**LAB 4**

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?

**Answer:** The IP address is 192.168.1.102

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

**Answer:** ICMP (1)

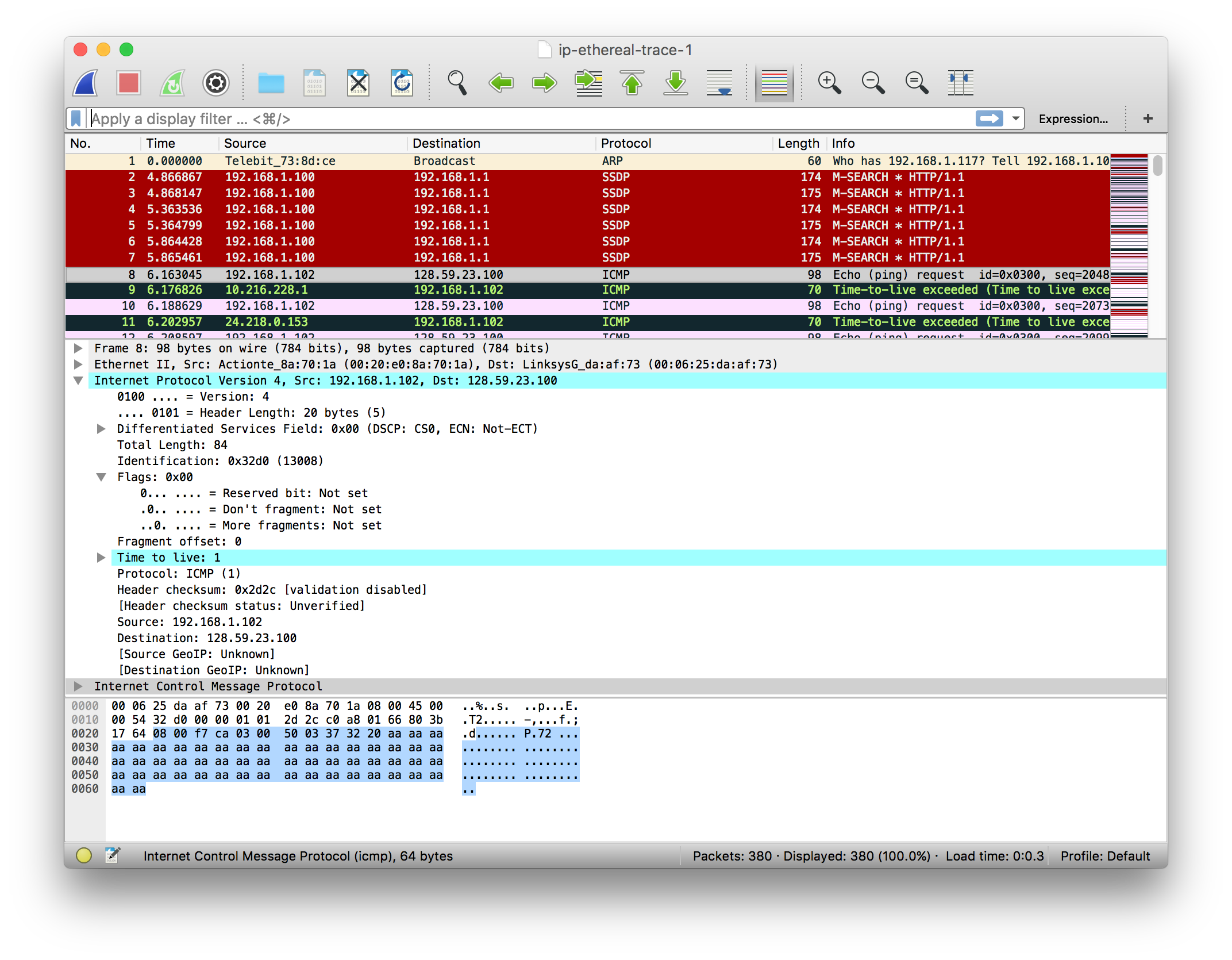
3. How many bytes are in the IP header? How many bytes are in the payload *of the IP datagram*? Explain how you determined the number of payload bytes.

**IP header:** 20 bytes

**Payload bytes:**  64 bytes, we know this since the header was 20 bytes and the length of the message was 84. Thus, 84 – 20 gives us a payload of 64 bytes.

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

**Answer:** The fragment bit is 0 so the IP datagram has not been fragmented.



4

3

2

1

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?

