Assignment -4

CS 470 Spring 2018

Note: Only typed submissions are accepted.

Q1. (8 points)

Consider a datagram network using 32-bit host addresses. Suppose a router has four links, numbered 0 through 3, and packets are to be forwarded to the link interfaces as follows:

Destination Address Range	Link Interface
11100000 00000000 00000000 00000000 Through 11100000 00111111 11111111 11111111	0
11100000 01000000 00000000 00000000 through 11100000 01000000 11111111 11111111	1
11100000 01000001 00000000 00000000 through 11100001 01111111 11111111 11111111	2
otherwise	3

a. Provide a forwarding table that has five entries, uses longest prefix matching, and forwards packets to the correct link interfaces.

b. Describe how your forwarding table determines the appropriate link interface for datagrams with destination addresses:

11001000 10010001 01010001 01010101

11100001 01000000 11000011 00111100

11100001 10000000 00010001 01110111

Q2. (7 points)

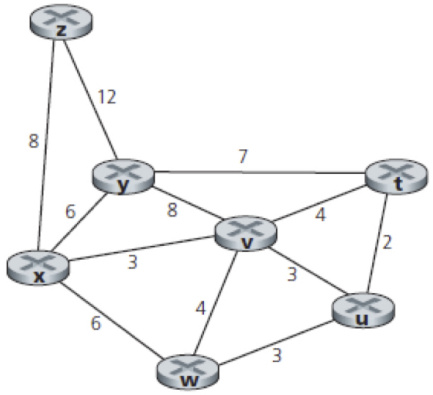
a. Consider sending a 2400-byte datagram into a link that has an MTU of 700 bytes. Suppose the original datagram is stamped with the identification number 422. How many fragments

are generated? What are the values in the various fields in the IP datagram(s) generated related to fragmentation?

b. Suppose datagrams are limited to 1,500 bytes (including header) between source Host A and destination Host B. Assuming a 20-byte IP header, how many datagrams would be required to send an MP3 consisting of 5 million bytes? Explain how you computed your answer.

Q3. (10 points)

Consider the following network. With the indicated link costs, use Dijkstra's shortest-path algorithm to compute the shortest path from x to all network nodes. Show how the algorithm works by computing a table similar to your textbook.



Q4. Use this link: https://gaia.cs.umass.edu/wireshark-labs/Wireshark_IP_v7.0.pdf to find the Wireshark lab for IP. Read the document carefully and answer the questions from the document. In addition attach the print screen of your screen capture with each answer (or set of answers). (15 points)

Q5. Use this link: https://gaia.cs.umass.edu/wireshark-labs/Wireshark_NAT_v7.0.pdf to find the Wireshark lab for NAT. Read the document carefully and answer the questions from the document. In addition attach the print screen of your screen capture with each answer (or set of answers). (10 points) (Extra credit: 3 points, see at the end of the pdf)

Q6. Use this link: https://gaia.cs.umass.edu/wireshark-labs/Wireshark_ICMP_v7.0.pdfto find the Wireshark lab for ICMP. Read the document carefully and answer the questions

Solutions Only

Question 1.

A. Datagram forwarding table that has five entries, uses longest prefix matching, and forwards packets to the correct link interfaces:

Longest Prefix Match	Link Interface
11100000 00	0
11100000 01000000	1
1110000 or 1110001 0	2
11100001 1	3
otherwise	3

B. The link interface following datagrams are:

a. 11001000 10010001 01010001 01010101: The given address doesn't match to any of the destination address range for link interface 0,1,2. So, it goes to the otherwise prefix match which is link interface 3.

b. 11100001 01000000 11000011 00111100: The first 8 bits matches to link interface 2 but also to the link interface 3. But, the next 8 bits after it falls in the address range of 2. So, this address has the link interface 2.

c. 11100001 10000000 00010001 01110111: It solely falls in the otherwise address range of link interface 3.

Question 2.

A. The link has an MTU of 700 bytes and the total size of the datagram being sent it of 2400 bytes excluding the IP header it is: 2380 bytes. But, the maximum size of the data field in the segment excluding the IP header is : 700 bytes - 20 = 680 bytes

2380/ 680 ~ 4. Therefore, we need 4 fragments.

Given, the original datagram is stamped with the identification number 422. The first segment will be of: 700 bytes, with flag set to 1 and offset set to 0. Similarly, the second and third segment will be of: 700 bytes with flag set to 1 and offset set to 85, 170. But the final

segment will be only of: 300 bytes with flag set to 0 and offset set to 255 which indicates that there are no more fragments left to be received.

B. Given,

MTU = 1500 bytes

IP header size = 20 bytes

Size of MP3 = $5 * 10^7$ bytes

Total bytes in data field = 1500 - 20 bytes = 1480 bytes

But, if the data is carried out in TCP for reliability, then each TCP segment has 20 bytes of header.

So, total bytes in the data field = 1480 bytes - 20 = 1460 bytes

Now, the total number of fragments required = $5 * 10^7$ bytes / 1460 bytes

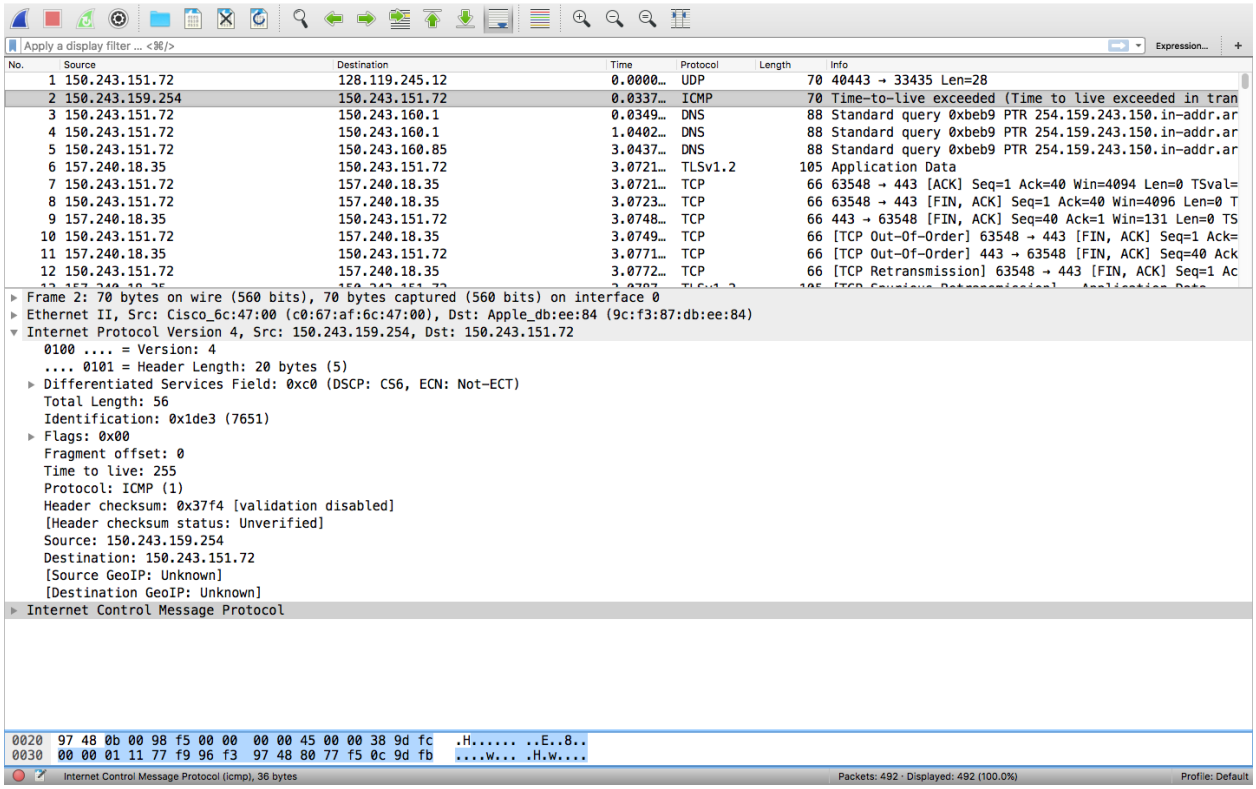
= 3425

Each 3424 will be of 1500 bytes and the last fragment 3425 will be of , $(5 * 10^7 - 1460 * 3424) +40 = 960+40$ bytes = 1000 bytes.

