IP LAB

In this lab, we will investigate the IP protocol, focusing on the IP datagram. We will do so by analyzing a trace of IP datagrams sent and received by an execution of the `traceroute` program (the `traceroute` program itself is explored in more detail in the Wireshark ICMP lab). We will investigate the various fields in the IP datagram, and study IP fragmentation in detail.

1. Capturing packets from an execution of traceroute

In order to generate a trace of IP datagrams for this lab, we'll use the traceroute program to send datagrams of different sizes towards some destination, X. Recall that traceroute operates by first sending one or more datagrams with the time-to-live (TTL) field in the IP header set to 1; it then sends a series of one or more datagrams towards the same destination with a TTL value of 2; it then sends a series of datagrams towards the same destination with a TTL value of 3; and so on. Recall that a router must decrement the TTL in each received datagram by 1 (actually, RFC 791 says that the router must decrement the TTL by at least one). If the TTL reaches 0, the router returns an ICMP message (type 11 – TTL-exceeded) to the sending host. As a result of this behavior, a datagram with a TTL of 1 (sent by the host executing traceroute) will cause the router one hop away from the sender to send an ICMP TTL-exceeded message back to the sender; the datagram sent with a TTL of 2 will cause the router two hops away to send an ICMP message back to the sender; the datagram sent with a TTL of 3 will cause the router three hops away to send an ICMP message back to the sender; and so on. In this manner, the host executing traceroute can learn the identities of the routers between itself and destination X by looking at the source IP addresses in the datagrams containing the ICMP TTL-exceeded messages. We will want to run traceroute and have it sent datagrams of various lengths.

Windows.

The tracert program provided with Windows does not allow one to change the size of the ICMP echo request (ping) message sent by the tracert program. A nicer Windows traceroute program is mturoute, available in free version at https://www.elifulkerson.com/projects/mturoute.php. Download and run mturoute in the command prompt and test it out by performing a few traceroutes to your favorite sites (Download mturoute-src-zip and mturoute.exe) (ex for reference: C:\users\Sanika\Downloads>mturoute). The size of the ICMP echo request message can be explicitly set in mturoute by using the command: 'mturoute.exe -t -f -m < length > google.com'

Linux/Unix/MacOS.

With the Unix/MacOS traceroute command, the size of the UDP datagram sent towards the destination can be explicitly set by indicating the number of bytes in the datagram; this value is entered in the traceroute command line immediately after the name or address of the destination. For example, to send traceroute datagrams of < length > bytes towards google.com, the command would be: 'traceroute google.com < length >'

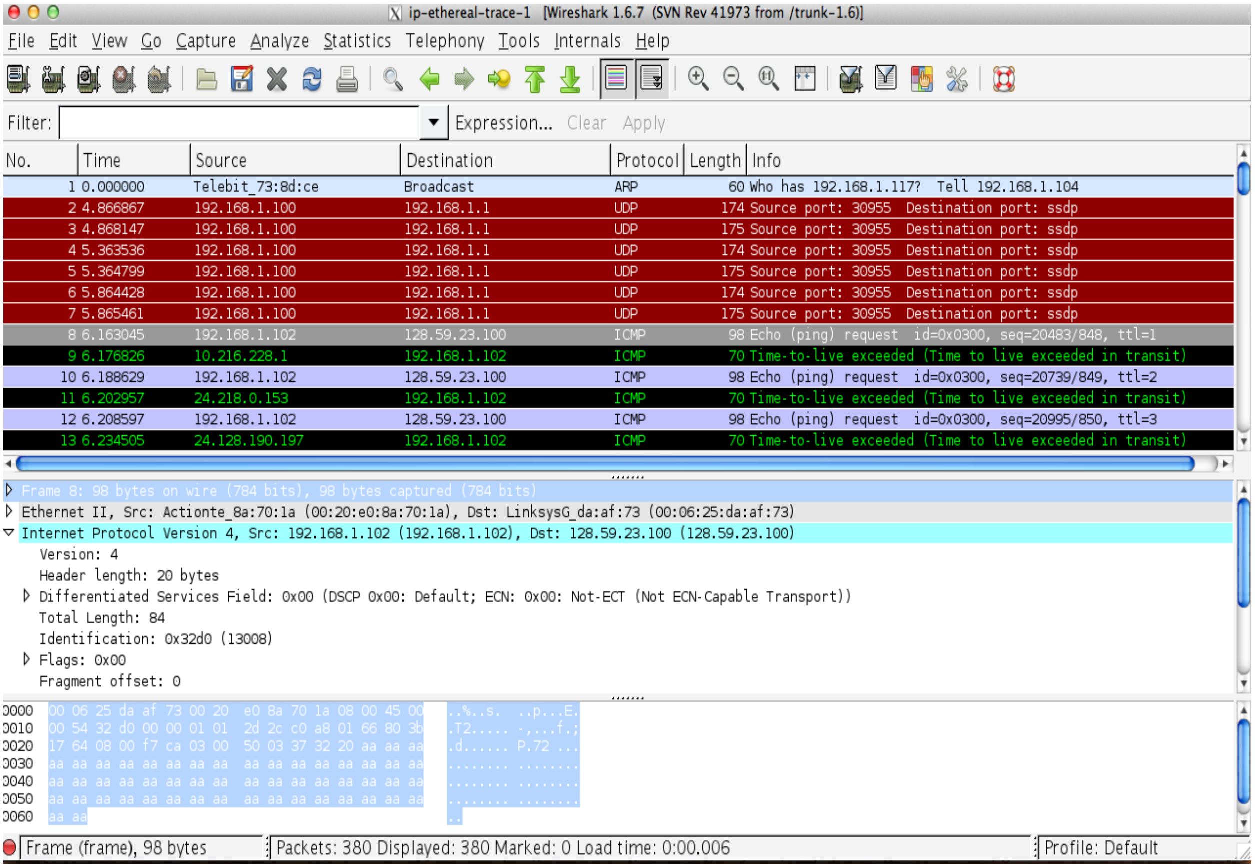
Do the following:

