ECS 152A - Project 1 - ICMP Wireshark Lab

Han Nguyen (917278789) & Blake McMurray (999162729)

1/17/20

1 ICMP and Ping

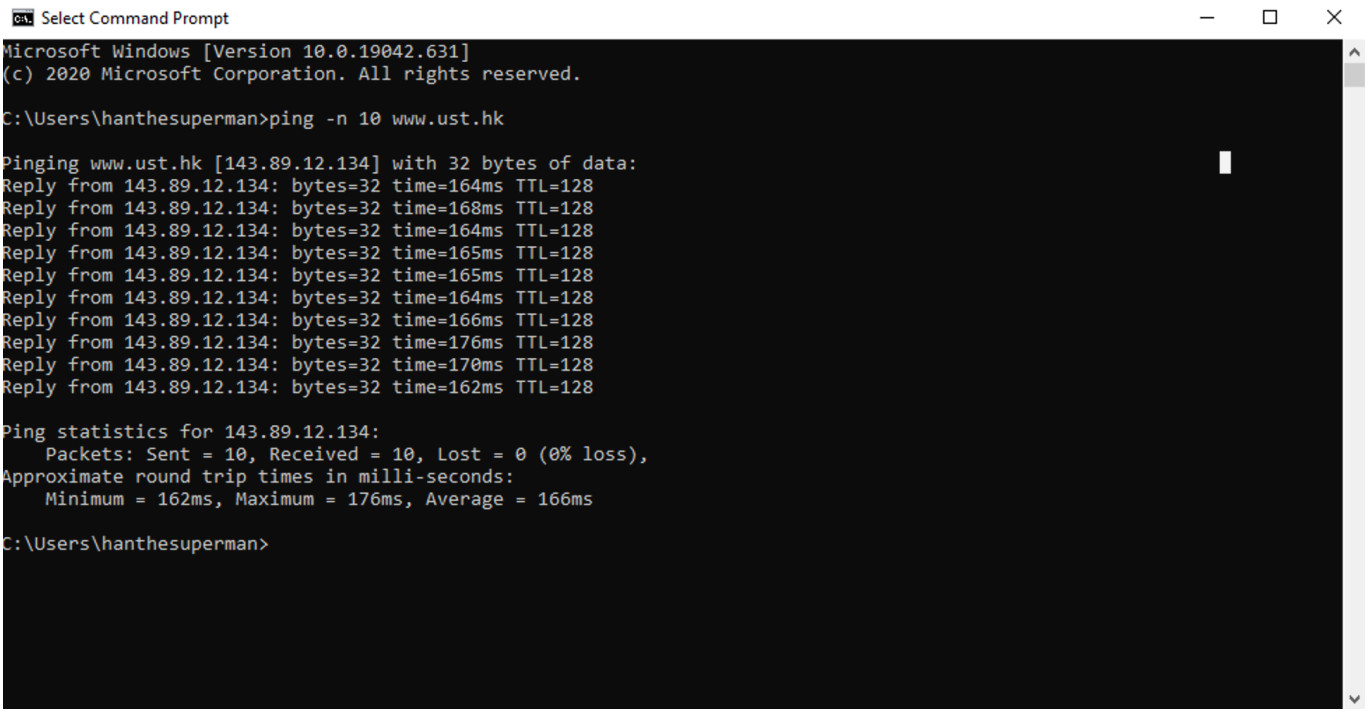

Figure 1: Screenshot of the *ping* command in Command Prompt

```
No.     Time           Source                Destination           Protocol Length Info
      4 0.148265       10.211.55.3           143.89.12.134         ICMP     74     Echo (ping) request  id=0x0001, seq=22/5632, ttl=128
(reply in 5)
Frame 4: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{4FC1B0C6-DEFD-4172-8260-8DA1C738ED37}, id
0
Ethernet II, Src: Parallel_c1:9a:54 (00:1c:42:c1:9a:54), Dst: Parallel_00:00:18 (00:1c:42:00:00:18)
Internet Protocol Version 4, Src: 10.211.55.3, Dst: 143.89.12.134
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x4d45 [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence Number (BE): 22 (0x0016)
    Sequence Number (LE): 5632 (0x1600)
    [Response frame: 5]
    Data (32 bytes)
0000  61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70   abcdefghijklmnop
0010  71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69   qrstuvwabcdefghi
```

Figure 2: Packet of ICMP Ping Request Message

Question 1

The IP address of the host (source) is 10.211.55.3. The IP address of the destination is 143.89.12.134, as shown in Figure 2.

Question 2

An ICMP packet does not have source and destination port numbers because the ICMP is used to communicate network-layer datagrams and information between the hosts and routers, while the ports are used to communicate between application layer processes. (Source: http://www.cs.toronto.edu/ ahchinaei/teaching/2016jan/csc358/Assignment4wSol.pdf)

Question 3

From the Figure 2 above, it can be observed that the ICMP type number is 8 and the code number is 0. The other fields that the ICMP packet has is the checksum, identifier, sequence number and data size. The number bytes in checksum, sequence number and identifier are 2 bytes in each field.

Question 4

The corresponding ping reply packet is as below:

```
No.     Time           Source                Destination           Protocol Length Info
      5 0.312683       143.89.12.134         10.211.55.3           ICMP     74     Echo (ping) reply    id=0x0001, seq=22/5632, ttl=128
(request in 4)
Frame 5: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{4FC1B0C6-DEFD-4172-8260-8DA1C738ED37}, id
0
Ethernet II, Src: Parallel_00:00:18 (00:1c:42:00:00:18), Dst: Parallel_c1:9a:54 (00:1c:42:c1:9a:54)
Internet Protocol Version 4, Src: 143.89.12.134, Dst: 10.211.55.3
Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0x5545 [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence Number (BE): 22 (0x0016)
    Sequence Number (LE): 5632 (0x1600)
    [Request frame: 4]
    [Response time: 164.418 ms]
    Data (32 bytes)
0000  61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70   abcdefghijklmnop
0010  71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69   qrstuvwabcdefghi
```

Figure 3: Packet of ICMP Ping Reply Message

From the Figure 3 above, it can be observed that the ICMP type number is 0 and the code number is 0.