Question 3

Step	N'	D(t),p(t)	D(u),p(u)	D(v),p(v)	D(w),p(w)	D(y),p(y)	D(z),p(z)
0	x	∞	∞	3, x	6,x	6, x	8, x
1	xv	7, v	6, v	3, x	6, x	6, x	8, x
2	xvu	7, v	6, v	3, x	6, x	6, x	8, x
3	xvuw	7,v	6, v	3, x	6, x	6, x	8,x
4	xvuwy	7, v	6, v	3, x	6, x	6, x	8,x
5	xvuwyt	7, v	6, v	3, x	6, x	6,x	8, x
6	xvuwytz	7, v	6, v	3, x	6, x	6, x	8, x

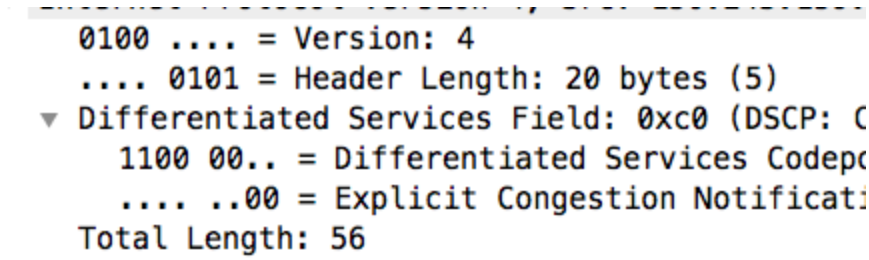
Question 4 Capturing packets from an execution of traceroute



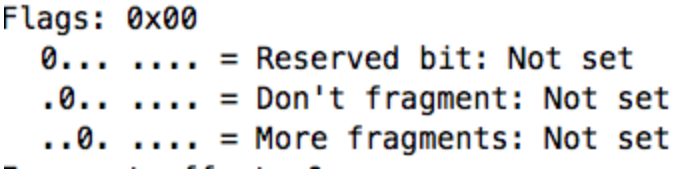
1. The IP address of my computer is:150.243.151.72

2. The value in the upper layer protocol field is: ICMP (1)

3. IP header bytes: 20 bytes , payload of the datagram: 56-20 = 36 bytes



4. The IP datagram has not been fragmented. Since, the more fragments = 0.



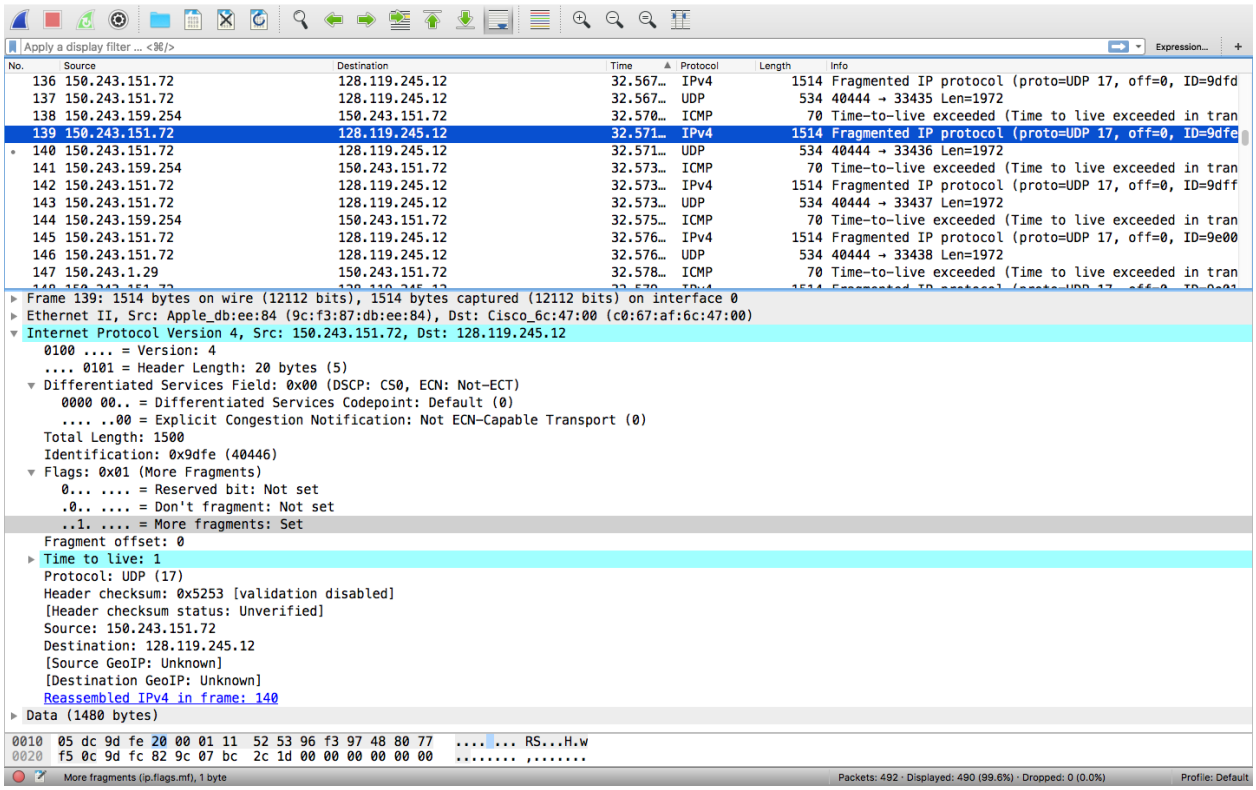
5. Identification, time to live, header checksum, (source IP and destination IP) always change.

6. Flags, version, fragment offset, protocol, differentiated services and header length always stays constant. These field should always remain constant as well because we are using the same source and destination, same version IPv4, same ICMP packets and they use the same type of ICMP services. And, the identification, ttl, header checksum should change because each packets have different identifications and since the header changes, so does the checksum.
7. With each ICMP requests, the identification field of the IP datagram increases.
8. The value in the Identification field and the TTL field are: 7651 and 255.
9. The identification of each packets are unique so it always changes. When more than 1 packets has the same identification then they are the fragments from the same packet.

Fragmentation

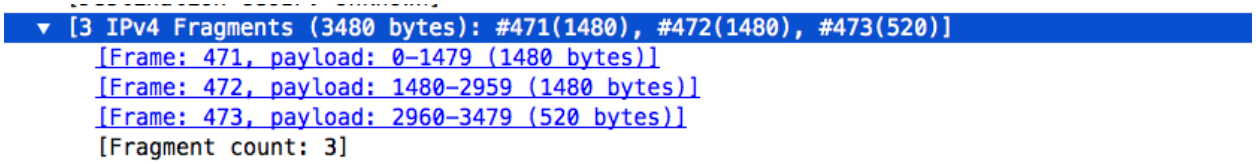
10. Couldn't find ICMP echo request as I used the MAC instead of the pingplotter. But, the more fragments is equal to 1.
11.The flag is set to more fragment bit. The fragment offset is set to 0 this means this is the first fragment. This datagram has the total length of 1500 including the header.

12. The fragment offset is not 0. It is not the last fragment because the fragment flag is set to more fragments.



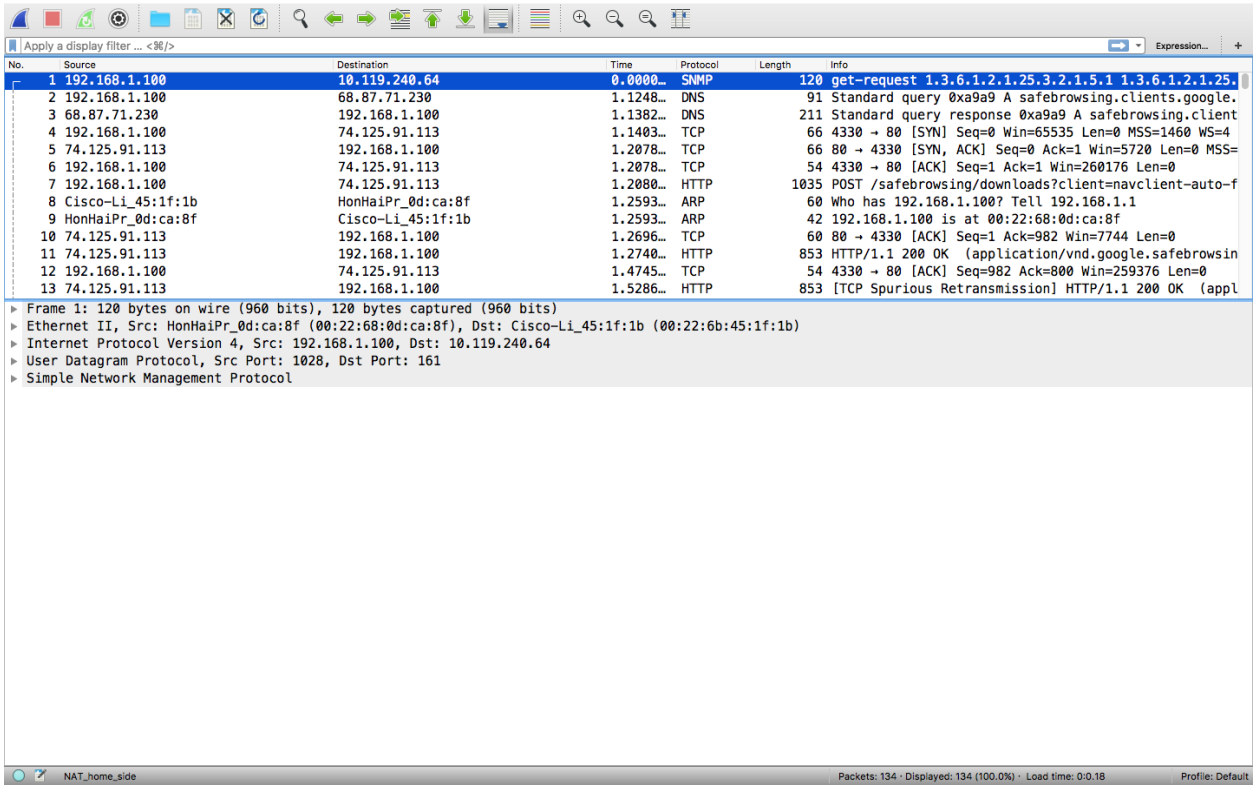
13. Total length, flags, fragment offset and checksum.

