IP 实验报告

胡延伸 PB22050983

traceroute

MacOS 上直接在终端输入 “traceroute + 长度” 即可发送特定长度的 UDP 数据报。

实验步骤

1. 打开 Wireshark 进行包捕捉
2. 利用 traceroute 依次发送 3 段不同长度（56， 2000， 3500）的数据报，如下图:

```
(base) huyanshen@huyanshens-MacBook-Air ~ % traceroute gaia.cs.umass.edu 56
traceroute to gaia.cs.umass.edu (128.119.245.12), 64 hops max, 56 byte packets
 1  100.64.128.1 (100.64.128.1)  14.766 ms  4.954 ms  4.958 ms
 2  100.65.0.9 (100.65.0.9)  15.171 ms  4.255 ms  4.920 ms
 3  * 222.195.81.1 (222.195.81.1)  10.323 ms *
 4  202.38.64.58 (202.38.64.58)  5.282 ms
    202.38.64.60 (202.38.64.60)  5.394 ms
    202.38.64.58 (202.38.64.58)  6.210 ms
 5  210.45.224.252 (210.45.224.252)  6.401 ms  7.019 ms  9.107 ms
 6  101.4.115.13 (101.4.115.13)  12.181 ms  6.066 ms  5.584 ms
 7  101.4.115.185 (101.4.115.185)  7.495 ms  7.278 ms  6.837 ms
 8  100.64.61.1 (100.64.61.1)  5.103 ms  4.874 ms  7.053 ms
 9  101.4.112.61 (101.4.112.61)  13.181 ms  12.604 ms  13.003 ms
10  101.4.117.38 (101.4.117.38)  20.989 ms  21.885 ms  21.025 ms
11  * * *
12  * *
```

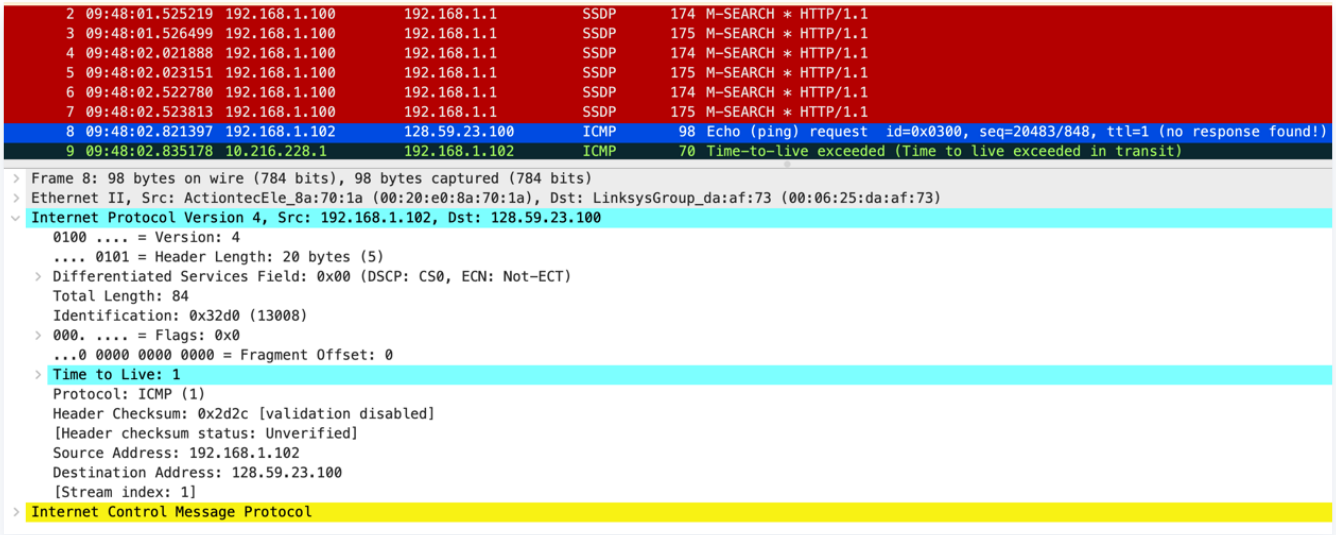
```
(base) huyanshen@huyanshens-MacBook-Air ~ % traceroute gaia.cs.umass.edu 2000
traceroute to gaia.cs.umass.edu (128.119.245.12), 64 hops max, 2000 byte packets
 1  100.64.128.1 (100.64.128.1)  13.252 ms  5.908 ms  5.779 ms
 2  100.65.0.9 (100.65.0.9)  4.963 ms *  10.889 ms
 3  222.195.81.1 (222.195.81.1)  5.346 ms  9.662 ms  13.773 ms
 4  202.38.64.60 (202.38.64.60)  5.061 ms  5.913 ms  4.318 ms
 5  210.45.224.252 (210.45.224.252)  4.970 ms  5.607 ms *
 6  * 101.4.115.13 (101.4.115.13)  10.975 ms  5.421 ms
 7  101.4.115.185 (101.4.115.185)  7.180 ms  7.107 ms  8.252 ms
 8  100.64.61.1 (100.64.61.1)  5.103 ms  5.947 ms  5.842 ms
 9  101.4.112.61 (101.4.112.61)  13.154 ms  12.744 ms  13.001 ms
10  101.4.117.38 (101.4.117.38)  20.967 ms  20.793 ms  20.694 ms
11  101.4.112.1 (101.4.112.1)  30.191 ms  30.691 ms  32.113 ms
```

```
[(base) huyanshen@huyanshens-MacBook-Air ~ % traceroute gaia.cs.umass.edu 3500   ]
traceroute to gaia.cs.umass.edu (128.119.245.12), 64 hops max, 3500 byte packets
 1  100.64.128.1 (100.64.128.1)  13.345 ms  5.209 ms  4.970 ms
 2  100.65.0.9 (100.65.0.9)  5.078 ms  6.834 ms  5.174 ms
 3  222.195.81.1 (222.195.81.1)  5.048 ms  4.978 ms *
 4  202.38.64.60 (202.38.64.60)  12.062 ms  5.074 ms  5.081 ms
 5  210.45.224.252 (210.45.224.252)  6.011 ms  6.342 ms *
 6  101.4.115.13 (101.4.115.13)  43.627 ms  6.320 ms  6.891 ms
 7  101.4.115.185 (101.4.115.185)  7.068 ms  6.993 ms  8.057 ms
 8  100.64.61.1 (100.64.61.1)  6.622 ms  5.214 ms  4.876 ms
 9  101.4.112.61 (101.4.112.61)  12.136 ms  12.702 ms  13.038 ms
10  101.4.117.38 (101.4.117.38)  21.046 ms  24.502 ms  21.264 ms
11  * 101.4.112.1 (101.4.112.1)  54.852 ms  37.856 ms
12  * * *
```

