Nathan Flack

Wireshark Lab #4

**Note: using provided trace (icmp-ethereal-trace-1)**

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: 192.168.1.102**

A screenshot of a cell phone

Description automatically generated



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

**Protocol: ICMP (1)**

A screenshot of a cell phone

Description automatically generated



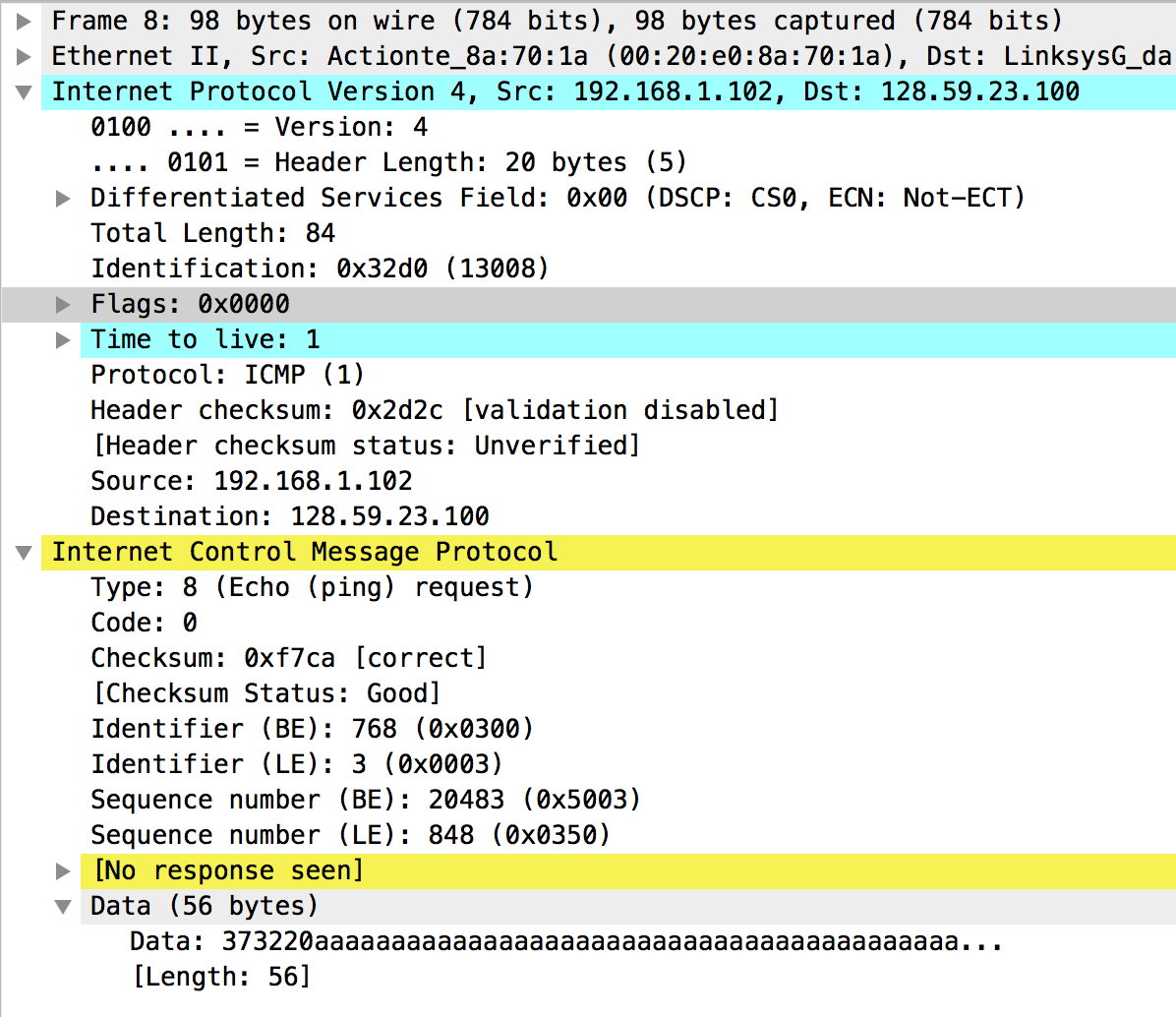
1. How many bytes are in the IP header? How many bytes are in the payload *of the IP datagram*? Explain how you determined the number of payload bytes.