14. There were 3 fragments created from the original datagram.



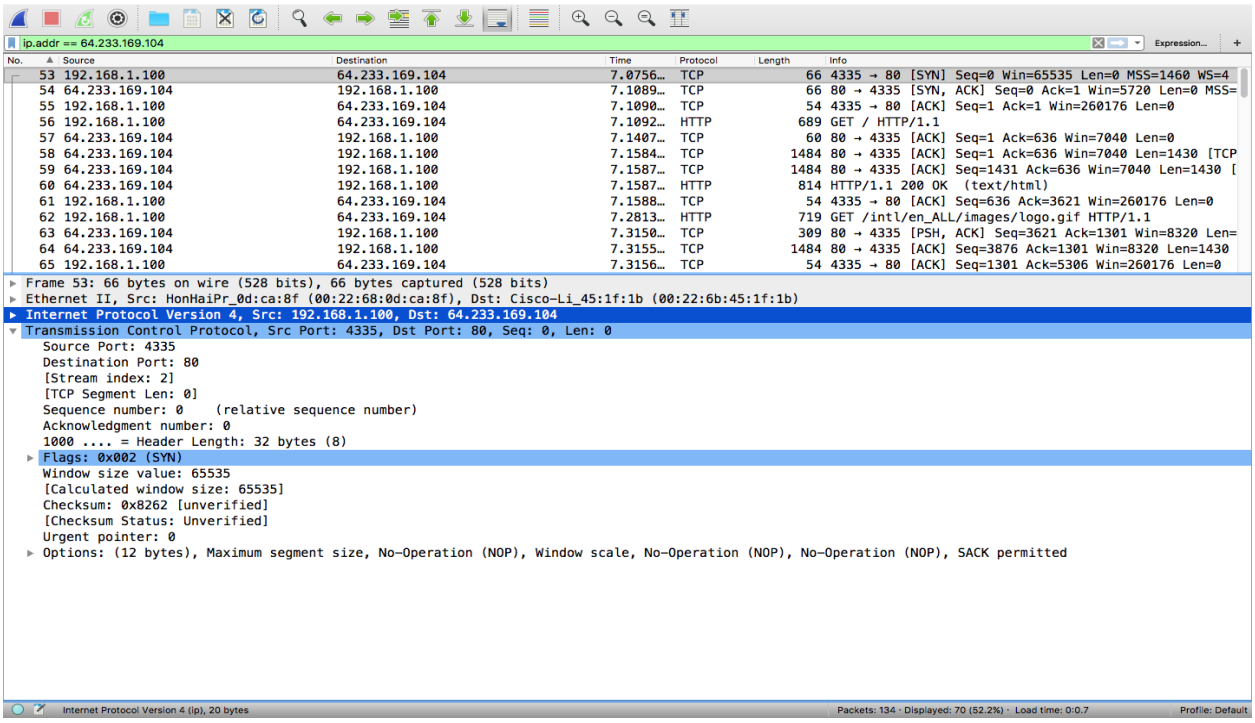
15. Fragment and offset.

Question 5. **Wireshark Lab: NAT v7.0**

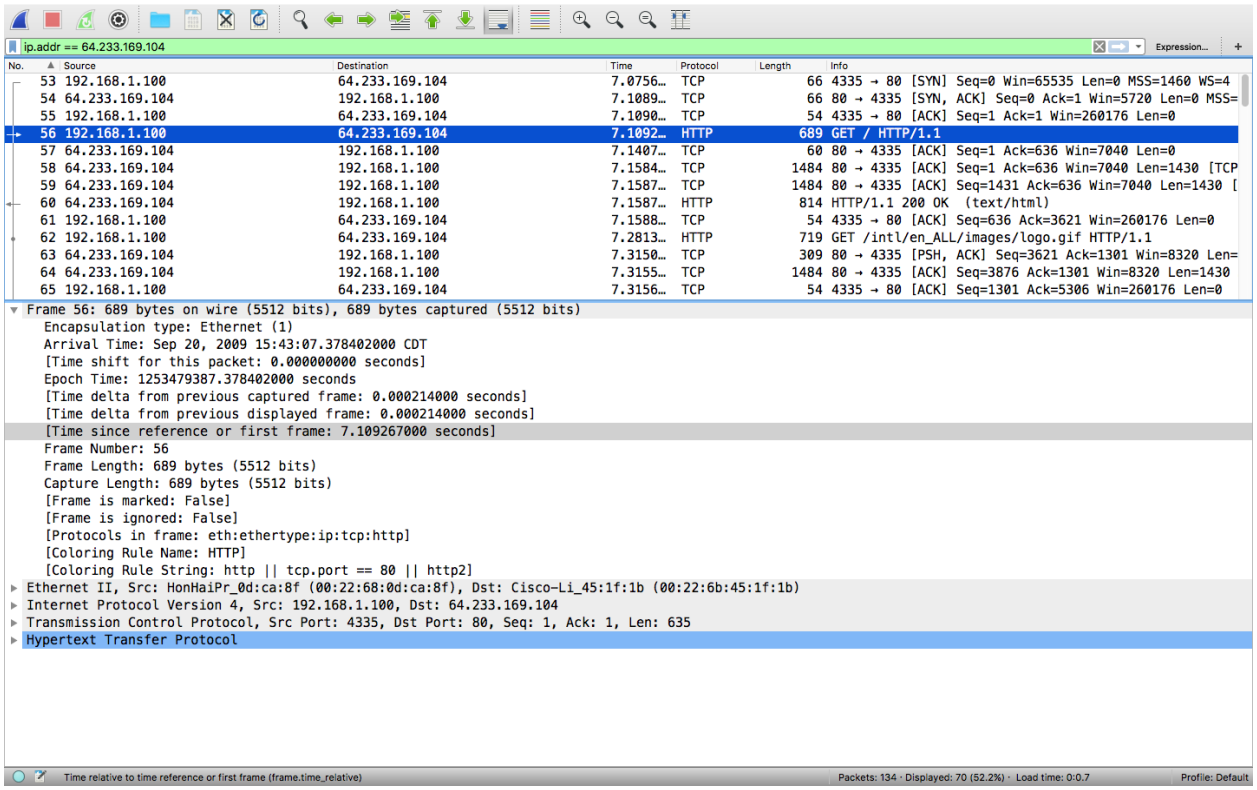


1. The IP address of the client is 192.168.1.100.

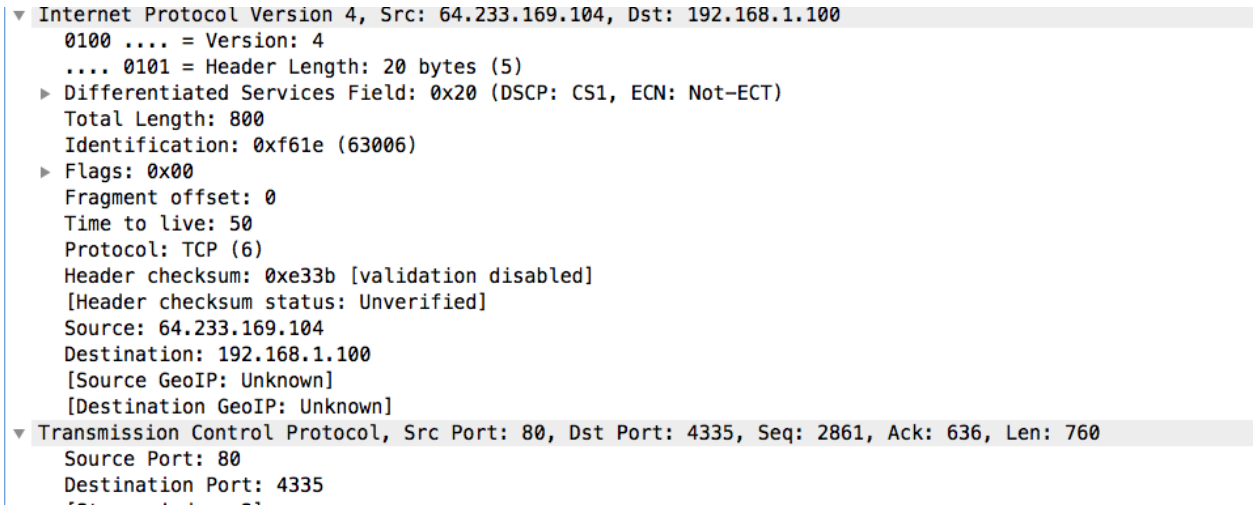
2.



