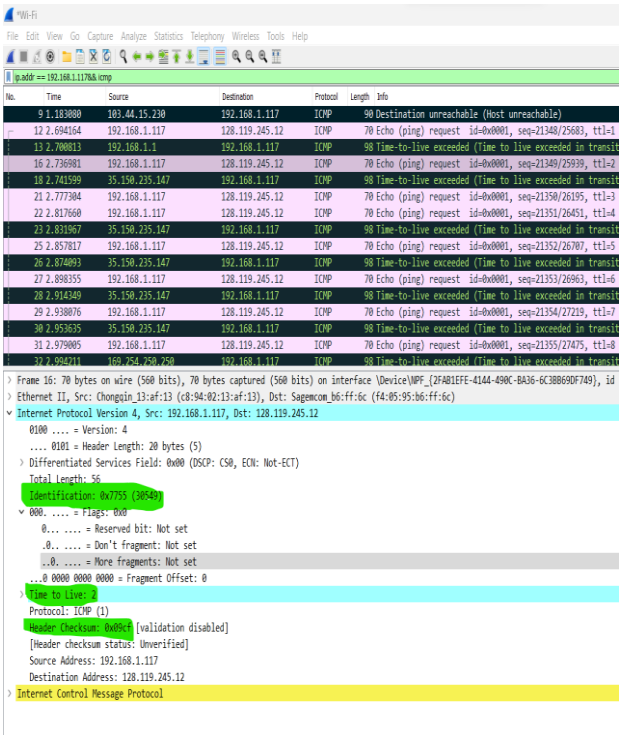
In flags section, the more fragments bit = 0 and fragment offset is also set to 0, so we can conclude that the data is not fragmented.



5. **Which fields in the IP datagram *always* change from one datagram to the next within this series of ICMP messages sent by your computer?**

From below two screenshots we can observe that the identification, time to live and the header checksum change from one datagram to next.





6. **Which fields stay constant? Which of the fields *must* stay constant? Which fields must change? Why?**

The fields that stay constant across the IP datagrams are:

- Version (IP version is 4 for all packets)
- Header Length (All are ICMP)
- Source IP (The requests are sent from same source)
- Destination IP (The requests are being sent to the same destination)
- Differentiated Services (ICMP packets use same Type of Service class)
- Upper Layer Protocol (All are ICMP packets)

The fields that must stay constant are:

- Version (IP version is 4 for all packets)
- Header Length (All are ICMP)
- Source IP (The requests are sent from same source)
- Destination IP (The requests are being sent to the same destination)
- Upper Layer Protocol (All are ICMP packets)
- Differentiated Services (ICMP packets use same Type of Service class)

The fields that must change are:

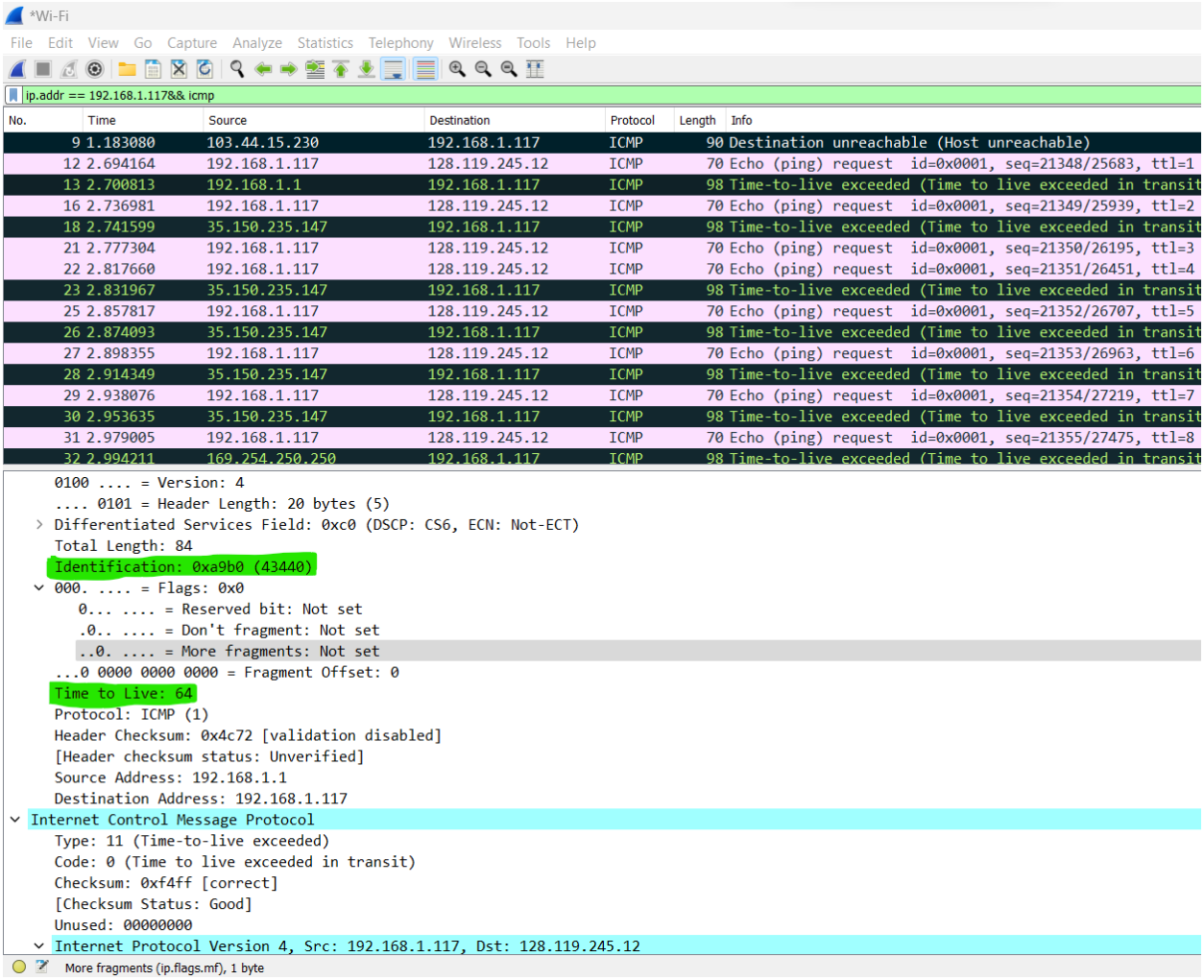
- Identification(IP packets must have different ids)
- Time to live (traceroute increments each subsequent packet)
- Header checksum (since header changes, so must checksum)

7. **Describe the pattern you see in the values in the Identification field of the IP datagram.**

The pattern we can observe from the IP datagram is that the IP header Identification field increments by one from one request to another request.

8. **What is the value in the Identification field and the TTL field?**

From the pattern in the ICMP TTL exceeded replies sent to my computer by the nearest (first hop) router, we can observe that the,
Time to Live: 64
Identification: 0xa9b0 (43440)



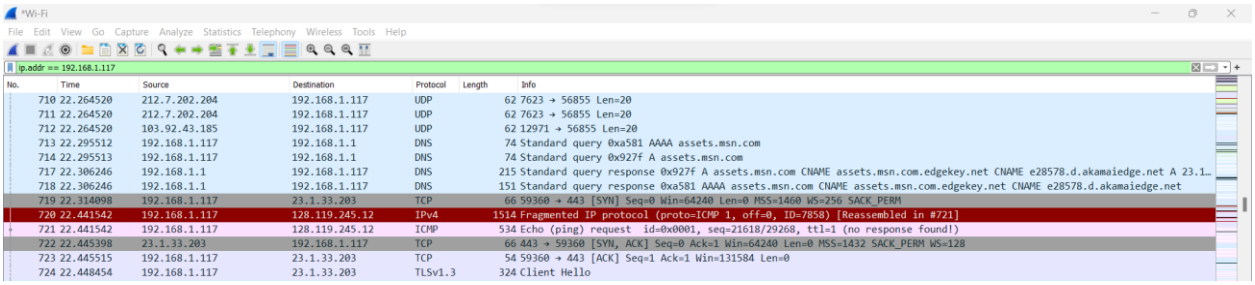
9. Do these values remain unchanged for all of the ICMP TTL-exceeded replies sent to your computer by the nearest (first hop) router? Why?

