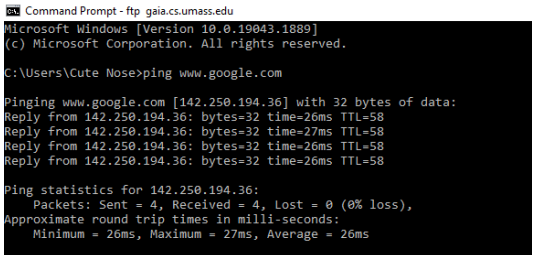
CNS Lab on WireShark

Date: 28/08/2022
Id: 2022PIS5083
Name: SUMITRA SHARMA
Branch: M.Tech (Computer Science and Information Security)
Batch: 2022-2024

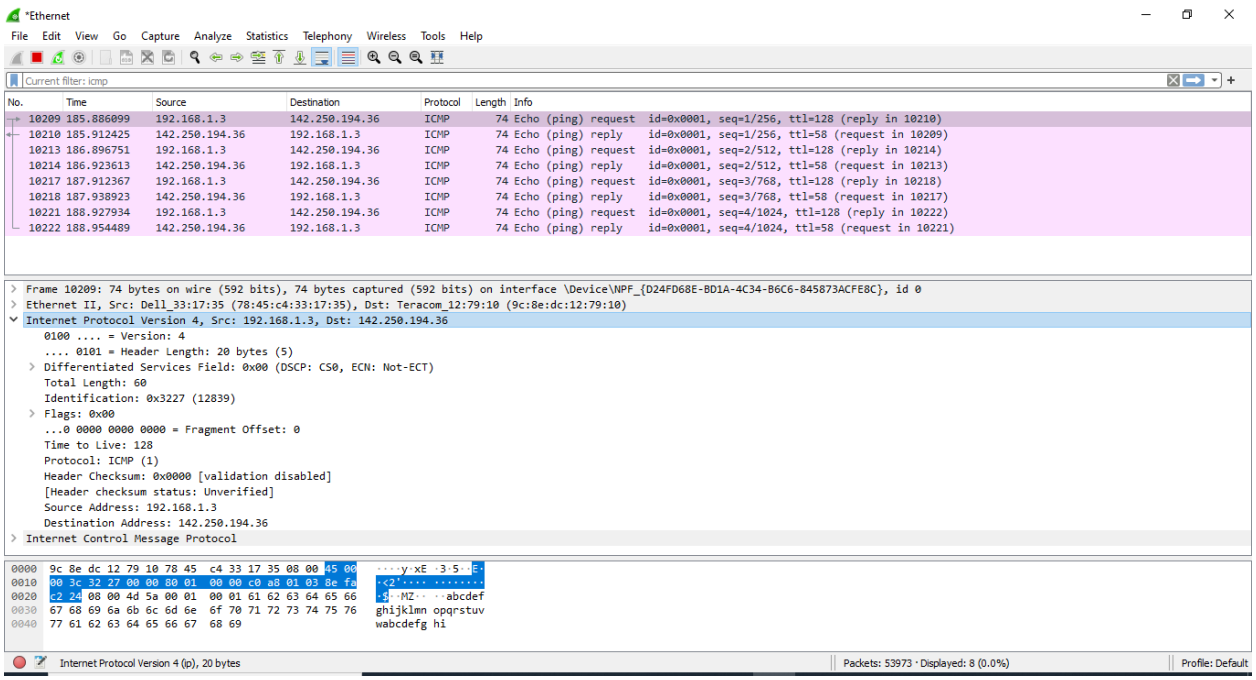
Networks Assignment:

ICMP protocol analysis prerequisites:-



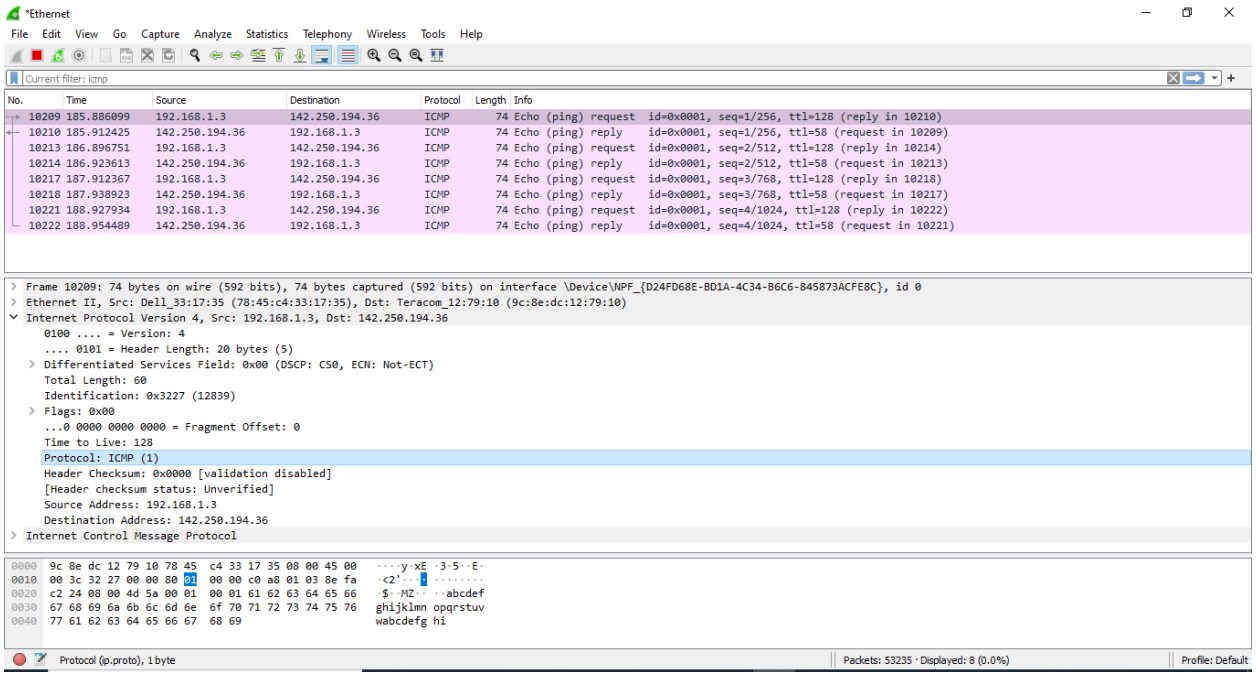
(a) Select the first ICMP Echo Request message sent by your computer, and expand the Internet Protocol part of the packet in the packet details window. What is the IP address of your Computer?

Solution: 192.168.1.3



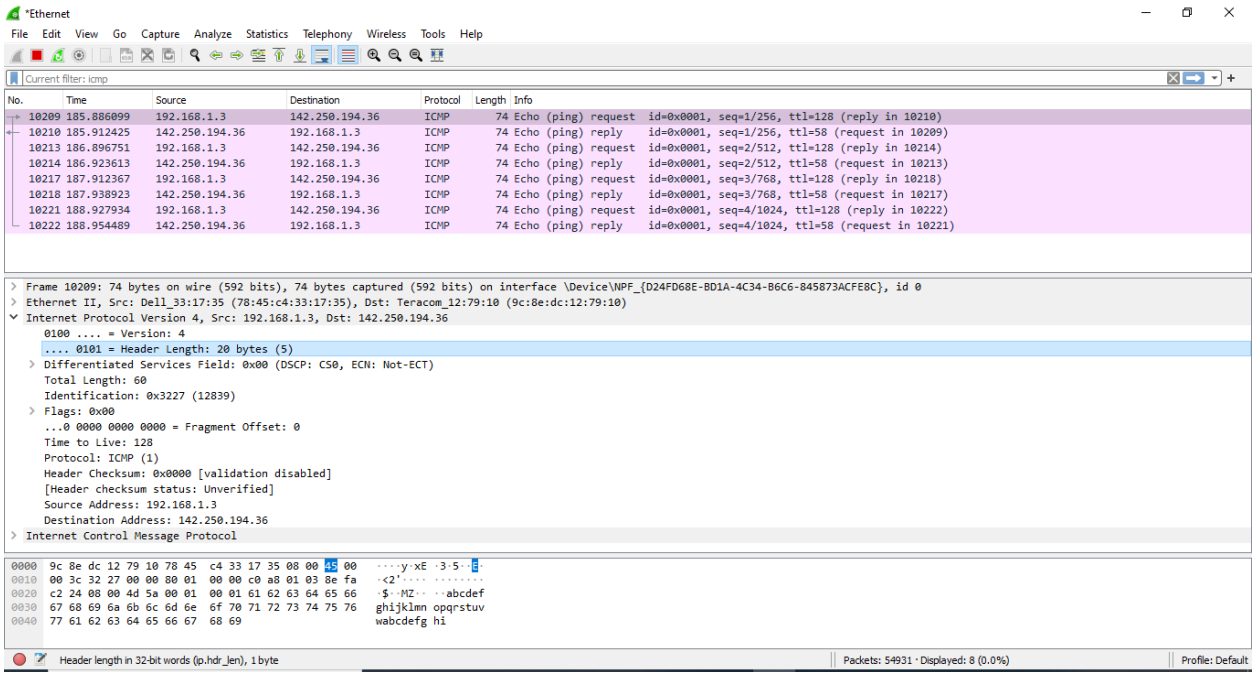
(b) Within the IP packet header, what is the value in the upper layer protocol field?