- Start up Wireshark and begin packet capture (Capture->Start) and then press OK on the Wireshark Packet Capture Options screen (we will not need to select any options here). Apply the filter ‘ip.proto == 1 or ip.proto == 17’ to capture only the traceroute-related packets.
- If you are using a Windows platform, enter three mturoute commands, first with a length of 56 bytes, second with a length of 2000 bytes, and third with a length of 3500 bytes. Stop Wireshark tracing (Reference ex for length 3500: ‘mturoute.exe -t -f -m 3500 google.com’)
- If you are using a Linux or Mac platform, enter three traceroute commands, one with a length of 56 bytes, one with a length of 2000 bytes, and one with a length of 3500 bytes. Stop Wireshark tracing (Reference ex for length 3500: ‘traceroute google.com 3500’)
- Save the capture once traceroute finishes, so that you easily locate the packets you need to answer the questions.

2. A look at the captured trace

In your trace, you should be able to see the series of ICMP Echo Request (in the case of Windows machine) or the UDP segment (in the case of Unix) sent by your computer to the target IP address and the ICMP TTL-exceeded messages returned to your computer by the intermediate routers. **When answering each question, use File->Print, choose Selected packet only. Annotate the output so that it is clear where in the output you are getting the information for your answer. Also, attach the screenshot of the command line output of the above commands (Similar to Figure 1, Figure 2 and Figure 3 below).**

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.