3. 停止 Wireshark 的包捕捉。由于本机网络捕获的包过于繁杂，所以采用官方提供的 trace 文件进行分析.

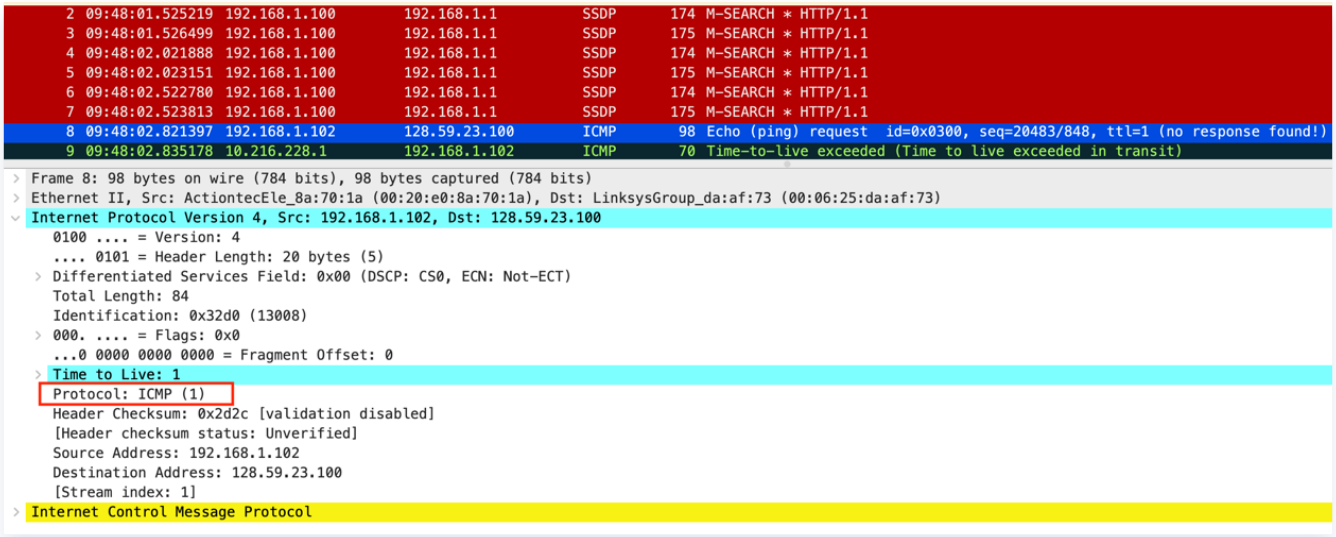
A look at the captured trace

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?