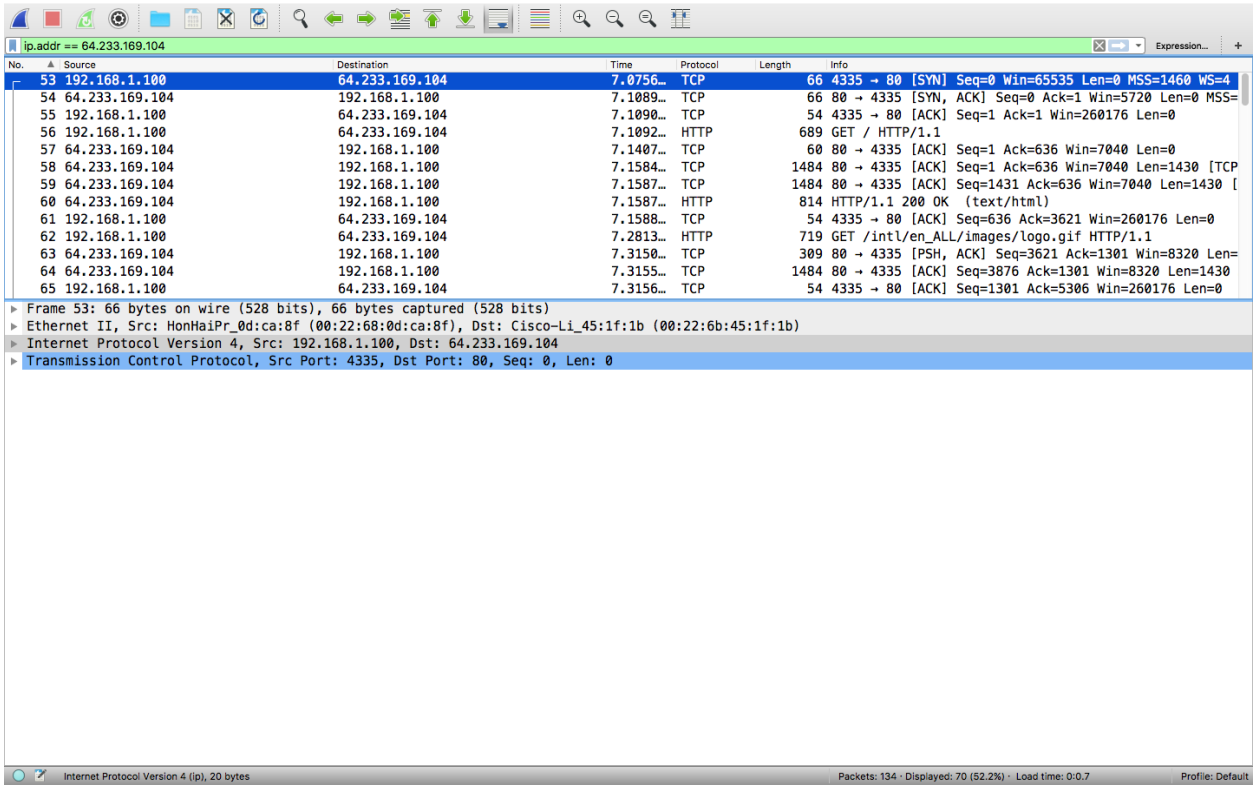
3. The source and destination IP addresses are: 192.168.1.100 at port 4335, Dst: 64.233.169.104 at port 80.



4. The corresponding 200 OK HTTP message was received at 7.158797. The source IP address is: 64.233.169.104 and port number 80, Dst IP address is : 192.168.1.100 and port number : 4335.

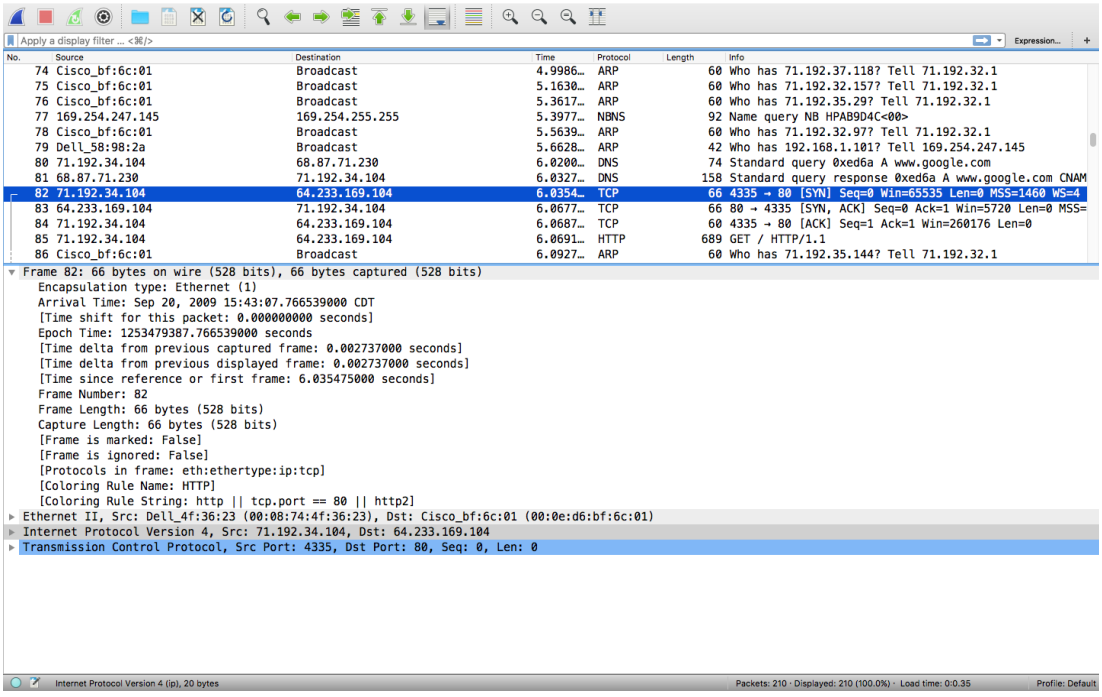


5. The client-to-server TCP SYN segment sent that sets up the connection used by the GET sent at time 7.109267 was at 7.075657 seconds.
The source IP address at TCP SYN segment is: 192.168.1.100 at port 4335 and Destination IP address is: 64.233.169.104 at port 80.



And the source IP address of the ACK segment is: 64.233.169.104 at port 80 and the destination IP address is: 192.168.1.100 at port 4335. The client received the ACK at 7.108986 seconds.

6.

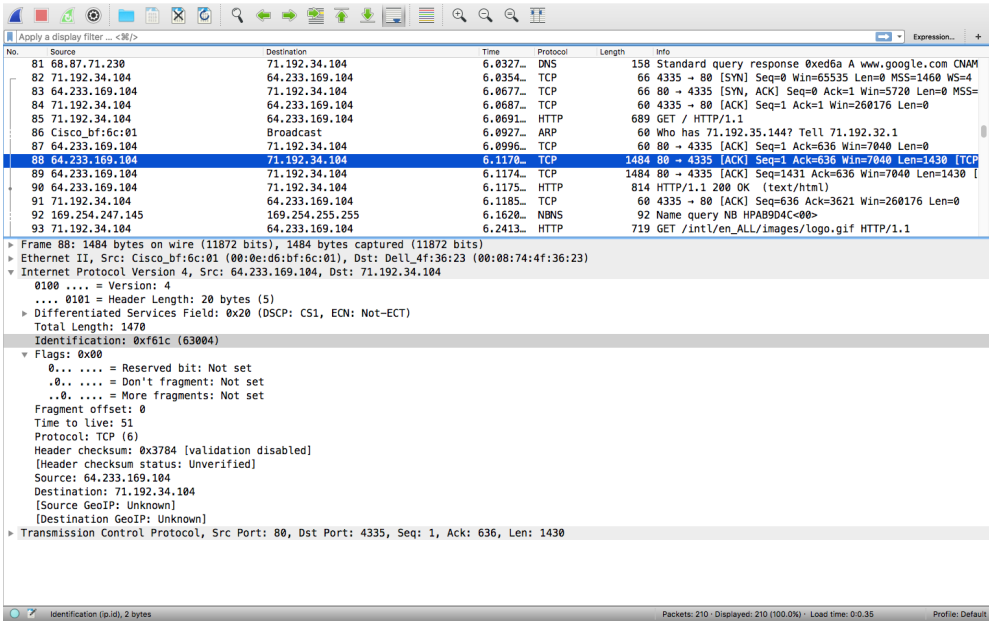


It was received at time: 6.067775 seconds

Source IP: 71.192.34.104 AT PORT 4335, Destination IP: 64.233.169.104 AT PORT 80

The source IP, destination IP and their port numbers are different from Q. 3.

7. No. The checksum and flags are changed.



8.