Solution: ICMP (1)



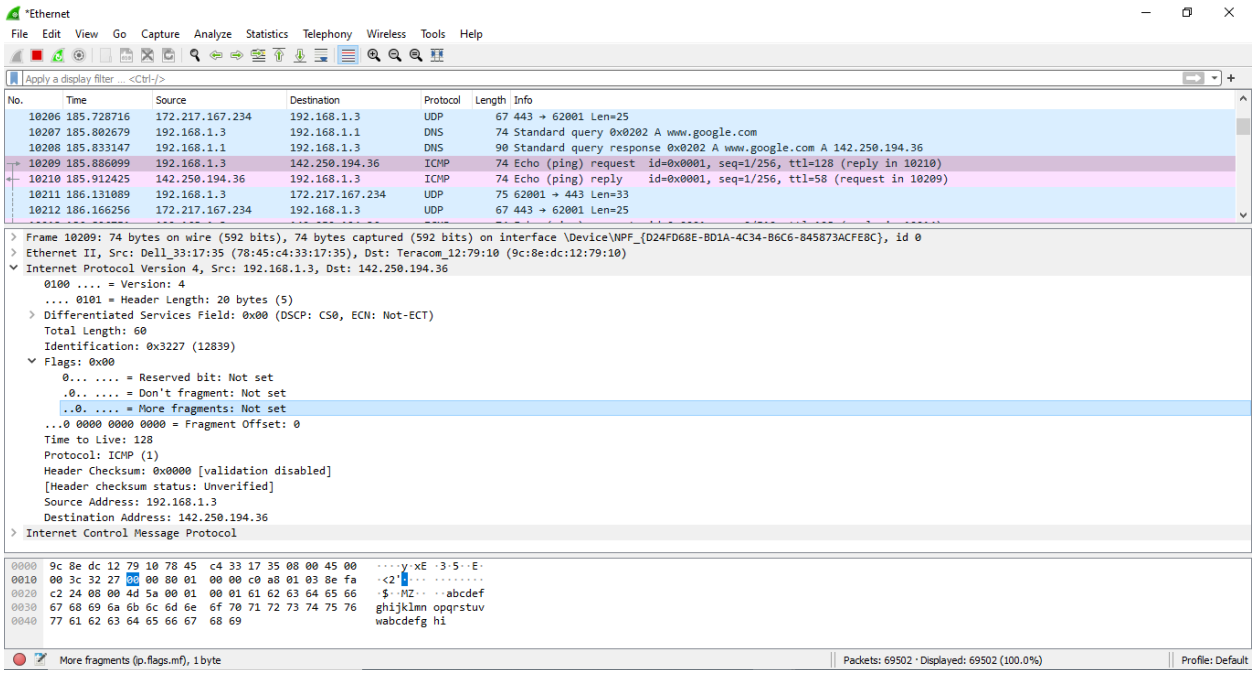
(c) How many bytes are in the IP header? How many bytes are in the payload of the IP datagram? Explain how you determined the number of payload bytes.

Solution: IP header length = 20 bytes (as seen in screenshot)
Payload bytes: 40 (total length 60 - the 20 header bytes = 40)



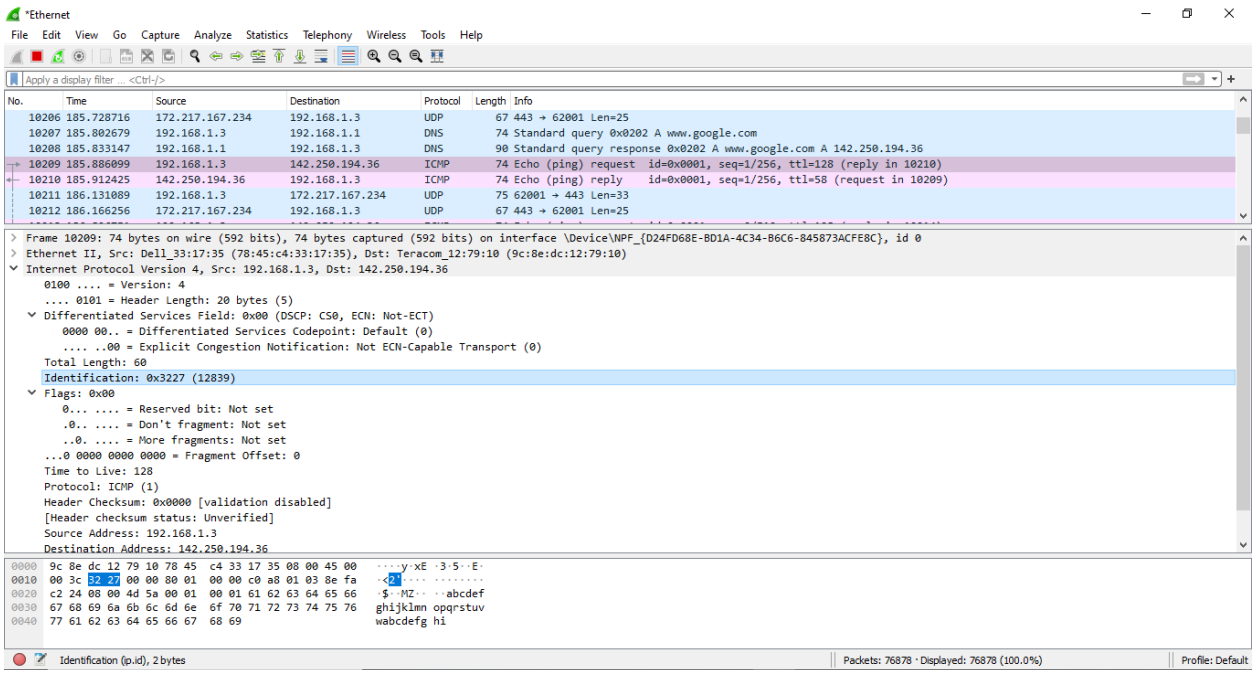
(d) Has this IP datagram been fragmented? Explain how you determined whether or not the datagram has been fragmented.

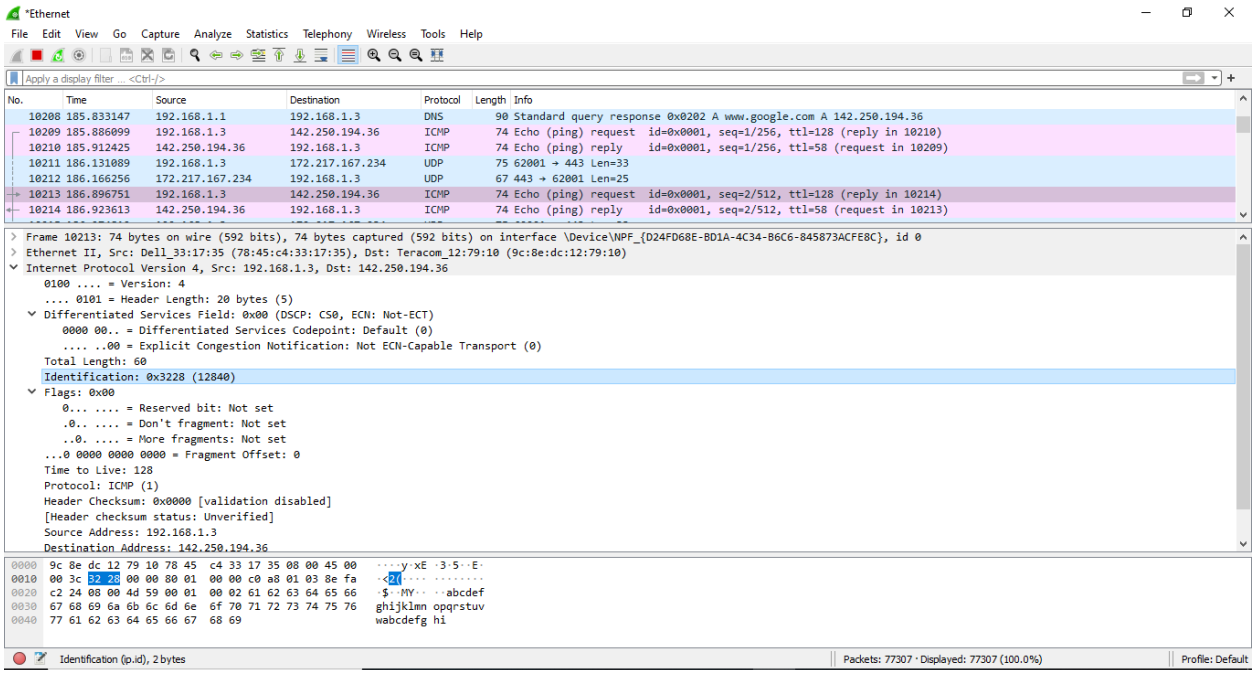
Solution: No, the IP datagram has not been fragmented. Observe the More fragments field. Notice that it is not set, indicating no more fragments will follow.



(e) Which fields in the IP datagram always change from one datagram to the next within this series of ICMP messages sent by your computer?

Solution: Identification field.





(f) Which fields stay constant? Which of the fields must stay constant? Which fields must change? Why?

Solution: The following fields remain constant:

- version (IPv4 always used)
- header length (doesn't change since we are always using IPv4)
- source IP (my computer's IP address doesn't change)
- destination IP (www.google .com IP address doesn't change)

• differentiated services (same protocol every time)
• upper layer protocol (same protocol every time)
• header checksum (verification disabled in my tests)
• TTL (remains same in my test) because it represents the maximum number of IP routers that the packet can go through before being discarded.

The following fields change:
• Identification field is incrementing (each IP datagram has a different ID). The identification field changes for all the ICMP TTL-exceeded replies **because the identification field is a unique value**. When two or more IP datagrams have the same identification value, then it means that these IP datagrams are fragments of a single large IP datagram.