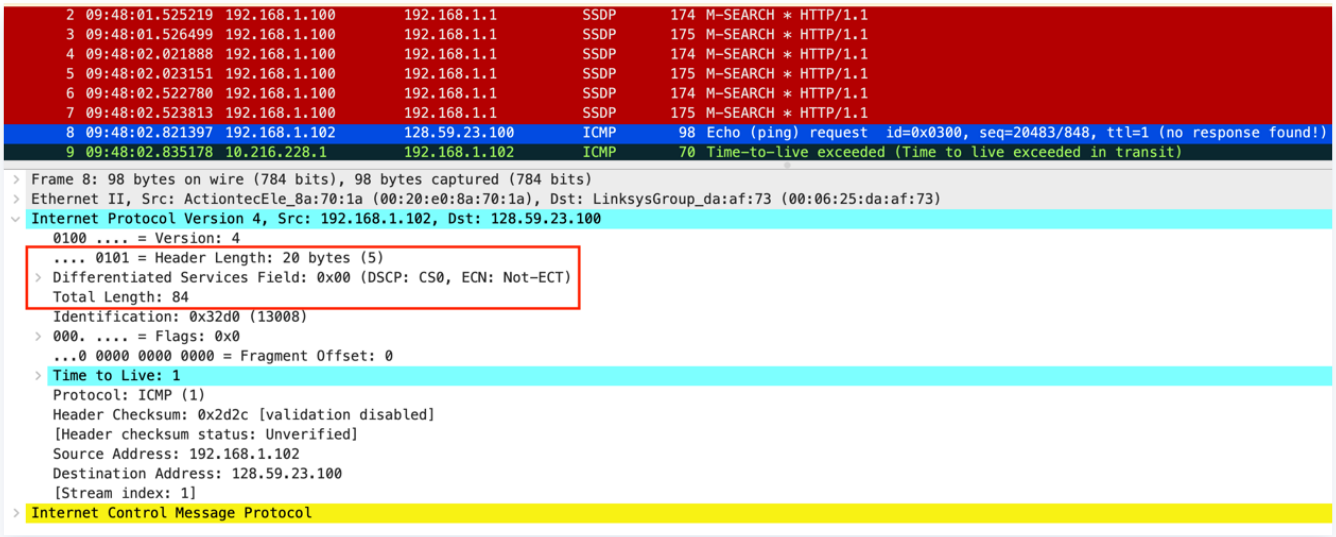
如上图，本地电脑的 IP 地址为: 192.168.1.102

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



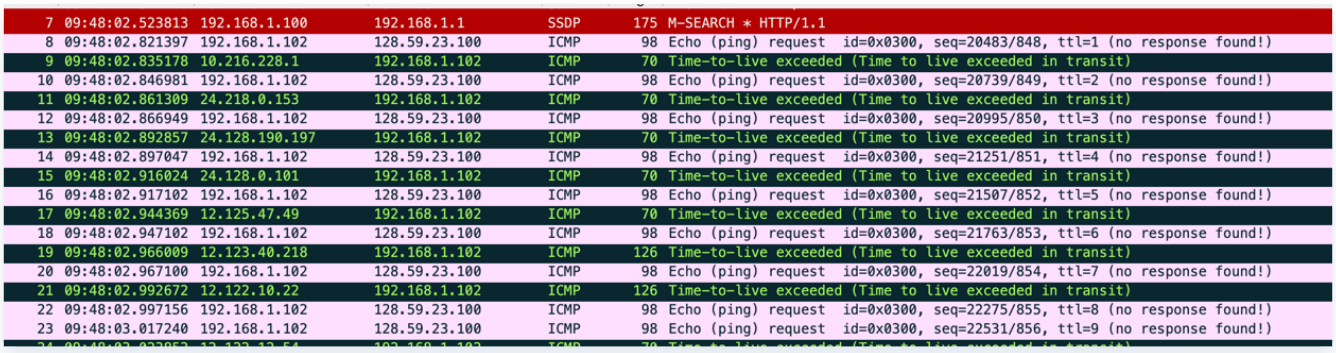
如上图，上层协议字段为1.

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.



如上图，头部大小为 20 bytes; 总大小为 84 bytes, 故有效负载大小为 64 bytes.