The first message was received at 6.117078 seconds

HTTP 200 OK message source IP: 64.233.169.104, Port: 80

HTTP 200 OK message Destination IP: 71.192.34.104, Port: 4335

Version and flag did not change while ttl, header checksum changed.

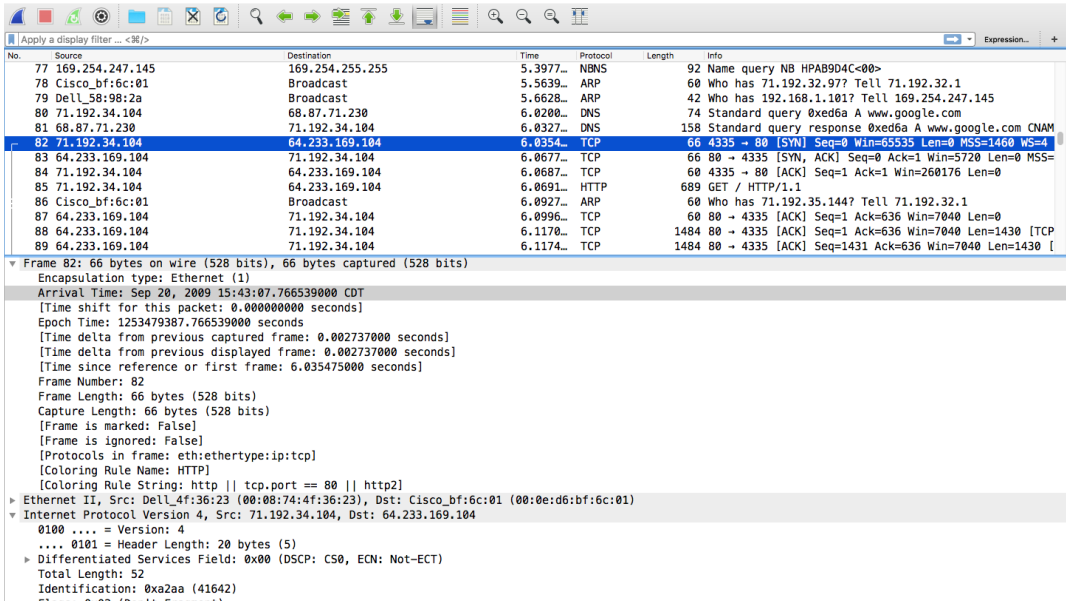
9. <u>For SYN:</u>

Time: 6.035475000 seconds

Source IP Address: 71.192.34.104

Destination IP Address: 64.233.169.104

Ttl changed.



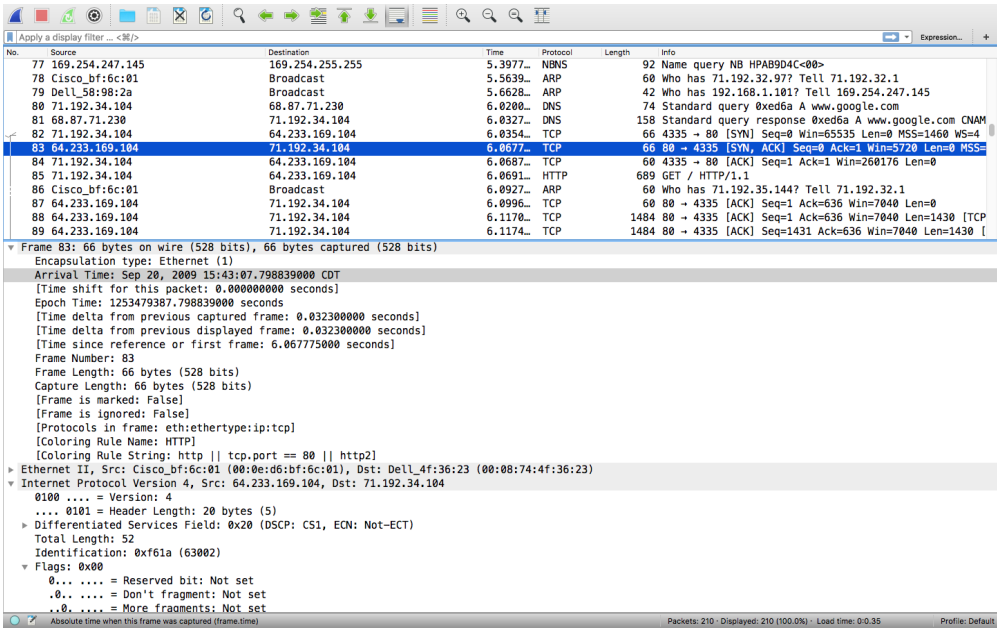
For ACK:

Time: 6.035475000 seconds

Source IP Address: 71.192.34.104

Destination IP Address: 64.233.169.104

Time to live changed



10.

NAT translation table in the NAT router	
WAN SIDE	LAN SIDE
71.192.34.104 Port: 4335	192.168.1.100 port: 4335

EXTRA CREDIT

The trace files investigated above have additional connections to Google servers above and beyond the HTTP GET, 200 OK request/response studied above. For example, in the NAT_home_side trace file, consider the client-to-server GET at time 1.572315, and the GET at time 7.573305.

ANSWER:

HTTP request messages are used when we directly want to invoke our functions via HTTP. To allow for specific segmentation the HTTP takes arguments: request and response. They represent the HTTP request sent to the server and the response that will be returned to the client. The GET at time 1.572315, and the GET at time 7.573305 in the NAT_homw_side trace file represent that the site is not malicious and safe to visit. It represents the safe browsing check that Google implemented. The first URL irects the client to safebrowsing and after making sure that the website is safe, the second URL directs the client to the targeted website. Also, the destination IP address changed in different frames which means that there must be DNS queries in different segments.