* Identification
* Time to live
* Header Checksum

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

**Answer:** The fields that stay constant and must stay constant are:

* header length – since these are all ICMP packets
* source/dest ip –since we are sending the data to the same spot
* upper layer protocol – since these are all ICMP packets
* version – We are using IPv4

The fields that change:

* Identification – packets need different ids
* TTL – traceroute increases the TTL
* Header checksum – since the header data changes.

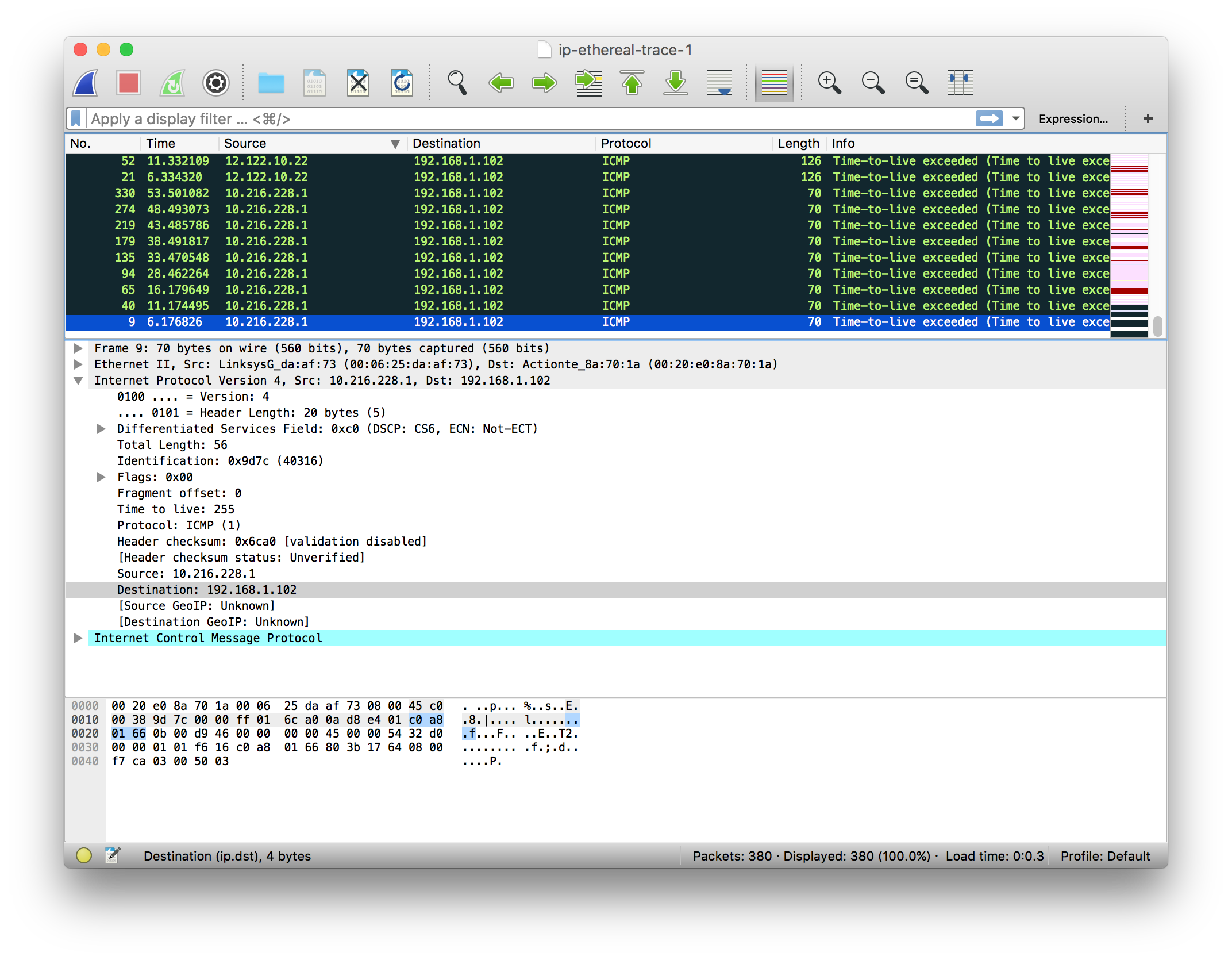
7. Describe the pattern you see in the values in the Identification field of the IP datagram.

**Answer:** The identification field increases by 1.

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

**Answer:**

* Identification: 40316
* TTL: 255



8

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?