From the data captured in Wireshark, we see that the identification field values for all the ICMP TTL-exceeded replies are different because a unique value is given to identify it for every response. If two or more IP datagrams have the same identification value, it means that they are fragments of one large IP packet.

The TTL field remains unchanged because all the responses generated by the same hop router are always the same.

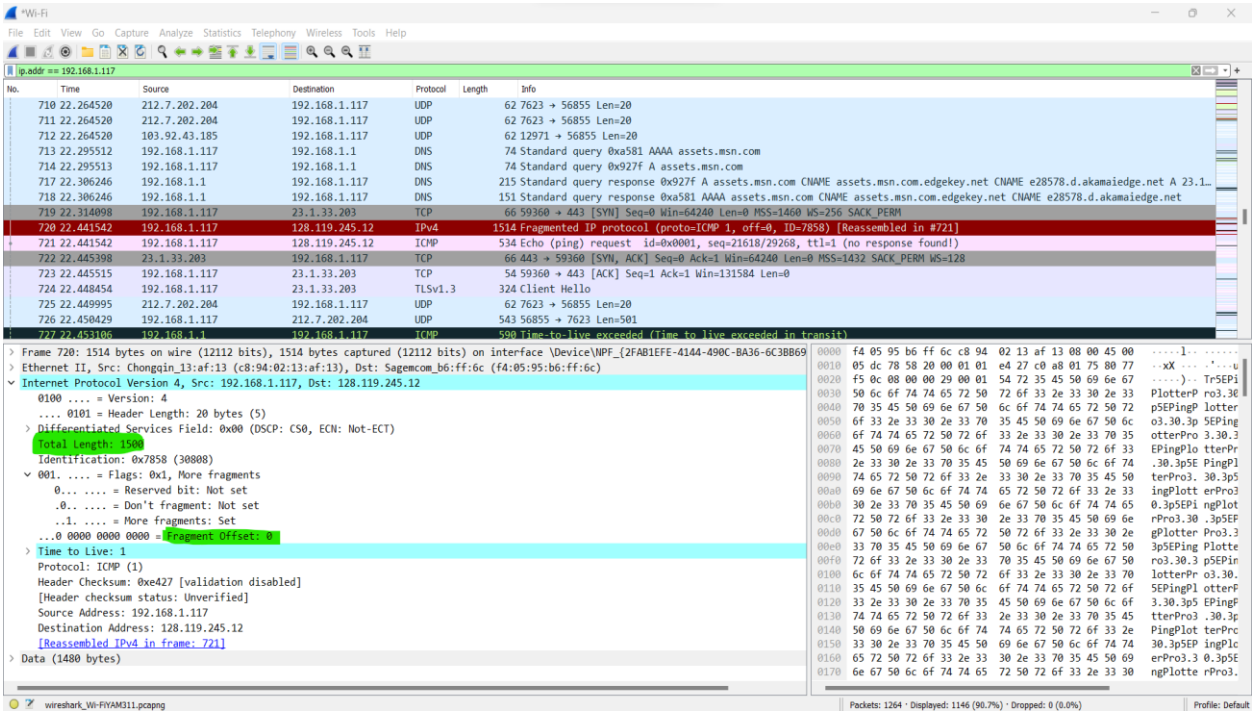
10. Find the first ICMP Echo Request message that was sent by your computer after you changed the *Packet Size* in *pingplotter* to be 2000. Has that message been fragmented across more than one IP datagram?

From the screenshot we can observe that the packet has been fragmented more than once in 720 line.



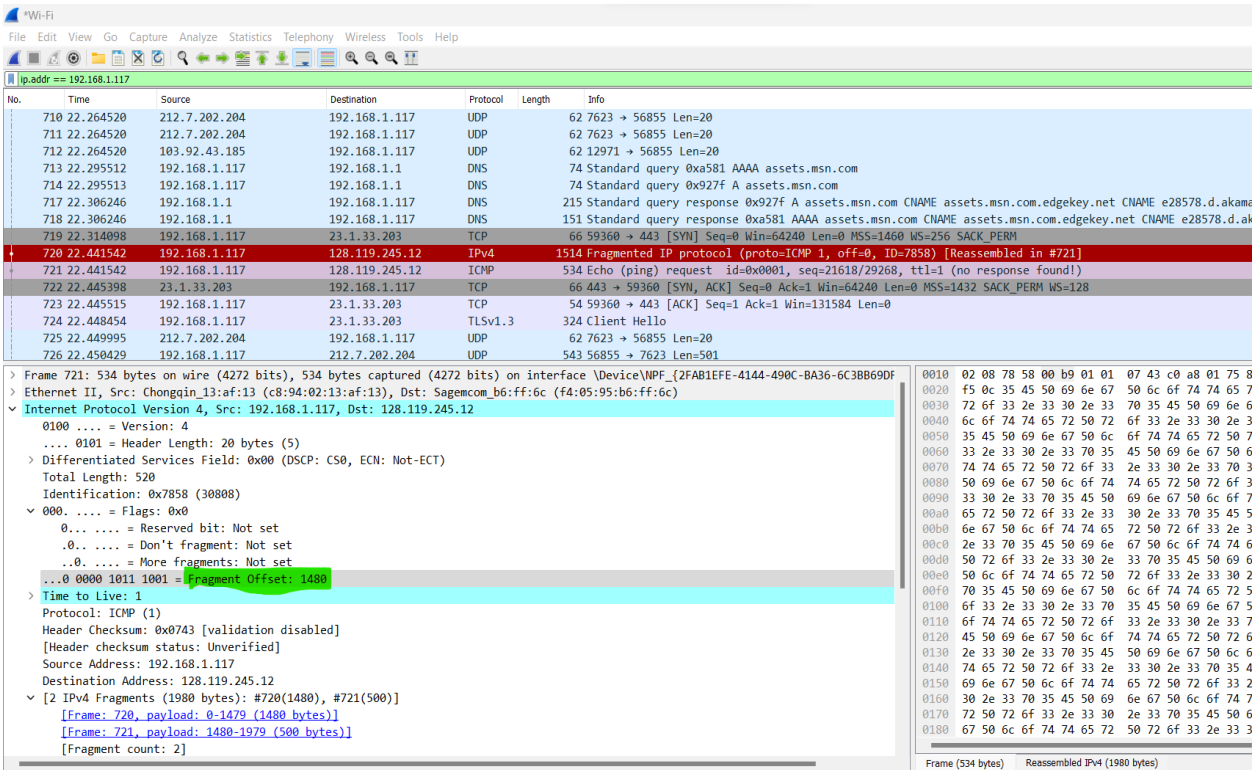
11. Print out the first fragment of the fragmented IP datagram. What information in the IP header indicates that the datagram been fragmented? What information in the IP header indicates whether this is the first fragment versus a latter fragment? How long is this IP datagram?

In the below screenshot, we can observe that more fragments field has the value “set” under the flags section. The fragment offset is set to ‘0’ which indicates that it is the first segment. The first datagram has total length 1500.