**Header: 20 Bytes**

A screenshot of a cell phone

Description automatically generated



Payload: 64 Bytes. I subtracted the size of the header (20 bytes) from the total length (84 bytes)





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

**No, the IP datagram has not been fragmented since the “More fragments” flag is 0. If this flag was set to 1, that means there would be at least one more fragment expected.**

A screenshot of a cell phone

Description automatically generated



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

**The “Identification” field is changing as well as the Time to Live (TTL). This of course, drives changes to the checksum.**

A screenshot of a cell phone

Description automatically generated



A screenshot of a cell phone

Description automatically generated



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



**These fields stay constant: Flags, data/header length, protocol, IPs (for this block of requests), and length. Fields that must remain consistent are at least the protocol (ICMP), the length/size of the data and header. The length/size cannot change because this would drive changes to the timing that traceroute is trying to measure.**



A screenshot of a cell phone

Description automatically generated



A screenshot of a cell phone

Description automatically generated



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

**Identification Field values: 13126, 13127, 13128, 13129, 13130, …**

**The value is increasing for each echo ping request sent.**

A screenshot of a cell phone

Description automatically generated



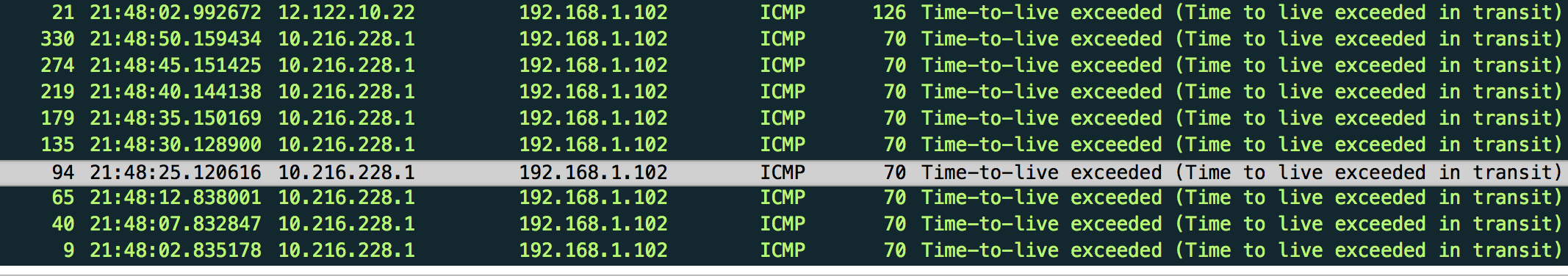
A screenshot of a cell phone

Description automatically generated



Next (with the packets still sorted by source address) find the series of ICMP TTL-exceeded replies sent to your computer by the nearest (first hop) router.

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

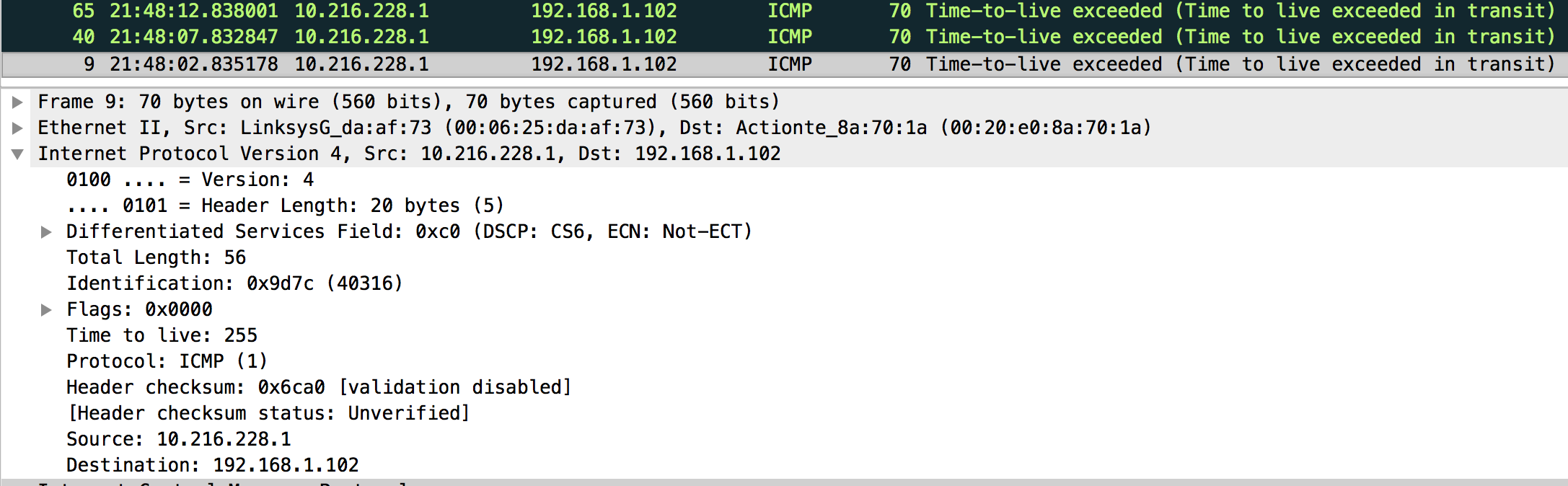
First TTL exceeded packet sent from 10.216.228.1