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

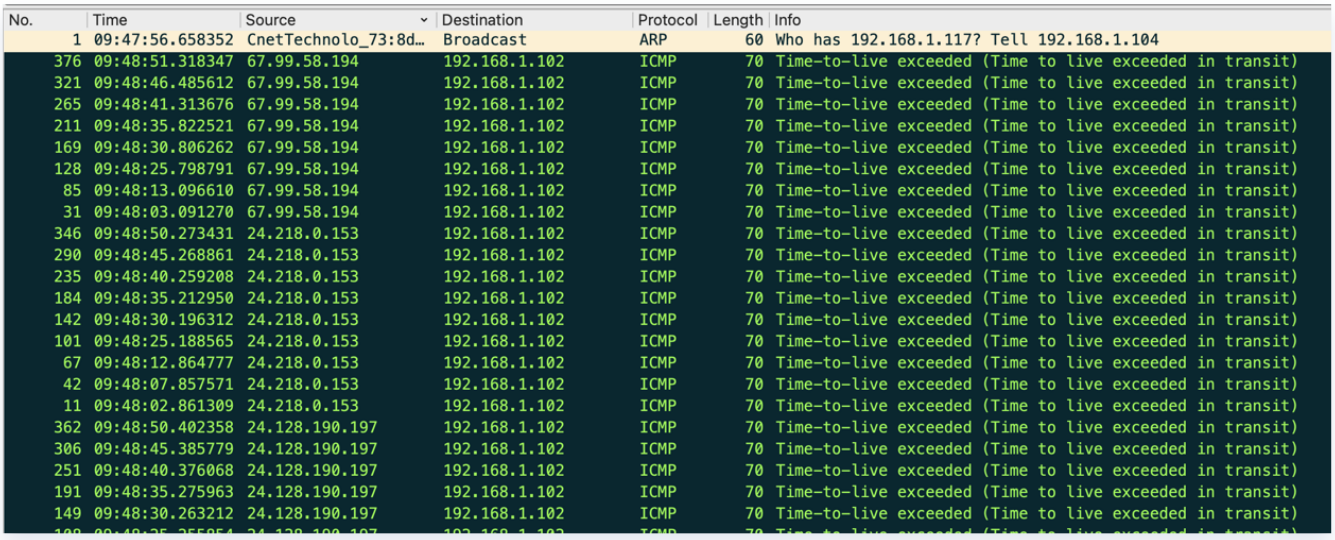


7	09:48:02.523813	192.168.1.100	192.168.1.1	SSDP	175	M-SEARCH * HTTP/1.1
8	09:48:02.821397	192.168.1.102	128.59.23.100	ICMP	98	Echo (ping) request id=0x0300, seq=20483/848, ttl=1 (no response found!)
9	09:48:02.835178	10.216.228.1	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
10	09:48:02.846981	192.168.1.102	128.59.23.100	ICMP	98	Echo (ping) request id=0x0300, seq=20739/849, ttl=2 (no response found!)
11	09:48:02.861309	24.218.0.153	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
12	09:48:02.866949	192.168.1.102	128.59.23.100	ICMP	98	Echo (ping) request id=0x0300, seq=20995/850, ttl=3 (no response found!)
13	09:48:02.892857	24.128.190.197	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
14	09:48:02.897047	192.168.1.102	128.59.23.100	ICMP	98	Echo (ping) request id=0x0300, seq=21251/851, ttl=4 (no response found!)
15	09:48:02.916024	24.128.0.101	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
16	09:48:02.917102	192.168.1.102	128.59.23.100	ICMP	98	Echo (ping) request id=0x0300, seq=21507/852, ttl=5 (no response found!)
17	09:48:02.944369	12.125.47.49	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
18	09:48:02.947102	192.168.1.102	128.59.23.100	ICMP	98	Echo (ping) request id=0x0300, seq=21763/853, ttl=6 (no response found!)
19	09:48:02.966009	12.123.40.218	192.168.1.102	ICMP	126	Time-to-live exceeded (Time to live exceeded in transit)
20	09:48:02.967100	192.168.1.102	128.59.23.100	ICMP	98	Echo (ping) request id=0x0300, seq=22019/854, ttl=7 (no response found!)
21	09:48:02.992672	12.122.10.22	192.168.1.102	ICMP	126	Time-to-live exceeded (Time to live exceeded in transit)
22	09:48:02.997156	192.168.1.102	128.59.23.100	ICMP	98	Echo (ping) request id=0x0300, seq=22275/855, ttl=8 (no response found!)
23	09:48:03.017240	192.168.1.102	128.59.23.100	ICMP	98	Echo (ping) request id=0x0300, seq=22531/856, ttl=9 (no response found!)

没有，因为如果数据报分段，应该会出现多个 TTL = 1 的包，但图中没有。

Sort the traced packets according to IP source address by clicking on the Sourcecolumn header; a small downward pointing arrow should appear next to the word Source. If the arrow points up, click on the Source column header again.

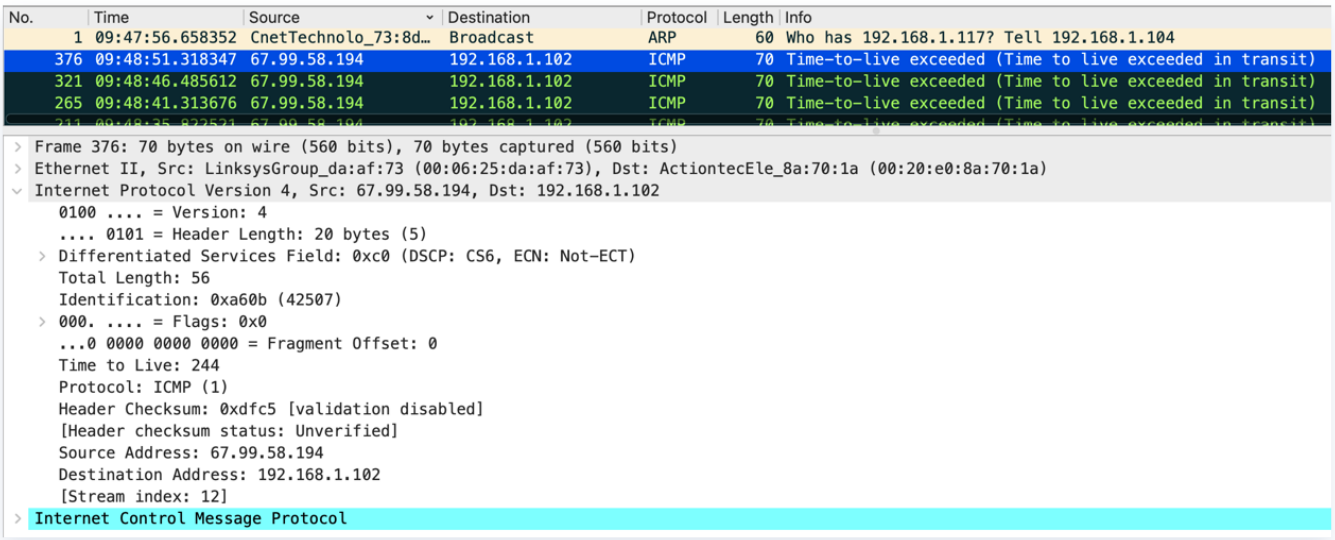
如下图所示，按照 IP 地址降序排列的界面:



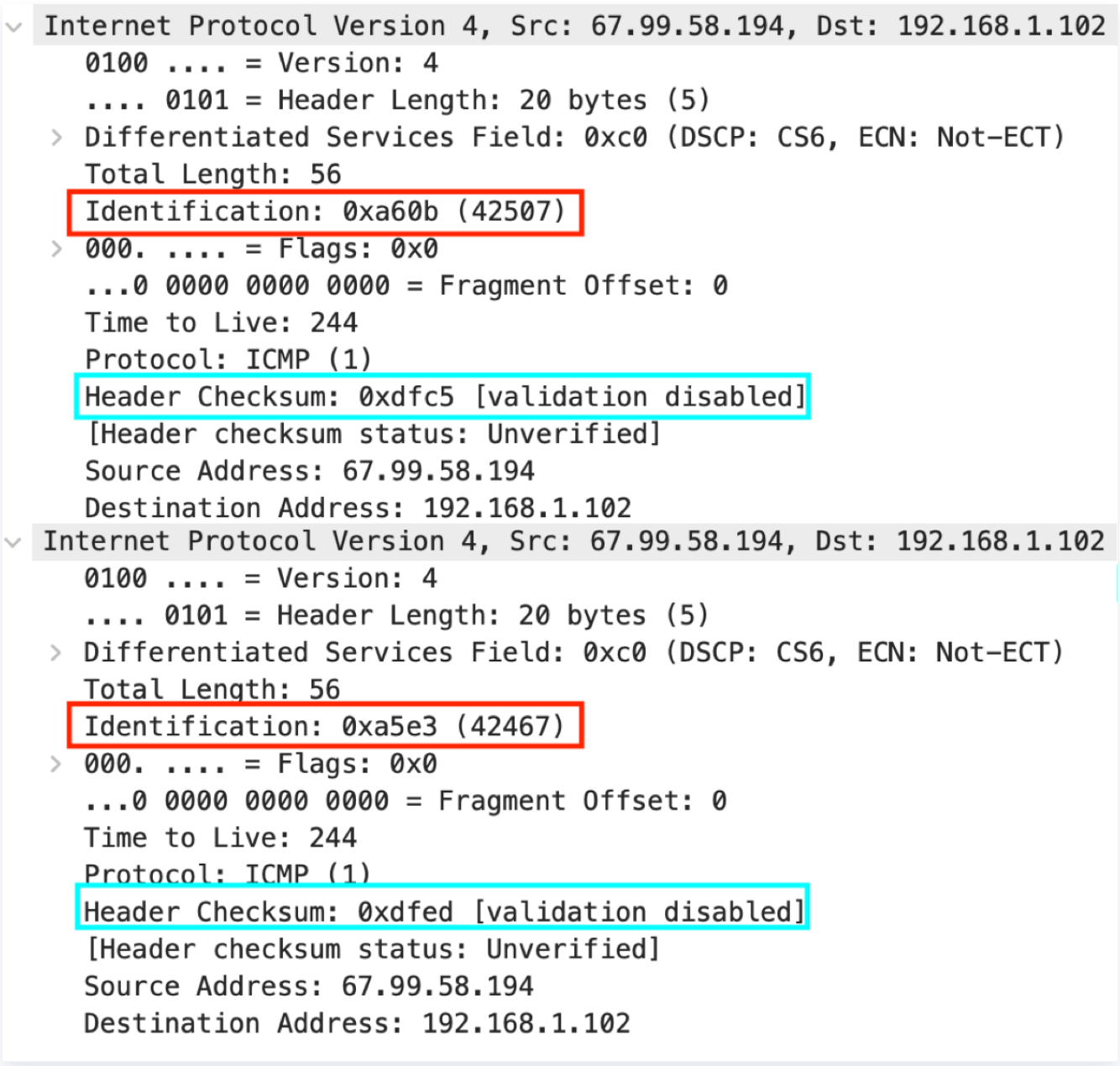
No.	Time	Source	Destination	Protocol	Length	Info
1	09:47:56.658352	CnetTechnolo_73:8d…	Broadcast	ARP	60	Who has 192.168.1.117? Tell 192.168.1.104
376	09:48:51.318347	67.99.58.194	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
321	09:48:46.485612	67.99.58.194	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
265	09:48:41.313676	67.99.58.194	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
211	09:48:35.822521	67.99.58.194	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
169	09:48:30.806262	67.99.58.194	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
128	09:48:25.798791	67.99.58.194	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
85	09:48:13.096610	67.99.58.194	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
31	09:48:03.091270	67.99.58.194	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
346	09:48:50.273431	24.218.0.153	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
290	09:48:45.268861	24.218.0.153	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
235	09:48:40.259208	24.218.0.153	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
184	09:48:35.212950	24.218.0.153	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
142	09:48:30.196312	24.218.0.153	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
101	09:48:25.188565	24.218.0.153	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
67	09:48:12.864777	24.218.0.153	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
42	09:48:07.857571	24.218.0.153	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
11	09:48:02.861309	24.218.0.153	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
362	09:48:50.402358	24.128.190.197	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
306	09:48:45.385779	24.128.190.197	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
251	09:48:40.376068	24.128.190.197	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
191	09:48:35.275963	24.128.190.197	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
149	09:48:30.263212	24.128.190.197	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)

Select the first ICMP Echo Request message sent by your computer, and expand the Internet Protocol portion in the “details of selected packet header” window.

如下图:



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?



如上图，Identifaicaton, 和 Header Checksum 一直在改变。

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

保持不变：

1. Explicit Congestion Notification，ECN：允许在不丢弃报文的同时通知对方网络拥塞的发生。
2. Total Length：占 16 位字段，定义了报文总长，包含首部和数据，单位为字节。这个字段的最小值是 20（0 字节数据），最大值是65535。
3. Fragment Offset：这个13位字段指明了每个分片相对于原始报文开头的偏移量，以8字节作单位。
4. Source：报文的发送端；

5. Destination：报文的接收端；
6. Option：附加的首部字段可能跟在目的地址之后；

必须保持不变：

1. 版本（Version）：占 4 bit，通信双方使用的版本必须一致，对于 IPv4 字段的值是4.
2. 首部长度（Internet Header Length， IHL）：占 4 bit，首部长度说明首部有多少 32 位字（4字节）.
3. 区分服务（Differentiated Services，DS）：占 6 bit，只有在使用区分服务时，这个字段才起作用，在一般的情况下都不使用这个字段.