12. Print out the second fragment of the fragmented IP datagram. What information in the IP header indicates that this is not the first datagram fragment? Are the more fragments? How can you tell?

Below is the screenshot of the second fragment of the datagram. The fragment offset has the value 1480 unlike the first fragment which has value 0 indicating that it is the second fragment. There are not more fragments because under the flag field the more segments has “Not set” value.

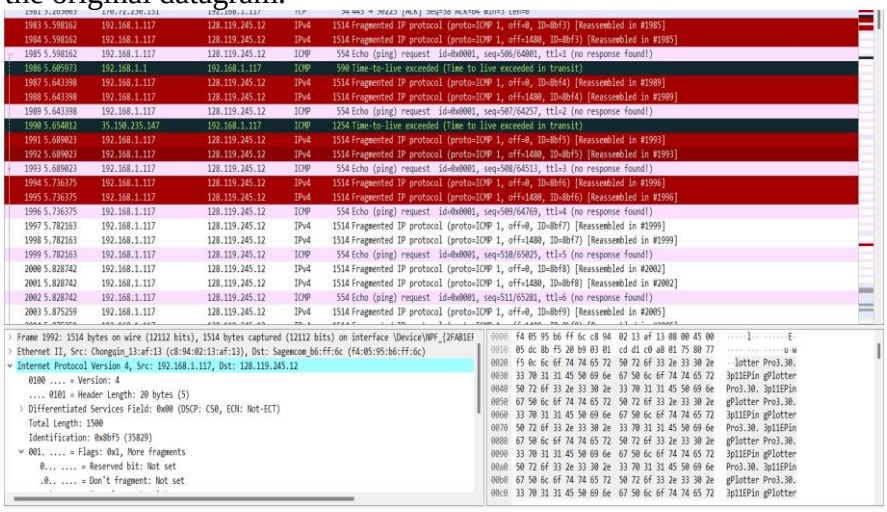


13. What fields change in the IP header between the first and second fragment?

The fields in the IP header that has changed between the first and second fragments are the total length, flags, checksum, and the fragment offset.

14. How many fragments were created from the original datagram?

When the number of bytes is changed from 2000 to 3500 bytes, three fragments are created from the original datagram.



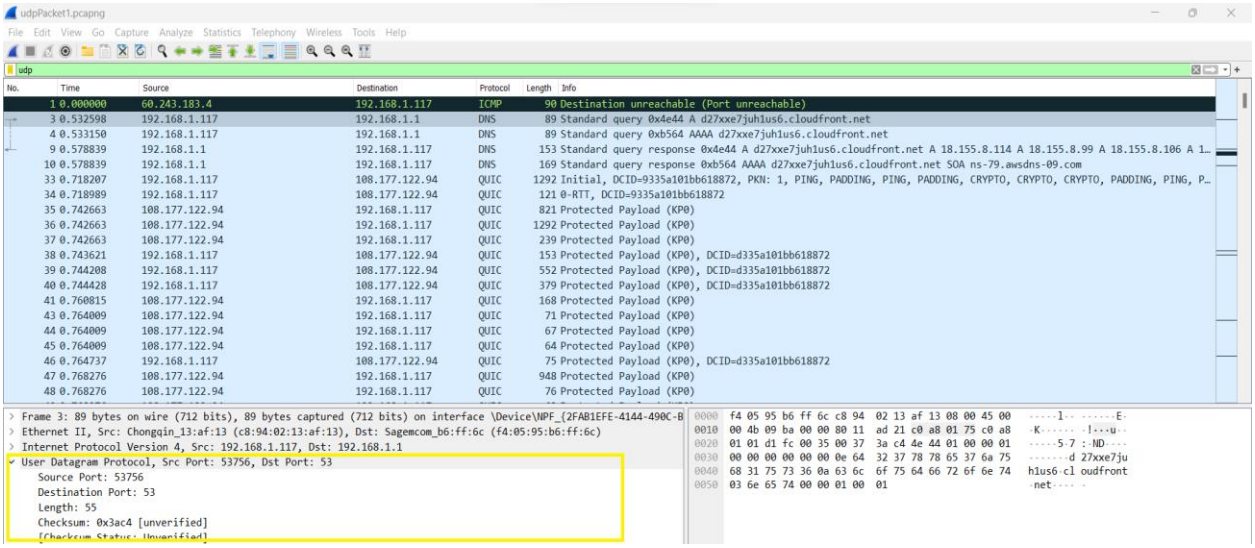
15. What fields change in the IP header among the fragments?

There are two fields that change: checksums and fragment offsets (0, 1480, 2960 for first, second, and third fragments, respectively). The first and second packets are 1500 bytes in length, and their

more fragments flags are both set to "Set", but the third fragment is 540 bytes in length, and its flag value is "Not Set", indicating that it is the last fragment.

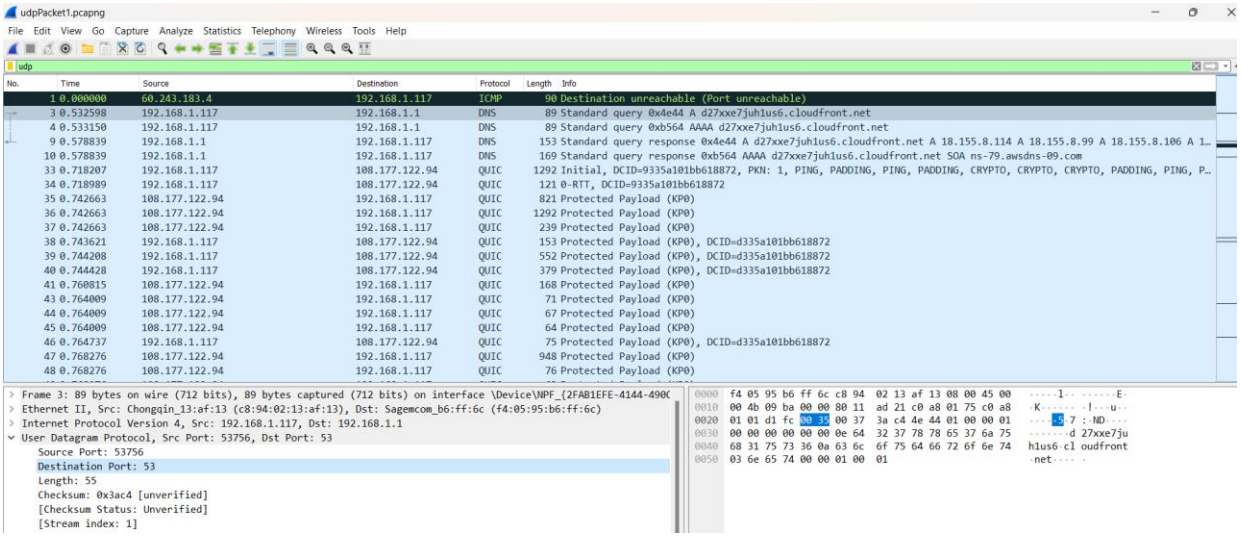
1. **Select *one* UDP packet from your trace. From this packet, determine how many fields there are in the UDP header. Name these fields.**

   The UDP packet has the four fields Source Port, Destination port, Length and checksum values in the header .