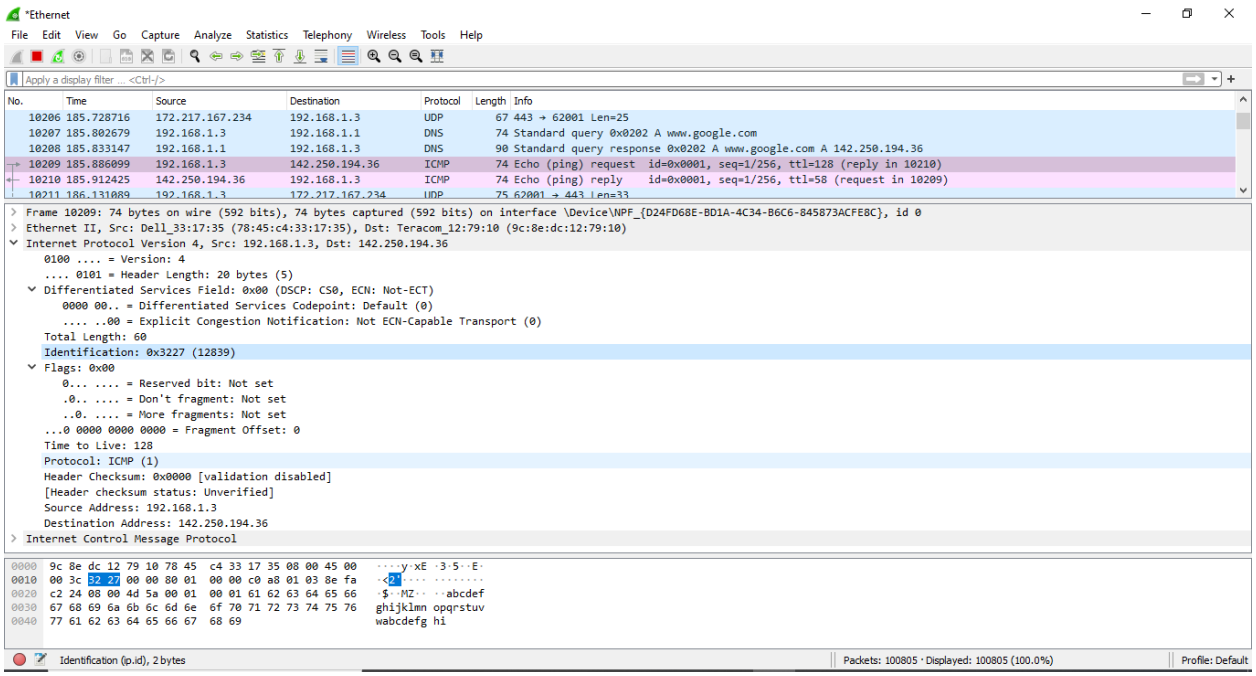
(g) Describe the pattern you see in the values in the Identification field of the IP datagram.

Solution: According to the below table. We can conclude that in the series of IP datagram ping request packets the identification field increments by 1. But in the series of IP datagram ping reply packets the identification field remains the same.

Packet no.	Request Identification field	Reply Identification field
1.	0x3227 (12839)	0x000 (0)
2.	0x3228 (12840)	0x000 (0)
3.	0x3229 (12841)	0x000 (0)
4.	0x322a (12842)	0x000 (0)

(h) What is the value in the Identification field and the TTL field?

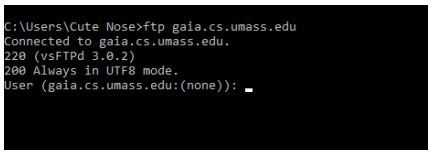
Solution: Identification field = 0x3227 (12839), TTL field = 128.



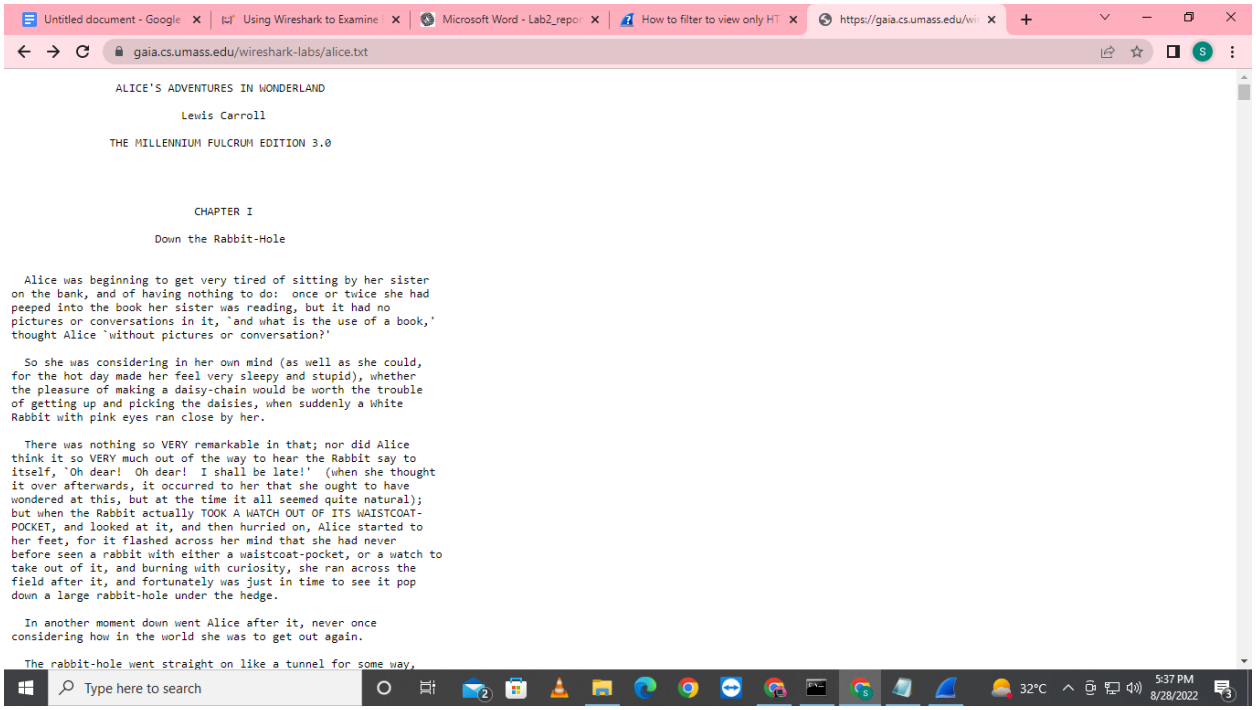
(i) Do these values remain unchanged for all of the ICMP TTL-exceeded replies sent to your computer by the nearest (first hop) router? Why?

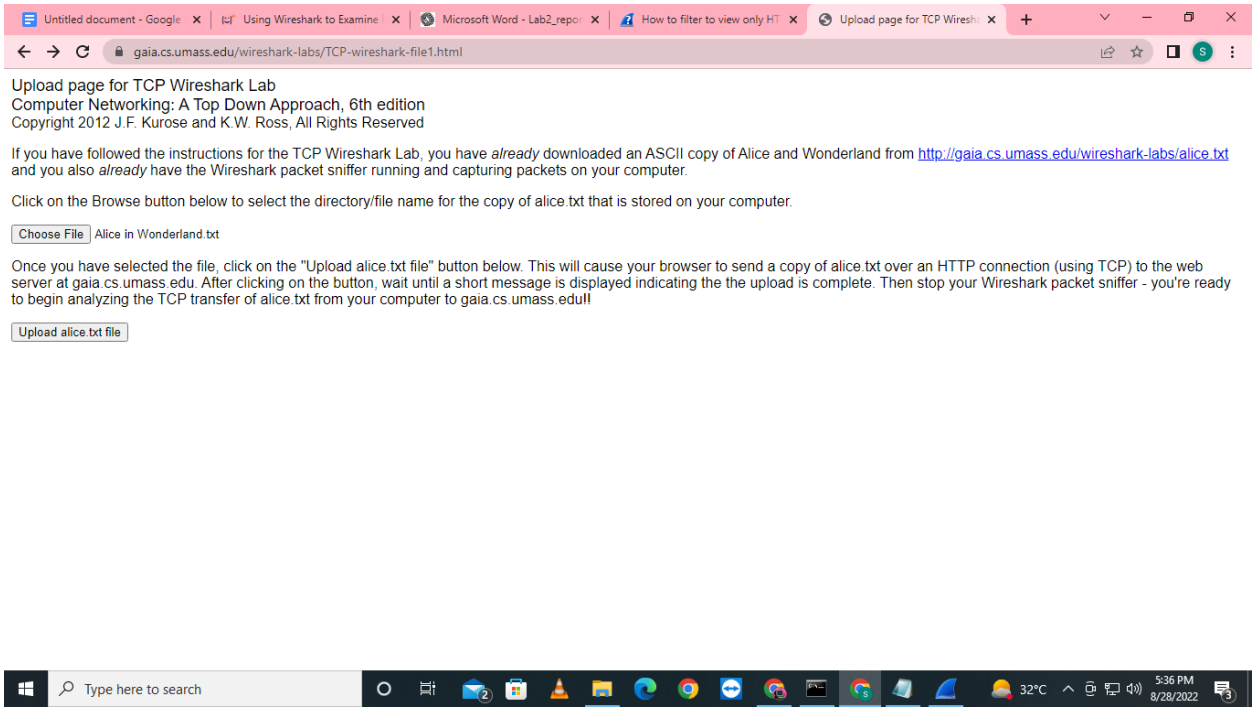
Solution: Yes, the values for Identification field and TTL remain unchanged for all of the ICMP TTL-exceeded replies sent to my computer because it represents the maximum number of IP routers that the packet can go through before being discarded.

FTP protocol analysis prerequisites:-

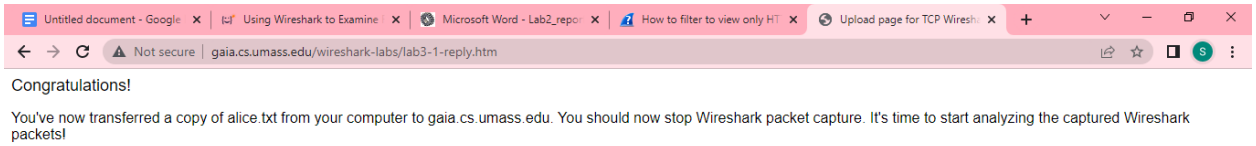


TCP protocol analysis prerequisites:-
Download the below file and save as “Alice in Wonderland”.