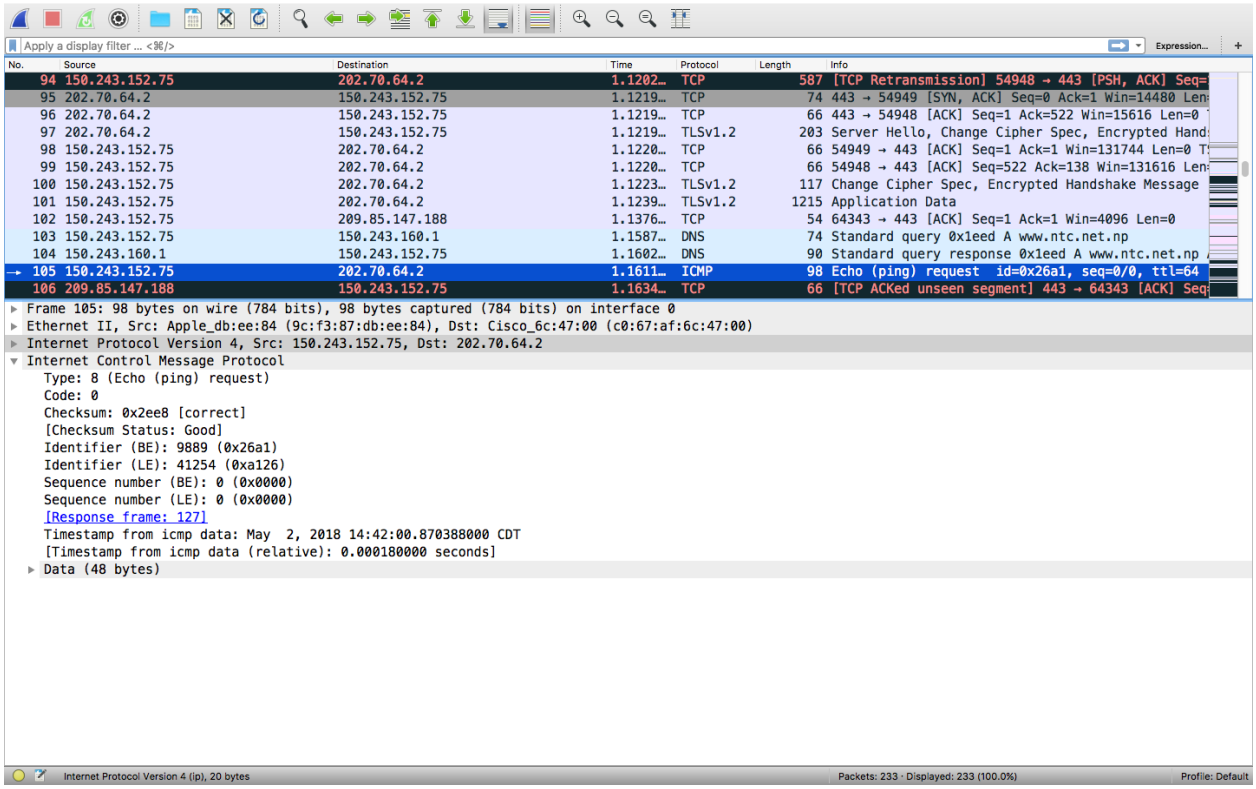
Question 6. **ICMP and PING**

Url used:
www.ntc.net.np

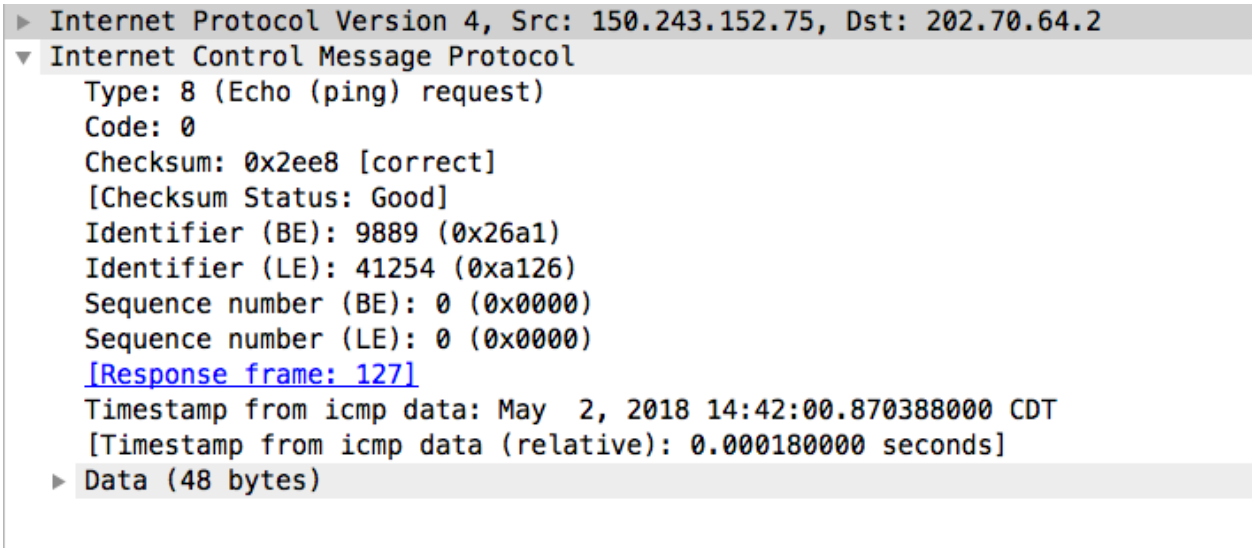
```
dikshyas-air:~ dikshya96$ ping -c 10 www.ntc.net.np
PING www.ntc.net.np (202.70.64.2): 56 data bytes
64 bytes from 202.70.64.2: icmp_seq=0 ttl=48 time=750.413 ms
64 bytes from 202.70.64.2: icmp_seq=1 ttl=48 time=305.834 ms
Request timeout for icmp_seq 2
64 bytes from 202.70.64.2: icmp_seq=2 ttl=48 time=1001.550 ms
64 bytes from 202.70.64.2: icmp_seq=3 ttl=48 time=1011.629 ms
64 bytes from 202.70.64.2: icmp_seq=4 ttl=48 time=1006.666 ms
64 bytes from 202.70.64.2: icmp_seq=5 ttl=48 time=454.685 ms
64 bytes from 202.70.64.2: icmp_seq=6 ttl=48 time=319.680 ms
64 bytes from 202.70.64.2: icmp_seq=7 ttl=48 time=1006.877 ms
64 bytes from 202.70.64.2: icmp_seq=8 ttl=48 time=358.350 ms
64 bytes from 202.70.64.2: icmp_seq=9 ttl=48 time=1406.426 ms

--- www.ntc.net.np ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 305.834/762.211/1406.426/362.775 ms
dikshyas-air:~ dikshya96$
```

1. The IP address of the host and destination are: 150.243.152.75 and 202.70.64.2

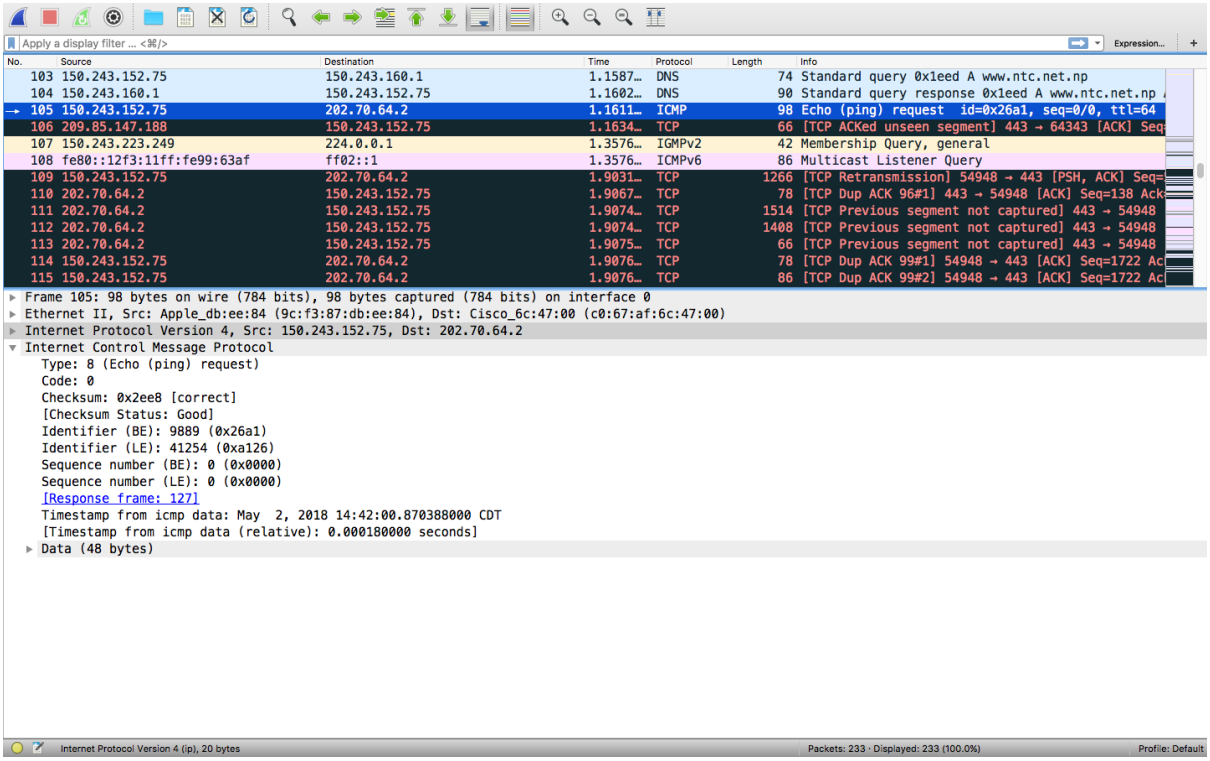


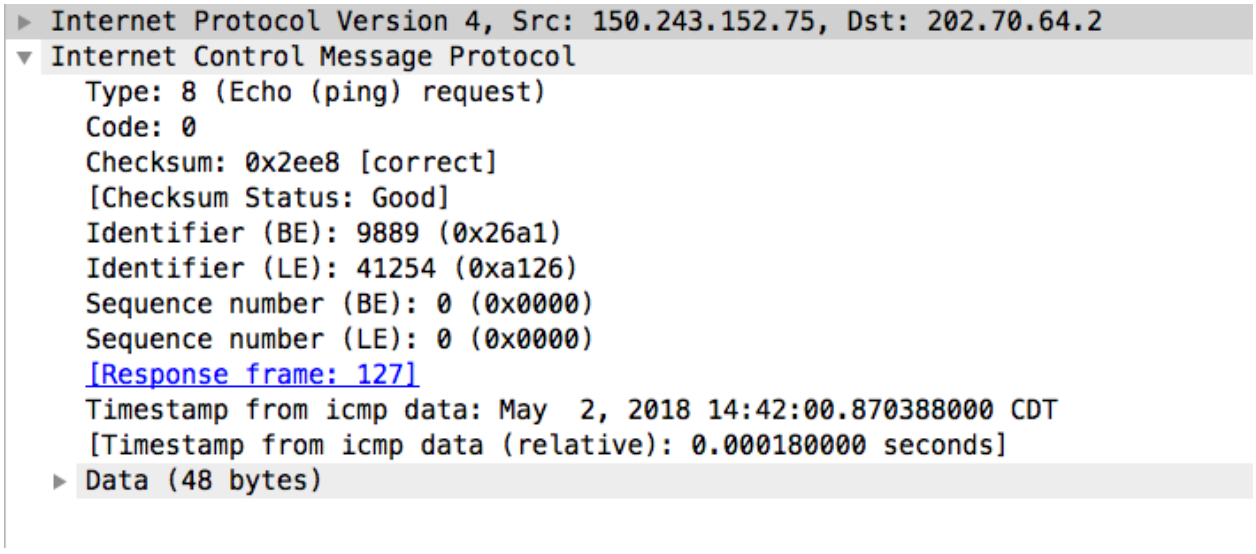
2. An ICMP packet does not have source and destination port numbers because ICMP only communicates between the host and the routers, not between the application layer processes. The network software itself interprets all ICMP messages, no port numbers are needed to direct the ICMP message to an application layer process.