2. **By consulting the displayed information in Wireshark's packet content field for this packet, determine the length (in bytes) of each of the UDP header fields.**

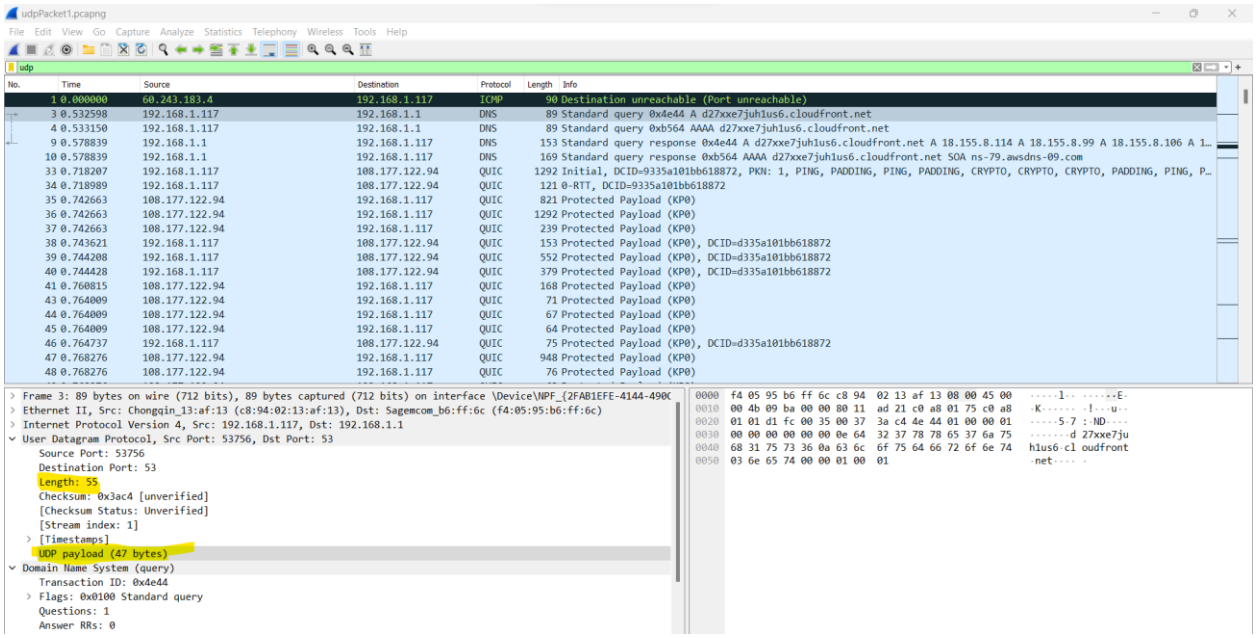
   The total UDP header is of 8 bytes where each field has length of 2 bytes.



3. **The value in the Length field is the length of what? (You can consult the text for this answer). Verify your claim with your captured UDP packet.**

   The length field in the UDP segment specifies the number of bytes required for the header and also the data. The length value has to be specified because the size of the data field might be

different from one UDP segment to another. The length of UDP payload for selected packet is 55 bytes. 55 bytes - 8 bytes = 47 bytes.



4. **What is the maximum number of bytes that can be included in a UDP payload?(Hint: the answer to this question can be determined by your answer to 2. above)**

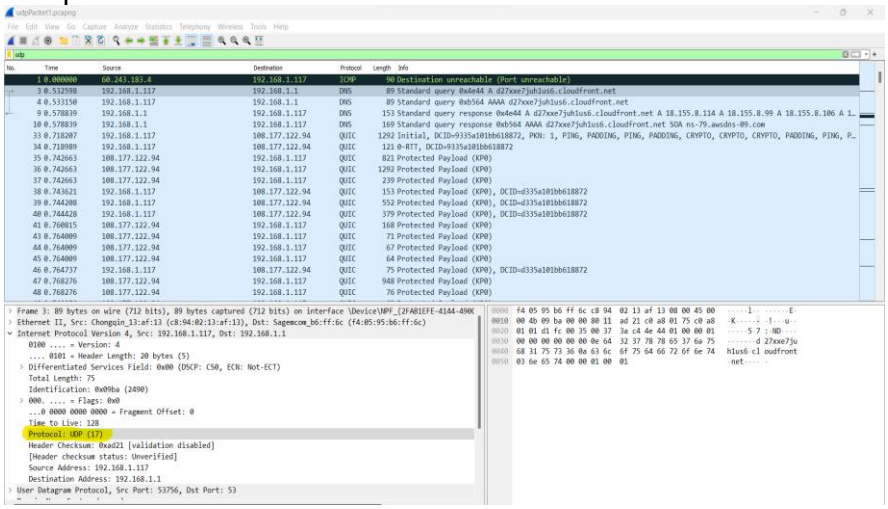
   The maximum number of bytes that can be included in the payload is 2^{16}- the bytes used by the header field which is 8 bytes. So, the maximum payload is 65535-8= 65527 bytes

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

   The possible largest source port number can be 2^{16} = 65535.

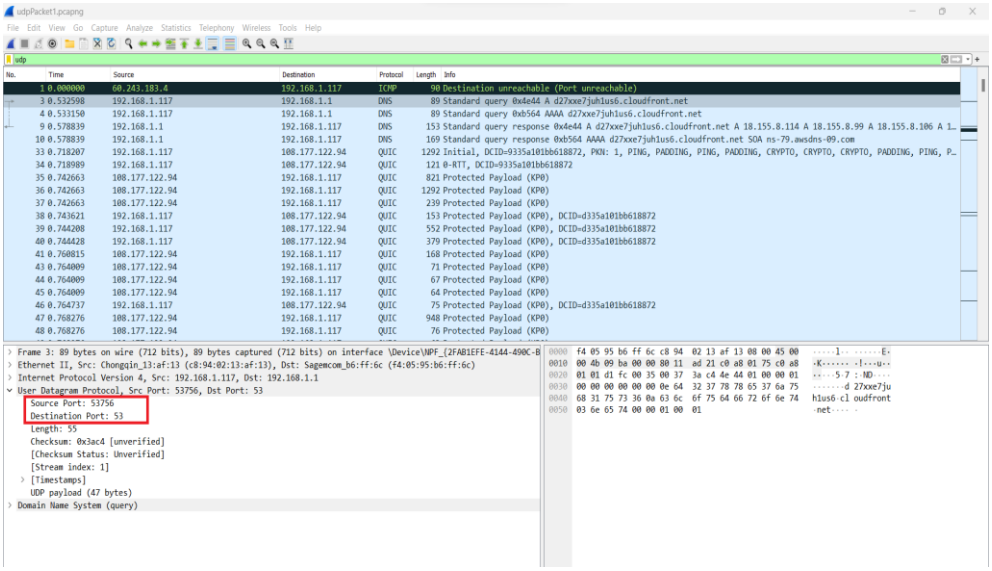
6. **What is the protocol number for UDP? Give your answer in both hexadecimal and decimal notation. To answer this question, you'll need to look into the Protocol field of the IP datagram containing this UDP segment (see Figure 4.13 in the text, and the discussion of IP header fields).**

   The protocol number for UDP is 17 in decimal notation and is 0x11 in hexadecimal.