ip-ethereal-trace-1 [Wireshark 1.6.7 (SVN Rev 41973 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply

```
No.  Time      Source           Destination     Protocol Length Info
   1 0.000000  Telebit_73:8d:ce Broadcast       ARP         60 Who has 192.168.1.117?  Tell 192.168.1.104
   2 4.866867  192.168.1.100    192.168.1.1     UDP        174 Source port: 30955  Destination port: ssdp
   3 4.868147  192.168.1.100    192.168.1.1     UDP        175 Source port: 30955  Destination port: ssdp
   4 5.363536  192.168.1.100    192.168.1.1     UDP        174 Source port: 30955  Destination port: ssdp
   5 5.364799  192.168.1.100    192.168.1.1     UDP        175 Source port: 30955  Destination port: ssdp
   6 5.864428  192.168.1.100    192.168.1.1     UDP        174 Source port: 30955  Destination port: ssdp
   7 5.865461  192.168.1.100    192.168.1.1     UDP        175 Source port: 30955  Destination port: ssdp
   8 6.163045  192.168.1.102    128.59.23.100   ICMP        98 Echo (ping) request  id=0x0300, seq=20483/848, ttl=1
   9 6.176826  10.216.228.1     192.168.1.102   ICMP        70 Time-to-live exceeded (Time to live exceeded in transit)
  10 6.188629  192.168.1.102    128.59.23.100   ICMP        98 Echo (ping) request  id=0x0300, seq=20739/849, ttl=2
  11 6.202957  24.218.0.153     192.168.1.102   ICMP        70 Time-to-live exceeded (Time to live exceeded in transit)
  12 6.208597  192.168.1.102    128.59.23.100   ICMP        98 Echo (ping) request  id=0x0300, seq=20995/850, ttl=3
  13 6.234505  24.128.190.197   192.168.1.102   ICMP        70 Time-to-live exceeded (Time to live exceeded in transit)
```

```
▷ Frame 8: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
▷ Ethernet II, Src: Actionte_8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG_da:af:73 (00:06:25:da:af:73)
▽ Internet Protocol Version 4, Src: 192.168.1.102 (192.168.1.102), Dst: 128.59.23.100 (128.59.23.100)
    Version: 4
    Header length: 20 bytes
  ▷ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
    Total Length: 84
    Identification: 0x32d0 (13008)
  ▷ Flags: 0x00
    Fragment offset: 0
```

```
0000  00 06 25 da af 73 00 20  e0 8a 70 1a 08 00 45 00   ..%..s.  ..p...E.
0010  00 54 32 d0 00 00 01 01  2d 2c c0 a8 01 66 80 3b   .T2..... -,...f.;
0020  17 64 08 00 f7 ca 03 00  50 03 37 32 20 aa aa aa   .d...... P.72 ...
0030  aa aa aa aa aa aa aa aa  aa aa aa aa aa aa aa aa   ........ ........
0040  aa aa aa aa aa aa aa aa  aa aa aa aa aa aa aa aa   ........ ........
0050  aa aa aa aa aa aa aa aa  aa aa aa aa aa aa aa aa   ........ ........
0060  aa aa                                              ..
```

```
Frame (frame), 98 bytes | Packets: 380 Displayed: 380 Marked: 0 Load time: 0:00.006 | Profile: Default
```

Figure 1 showing first ICMP echo request message and packet details of internet protocol part.

1. What is the IP address of your computer? How did you identify that?

2. Within the IP packet header, what is the value in the upper layer protocol field? What is the purpose of including it in the IP header?

3. Within the IP packet header, what is the value of the Flags field? Has this IP datagram been fragmented? Explain how you determined whether or not the datagram has been fragmented.

Next, sort the traced packets according to IP destination address by clicking on the Destination column header; a small arrow should appear next to the word Destination. Select the first ICMP Echo Request message sent by your computer to the target destination address and expand the Internet Protocol portion in the "details of selected packet header" window. In the "listing of captured packets" window, you should see all the subsequent ICMP messages (perhaps with additional interspersed packets sent by other protocols running on your computer) below this first ICMP. Use the down arrow to move through the ICMP messages sent by your computer.