Identification field: 0x9D7C (40316)

TTL: 255





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

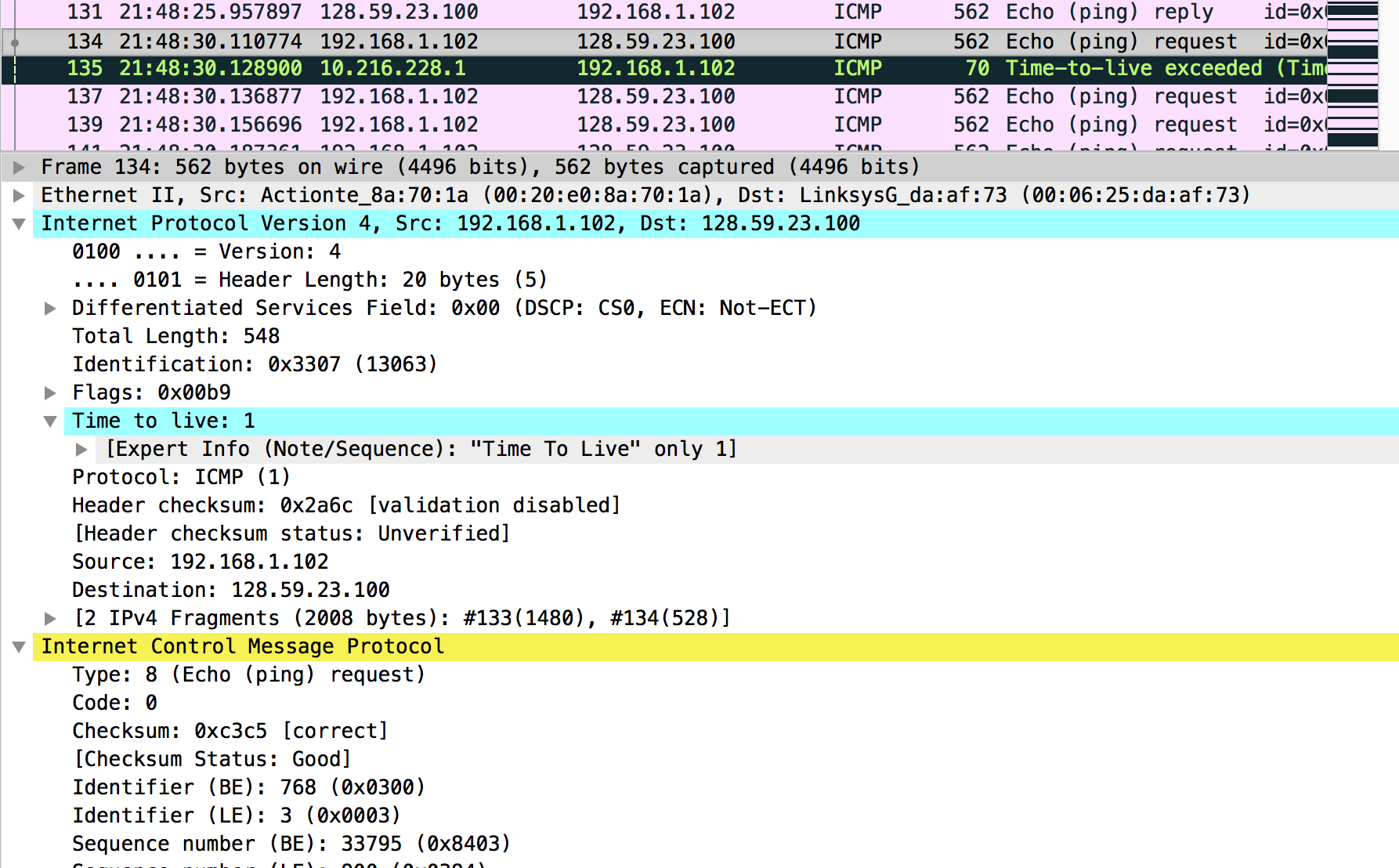
**The TTL field (set to 255 by the router when it sends the TTL-exceeded reply) does not change because the packet is coming from one hop away. Therefore, the TTL field has not been decremented by any other routers before it is received by my computer.**