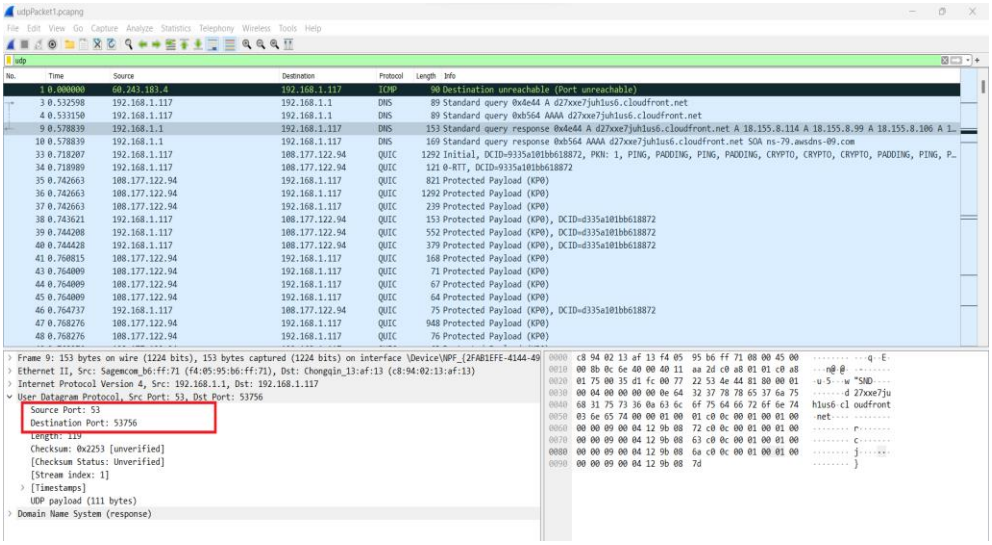


7. **Examine a pair of UDP packets in which your host sends the first UDP packet and the second UDP packet is a reply to this first UDP packet. Describe the relationship between the port numbers in the two packets.**

The port numbers for sending request are:
Source Port Number: 53756
Destination Port Number: 53



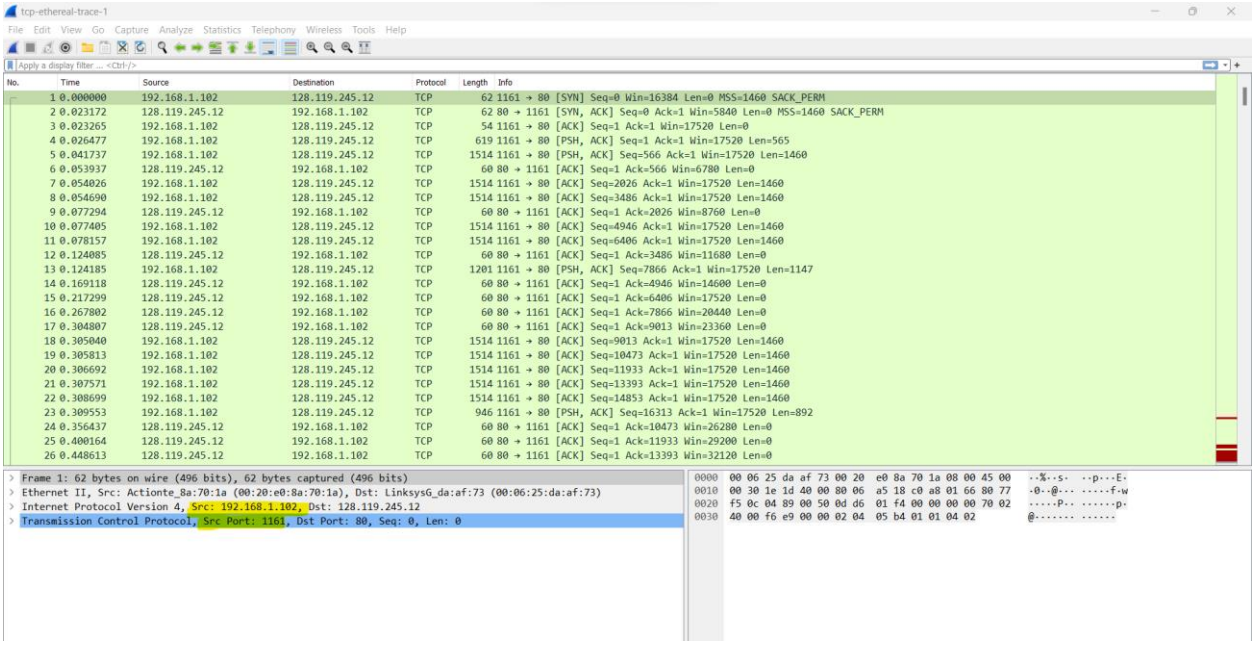
The port numbers for response received by host are:
Source Port Number: 53
Destination Port Number: 53756



The relationship between the port numbers is that the source port number for request is same as the destination port number of response and also the destination port number of the request is same as the source port number of the response which indicates that the messages are being transmitted between them.

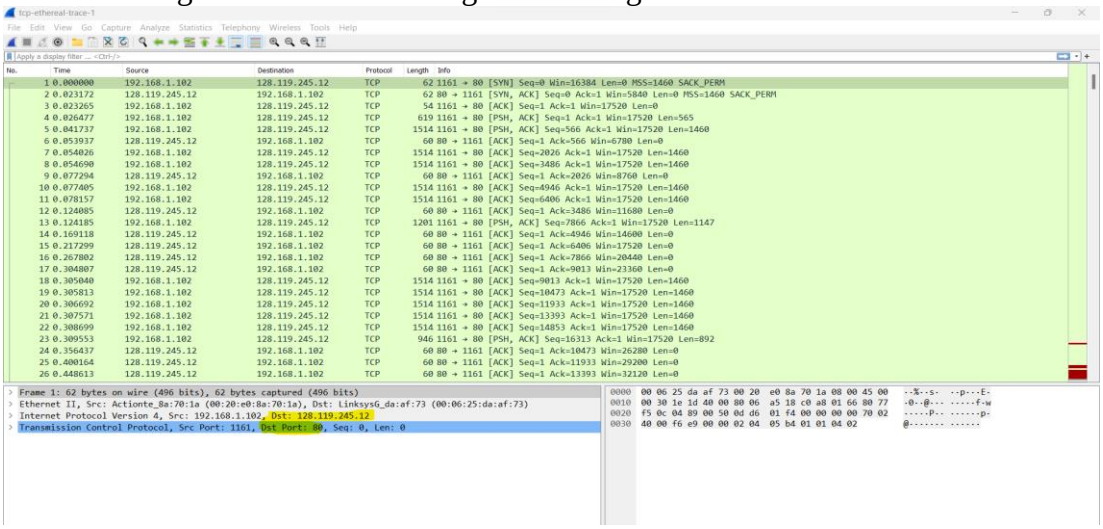
1. **What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu? To answer this question, it's probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the "details 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.**

   The IP address of the source is 192.168.1.102 and the Source Port through which the file is being transferred is 1161.