3. The Ping request packet:

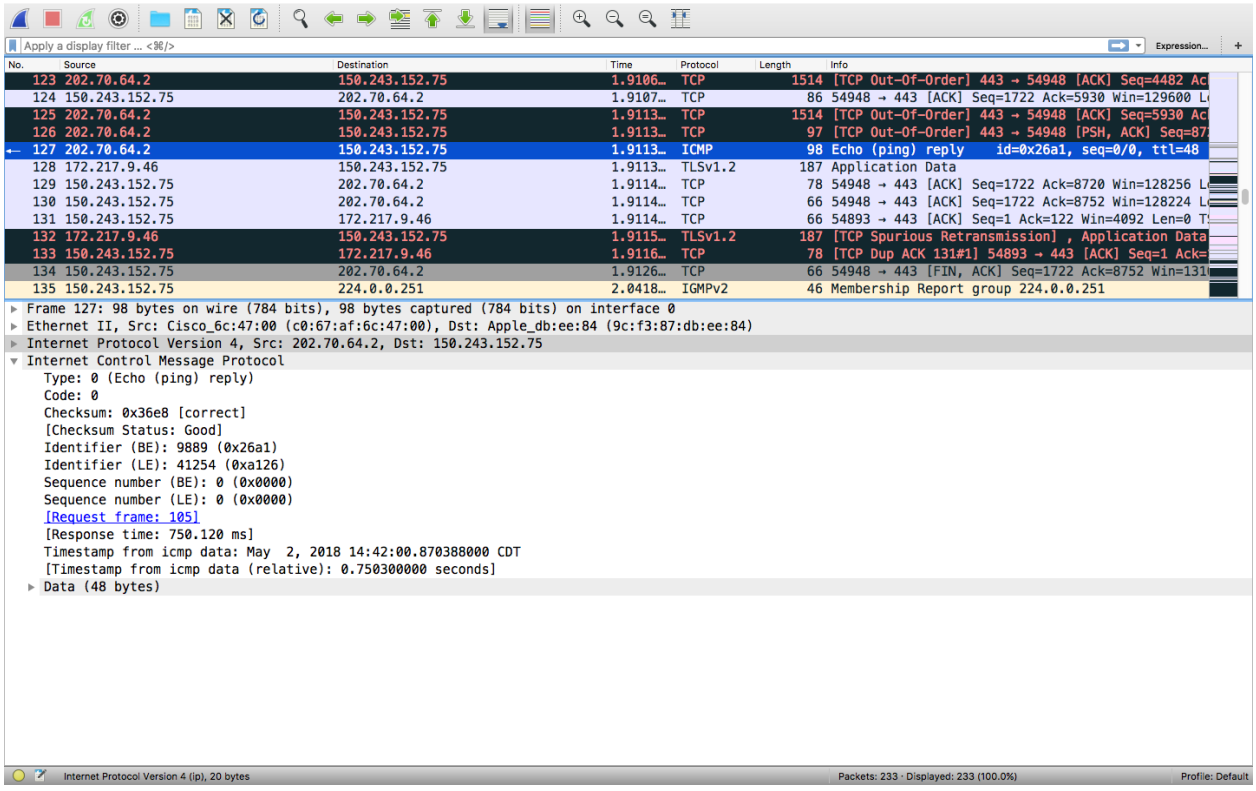
The ICMP type and code numbers are: 8 and 0. The ICMP packets have other fields like checksum, identifier, sequence numbers and timestamp. The checksum, identifier and sequence number each has 2 bytes.





4. The corresponding Ping reply Packet:

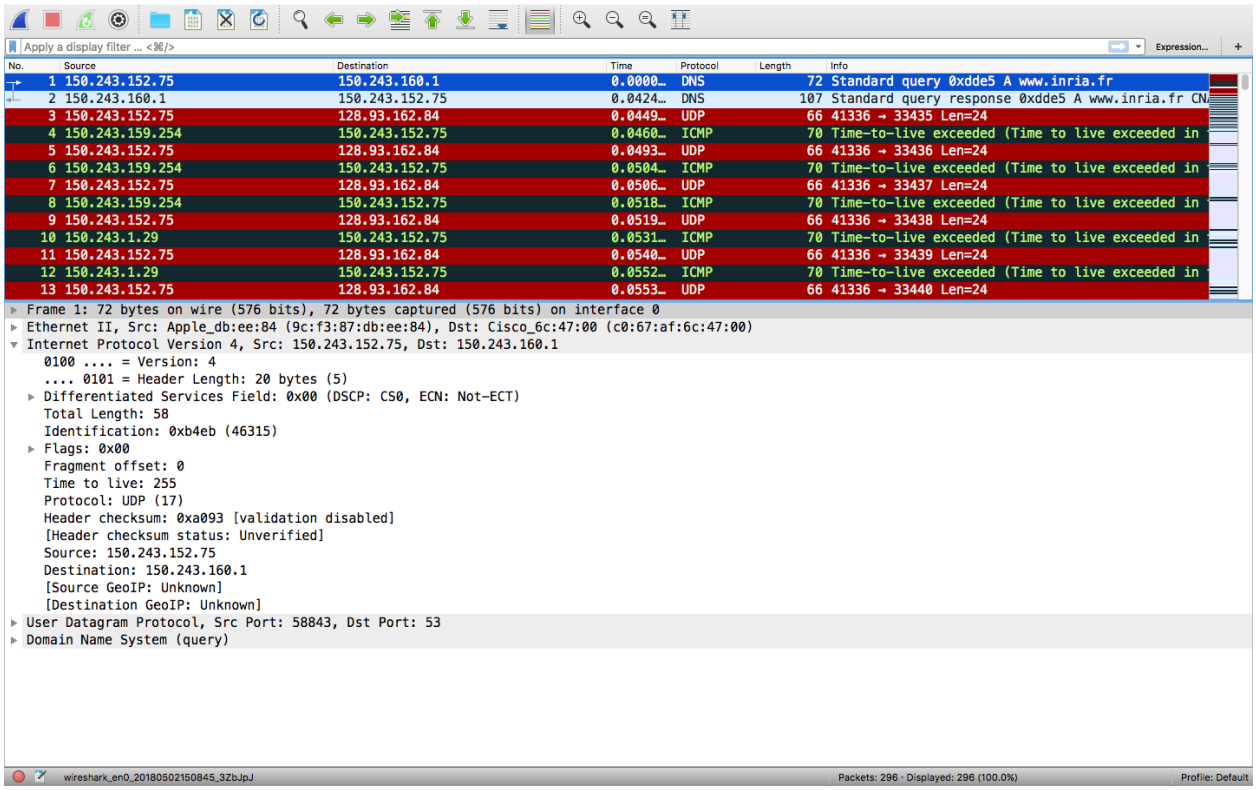
```
Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0x36e8 [correct]
    [Checksum Status: Good]
    Identifier (BE): 9889 (0x26a1)
    Identifier (LE): 41254 (0xa126)
    Sequence number (BE): 0 (0x0000)
    Sequence number (LE): 0 (0x0000)
    [Request frame: 105]
    [Response time: 750.120 ms]
    Timestamp from icmp data: May  2, 2018 14:42:00.870388000 CDT
    [Timestamp from icmp data (relative): 0.750300000 seconds]
```



The ICMP type and code are: 0,0. The ICMP packets have other fields like: checksum, identifier, sequence numbers and timestamps. the checksum, sequence number and identifier fields have 2 bytes.

The ICMP and TRACEROUTE

```
dikshyas-air:~ dikshya96$ traceroute www.inria.fr
traceroute to ezp3.inria.fr (128.93.162.84), 64 hops max, 52 byte packets
 1  150.243.159.254 (150.243.159.254)  1.756 ms  1.250 ms  1.272 ms
 2  150.243.1.29 (150.243.1.29)  1.292 ms  1.216 ms  1.321 ms
 3  209.152.144.201 (209.152.144.201)  9.389 ms  9.045 ms  11.675 ms
 4  * * bluebird-network.10gigabitethernet5-4.core1.mci3.he.net (216.66.74.38)  16.843 ms
 5  10ge4-4.core1.mci3.he.net (216.66.74.37)  17.772 ms  16.301 ms  24.439 ms
 6  100ge8-1.core2.chi1.he.net (184.105.81.210)  36.219 ms  28.455 ms  29.313 ms
 7  100ge16-1.core1.nyc4.he.net (184.105.223.162)  50.205 ms  44.854 ms  45.221 ms
 8  100ge4-1.core1.par2.he.net (184.105.81.78)  120.110 ms  122.756 ms  129.347 ms
 9  renater.equinix-ix.fr (195.42.145.38)  120.969 ms
    renater.par.franceix.net (37.49.236.19)  117.884 ms  117.407 ms
10  193.51.180.44 (193.51.180.44)  123.230 ms
    xe0-1-2-paris1-rtr-131.noc.renater.fr (193.51.177.88)  119.226 ms  124.270 ms
11  * * *
12  inria-rocquencourt-te1-4-inria-rtr-021.noc.renater.fr (193.51.184.177)  131.284 ms  117.066 ms  123.747 ms
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
```