Fragmentation

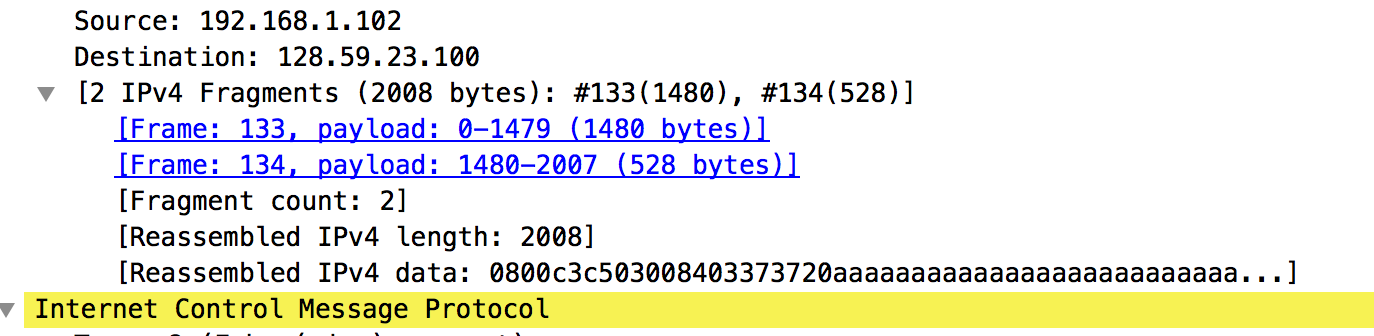
Sort the packet listing according to time again by clicking on the *Time* column.

1. 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?

**Yes, the message has been fragmented (however, Wireshark is showing me only 1 packet (134) instead of both packets (133 and 134)**