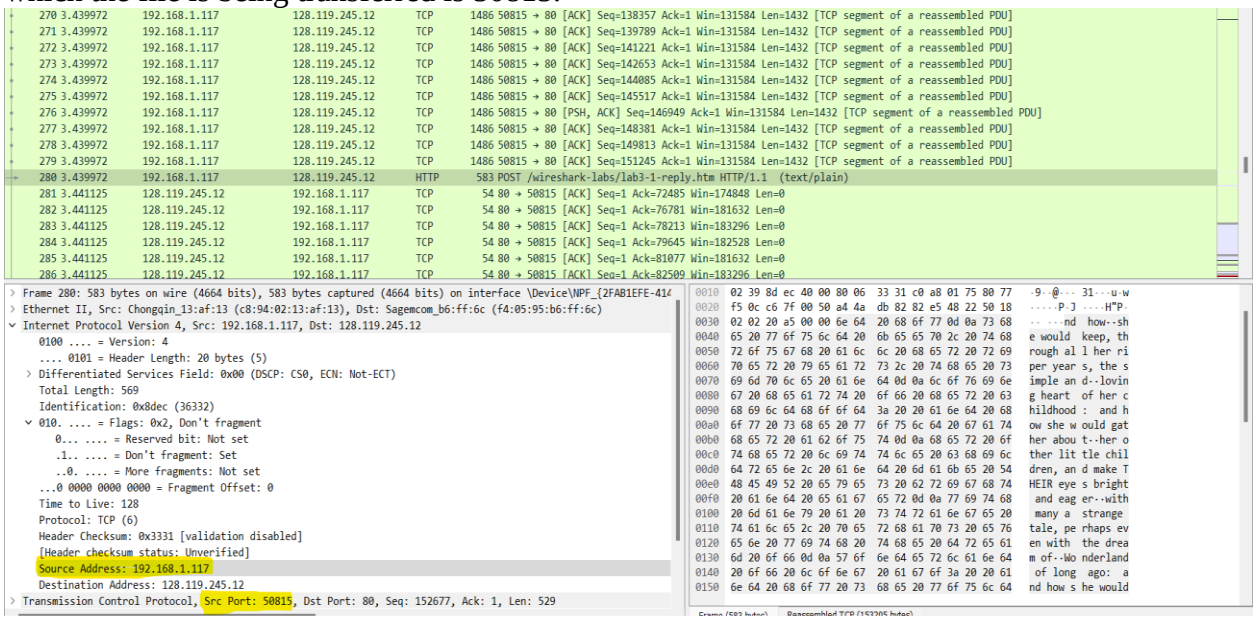
2. **What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?**

   The IP address of the destination computer i.e., gaia.cs.umass.edu is 128.119.245.12 TCP port number through which it is receiving the TCP segments is 80.



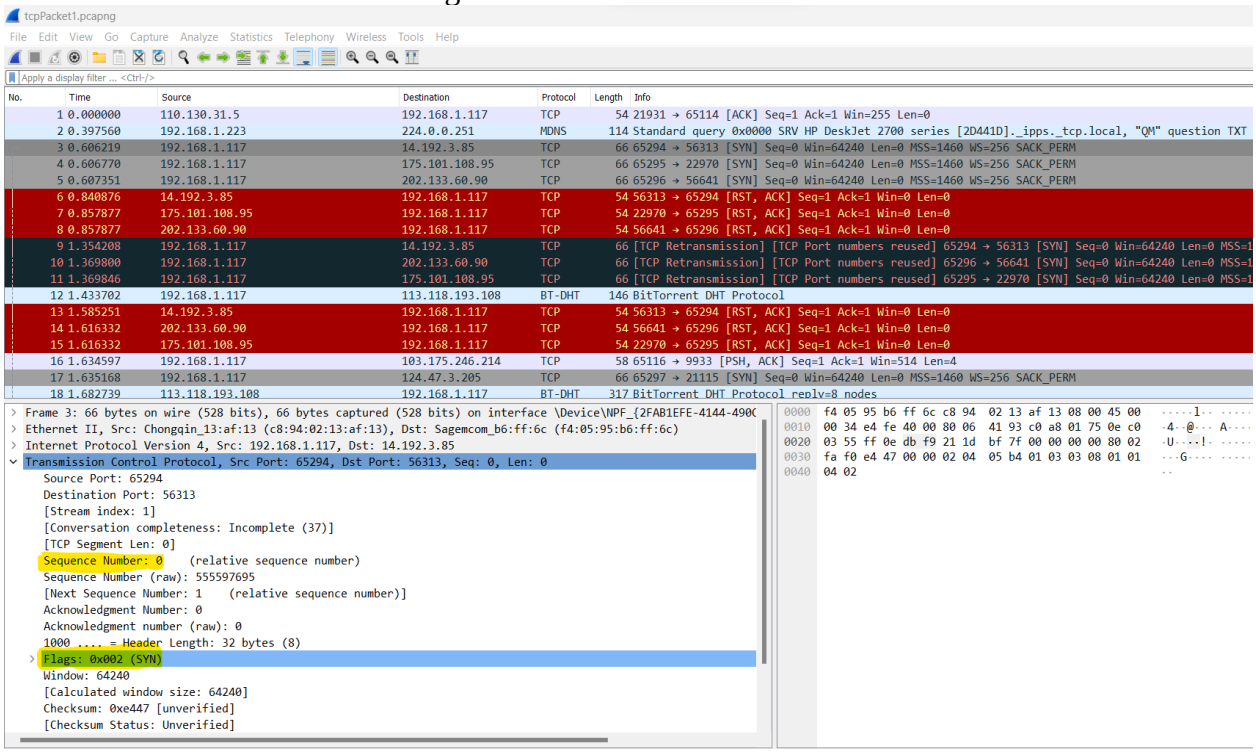
3. **What is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu?**

The IP address of the source i.e., my computer is 192.168.1.117 and the Source Port through which the file is being transferred is 50815.



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 of TCP SYN segment is “0” and the notation “SYN” near the Flags section indicate that it is a SYN segment.

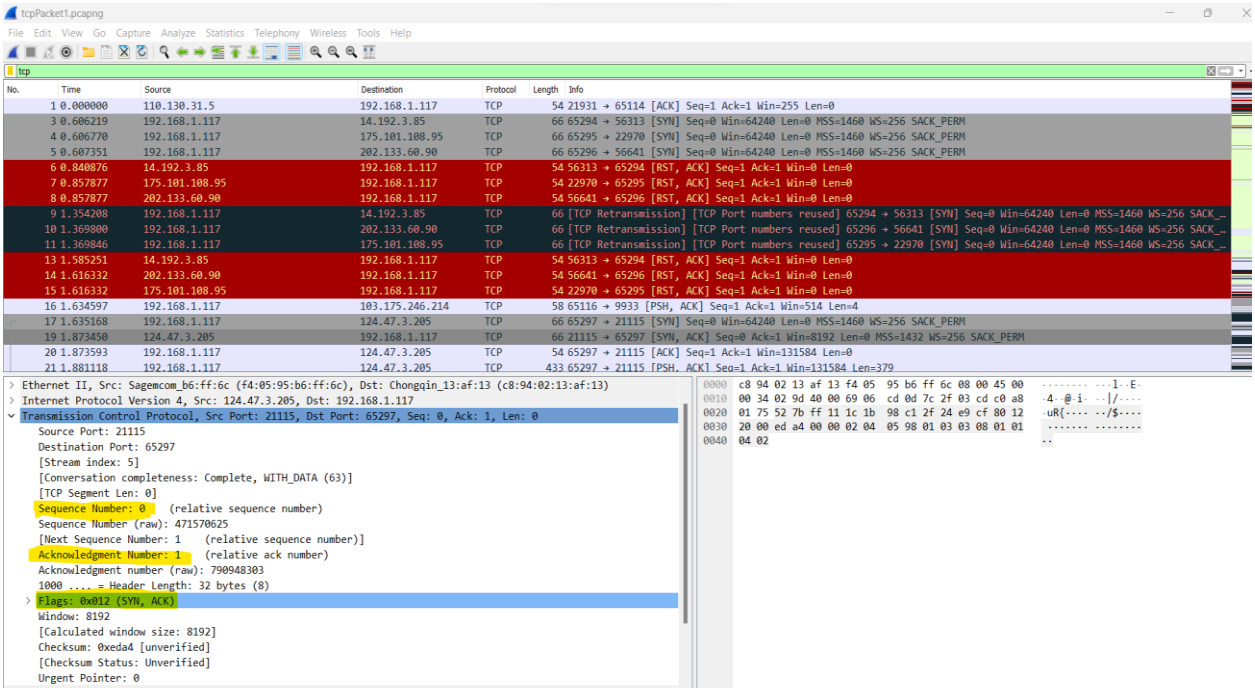


5. **What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did gaia.cs.umass.edu determine that value? What is it in the segment that identifies the segment as a SYNACK segment?**

   The sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN is "0".