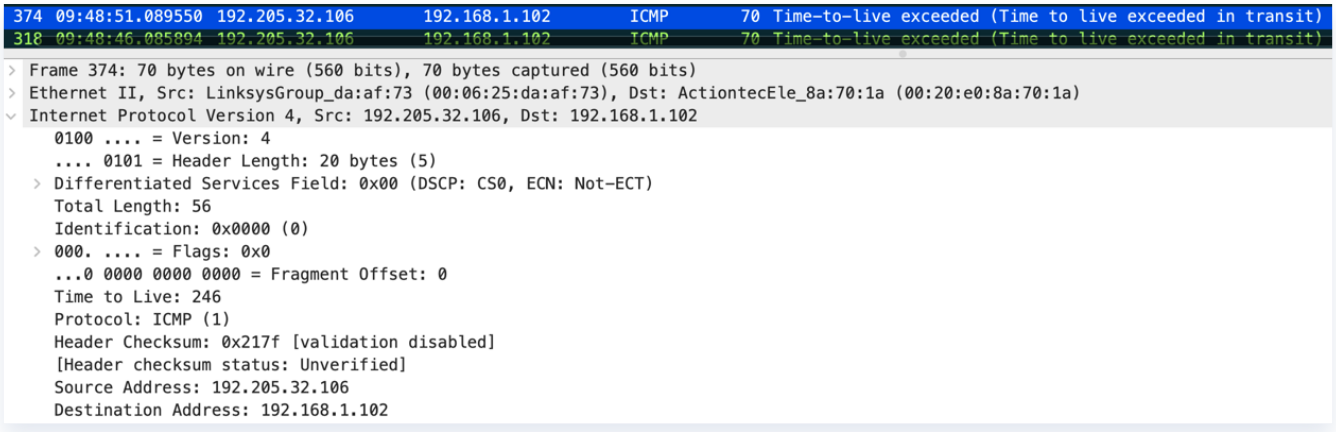
必须改变:

1. Identification：占 16 位，用来唯一地标识一个报文的所有分片；
2. Time To Live，TTL：占 8 位，避免报文在互联网中永远存在。
3. Header Checksum：占 16 位，检验和字段只对首部查错，在每一跳，路由器都要重新计算出的首部检验和并与此字段进行比对，如果不一致，此报文将会被丢弃；
4. 数据

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

这用于唯一地标识一条报文的所有fragment，因此如果报文不同，则必须更改此值以便它能够唯一地标识报文。

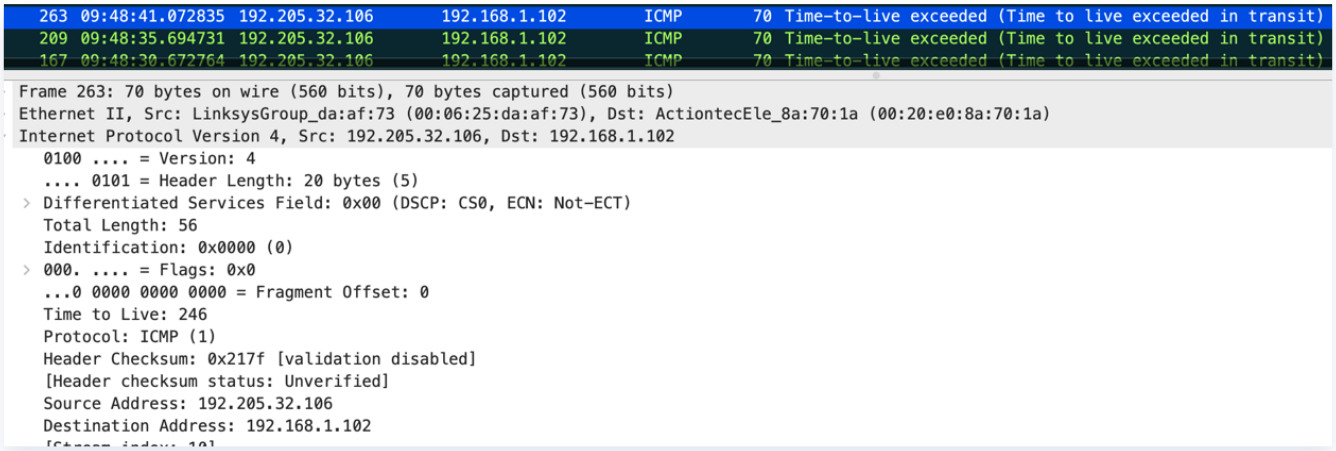
8. What is the value in the Identification field and the TTL field?