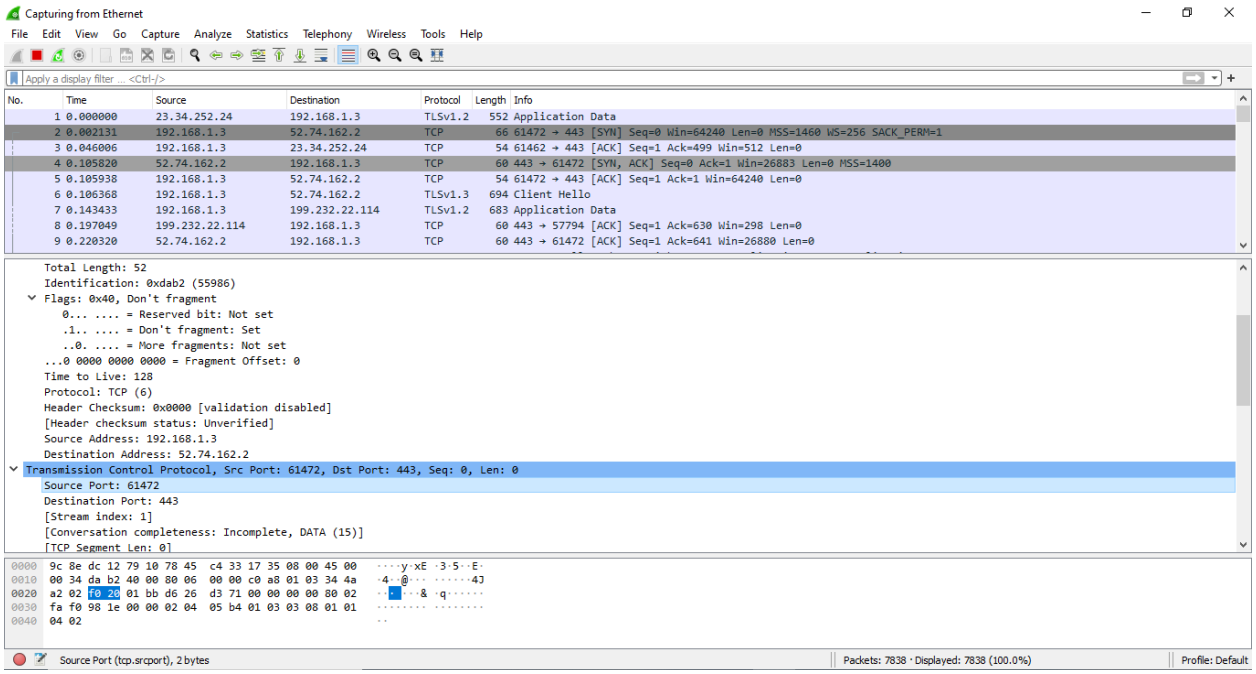


Restart Wireshark and then after upload the file.



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

Solution: Src IP address = 192.168.1.3
Src Port = 61472



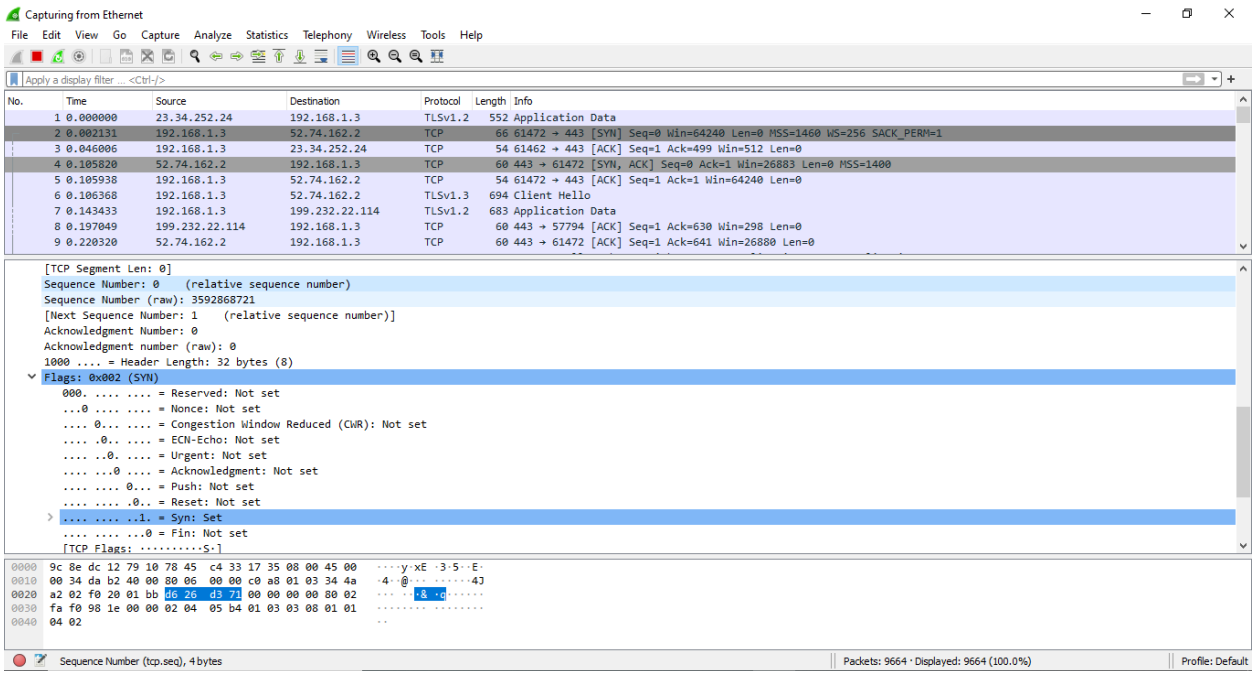
(k) What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in the segment that identifies the segment as a SYN segment?

Solution: Sequence Number is a 32-bit field that holds a number for the first byte sent in a particular segment. This number helps in the identification of the messages received in order.

Sequence number (relative) = 0

Sequence number (raw) = 3592868721

The first TCP datagram for the ftp session initiation only sets **SYN** bit to **1**.



(l) 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?

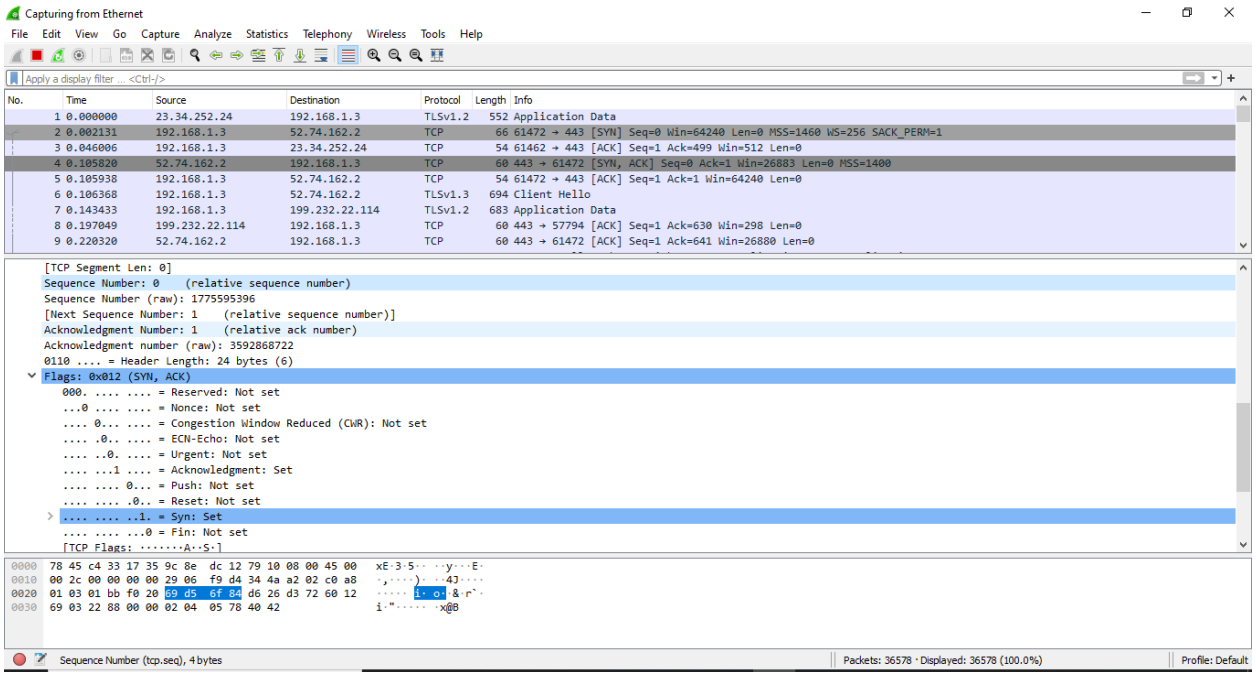
Solution: Sequence number (relative) = 0

Sequence number (raw) = 1775595396

Acknowledgement number (relative) = 1

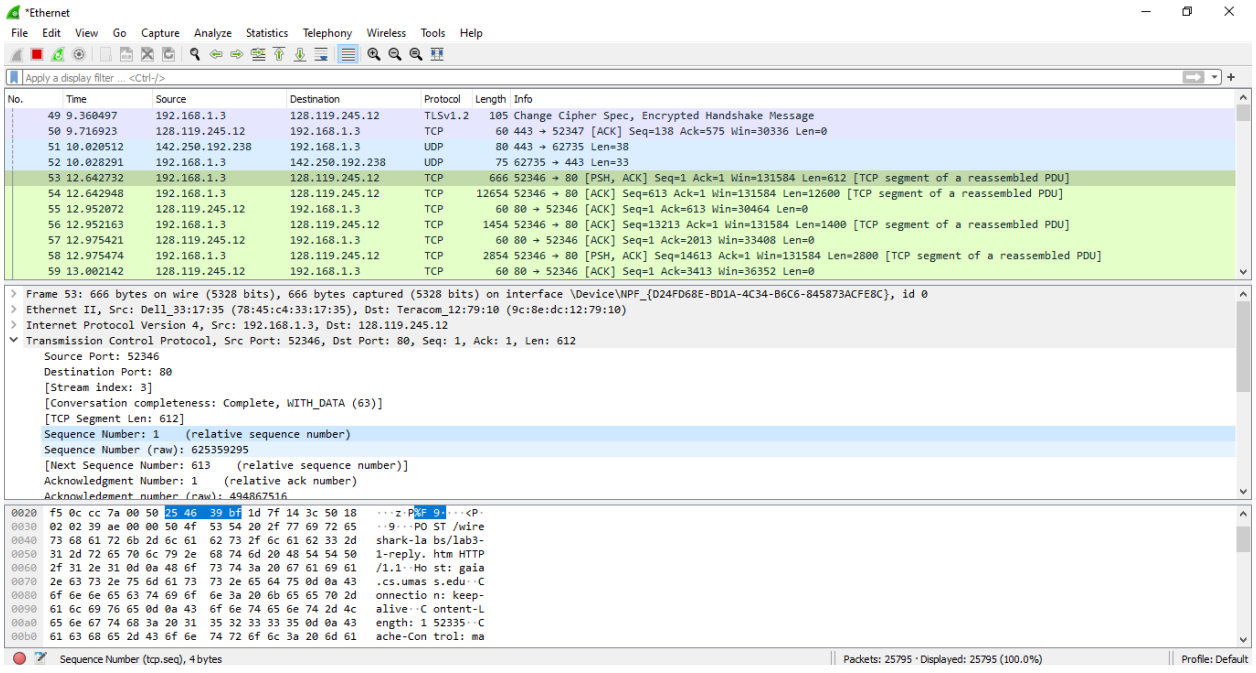
Acknowledgement number (raw) = 3592868722. The value of the ACKnowledgement field in the SYNACK segment is determined by gaia.cs.umass.edu **by adding 1 to the initial sequence number of SYN segment from the client computer** (i.e. the sequence number of the SYN segment initiated by the client computer is 0.).