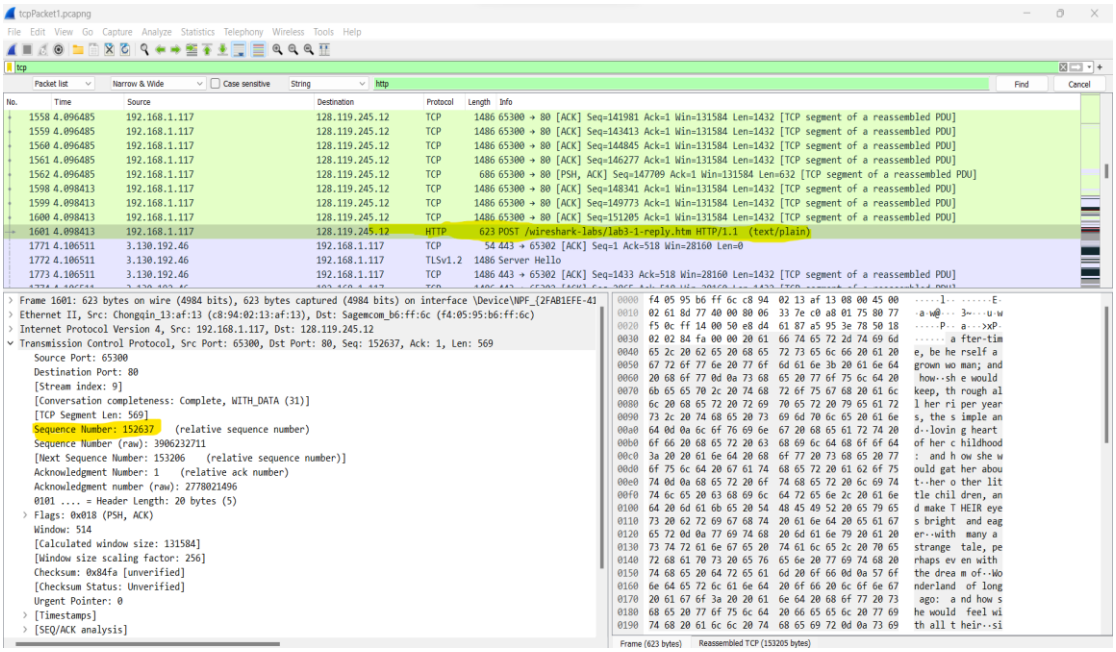
   The value of the Acknowledgement field in the SYNACK segment is "1" and this value is determined by adding +1 to the initial Seq number.

   The Flags field represented with (SYN,ACK) in the segment identifies that the segment as a SYNACK segment.



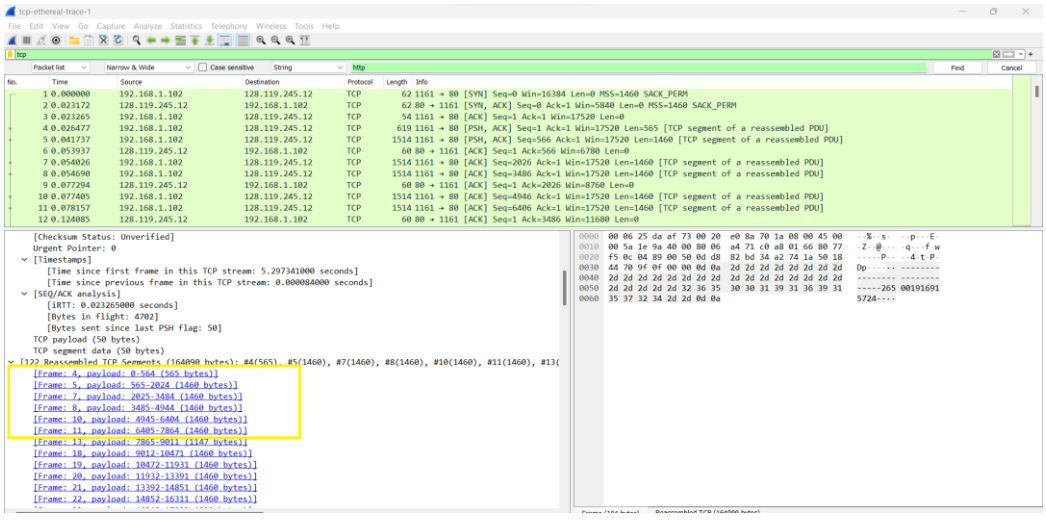
6. **What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you'll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a "POST" within its DATA field.**

   The sequence number of the TCP segment containing the HTTP POST command is 152637.



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 242 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 242 for all subsequent segments.

The first six segments in the TCP connection containing the HTTP POST are at the Frames 4,5,7,8,10,11 with sequence numbers Seq=1, Seq= 566, Seq= 2026, Seq= 3486, Seq= 4946, Seq= 6406 respectively.



The sending time and the received time of ACKs are tabulated in the following table using the flow graph generated using wireshark tool.

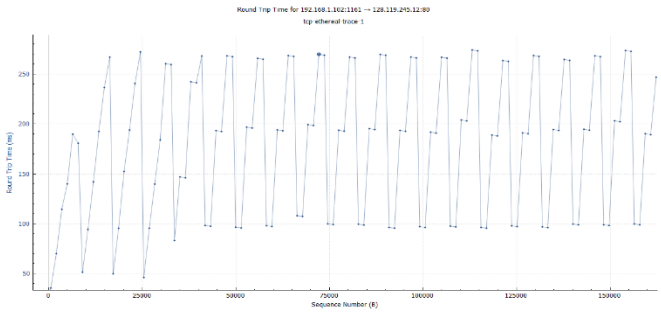
Frame Number	Sent Time	ACK received time	RTT Time
Frame 4	0.026477	0.053937	0.02746
Frame 5	0.041737	0.077294	0.035557
Frame 7	0.054026	0.124085	0.070059
Frame 8	0.054690	0.169118	0.11443
Frame 10	0.077405	0.217299	0.13989
Frame 11	0.078157	0.267802	0.18964