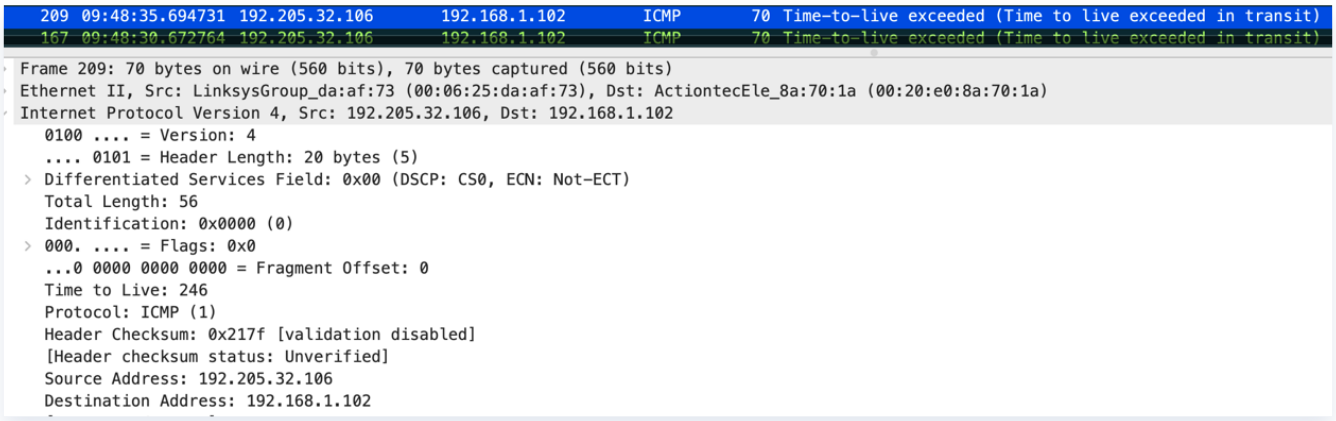


如上图, ID 为42507， TTL = 244

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?

如下两张图：

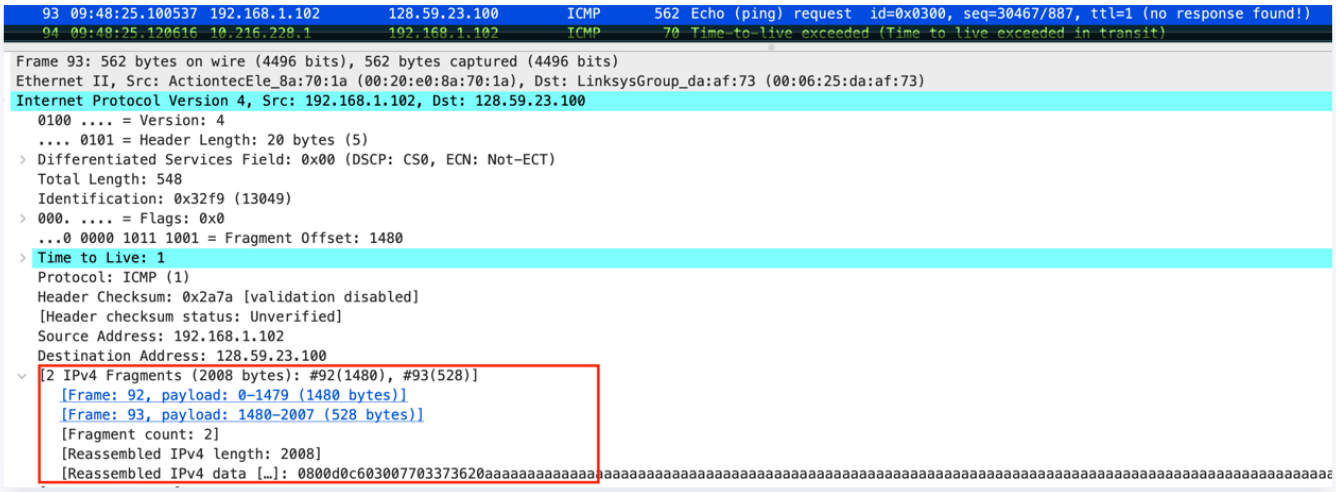




均保持不变；因为它们都是属于同一个报文的不同分片.

Fragmentation

D. 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?

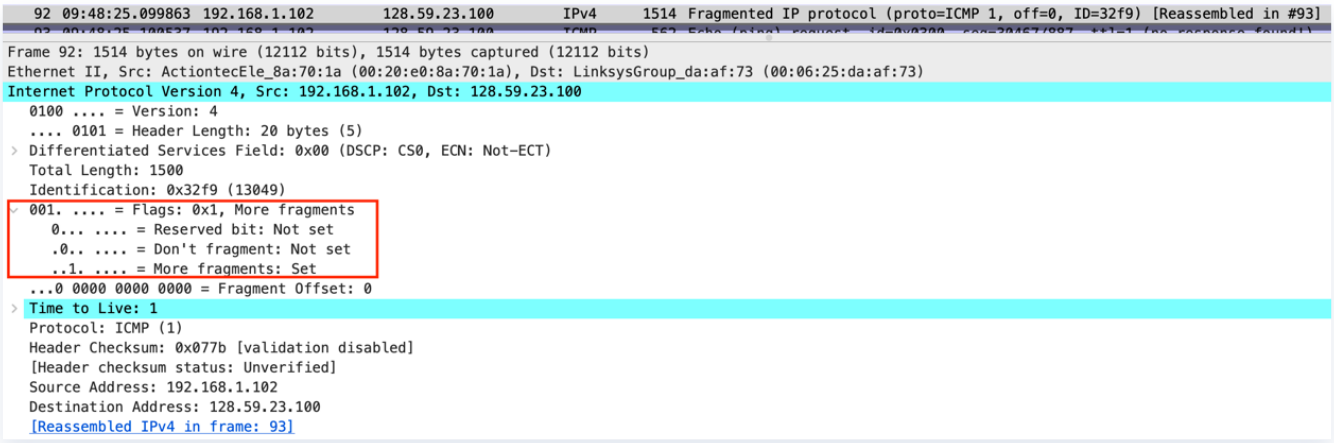


如上图，分成了两个分片。

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?