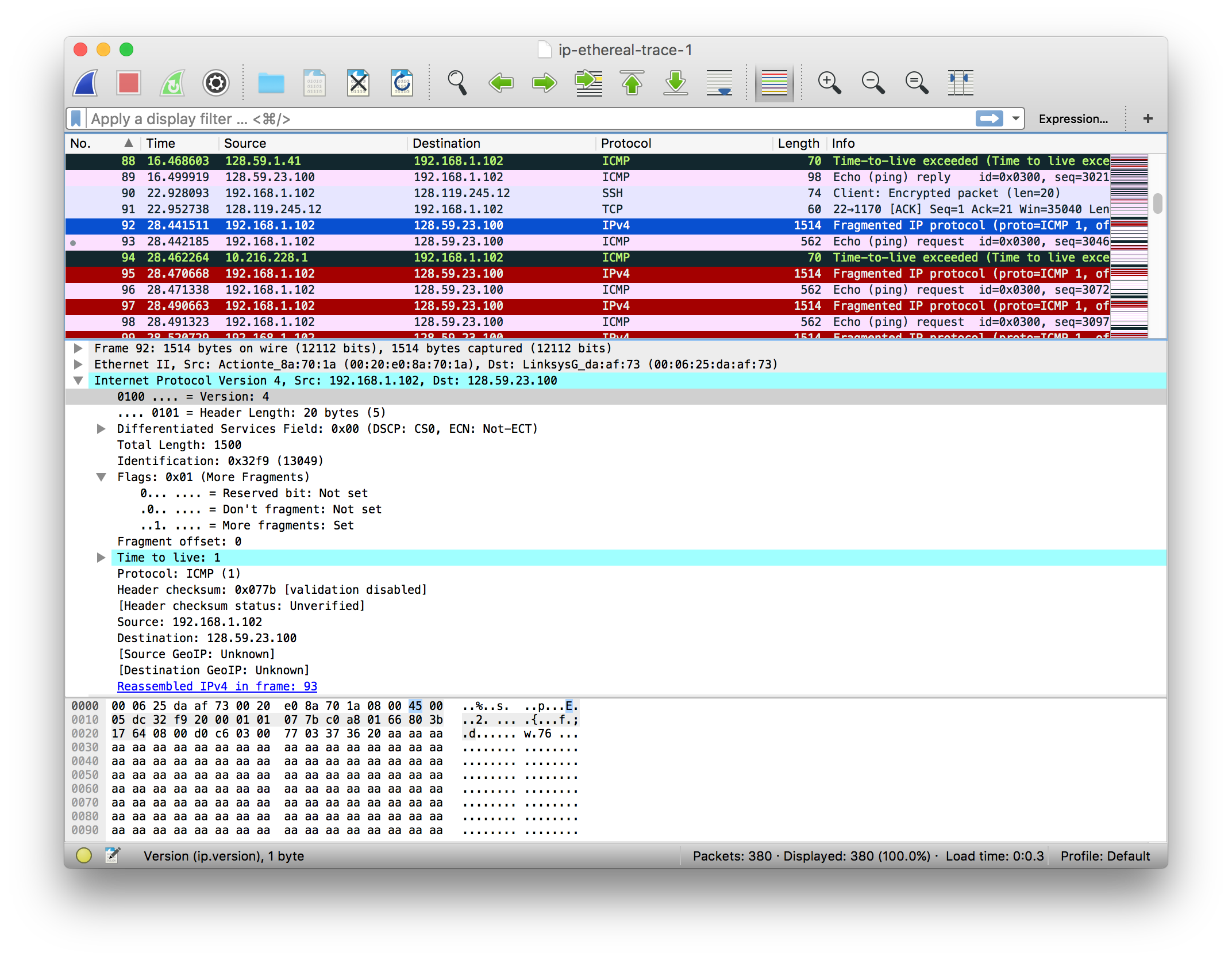
**Answer:** The identification field changes for each request because it must be unique. The TTL field stays the same since the TTL from the first hop router does not change.

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?