The other fields that the ICMP packet has is the checksum, identifier, sequence number and data size.

The number bytes in checksum, sequence number and identifier are 2 bytes in each field.

2 ICMP and Traceroute

As shown in Figure 4, my Mac/Linux Terminal was having issue with *traceroute command*, leading to the fact that there will be no Echo (ping) request or reply packets, so I will use the author's Wireshark *icmp-ethereal-trace-2*.

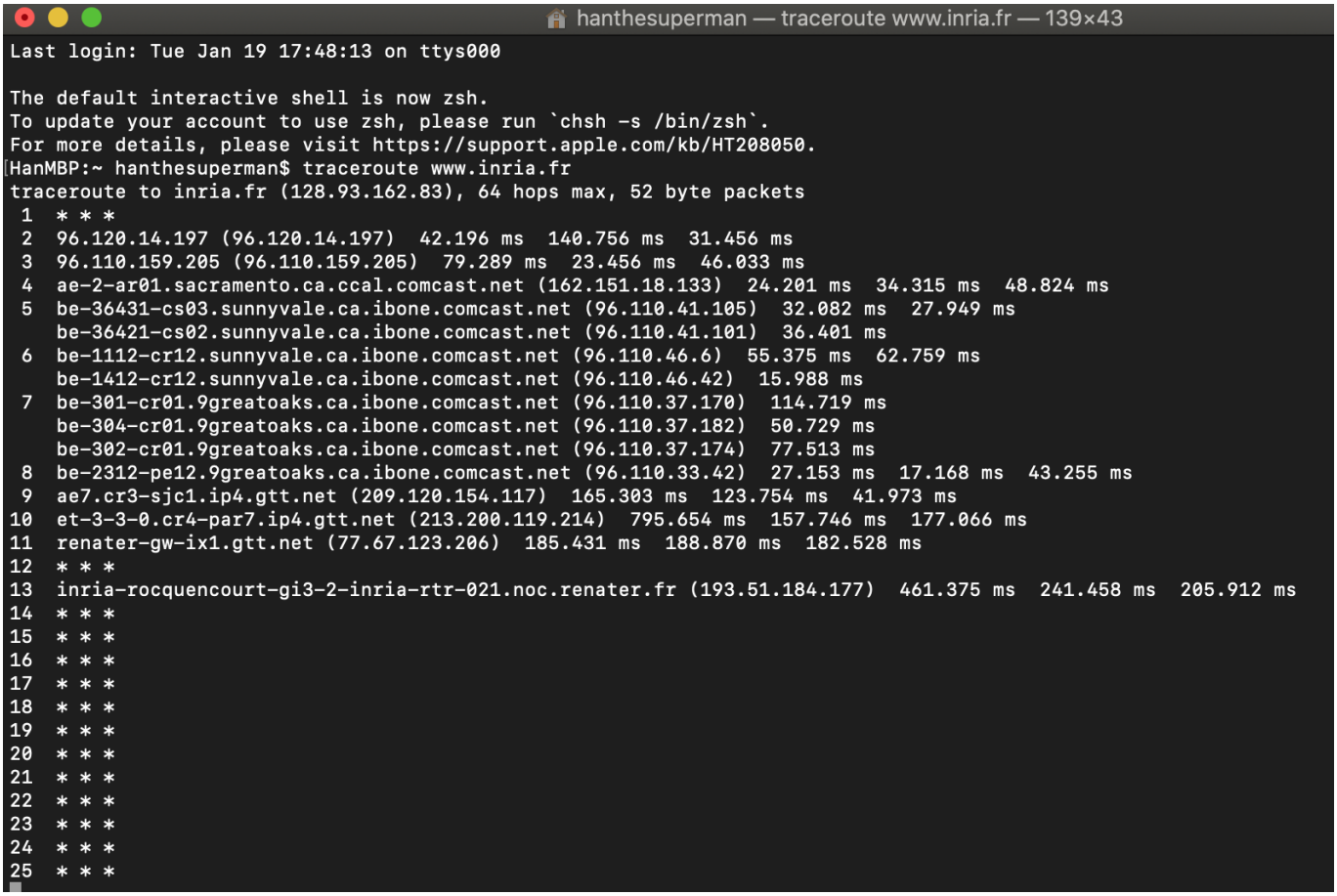

Figure 4: Mac/Linux Terminal when *traceroute* to www.inria.fr

```
No.     Time           Source                Destination           Protocol Length Info
      1 0.000000       192.168.1.101         138.96.146.2          ICMP     106    Echo (ping)
request  id=0x0200, seq=41985/420, ttl=1 (no response found!)
Frame 1: 106 bytes on wire (848 bits), 106 bytes captured (848 bits)
Ethernet II, Src: Dell_4f:36:23 (00:08:74:4f:36:23), Dst: LinksysG_da:af:73 (00:06:25:da:af:73)
Internet Protocol Version 4, Src: 192.168.1.101, Dst: 138.96.146.2
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 92
    Identification: 0xd2d5 (53973)
    Flags: 0x00
    Fragment Offset: 0
    Time to Live: 1
    Protocol: ICMP (1)
    Header Checksum: 0x085c [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.1.101
    Destination Address: 138.96.146.2
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x51fe [correct]
    [Checksum Status: Good]
    Identifier (BE): 512 (0x0200)
    Identifier (LE): 2 (0x0002)
    Sequence Number (BE): 41985 (0xa401)
    Sequence Number (LE): 420 (0x01a4)
    [No response seen]
    Data (64 bytes)
0000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

Figure 5: ICMP Packet information from the author's Wireshark

Question 5

Based on the Figure 5, the IP address of the source host is 192.168.1.101, and the IP address of the destination host is 138.96.146.2

Question 6

If the ICMP sent UDP packets instead, the IP protocol number is not 01. Rather, it will become 17 in decimal or 0x11 in hexa.

Question 7

There is no difference in the content of the query fields compared to the first lab.

Question 8

As shown in Figure 6, the ICMP error packet has an extra field of Internet Control Message Protocol, an extra header checksum and the unused 8 bytes that the error is generated compared to the original Echo (ping) request packet.

Question 9

Based on the last three Echo (ping) reply packets, they are different from the ICMP error packets in the field of Type. Echo (ping) reply packets have type 0, while ICMP error packets have type 11.

The reason for this difference is that when the packets are sent through routers, the Time-to-Live (TTL) determines the number of packets that are passed through the routers infinitely without being received. As the Echo (ping) reply packets are marked as received the by the routers and reached to the destination, it will receive Type 0.

Question 10

Based on the Figure 4 mentioned in the assignment description, the link between router 9 and router 10 has significantly longer delay compared to others. Based on the name of the router 9 and 10, I can guess that the router 9 is located in New York City, United States, where the other end could possibly be located in France, causing the longer delay across continent.

3 Extra Credit

```
No.     Time           Source                Destination           Protocol Length Info
      2 0.013151       10.216.228.1          192.168.1.101         ICMP     70     Time-to-live
exceeded (Time to live exceeded in transit)
Frame 2: 70 bytes on wire (560 bits), 70 bytes captured (560 bits)
Ethernet II, Src: LinksysG_da:af:73 (00:06:25:da:af:73), Dst: Dell_4f:36:23 (00:08:74:4f:36:23)
Internet Protocol Version 4, Src: 10.216.228.1, Dst: 192.168.1.101
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
    Total Length: 56
    Identification: 0x9d45 (40261)
    Flags: 0x00
    Fragment Offset: 0
    Time to Live: 255
    Protocol: ICMP (1)
    Header Checksum: 0x6cd8 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.216.228.1
    Destination Address: 192.168.1.101
Internet Control Message Protocol
    Type: 11 (Time-to-live exceeded)
    Code: 0 (Time to live exceeded in transit)
    Checksum: 0x2c16 [correct]
    [Checksum Status: Good]
    Unused: 00000000
    Internet Protocol Version 4, Src: 192.168.1.101, Dst: 138.96.146.2
        0100 .... = Version: 4
        .... 0101 = Header Length: 20 bytes (5)
```

/Users/hanthesuperman/Desktop/wireshark-traces/icmp-ethereal-trace-2 102 total packets, 102 shown

