COMP416
PROJECT #03
PART 1

Emir Atışay
53590

Table of Contents

Introduction:

In this Project, it is aimed to get familiar with the network layer protocols and data plane of the layer. IP datagram, ICMP, fragmentation and DHCP concepts are analyzed and network traffic is observed to see the packet behaviors. WireShark and PingPlotter tools are used through the experiments.

Questions:

For the ICMP and fragmentation part, with the help of the pingplotter tool, traceroute executed, from my machine to www.unc.edu, assigned URL target, and packets are analyzed. Having various packet sizes for the experiment, default size of 56 byte is used in the first part of the experiment. While ICMP messages are exchanged, packets are captured and analyzed via WireShark.

ICMP:

1. Match each part (indicated in Figure 3) with its protocol stack (Physical / Data Link / Network / Transport / Application). (For instance, is Internet Control Message Protocol (ICMP) a physical layer protocol, or a data link layer protocol, or a network layer protocol? Likewise; Internet protocol, Ethernet II, and Frame1?)

At the beginning of the experiment, firstly, the first captured information is taken as in the Figure 3 of the project instruction. The capture which can be seen in Figure 1 is examined for answering this question. Frame 1 does not correspond to any layer, just an indicator, where Ethernet II to data link layer protocol and Internet Protocol to network layer protocol. Followingly, in my first observations, I assumed that ICMP is a transport layer protocol. However, as it has no source or destination as TCP and UDP, I decided to examine over it. In the conclusion of my research, I observed that the ICMP is network layer protocol even though it is encapsulated in the IP.

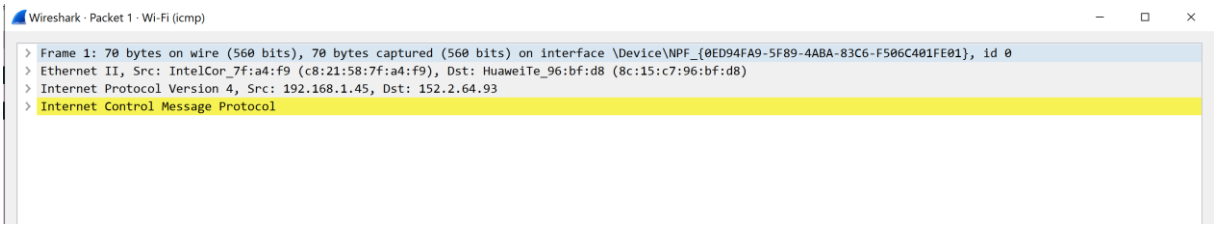


Figure 1

2. What is ICMP used for? Select the first ICMP Echo Request message sent by your computer (you will also use that message to answer question 4, 5 and 6), and expand the Internet Protocol part (and all subparts) of the packet in the packet details window. (Print the expanded window so that we can evaluate your answers from that output figure.) For this message, draw and fill the fields of IP datagram. (A typical IPv4 datagram format is available in the course book (Computer Networking: A Top-Down Approach, 6/E), Chapter 4)

Internet Control Message Protocol(ICMP) is a network layer protocol that used for detecting the network related problems. Through the destination from the source, ICMP used for utilizing the path, whether the routers or other devices are avaible or not, or in other words, whether the data can flow to destination or not.

Using the books Ipv4 datagram format, IP datagram can be filled as follows.

Version=4	Header Length=20	Service=0x00	Total Length=56
Identification= 0xda94		Flags=0x0000	Fragmentation offset=0
TTL=255	Protocol=ICMP	Header Checksum=0x46fb	
	Source=192.168.1.45		
	Destination=152.2.64.93		

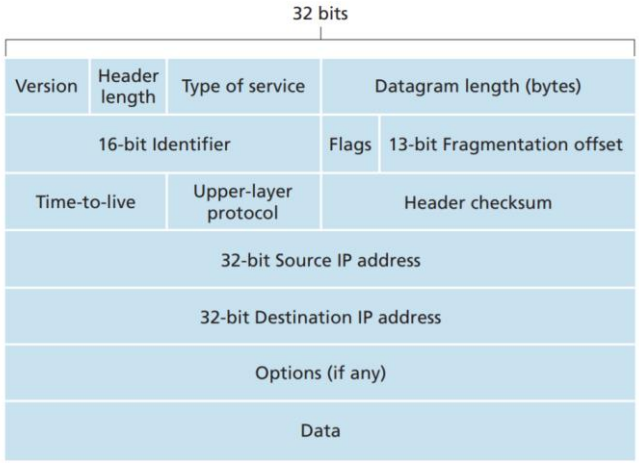


Figure 31[1]

```
Internet Protocol Version 4, Src: 192.168.1.45, Dst: 152.2.64.93
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 56
    Identification: 0xda94 (55956)
  > Flags: 0x0000
    Fragment offset: 0
    Time to live: 255
    Protocol: ICMP (1)
    Header checksum: 0x46fb [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.45
    Destination: 152.2.64.93
```

Figure 2

3. Why is it that an ICMP packet does not have source and destination port numbers? Examine one of the ping request packets sent by your host. What are the ICMP type and code numbers? What other fields does this ICMP packet have? How many bytes are the checksum, sequence number and identifier fields?