**Answer:** Yes, it has been fragmented.

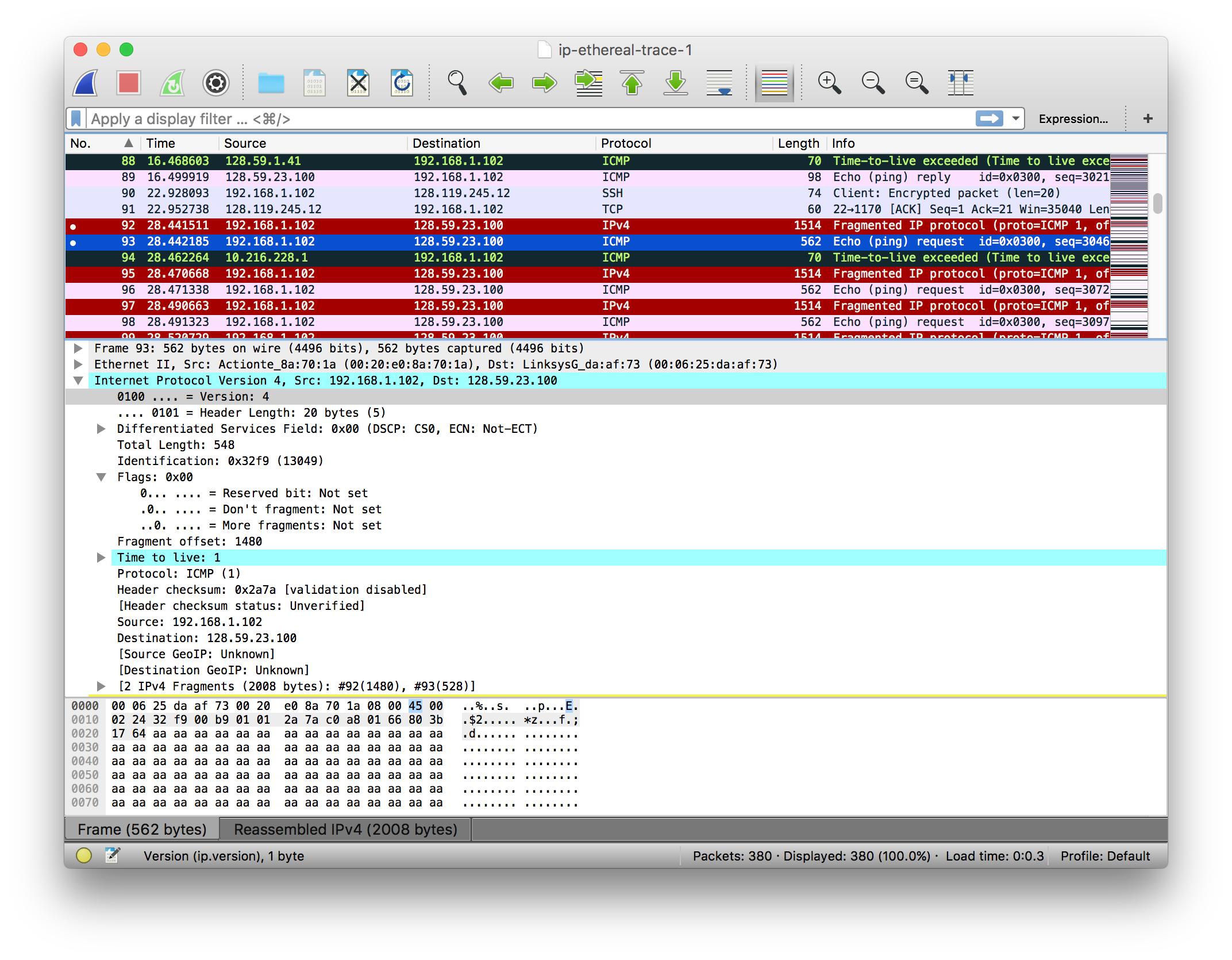
11. Screenshot the first fragment of the fragmented IP datagram (with sufficient details to answer these questions). What information in the IP header indicates that the datagram been fragmented? What information in the IP header indicates whether this is the first fragment versus a latter fragment? How long is this IP datagram?

**Answer:** The more fragments flag has been set to 1, which indicates the data has been fragmented. We can also tell which fragment is first by looking at the fragment offset. For the first fragment, it is 0. The length of this datagram is 1500 bytes, including the header.



12. Screenshot the second fragment of the fragmented IP datagram (with sufficient details to answer these questions). What information in the IP header indicates that this is not the first datagram fragment? Are the more fragments? How can you tell?

**Answer:** The fragment offset is 1480, which tells us this is not the first fragment. We can tell that this is the last fragment since the more fragments flag is 0.



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

* total length
* flags
* fragment offset
* checksum

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

**Answer:** 3 fragments were created.

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

**Answer:** The first two fragments contain the same flags and total length (1500), while the third fragment does not have any flags set and a length of 568. The fragment offset and checksum are different for all 3 fragments.