```
        Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
        Total Length: 92
        Identification: 0xd2d5 (53973)
        Flags: 0x00
        Fragment Offset: 0
        Time to Live: 1
        Protocol: ICMP (1)
        Header Checksum: 0xd145 [validation disabled]
        [Header checksum status: Unverified]
        Source Address: 192.168.1.101
        Destination Address: 138.96.146.2
    Internet Control Message Protocol
        Type: 8 (Echo (ping) request)
        Code: 0
        Checksum: 0x51fe [unverified] [in ICMP error packet]
        [Checksum Status: Unverified]
        Identifier (BE): 512 (0x0200)
        Identifier (LE): 2 (0x0002)
        Sequence Number (BE): 41985 (0xa401)
        Sequence Number (LE): 420 (0x01a4)
```

Figure 6: ICMP Packet information of the Time-to-live exceeded error

ECS 152A - Project 1 - IP Wireshark Lab

Han Nguyen (917278789) & Blake McMurray (999162729)

1/17/2020

1 Capturing packets from an execution of traceroute

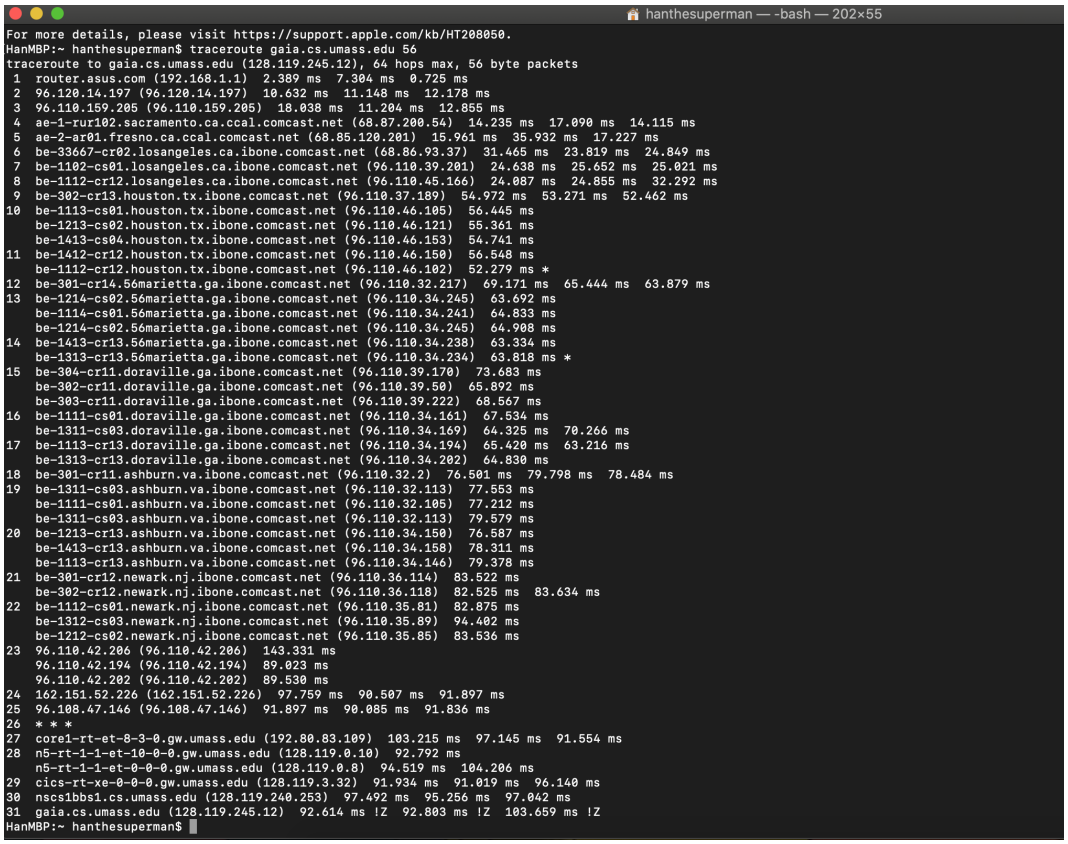

Figure 1: IP Traceroute with 56 bytes of packet

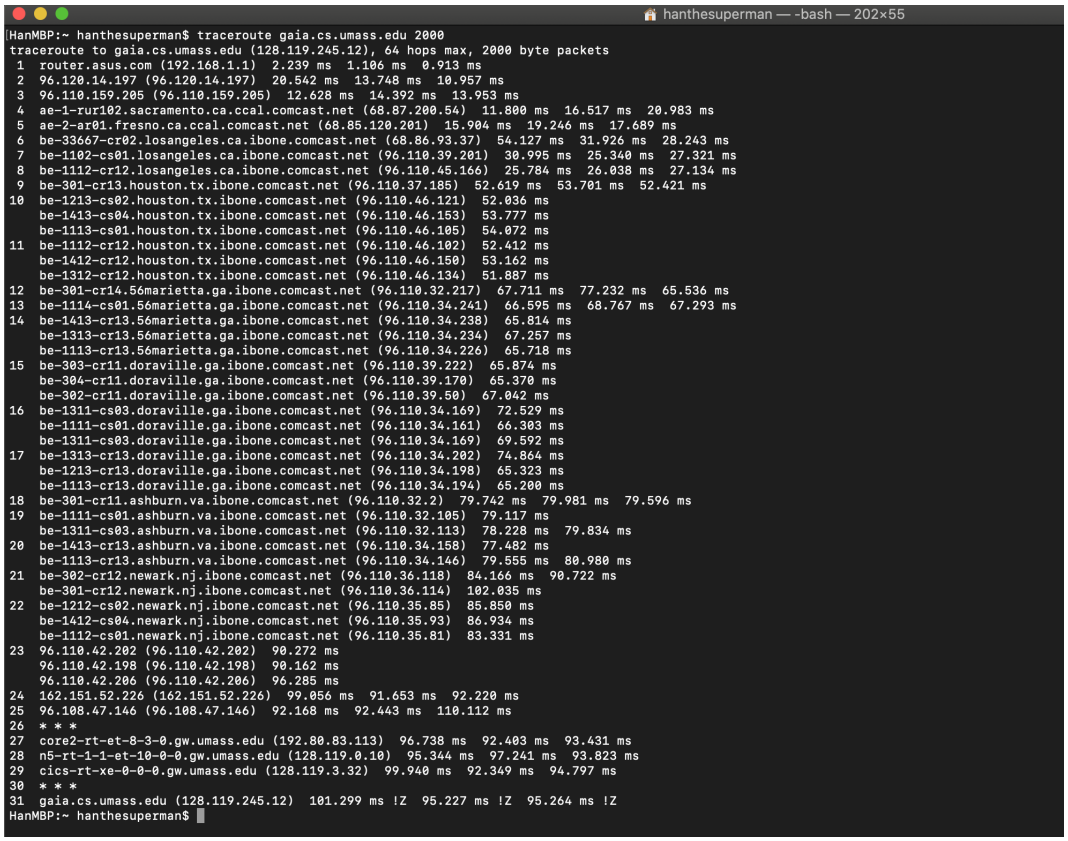

Figure 2: IP Traceroute with 2000 bytes of packet

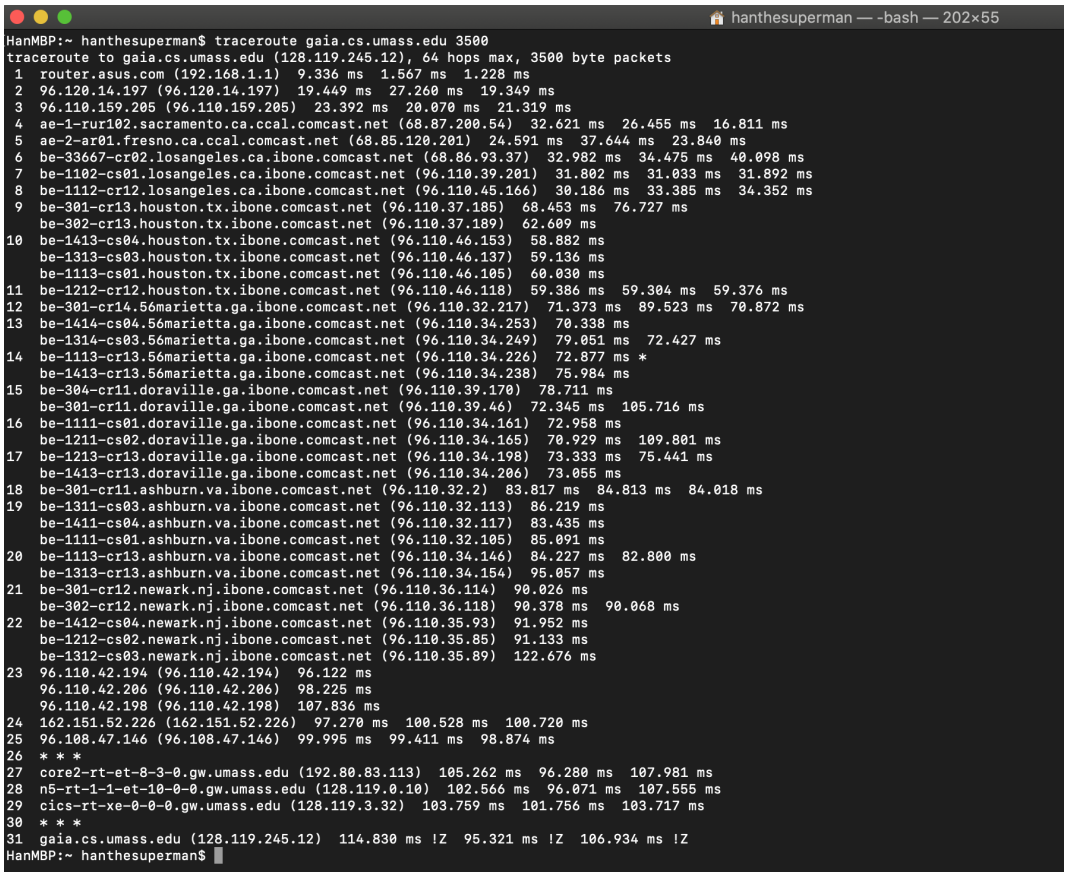

Figure 3: IP Traceroute with 3500 bytes of packet

2 A look at the captured trace