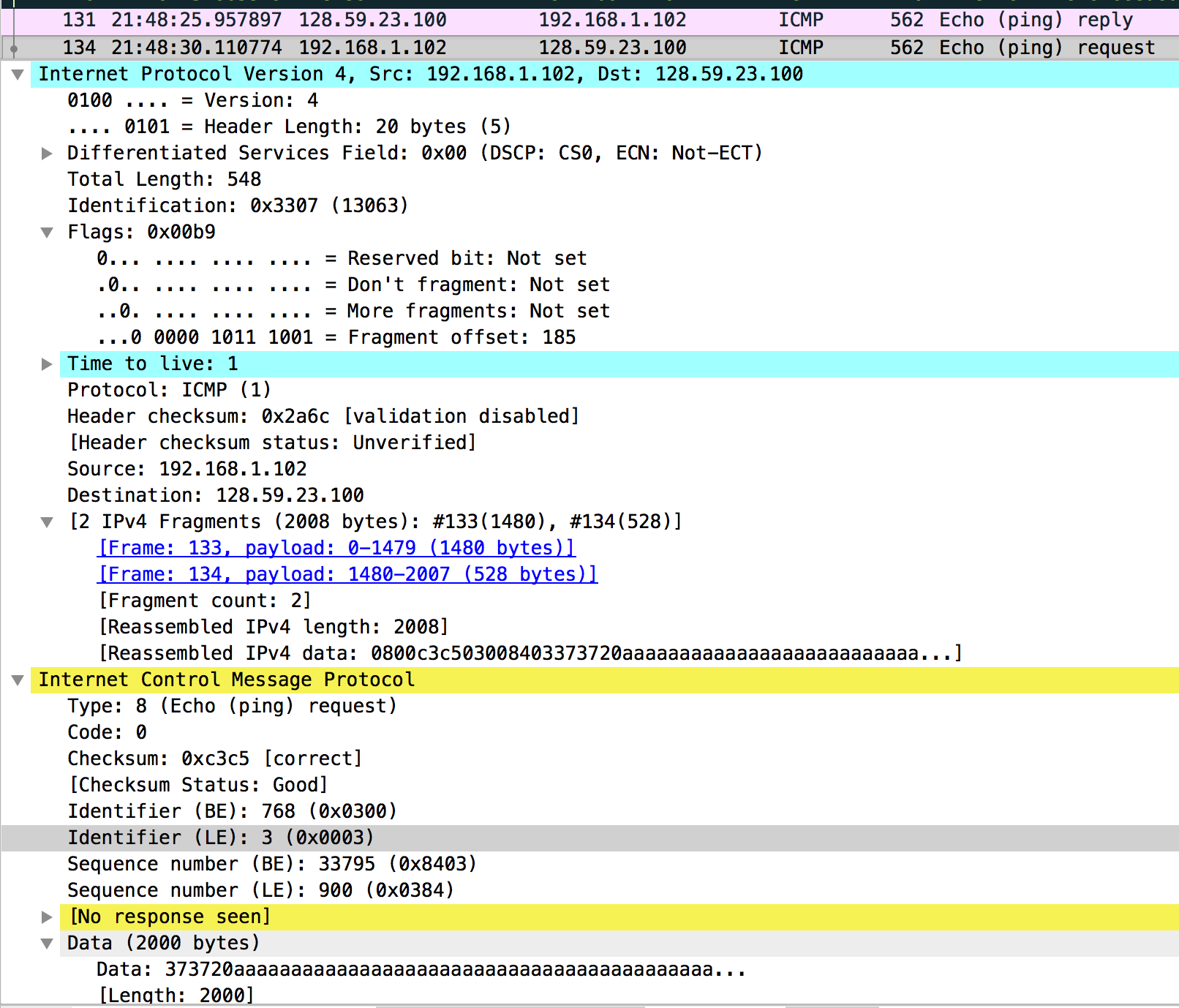




1. Print out the first fragment of the fragmented IP datagram. What information in the IP header indicates that the datagram been fragmented? What information in the IP header indicates whether this is the first fragment versus a latter fragment? How long is this IP datagram?

**The Flags show us the datagram has been fragmented. In this case Wireshark tells us that the datagram was fragmented, but puts it back together for us without showing the individual packets. IF I could show the first packet it would set the “More fragments:” flag (value of 1). This tells us that at least one fragmented packet exists. We would know it was the first fragment in the set because the offset flag would be set to ‘0’.**

**The IP datagram size (for the first packet) would equal data + header = 1480 + 20 = 1500 bytes**





Because there were two fragments, the “More fragments” would equal 1 for the first fragment and 0 for the second fragments.