The TCP datagram for the ftp sets **SYN & ACK** bit to **1**.



(m) 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.

Solution: Sequence number (relative) = 1
Sequence number (raw) = 625359295



(n) 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

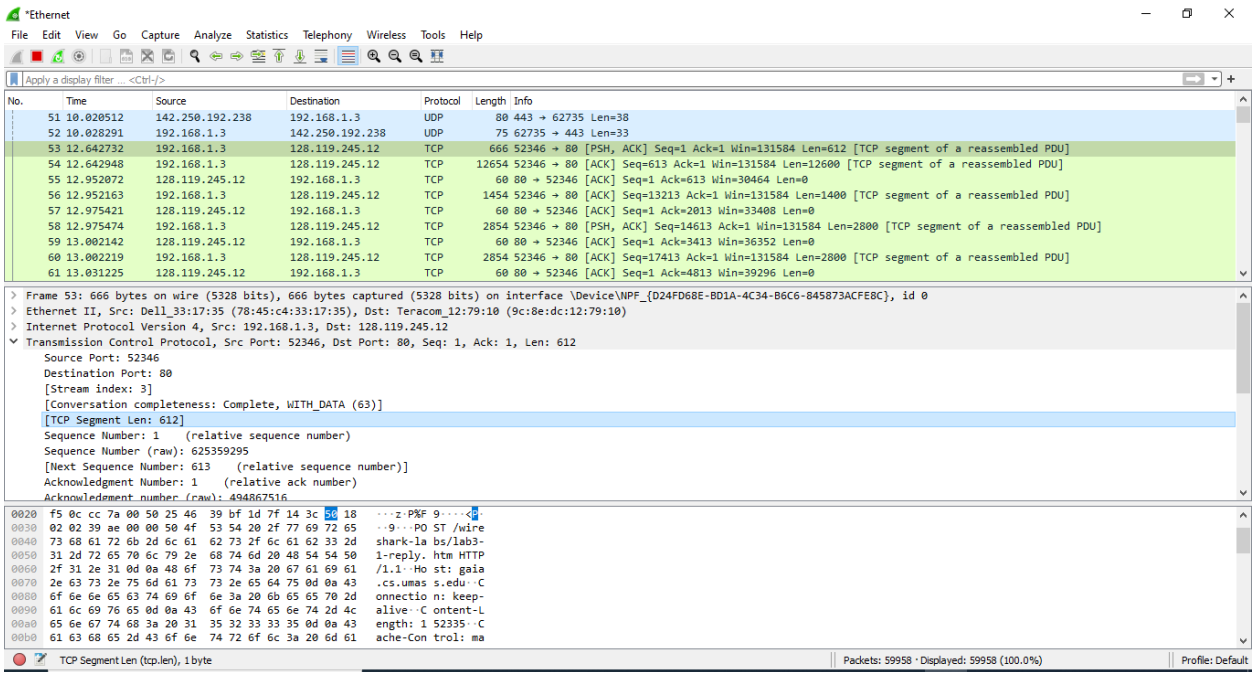
Solution:

Segment no.	Sequence number	Send time	Received time	RTT
1.	1	4.0255		4.0255
2.	613	4.02571	4.33484	0.30913
3.	13213	4.334931	4.358189	0.023258
4.	14613	4.358242	4.38491	0.26668
5.	17413	4.384987	4.413993	0.29006
6.	20213	4.414054	4.437691	0.023637

According to the formula: EstimatedRTT = 0.875 * EstimatedRTT + 0.125 * SampleRTT
EstimatedRTT after the receipt of the ACK of segment 1:
EstimatedRTT = RTT for Segment 1 = 0.095674 s
EstimatedRTT after the receipt of the ACK of segment 2:
EstimatedRTT = 0.30913 s
EstimatedRTT after the receipt of the ACK of segment 3:
EstimatedRTT = 0.875 * 0.30913 + 0.125 * 0.023258 = 0.273396 s
EstimatedRTT after the receipt of the ACK of segment 4:
EstimatedRTT = 0.875 * 0.273396 + 0.125 * 0.26668 = 0.2725565 s
EstimatedRTT after the receipt of the ACK of segment 5:
EstimatedRTT = 0.875 * 0.2725565 + 0.125 * 0.29006 = 0.2747444375 s
EstimatedRTT after the receipt of the ACK of segment 5:
EstimatedRTT = 0.875 * 0.2747444375 + 0.125 * 0.023637 = 0.2433560078 s

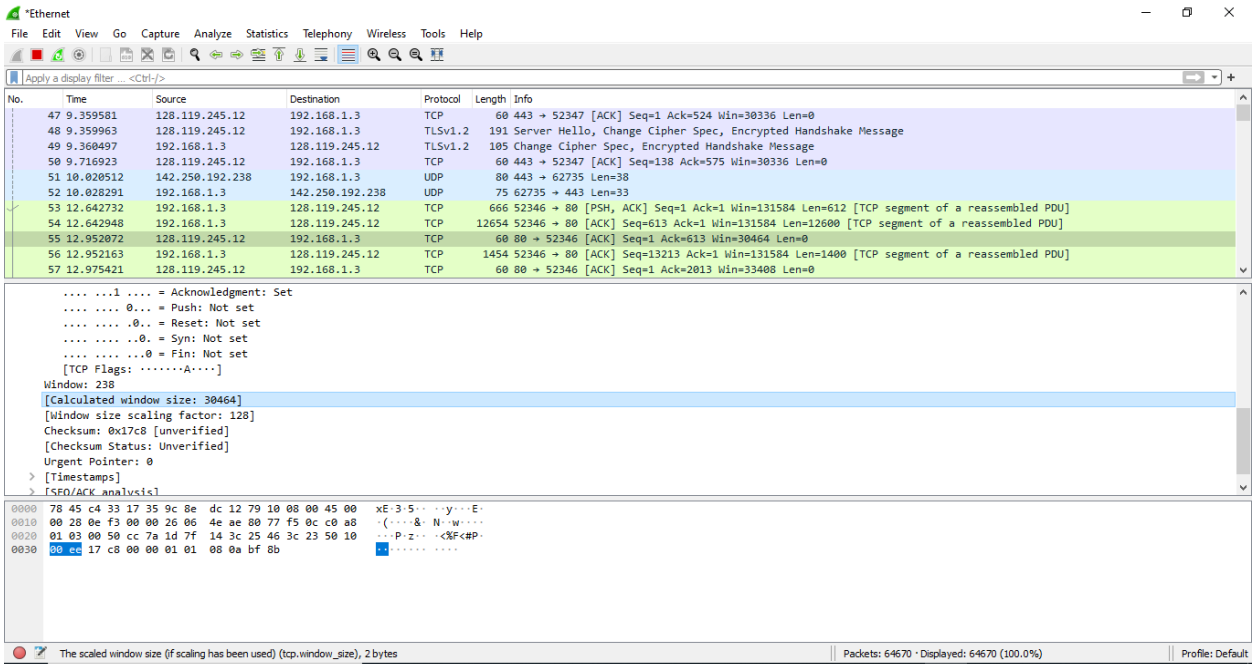
(o) What is the length of each of the first six TCP segments?

Solution: The length of the first TCP segment is 612 bytes, the length of the second TCP segment is 12600 bytes, the length of the third TCP segment is 1400 bytes. The length of each of the remaining TCP segments is 2800 bytes.

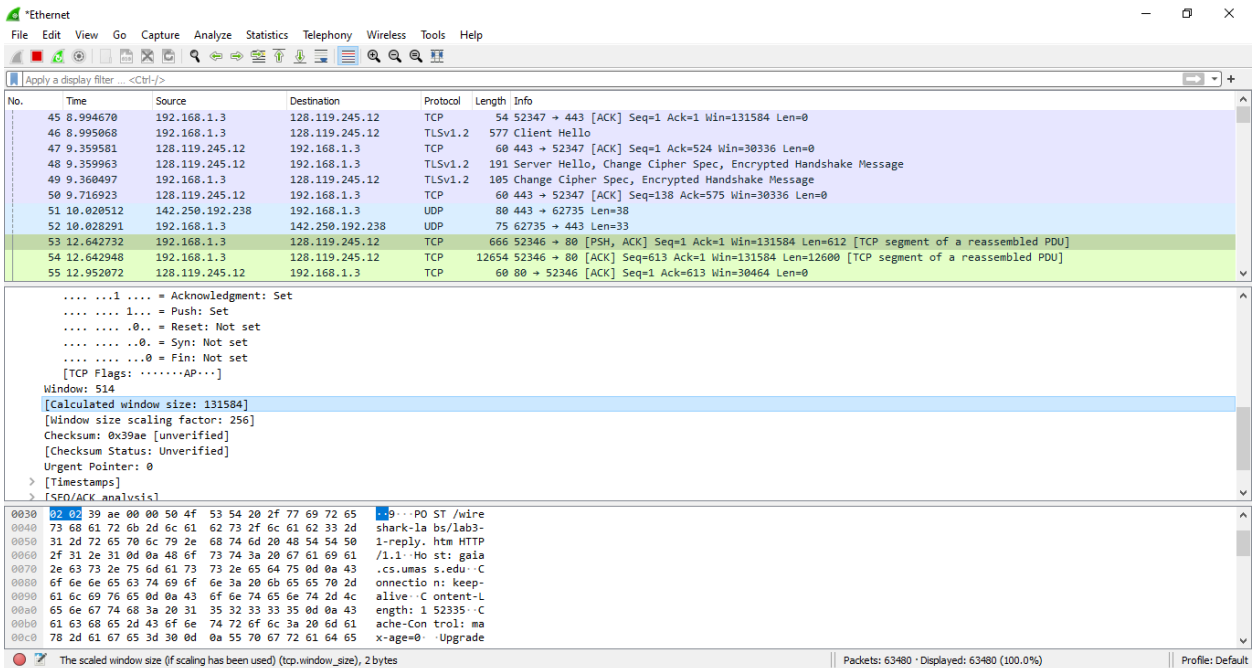


(p) What is the minimum amount of available buffer space advertised at the received for the entire trace? Does the lack of receiver buffer space ever throttle the sender?