```
No.     Time           Source                Destination           Protocol Length Info
      1 0.000000       192.168.1.39          239.255.255.250       SSDP     167    M-SEARCH *
HTTP/1.1
Frame 1: 167 bytes on wire (1336 bits), 167 bytes captured (1336 bits) on interface en0, id 0
Ethernet II, Src: fa:7b:8c:4e:d7:f3 (fa:7b:8c:4e:d7:f3), Dst: IPv4mcast_7f:ff:fa (01:00:5e:
7f:ff:fa)
Internet Protocol Version 4, Src: 192.168.1.39, Dst: 239.255.255.250
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 153
    Identification: 0x6e2b (28203)
    Flags: 0x00
    Fragment Offset: 0
    Time to Live: 1
    Protocol: UDP (17)
    Header Checksum: 0x995f [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.1.39
    Destination Address: 239.255.255.250
User Datagram Protocol, Src Port: 57917, Dst Port: 1900
Simple Service Discovery Protocol
```

Figure 4: First ICMP Echo Request Message

Question 1

As shown in Figure 4, my computer's IP address is: 192.168.1.39

Question 2

As shown in Figure 4, within the IP packet header, the value in the upper layer protocol field is UDP (17) in Unix/Linux.

Question 3

As shown in Figure 4, there are 20 bytes in the IP Header. Since there are also 153 bytes in total length, the payload of the IP datagram is $153 - 20 = 133$ bytes.

Question 4

As shown in Figure 5, the fragment offset bit is 0 and the bit of the *More fragments* section is also 0, hence there are no datagram fragments.

Question 5