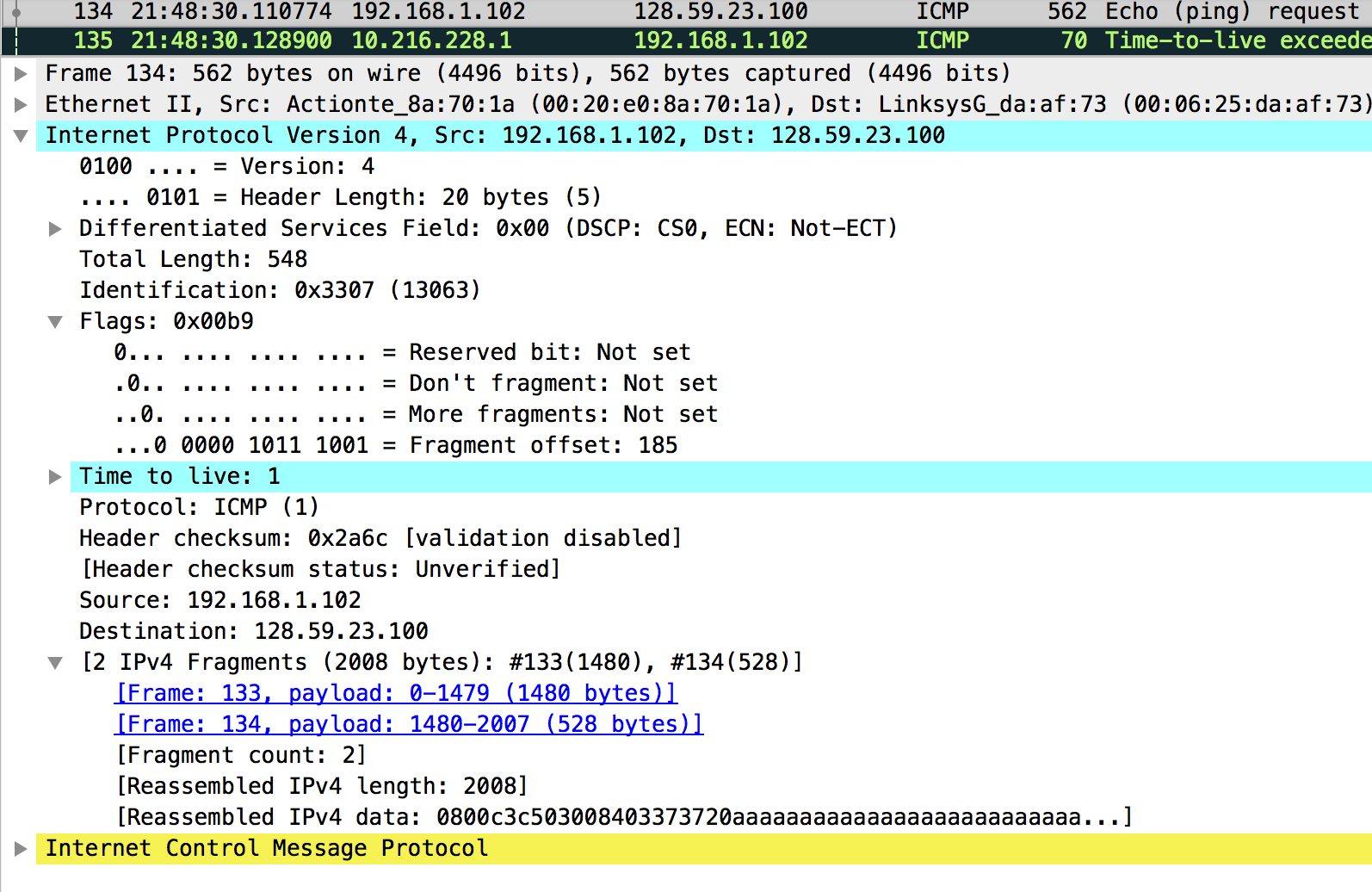
1. Print out the second fragment of the fragmented IP datagram. What information in the IP header indicates that this is not the first datagram fragment? Are there more fragments? How can you tell?

**I cannot print the second fragment because Wireshark has put them back together for me (I think it is a feature ☺). We know it is not the first fragment because of the “Fragment Offset” is not zero. The last fragment would show a ‘0’ for the “More fragments” flag communicating that it is the last fragment of the datagram. Therefore we would know there are no more fragments**.