Solution: The minimum amount of available buffer space advertised at the received for the entire trace is indicated first ACK from the server, its value is 30464 bytes (shown in above figure).

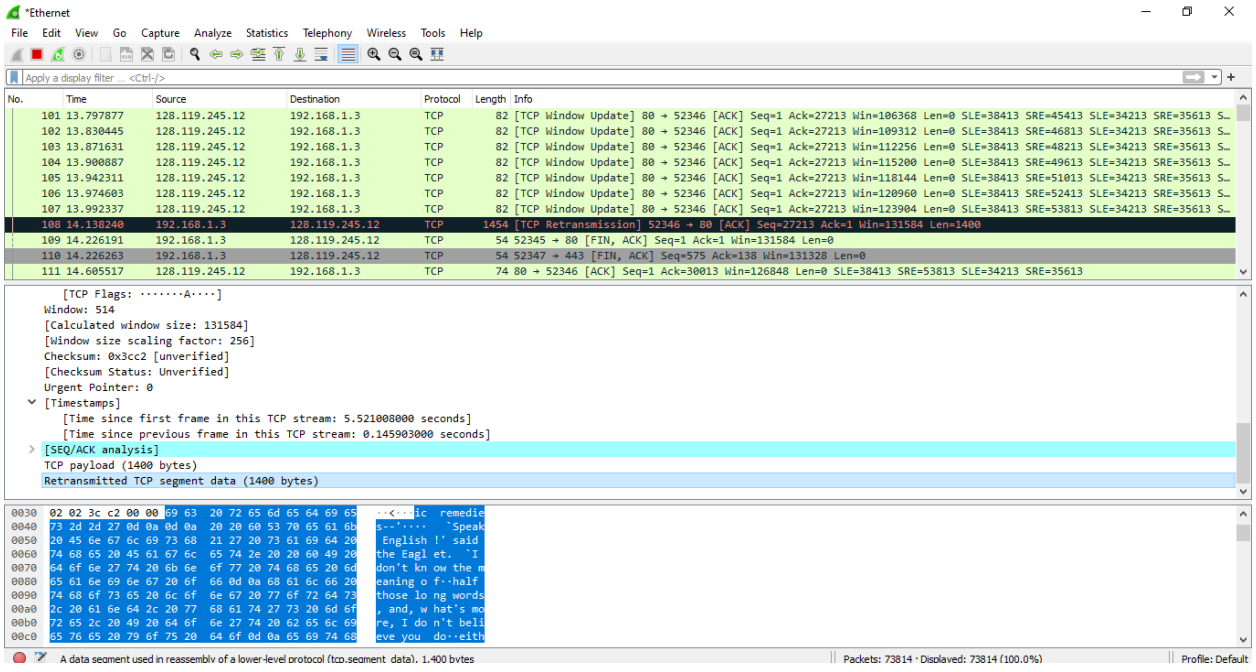


This reviver window grows until it reaches the maximum receiver buffer size of 131584 bytes. According to the trace, the sender is never throttled due to the lack of receiver buffer space.



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

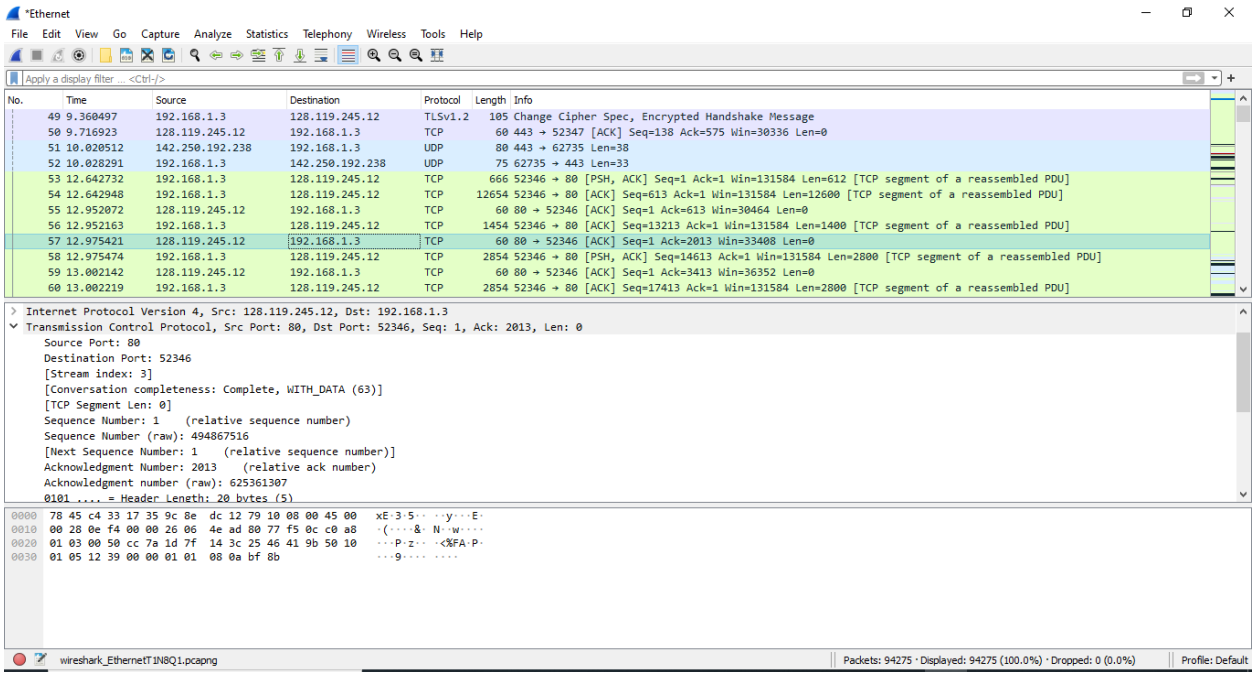
Solution: Yes, there are retransmitted segments in the trace file. When an outbound segment is handed down to an IP and there's no acknowledgment for the data before TCP's automatic timer expires, the segment is retransmitted.



(r) 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 250 in the

text).

Solution: The difference between the acknowledged sequence numbers of two consecutive ACKs indicates the data received by the server between these two ACKs. The receiver is ACKing every other segment. For example, segment of No. 57 acknowledged data with 2013 - 613 = 1400 bytes.

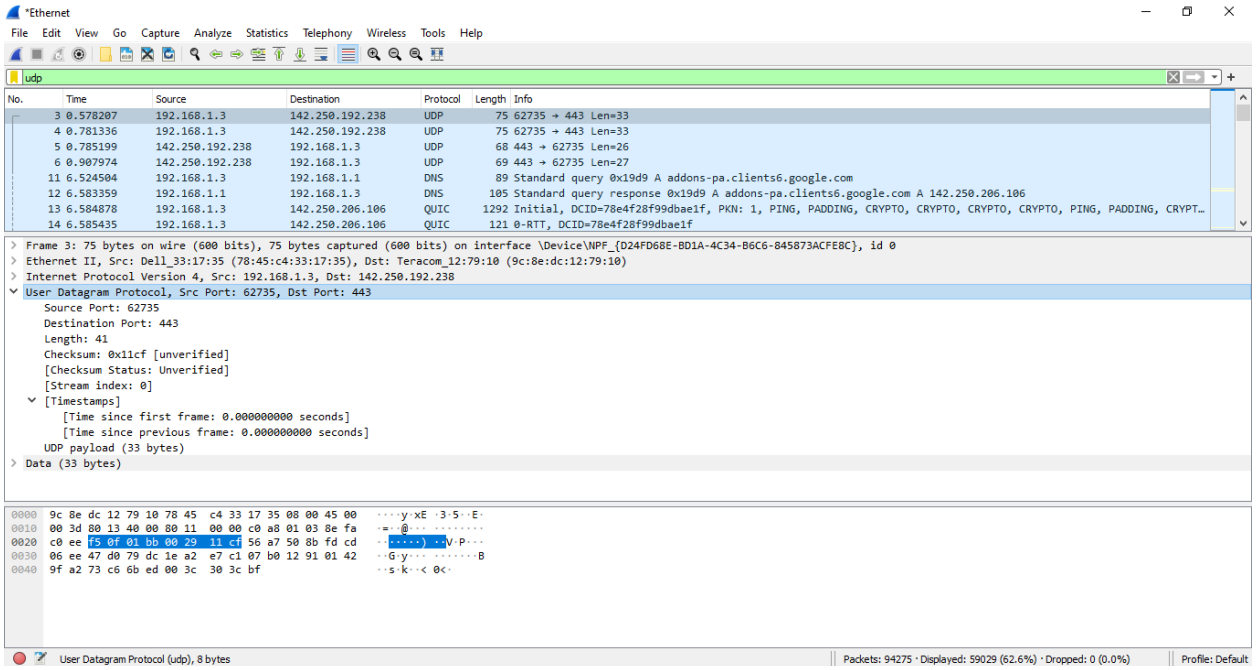


(s) What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

Solution: The alice.txt on the hard drive is 152,138 bytes, and the download time is 4.025500000(First TCP segment) - 0.000043000(last ACK) = 4.025457 second. Therefore, the throughput for the TCP connection is computed as 152,138/4.025457=37,793.969728 bytes/second.