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

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

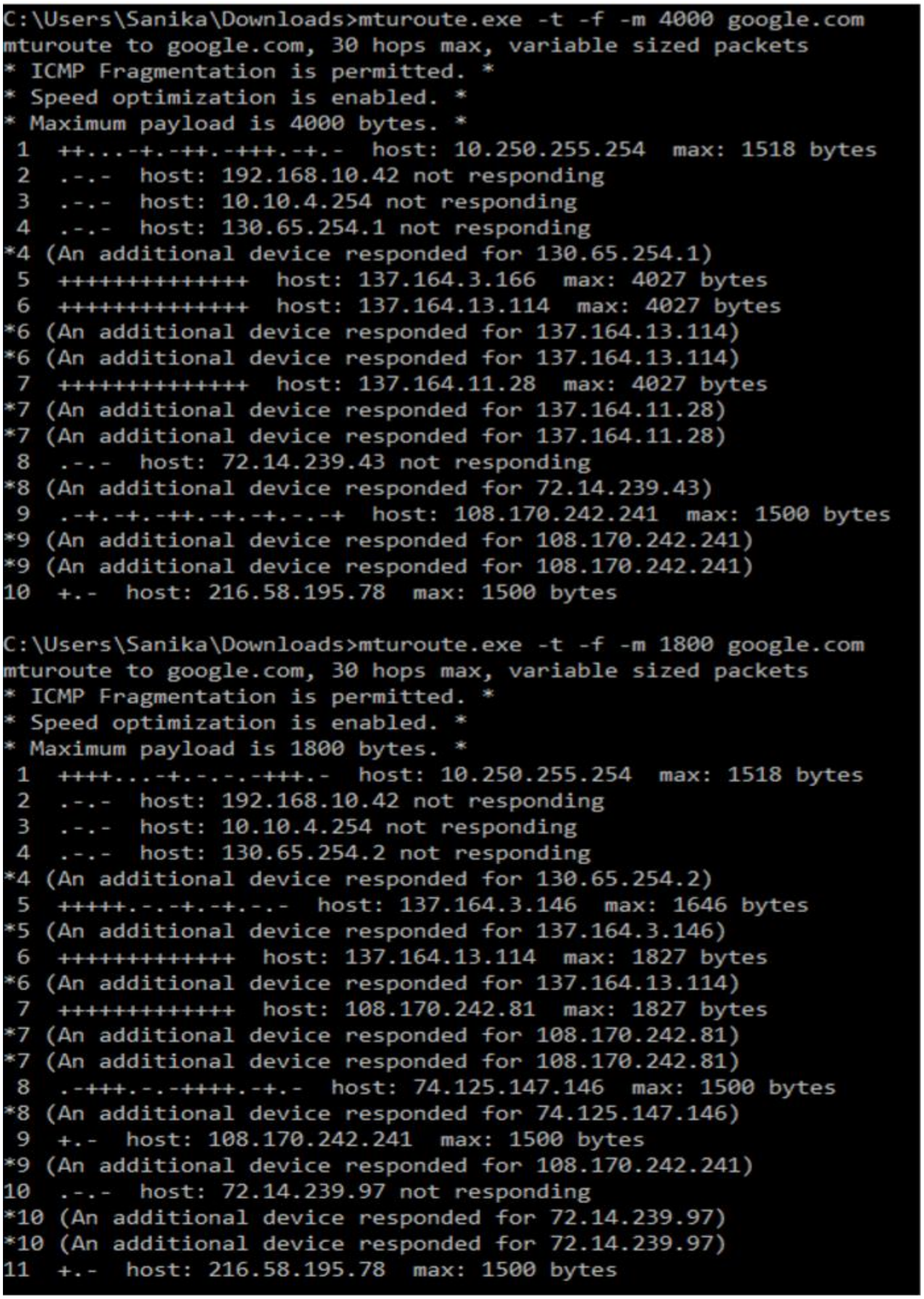
6. Describe the pattern you see in the values in the Identification field of the IP datagram.

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

7. What is the value in the Identification field, length field and the TTL field?

8. Do these values remain unchanged for all the ICMP TTL-exceeded replies sent to your computer by the nearest (first hop) router? Do these values change for packets sent by other intermediate routers?