The fields in the IP datagram that always change from one datagram to the next within this series of ICMP messages sent by my computer are:

- Total Length in bytes (found 5 cases so far)

- Identification (since different IP packets have different identification numbers).

- Time to Live (since the router might increment the TTL count if a packet is passed through routers infinitely: Source: https://www.cloudflare.com/learning/cdn/glossary/time-to-live-ttl/)

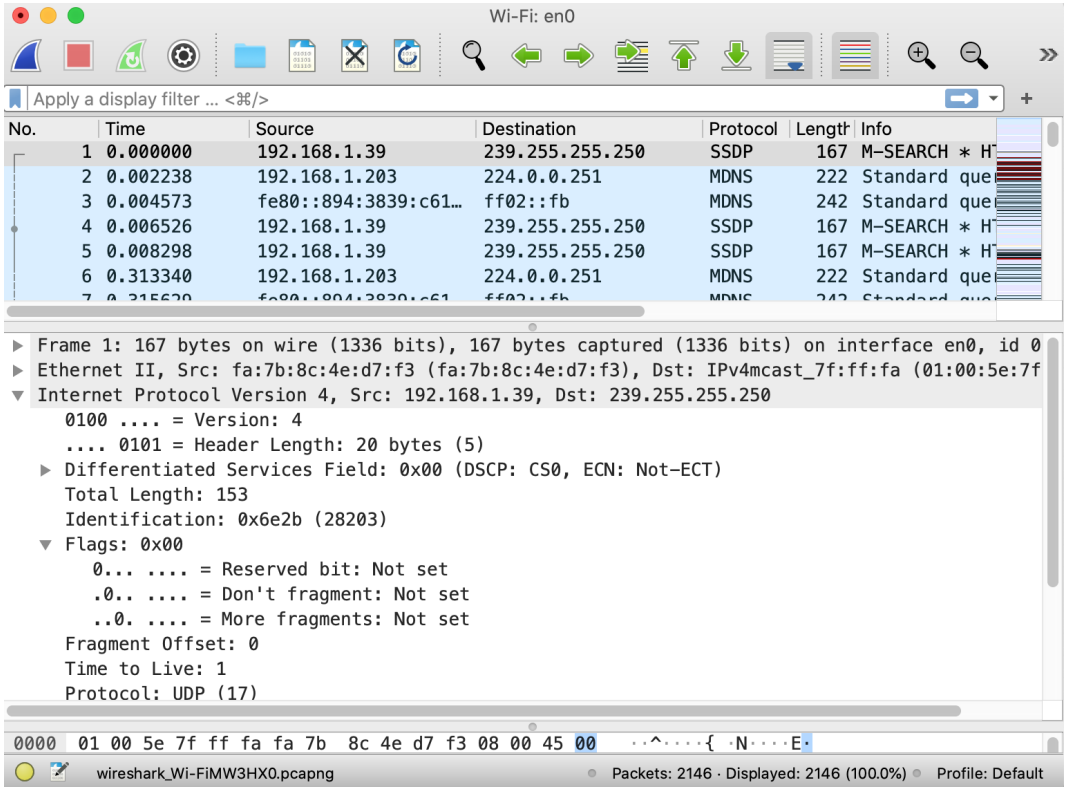

Figure 5: Data of Datagram Fragments

- Header Checksum (since the router change the checksum if the IP header changes when decrementing TTL count; Source: https://tools.ietf.org/html/rfc1812#section-4.2.2.5)

Question 6

The fields that stay constant are:

- Version (since the packets are captured in IPv4 version)
- Header Length (since the packets are captured in the ICMP)
- Differentiated Services (since the packets are captured in ICMP, they will have the same differentiated services)
- Flags and Fragment Offset (since the packets are not set to be reserved or fragmented)
- Protocol (sine the packets are captured in ICMP prototcol)
- Source and Destination IP Address (since the packets are sent and received by the same source from the *traceroute*

The fields that *must* stay constant and the reasons are described as the above. The fields that *must* change and the reasons are described in the Section 2.5, except for the *Total Length in bytes*.