The EstimatedRTT can be calculated by using the following formula,

EstimatedRTT = 0.875 * EstimatedRTT + 0.125 * SampleRTT

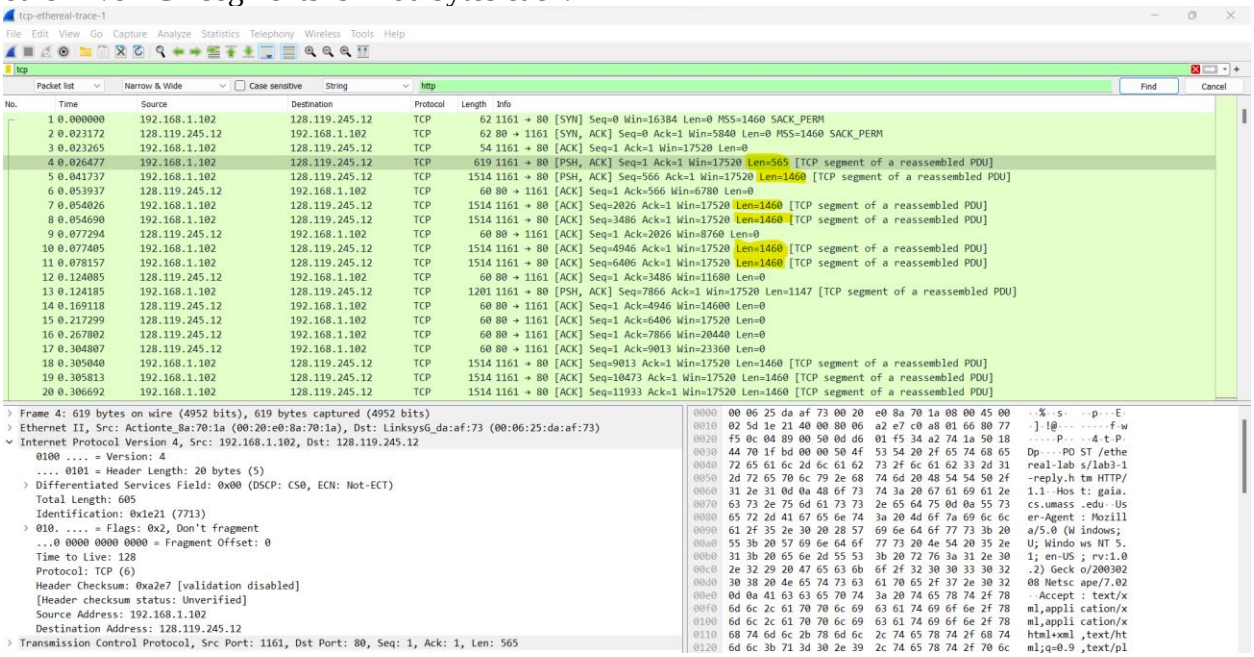
- EstimatedRTT after the receipt of the ACK of Frame 4 = RTT for Frame 4 = 0.02746 seconds
- EstimatedRTT after the receipt of the ACK of Frame 5 = 0.875 * 0.02746 + 0.125 * 0.035557 = 0.0285 seconds
- EstimatedRTT after the receipt of the ACK of Frame 7 = 0.875 * 0.0285 + 0.125 * 0.070059 = 0.0337 seconds
- EstimatedRTT after the receipt of the ACK of Frame 8 = 0.875 * 0.0337+ 0.125 * 0.11443 = 0.0438 seconds
- EstimatedRTT after the receipt of the ACK of Frame 10 = 0.875 * 0.0438 + 0.125 * 0.13989 = 0.0558 seconds
- EstimatedRTT after the receipt of the ACK of Frame 11 = 0.875 * 0.0558 + 0.125 * 0.18964 = 0.0725 seconds

The RTTs graph can be obtained as follows



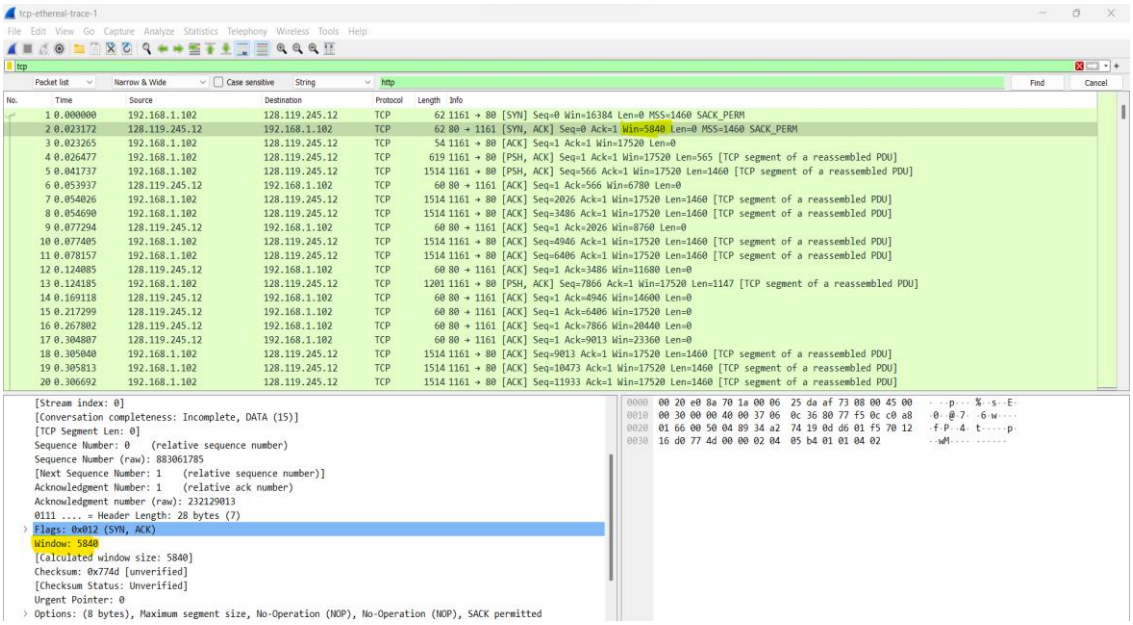
8. What is the length of each of the first six TCP segments?

The length of the first TCP segment that contains the HTTP POST is 565 bytes and the length of other five TCP segments is 1460 bytes each.



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 minimum amount of receiver window buffer advertised for gaia.cs.umass.edu for the entire trace is 5840 bytes, which can be obtained from the first acknowledgement received from the server. The receiver window grows steadily until a maximum receiver buffer size of 62780 bytes. This trace shows that the sender is never throttled due to a lack of receiver buffer space.

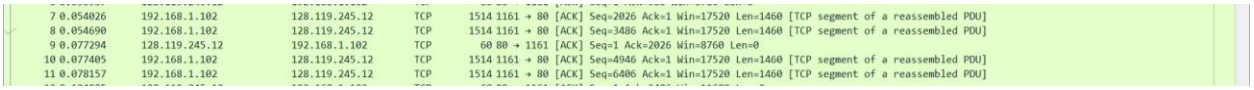


10. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?

By checking the sequence numbers of the TCP segments in the trace file, we can verify that there are no retransmitted segments in the trace file. Retransmitted segments should have smaller sequence numbers than their neighboring segments if there is a retransmitted segment. Furthermore, the old Acknowledgement number for former packets was never sent again.

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.

The difference between the acknowledged sequence numbers of two consecutive ACKs indica tes that the amount of data is received by the server between these two ACKs. The receiver is ACKing every other segment. In the below screenshot, we observe that the cumulative ACKs (No. 7 and 8) has the difference of 1460(3486-2026) which is equal to length of the segments which means the receiver is ACKing every other received segment.



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

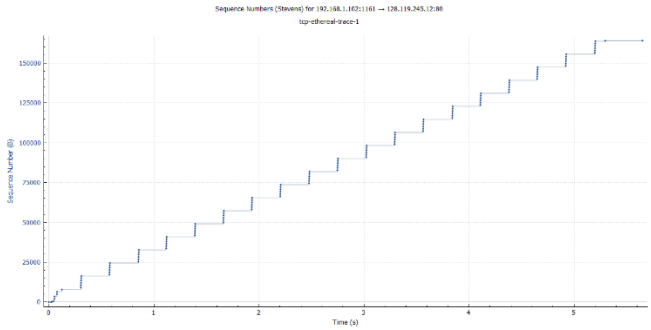
By subtracting the value of first sequence number from the value of the last acknowledgment number and this value is divided by the time since first frame to obtain the throughput. In our case, the last acknowledgment number is 164091 while the first seq number is 1. The time since first frame is 5.651141.

Throughput = (164091-1) / 5.651141 = 29,036.614

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.

The TCP slowstart phase begins at just above seq number 3600 and ends just after sequence number 7800. Congestion avoidance takes over at 150000.



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**

For the trace gathered on my computer when the file is transferred, the slow start of TCP begins at 0.2 seconds and then ends at about 0.3 seconds. Congestion avoidance starts at about 0.72 seconds because it cut down the amount being sent.