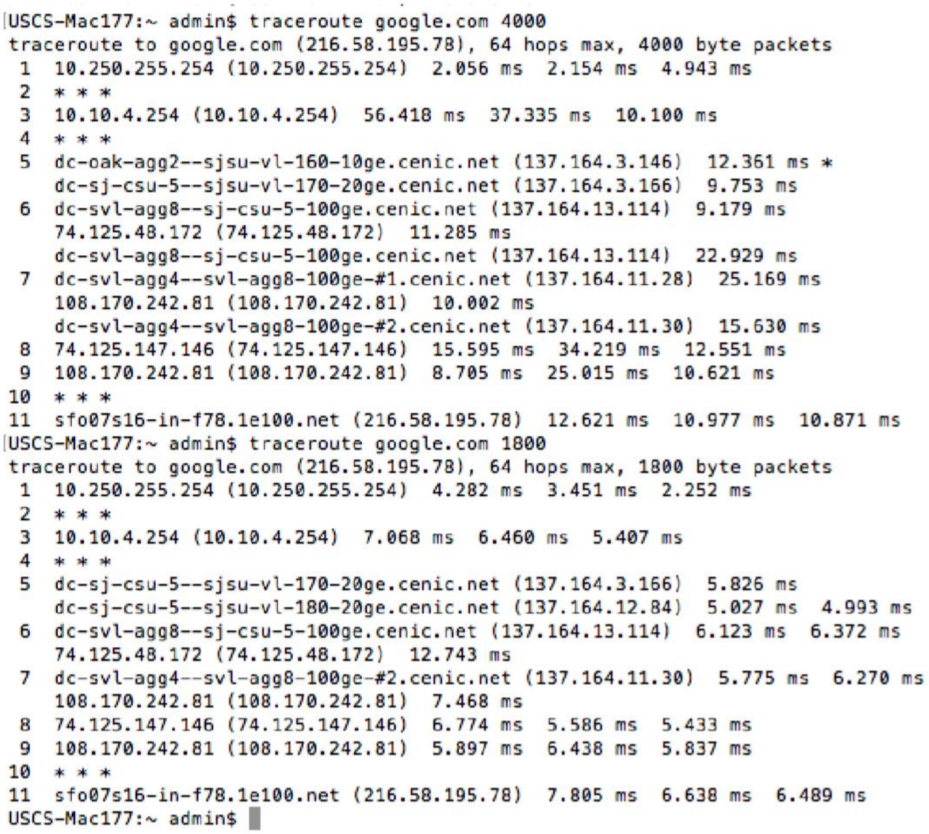
Fragmentation



```
C:\Users\Sanika\Downloads>mturoute.exe -t -f -m 4000 google.com
mturoute to google.com, 30 hops max, variable sized packets
* ICMP Fragmentation is permitted. *
* Speed optimization is enabled. *
* Maximum payload is 4000 bytes. *
 1  ++...-+.-++.-+++.-+.-  host: 10.250.255.254  max: 1518 bytes
 2  .-.-  host: 192.168.10.42 not responding
 3  .-.-  host: 10.10.4.254 not responding
 4  .-.-  host: 130.65.254.1 not responding
*4 (An additional device responded for 130.65.254.1)
 5  ++++++++++++++  host: 137.164.3.166  max: 4027 bytes
 6  ++++++++++++++  host: 137.164.13.114  max: 4027 bytes
*6 (An additional device responded for 137.164.13.114)
*6 (An additional device responded for 137.164.13.114)
 7  ++++++++++++++  host: 137.164.11.28  max: 4027 bytes
*7 (An additional device responded for 137.164.11.28)
*7 (An additional device responded for 137.164.11.28)
 8  .-.-  host: 72.14.239.43 not responding
*8 (An additional device responded for 72.14.239.43)
 9  .-+.-+.-++.-+.-+.-.-+  host: 108.170.242.241  max: 1500 bytes
*9 (An additional device responded for 108.170.242.241)
*9 (An additional device responded for 108.170.242.241)
10  +.-  host: 216.58.195.78  max: 1500 bytes

C:\Users\Sanika\Downloads>mturoute.exe -t -f -m 1800 google.com
mturoute to google.com, 30 hops max, variable sized packets
* ICMP Fragmentation is permitted. *
* Speed optimization is enabled. *
* Maximum payload is 1800 bytes. *
 1  ++++...-+.-.-.-+++.-  host: 10.250.255.254  max: 1518 bytes
 2  .-.-  host: 192.168.10.42 not responding
 3  .-.-  host: 10.10.4.254 not responding
 4  .-.-  host: 130.65.254.2 not responding
*4 (An additional device responded for 130.65.254.2)
 5  +++++.-.-+.-+.-.-  host: 137.164.3.146  max: 1646 bytes
*5 (An additional device responded for 137.164.3.146)
 6  +++++++++++++  host: 137.164.13.114  max: 1827 bytes
*6 (An additional device responded for 137.164.13.114)
 7  +++++++++++++  host: 108.170.242.81  max: 1827 bytes
*7 (An additional device responded for 108.170.242.81)
*7 (An additional device responded for 108.170.242.81)
 8  .-+++.-.-++++.-+.-  host: 74.125.147.146  max: 1500 bytes
*8 (An additional device responded for 74.125.147.146)
 9  +.-  host: 108.170.242.241  max: 1500 bytes
*9 (An additional device responded for 108.170.242.241)
10  .-.-  host: 72.14.239.97 not responding
*10 (An additional device responded for 72.14.239.97)
*10 (An additional device responded for 72.14.239.97)
11  +.-  host: 216.58.195.78  max: 1500 bytes
```