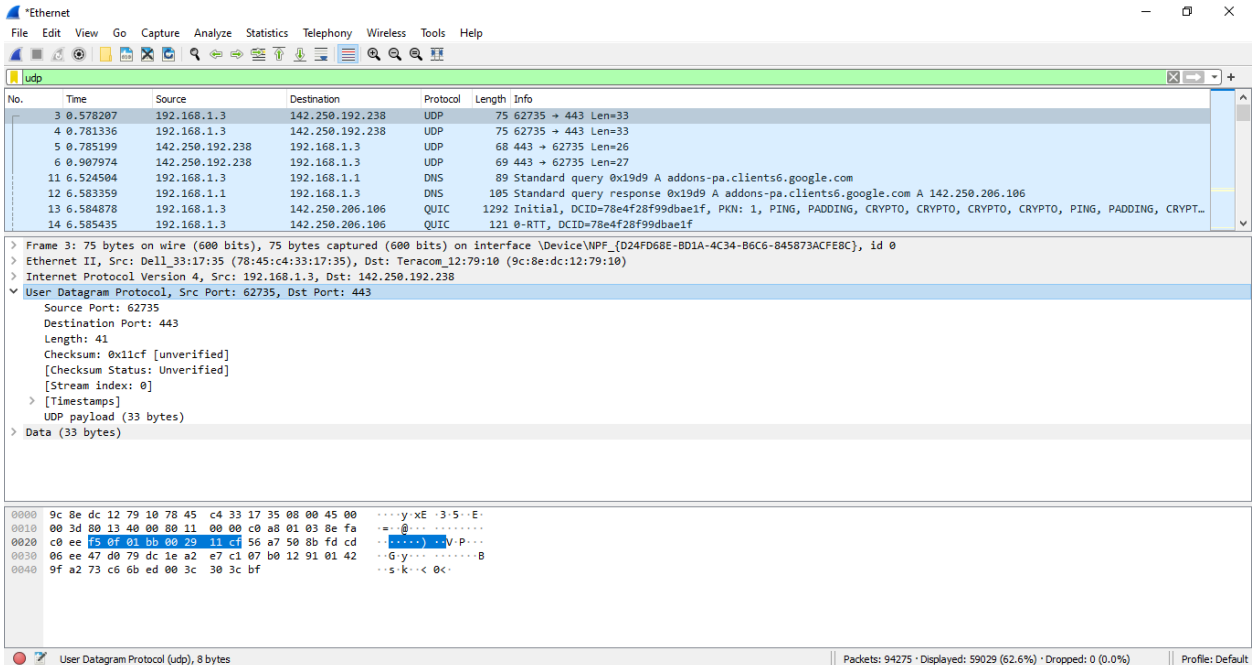
(t) Select one UDP packet from your trace. From this packet, determine how many fields there are in the UDP header. (You shouldn't look in the textbook! Answer these questions directly from what you observe in the packet trace.) Name these fields.

Solution: Source Port, Destination Port, Length, Checksum.



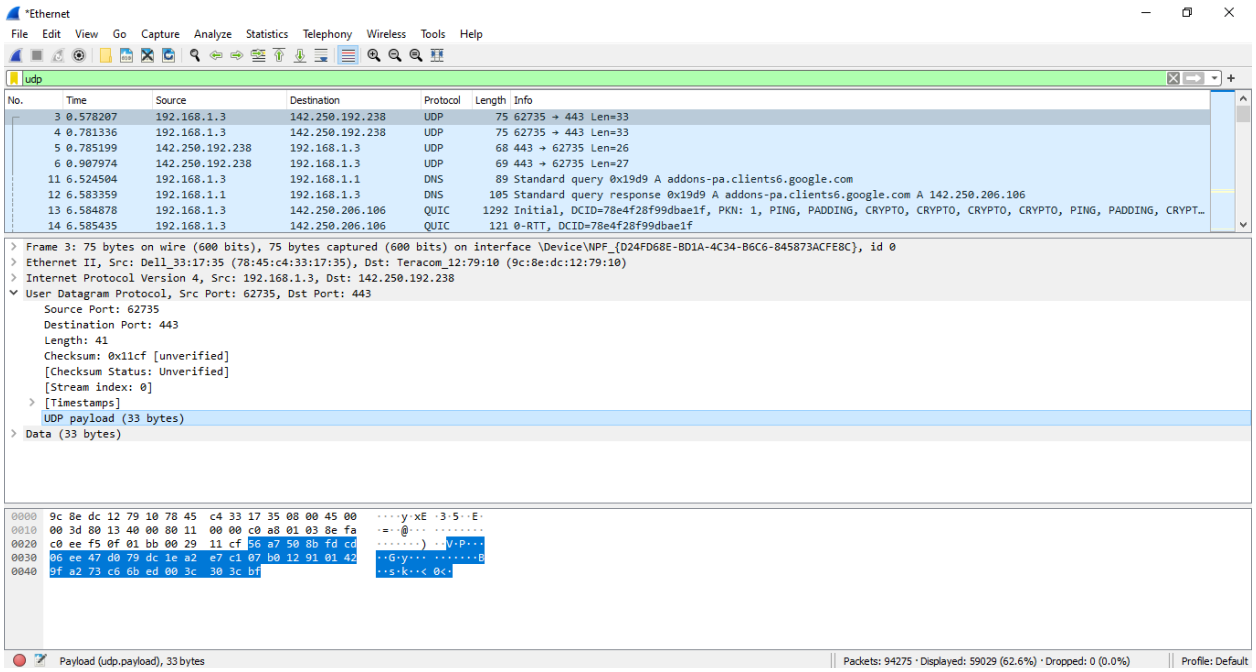
(u) By consulting the displayed information in Wireshark's packet content field for this packet, determine the length (in bytes) of each of the UDP header fields.

Solution: The UDP header has a fixed length of 8 bytes. Each of these 4 header fields is 2 bytes long.



(v) 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.

Solution: The length field specifies the number of bytes in the UDP segment (header plus data). An explicit length value is needed since the size of the data field may differ from one UDP segment to the next. The length of UDP payload for selected packet is 33 bytes. 41 bytes - 8 bytes = 33 bytes.



(w) 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)

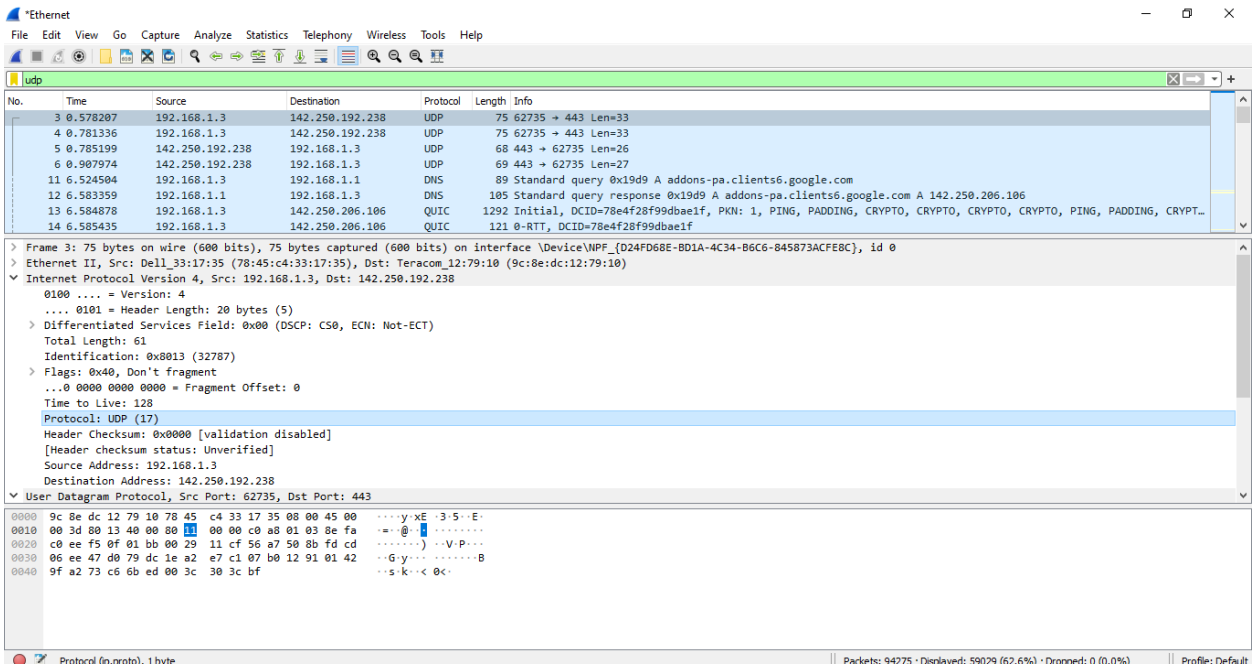
Solution: The maximum number of bytes that can be included in a UDP payload is (2^16 – 1) bytes plus the header bytes. This gives 65535 bytes – 8 bytes = 65527 bytes.

(x) What is the largest possible source port number? (Hint: see the hint in 4.)

Solution: The largest possible source port number is (2^16 – 1) = 65535.