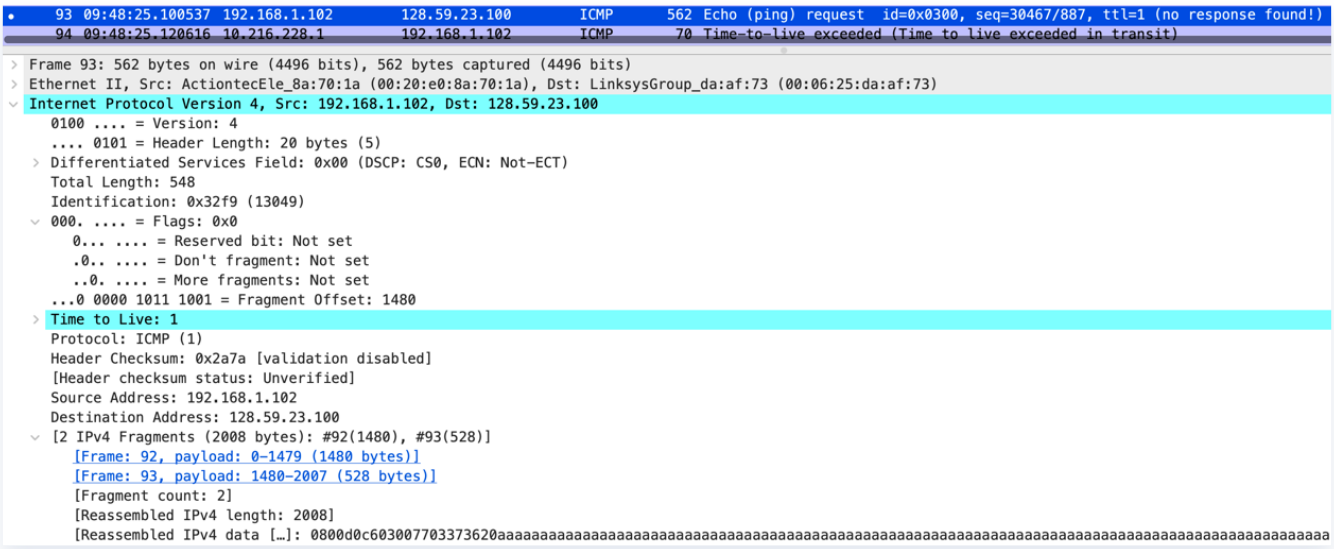
打印的信息见附件.

如下图:



More fragments字段为 1 表示 Set，即该数据包被分片。通过fragment offset = 0判断这是第一个片段;分片长度为 1480 bytes。

2. 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?
   打印信息见附件。
   如下图:

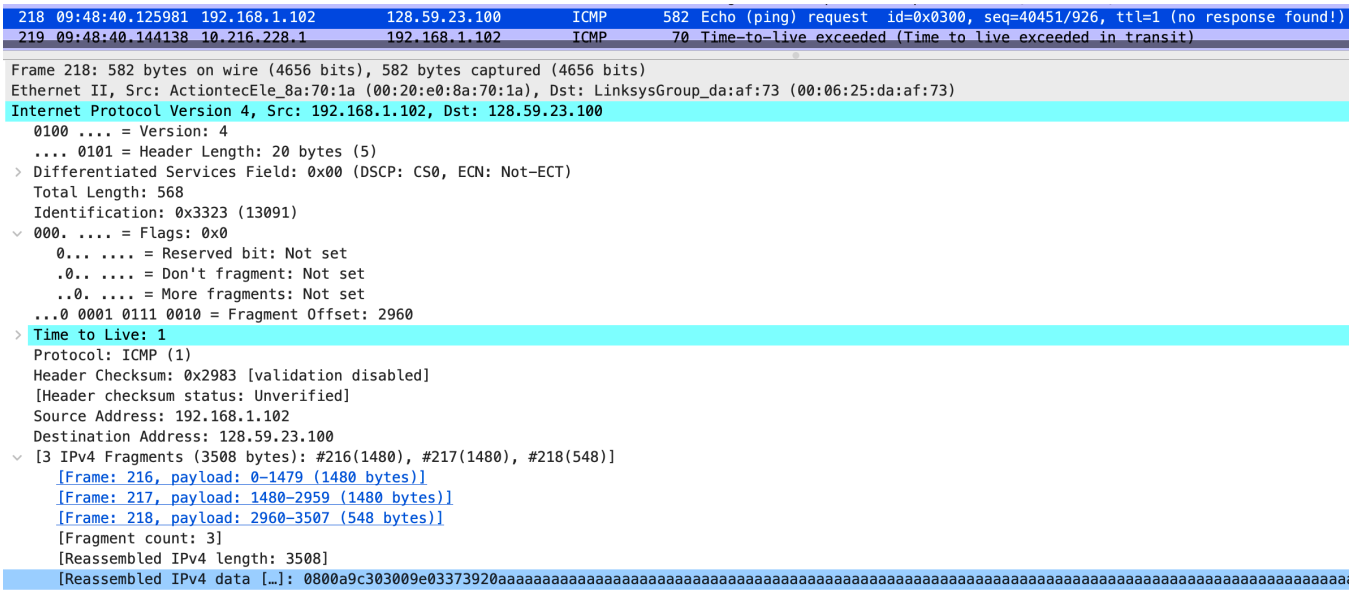


offset = 1480 表明这不是第一个分片；More fragments = Not set 表明没有更多的分片。

3. What fields change in the IP header between the first and second fragment?
   全长（Total Length）、标志 （Flags）和分片偏移 （Fragment Offset）

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.

4. How many fragments were created from the original datagram?
   如下图:



一共分成了 3 个分片

5. What fields change in the IP header among the fragments?
   和数据报大小为2000情况类似，全长（Total Length）、标志 （Flags）和分片偏移（Fragment Offset）会改变。