Question 7

The pattern is that the values in the Identification field of the IP datagram increment by each datagram.

Question 8

As shown in Figure 6, the Identification value is $0xfe4d$, or 65101. The Time-to-Live (TTL) value is 63.

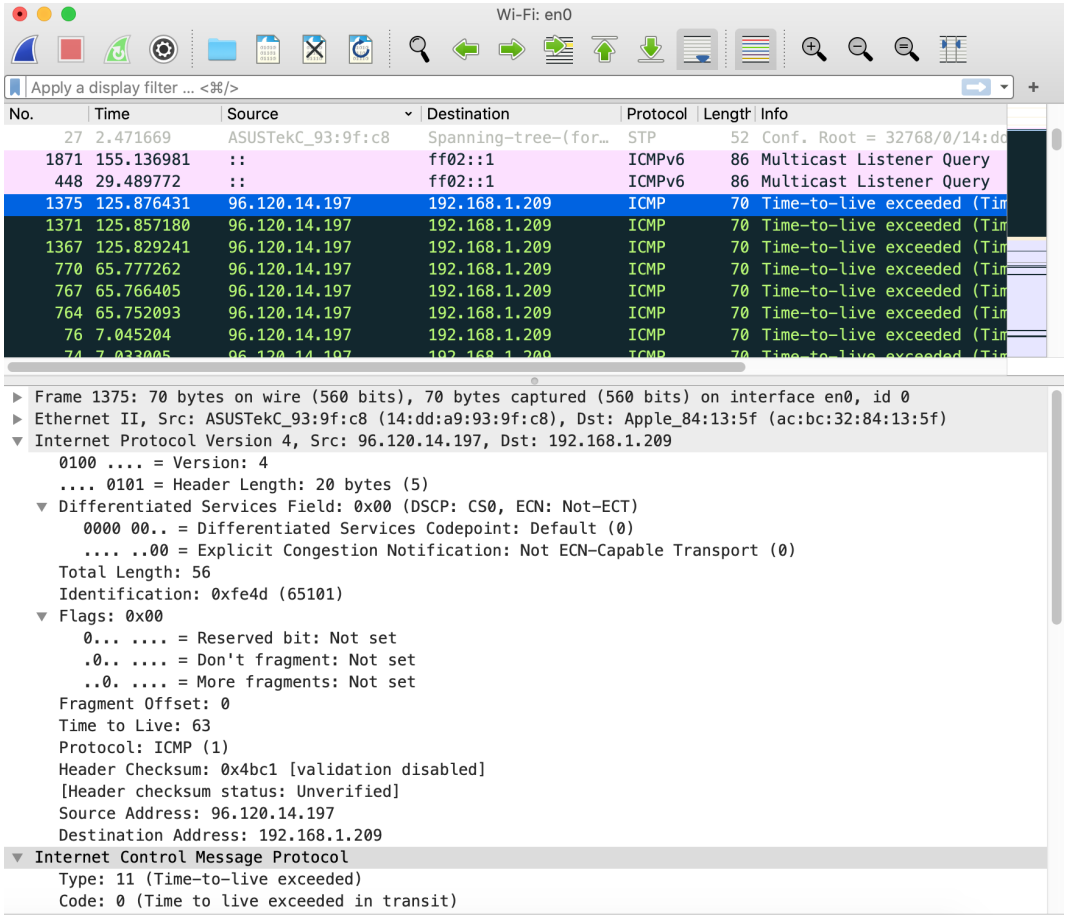

Figure 6: The nearest hop of TTL-exceeded packet

Question 9

The values in the Identification field of all TTL-exceeded packets at the nearest hop do not remain unchanged because different packets will have different Identification numbers. However, the values in the TTL field of all TTL-exceeded packets at the nearest hop remains unchanged since the TTL value changes only when a packet is passed through routers infinitely without being received. At the first hop, this number always remains the same until the next hop.

Fragmentation

Since we forgot to save the Wireshark captured packets after answering previous 9 questions, we will switch to use the author's *ip-ethereal-trace-1* to answer the Question 10 - 13

Question 10

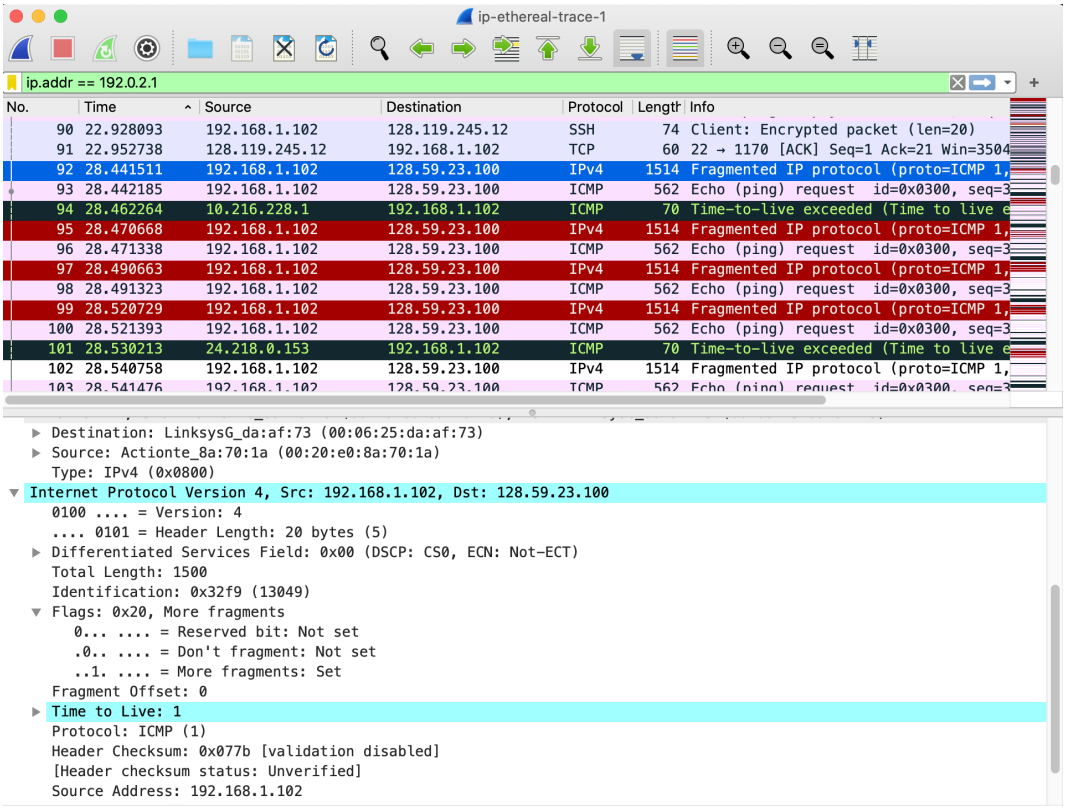

Figure 7: Fragmented IP Packet at pingplotter 1500

As shown in Figure 7, there are many IP addresses that say "Fragmented IP Protocol", so definitely the message has been fragmented across more than one IP address.

Question 11

As shown in Figure 7, the bit value of the field *More Fragments* has been set to 1 across many IP packets, which indicates that the IP packets have been fragmented.