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

**I cannot show a screenshot but here are the differences in the IP header for the first and second fragment:**

* **Total Length**
* **More Fragments flag**
* **Fragment offset field**
* **Checksum**

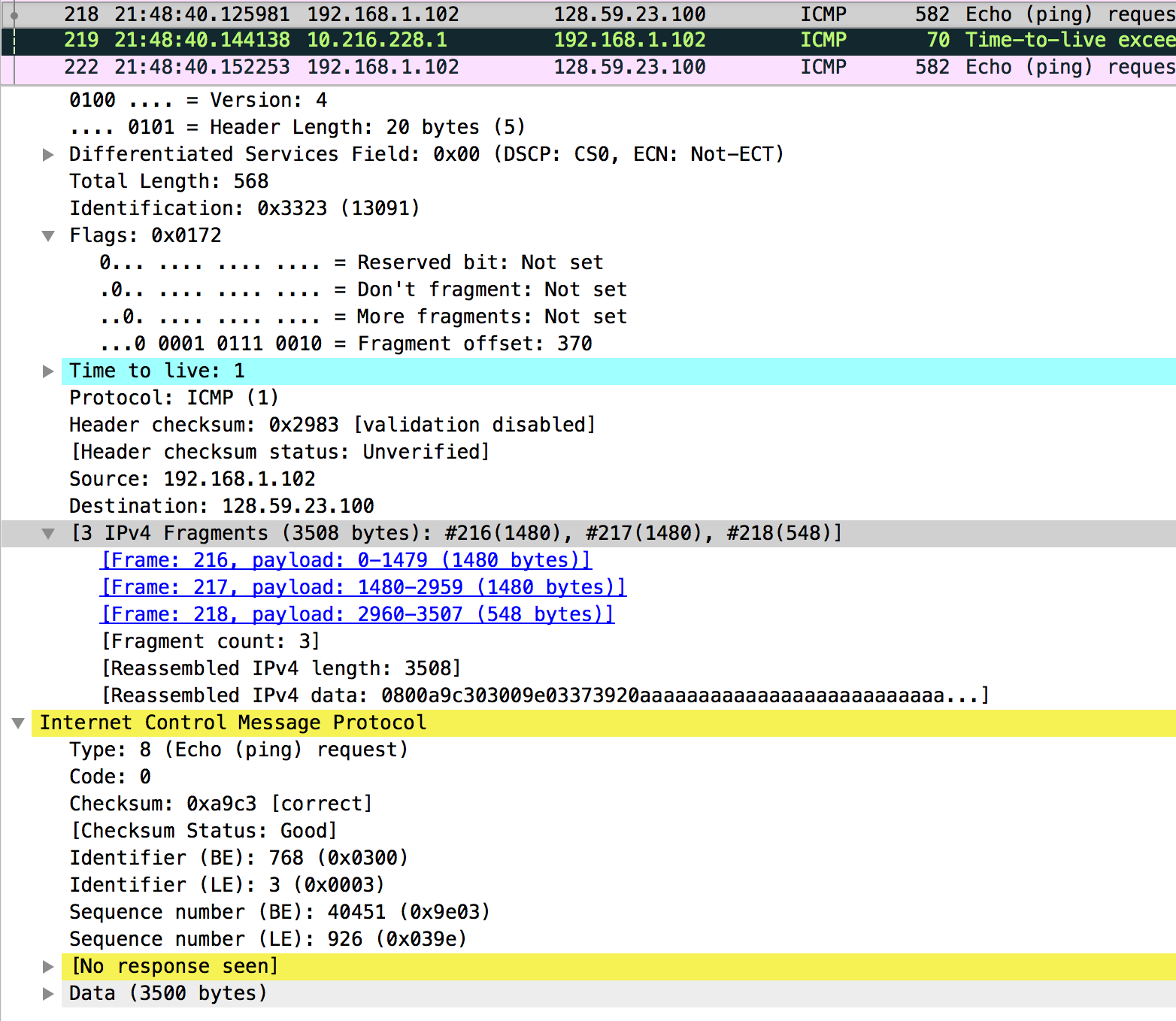


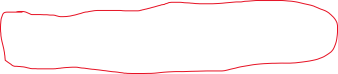


Now find the first ICMP Echo Request message that was sent by your computer after you changed the *Packet Size* in *pingplotter* to be 3500.

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

**Three.**





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

**Cannot take a screen shot, but these values change:**

* **More Fragments flag: (1st: ‘1’ 2nd: ‘1’ 3rd: ‘0’)**
* **Fragment Offset (1st: ‘0’ 2nd: ‘185’ 3rd: 370)**
* **Total Length (1st: 1480 2nd: 1480 3rd, 548)**

**Note: the packet above is only showing the total length for the last packet (548 + 20 bytes for the header)**