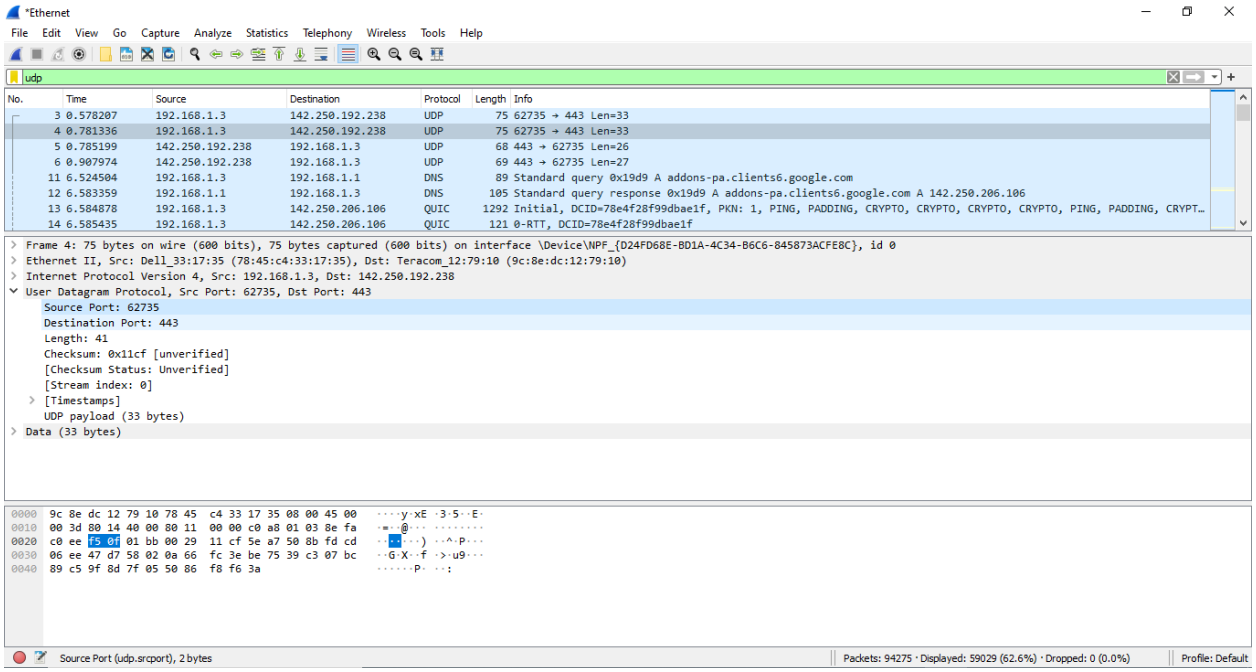
(y) What is the protocol number for UDP? Give your answer in both hexadecimal and decimal notation. To answer this question, you'll need to look into the Protocol field of the IP datagram containing this UDP segment (see Figure 4.13 in the text, and the discussion of IP header fields).

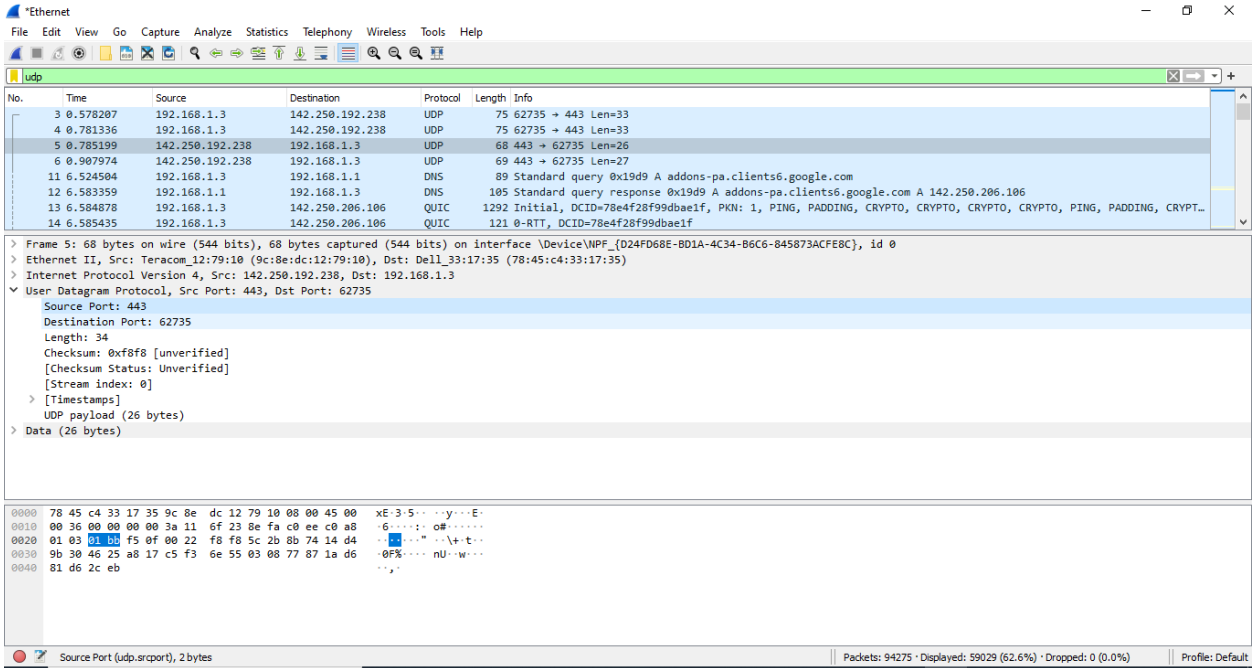
Solution: The IP protocol number for UDP is 0x11 hex, which is 17 in decimal value.



(z) 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. (Hint: for a second packet to be sent in response to a first packet, the sender of the first packet should be the destination of the second packet). Describe the relationship between the port numbers in the two packets.

Solution: The source port of the UDP packet sent by the host is the same as the destination port of the reply packet, and conversely the destination port of the UDP packet sent by the host is the same as the source port of the reply packet.


*Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
udp
No. Time Source Destination Protocol Length Info
3 0.578207 192.168.1.3 142.250.192.238 UDP 75 62735 → 443 Len=33
4 0.781336 192.168.1.3 142.250.192.238 UDP 75 62735 → 443 Len=33
5 0.785199 142.250.192.238 192.168.1.3 UDP 68 443 → 62735 Len=26
6 0.907974 142.250.192.238 192.168.1.3 UDP 69 443 → 62735 Len=27
11 6.524504 192.168.1.3 192.168.1.1 DNS 89 Standard query 0x19d9 A addons-pa.clients6.google.com
12 6.583359 192.168.1.1 192.168.1.3 DNS 105 Standard query response 0x19d9 A addons-pa.clients6.google.com A 142.250.206.106
13 6.584878 192.168.1.3 142.250.206.106 QUIC 1292 Initial, DCID=78e4f28f99dbae1f, PKN: 1, PING, PADDING, CRYPTO, CRYPTO, CRYPTO, CRYPTO, PING, PADDING, CRYPT…
14 6.585435 192.168.1.3 142.250.206.106 QUIC 121 0-RTT, DCID=78e4f28f99dbae1f
Frame 4: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface \Device\NPF_{D24FD68E-BD1A-4C34-B6C6-845873ACFE8C}, id 0
Ethernet II, Src: Dell_33:17:35 (78:45:c4:33:17:35), Dst: Teracom_12:79:10 (9c:8e:dc:12:79:10)
Internet Protocol Version 4, Src: 192.168.1.3, Dst: 142.250.192.238
User Datagram Protocol, Src Port: 62735, Dst Port: 443
Source Port: 62735
Destination Port: 443
Length: 41
Checksum: 0x11cf [unverified]
[Checksum Status: Unverified]
[Stream index: 0]
[Timestamps]
UDP payload (33 bytes)
Data (33 bytes)
0000 9c 8e dc 12 79 10 78 45 c4 33 17 35 08 00 45 00
0010 00 3d 80 14 40 00 80 11 00 00 c0 a8 01 03 8e fa
0020 c0 ee f5 0f 01 bb 00 29 11 cf 5e a7 50 8b fd cd
0030 06 ee 47 d7 58 02 0a 66 fc 3e be 75 39 c3 07 bc
0040 89 c5 9f 8d 7f 05 50 86 f8 f6 3a
Source Port (udp.srcport), 2 bytes
Packets: 94275 · Displayed: 59029 (62.6%) · Dropped: 0 (0.0%)
Profile: Default


*Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
udp
No. Time Source Destination Protocol Length Info
3 0.578207 192.168.1.3 142.250.192.238 UDP 75 62735 → 443 Len=33
4 0.781336 192.168.1.3 142.250.192.238 UDP 75 62735 → 443 Len=33
5 0.785199 142.250.192.238 192.168.1.3 UDP 68 443 → 62735 Len=26
6 0.907974 142.250.192.238 192.168.1.3 UDP 69 443 → 62735 Len=27
11 6.524504 192.168.1.3 192.168.1.1 DNS 89 Standard query 0x19d9 A addons-pa.clients6.google.com
12 6.583359 192.168.1.1 192.168.1.3 DNS 105 Standard query response 0x19d9 A addons-pa.clients6.google.com A 142.250.206.106
13 6.584878 192.168.1.3 142.250.206.106 QUIC 1292 Initial, DCID=78e4f28f99dbae1f, PKN: 1, PING, PADDING, CRYPTO, CRYPTO, CRYPTO, CRYPTO, PING, PADDING, CRYPT…
14 6.585435 192.168.1.3 142.250.206.106 QUIC 121 0-RTT, DCID=78e4f28f99dbae1f
Frame 5: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface \Device\NPF_{D24FD68E-BD1A-4C34-B6C6-845873ACFE8C}, id 0
Ethernet II, Src: Teracom_12:79:10 (9c:8e:dc:12:79:10), Dst: Dell_33:17:35 (78:45:c4:33:17:35)
Internet Protocol Version 4, Src: 142.250.192.238, Dst: 192.168.1.3
User Datagram Protocol, Src Port: 443, Dst Port: 62735
Source Port: 443
Destination Port: 62735
Length: 34
Checksum: 0xf8f8 [unverified]
[Checksum Status: Unverified]
[Stream index: 0]
[Timestamps]
UDP payload (26 bytes)
Data (26 bytes)
0000 78 45 c4 33 17 35 9c 8e dc 12 79 10 08 00 45 00
0010 00 36 00 00 00 00 3a 11 6f 23 8e fa c0 ee c0 a8
0020 01 03 01 bb f5 0f 00 22 f8 f8 5c 2b 8b 74 14 d4
0030 9b 30 46 25 a8 17 c5 f3 6e 55 03 08 77 87 1a d6
0040 81 d6 2c eb
Source Port (udp.srcport), 2 bytes
Packets: 94275 · Displayed: 59029 (62.6%) · Dropped: 0 (0.0%)
Profile: Default