Since the flag offset is 0, this must be the first fragmented packet, compared to latter fragmented packets where their flag offset must be greater than 0.

This fragmented IP datagram has total length of 1500.

Question 12

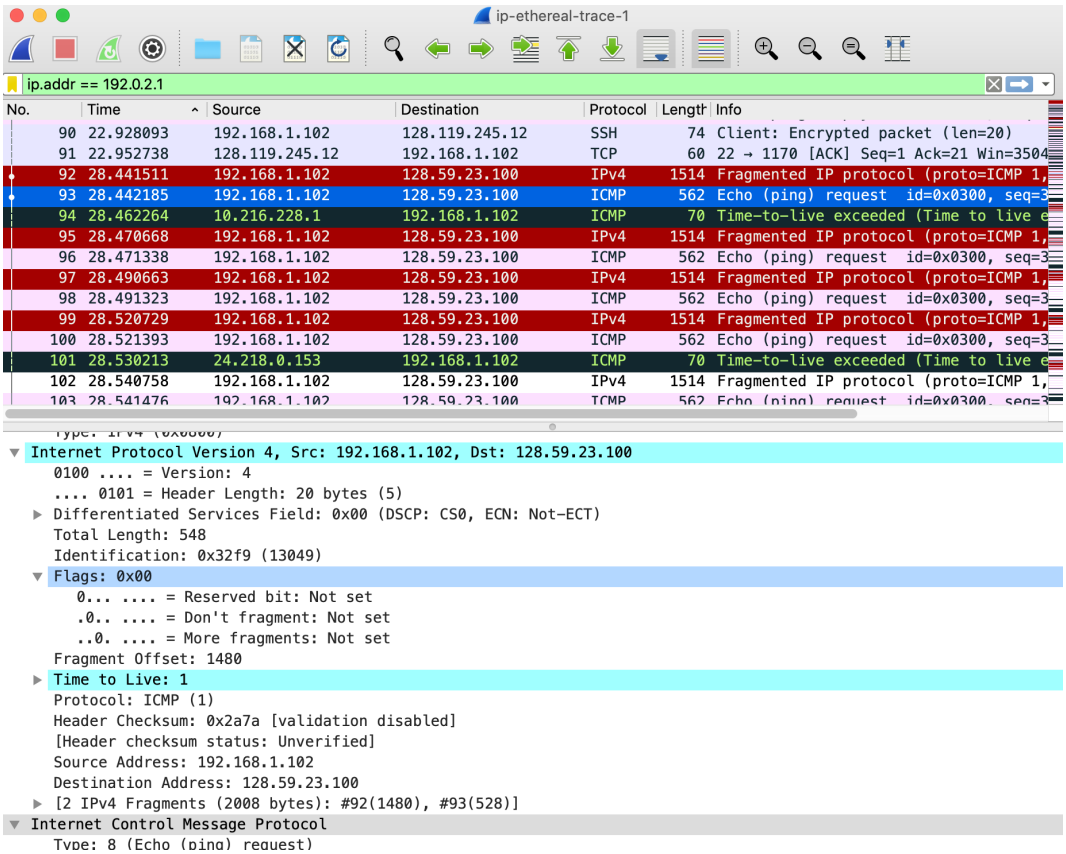

Figure 8: Second Fragmented IP Packet at pingplotter 1500

As shown in Figure 8, the second fragment of the fragmented IP packet has the fragment offset of 1480, which indicates that this is not the first fragment (first fragment must have fragment offset as 0). However, this is the final fragment of the fragmented IP packet, since the bit value of the *More Fragments* is set to 0.

Question 13

The fields that change between the first and the second fragment of the fragmented IP packet are:

- Total length (1500 in the first and 548 in the second)
- Flags, Fragment Offset and *More Fragments* bit (as above)
- Header Checksum (0x077b in the first and 0x2a7a in the second)

Question 14

As shown in Figure 9, there are three fragments of message after setting the pingplotter to 3500 (UDP and ICMP can be merged since it runs on Macbook Terminal).

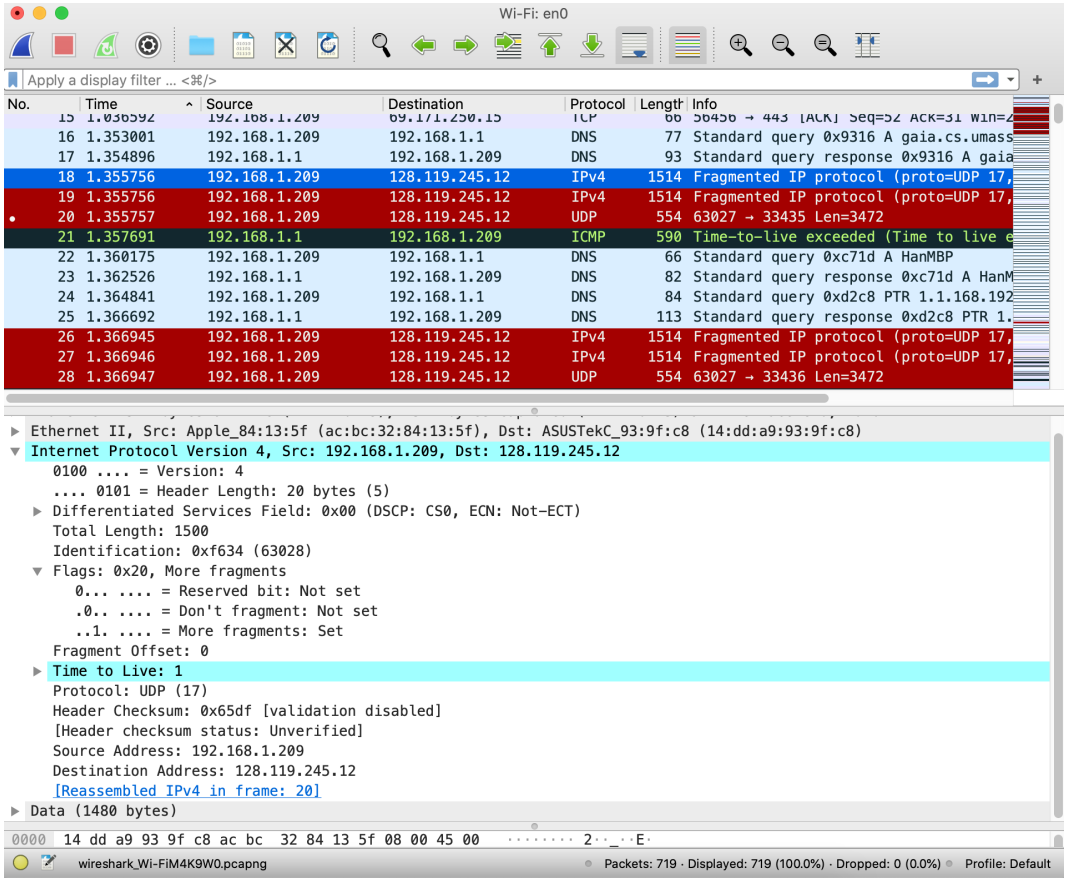

Figure 9: Fragmented IP Packet at pingplotter 3500

Question 15

The fields in the IP header that change across the fragments of the fragmented IP packet are:

- Fragment Offset (0, 1480, 2960)
- Total Length (1500, 1500, 540)
- Header Checksum (0x65df, 0x6526, 0x882d)