Figure 2 Windows Command prompt window showing the commands for change in packet size (Above examples use 4000 and 1800 packet sizes, change your commands according to the packet size requirement in the question)



```
USCS-Mac177:~ admin$ traceroute google.com 4000
traceroute to google.com (216.58.195.78), 64 hops max, 4000 byte packets
 1  10.250.255.254 (10.250.255.254)  2.056 ms  2.154 ms  4.943 ms
 2  * * *
 3  10.10.4.254 (10.10.4.254)  56.418 ms  37.335 ms  10.100 ms
 4  * * *
 5  dc-oak-agg2--sjsu-vl-160-10ge.cenic.net (137.164.3.146)  12.361 ms *
    dc-sj-csu-5--sjsu-vl-170-20ge.cenic.net (137.164.3.166)  9.753 ms
 6  dc-svl-agg8--sj-csu-5-100ge.cenic.net (137.164.13.114)  9.179 ms
    74.125.48.172 (74.125.48.172)  11.285 ms
    dc-svl-agg8--sj-csu-5-100ge.cenic.net (137.164.13.114)  22.929 ms
 7  dc-svl-agg4--svl-agg8-100ge-#1.cenic.net (137.164.11.28)  25.169 ms
    108.170.242.81 (108.170.242.81)  10.002 ms
    dc-svl-agg4--svl-agg8-100ge-#2.cenic.net (137.164.11.30)  15.630 ms
 8  74.125.147.146 (74.125.147.146)  15.595 ms  34.219 ms  12.551 ms
 9  108.170.242.81 (108.170.242.81)  8.705 ms  25.015 ms  10.621 ms
10  * * *
11  sfo07s16-in-f78.1e100.net (216.58.195.78)  12.621 ms  10.977 ms  10.871 ms
USCS-Mac177:~ admin$ traceroute google.com 1800
traceroute to google.com (216.58.195.78), 64 hops max, 1800 byte packets
 1  10.250.255.254 (10.250.255.254)  4.282 ms  3.451 ms  2.252 ms
 2  * * *
 3  10.10.4.254 (10.10.4.254)  7.068 ms  6.460 ms  5.407 ms
 4  * * *
 5  dc-sj-csu-5--sjsu-vl-170-20ge.cenic.net (137.164.3.166)  5.826 ms
    dc-sj-csu-5--sjsu-vl-180-20ge.cenic.net (137.164.12.84)  5.027 ms  4.993 ms
 6  dc-svl-agg8--sj-csu-5-100ge.cenic.net (137.164.13.114)  6.123 ms  6.372 ms
    74.125.48.172 (74.125.48.172)  12.743 ms
 7  dc-svl-agg4--svl-agg8-100ge-#2.cenic.net (137.164.11.30)  5.775 ms  6.270 ms
    108.170.242.81 (108.170.242.81)  7.468 ms
 8  74.125.147.146 (74.125.147.146)  6.774 ms  5.586 ms  5.433 ms
 9  108.170.242.81 (108.170.242.81)  5.897 ms  6.438 ms  5.837 ms
10  * * *
11  sfo07s16-in-f78.1e100.net (216.58.195.78)  7.805 ms  6.638 ms  6.489 ms
USCS-Mac177:~ admin$ 
```

Figure 3 MacOS Terminal window showing the commands you should enter for change in packet size (Above examples use 4000 and 1800 packet sizes, change your commands according to the packet size requirement in the question)

Sort the packet listing according to time again by clicking on the Time column. Find the first ICMP Echo Request message that was sent by your computer after you changed the Packet Size to be 2000.

Note for Windows users - mturoute will send a number of packets each of different size up to the max size mentioned in the command. As seen in Figure 1, packets with max 4027 bytes were only sent to hosts on lines numbered 5, 6 and 7 and packets with max 1827 were only sent to hosts on lines numbered 6 and 7. It might vary in your case. Try to use packets sent to these hosts while answering below questions. Try to spot the ICMP Echo request with the max packet size sent from your machine to the particular host. In any case, be sure to submit an annotation of the exact packet(s) you used for answering and also your pcap along with your answers.

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

10. 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?

11. Which fields in the IP header remain constant between fragments of the same packet?

Now find the first ICMP Echo Request message that was sent by your computer after you are changing the Packet Size to be 3500.

12. How many fragments was the original datagram fragmented into?

13. What fields change in the IP header among the fragments? Why?