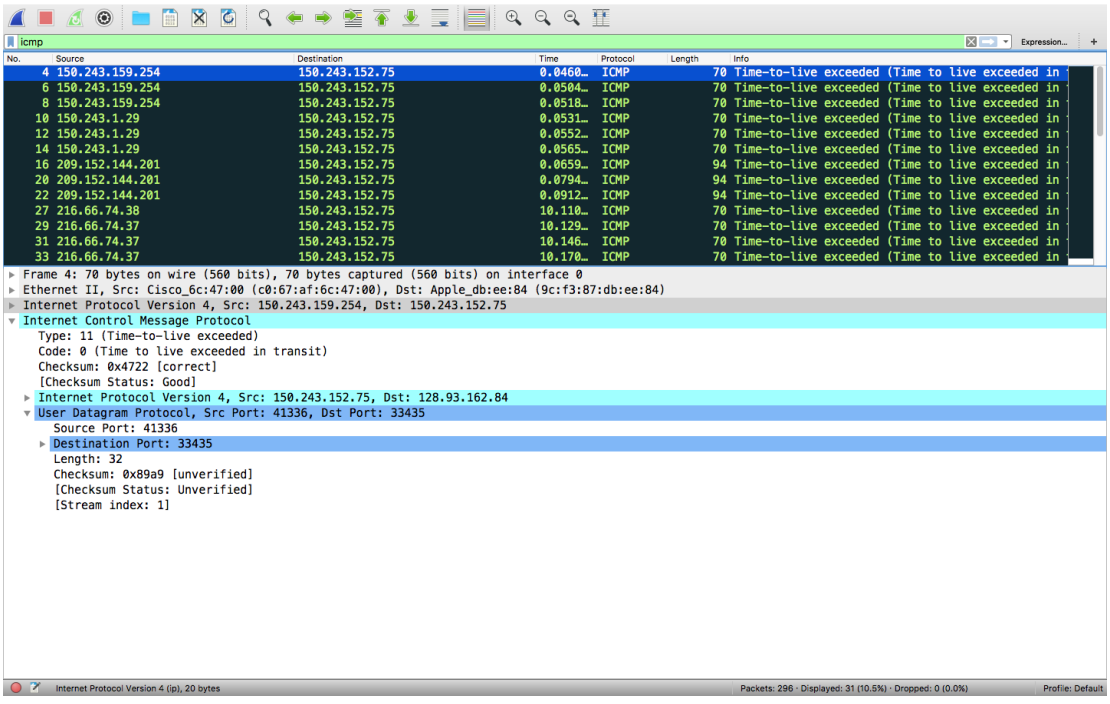
5. The IP address of the host and the target destination host is: 150.243.152.75 and 128.93.162.84

```
▼ Internet Protocol Version 4, Src: 150.243.152.75, Dst: 128.93.162.84
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 52
    Identification: 0xa19b (41371)
  ▶ Flags: 0x00
    Fragment offset: 0
    Time to live: 12
    Protocol: UDP (17)
    Header checksum: 0xbb2d [validation disabled]
    [Header checksum status: Unverified]
    Source: 150.243.152.75
    Destination: 128.93.162.84
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
▶ User Datagram Protocol, Src Port: 41336, Dst Port: 33469
▶ Data (24 bytes)
```

6. The IP protocol number will not be 01 for the probe packets instead it will be 17.

```
Flags: 0x00
Fragment offset: 0
Time to live: 12
Protocol: UDP (17)
Header checksum: 0xbb2e [validation disabled]
```

7. Yes, it is different from the ICMP ping query packets in the first half of this lab. It has just the “time-to-live extended” information

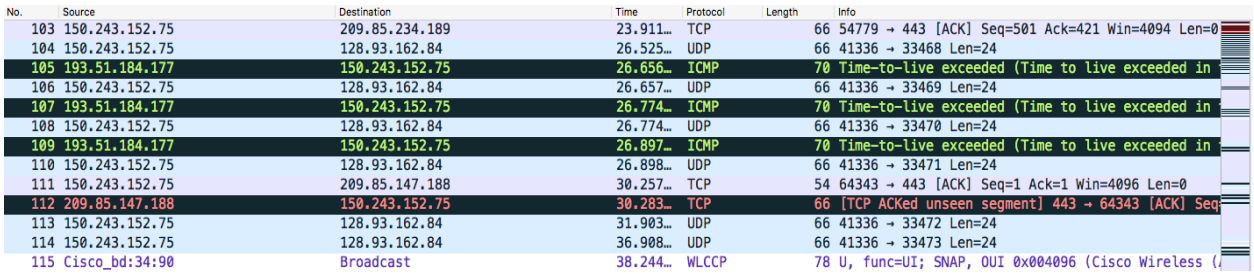


8. It includes the packet of the ping and also includes IPv4 and UDP.

```
▼ Internet Control Message Protocol
    Type: 11 (Time-to-live exceeded)
    Code: 0 (Time to live exceeded in transit)
    Checksum: 0x4722 [correct]
    [Checksum Status: Good]
  ▶ Internet Protocol Version 4, Src: 150.243.152.75, Dst: 128.93.162.84
  ▶ User Datagram Protocol, Src Port: 41336, Dst Port: 33435
```

9. It is different from the error packets as
    - the code =
    - type = 11
    - TTL = 245
    - Header checksum = unverified
    - Flags “don’t fragment” is not in the last three ICMP packet.

They are different as all the datagram made it to the host before TTl expired.



No.	Source	Destination	Time	Protocol	Length	Info
103	150.243.152.75	209.85.234.189	23.911…	TCP		66 54779 → 443 [ACK] Seq=501 Ack=421 Win=4094 Len=0
104	150.243.152.75	128.93.162.84	26.525…	UDP		66 41336 → 33468 Len=24
105	193.51.184.177	150.243.152.75	26.656…	ICMP		70 Time-to-live exceeded (Time to live exceeded in
106	150.243.152.75	128.93.162.84	26.657…	UDP		66 41336 → 33469 Len=24
107	193.51.184.177	150.243.152.75	26.774…	ICMP		70 Time-to-live exceeded (Time to live exceeded in
108	150.243.152.75	128.93.162.84	26.774…	UDP		66 41336 → 33470 Len=24
109	193.51.184.177	150.243.152.75	26.897…	ICMP		70 Time-to-live exceeded (Time to live exceeded in
110	150.243.152.75	128.93.162.84	26.898…	UDP		66 41336 → 33471 Len=24
111	150.243.152.75	209.85.147.188	30.257…	TCP		54 64343 → 443 [ACK] Seq=1 Ack=1 Win=4096 Len=0
112	209.85.147.188	150.243.152.75	30.283…	TCP		66 [TCP ACKed unseen segment] 443 → 64343 [ACK] Seq
113	150.243.152.75	128.93.162.84	31.903…	UDP		66 41336 → 33472 Len=24
114	150.243.152.75	128.93.162.84	36.908…	UDP		66 41336 → 33473 Len=24
115	Cisco_bd:34:90	Broadcast	38.244…	WLCCP		78 U, func=UI; SNAP, OUI 0x004096 (Cisco Wireless (

10. * Figure 4*

```
[dikshyas-air:~ dikshya96$ traceroute www.inria.fr
traceroute to ezp3.inria.fr (128.93.162.84), 64 hops max, 52 byte packets
 1  150.243.159.254 (150.243.159.254)  1.756 ms  1.250 ms  1.272 ms
 2  150.243.1.29 (150.243.1.29)  1.292 ms  1.216 ms  1.321 ms
 3  209.152.144.201 (209.152.144.201)  9.389 ms  9.045 ms  11.675 ms
 4  * * bluebird-network.10gigabitethernet5-4.core1.mci3.he.net (216.66.74.38)  16.843 ms
 5  10ge4-4.core1.mci3.he.net (216.66.74.37)  17.772 ms  16.301 ms  24.439 ms
 6  100ge8-1.core2.chi1.he.net (184.105.81.210)  36.219 ms  28.455 ms  29.313 ms
 7  100ge16-1.core1.nyc4.he.net (184.105.223.162)  50.205 ms  44.854 ms  45.221 ms
 8  100ge4-1.core1.par2.he.net (184.105.81.78)  120.110 ms  122.756 ms  129.347 ms
 9  renater.equinix-ix.fr (195.42.145.38)  120.969 ms
    renater.par.franceix.net (37.49.236.19)  117.884 ms  117.407 ms
10  193.51.180.44 (193.51.180.44)  123.230 ms
    xe0-1-2-paris1-rtr-131.noc.renater.fr (193.51.177.88)  119.226 ms  124.270 ms
11  * * *
12  inria-rocquencourt-te1-4-inria-rtr-021.noc.renater.fr (193.51.184.177)  131.284 ms  117.066 ms  123.747 ms
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
```

In Line 7 and Line 8, we can find a huge time difference. Line 8 has significantly longer delay. The IP address on Line 7; 184.105.223.162 is from New York, USA while the IP address in Line 8; 184.105.81.78 is from Paris, France. This is a transatlantic.