Rather than constructing a communication channel between two application layer processes, main objective of ICMP is to diagnose if there is a network related issue. In that manner, ICMP's communication is between the hosts and routers and ICMP does not need source and destination port numbers. From the Figure 3, one can see the typical echo request sent from my machine. Type of the ICMP is 8 and the code of it is 0. These are the main identifiers of the ICMP message and indiciates that it is a echo request. In addition to type and code, there are also checksum, identifier and sequence number fields which are both 2 bytes. And rest is the data of the ICMP message which is 28 bytes.

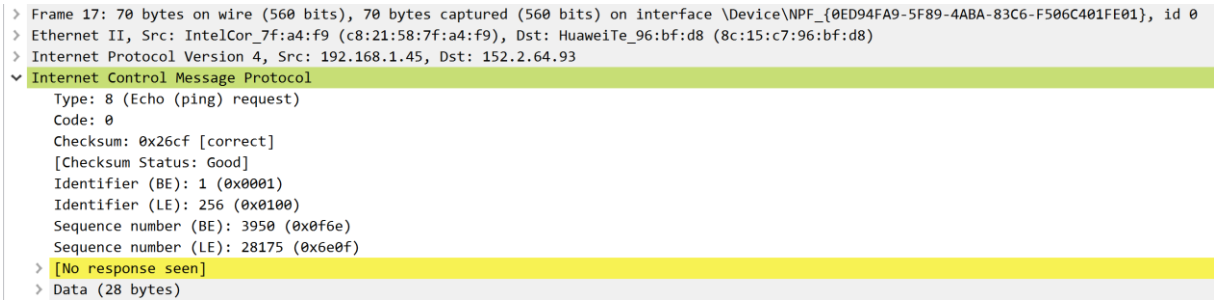


Figure 3

4. What is the IP address of your computer? What is the IP address of the nearest router?

From the figure 4 and 5, it can be seen that the IP address of my computer is 192.168.1.45. This information can be also obtained with the ipconfig command. In addition to that IP address, as far as I concern, 195.87.128.38 is the IP address of the nearest router as the figure 5, caption from the pingplotter, indicates it as the destination of the first hop.

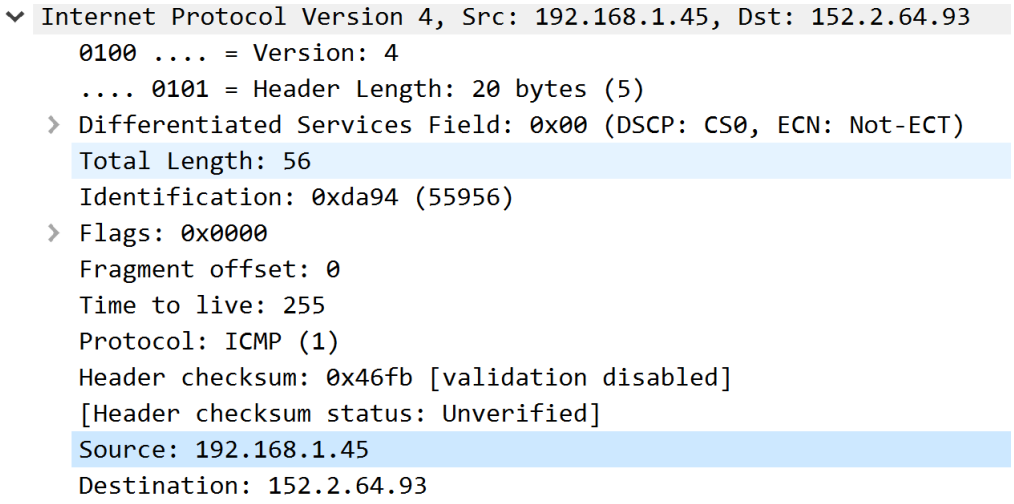


Figure 4

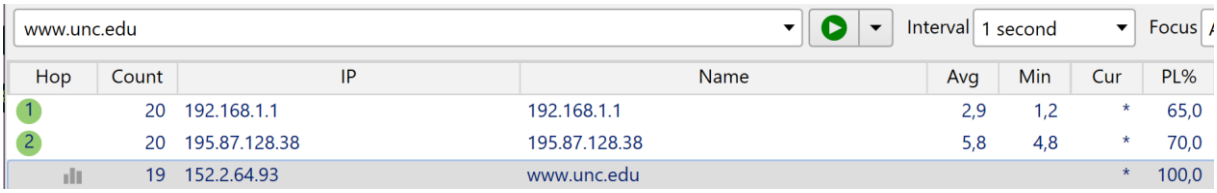


Figure 5

5. 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.

From the figure 4, it can be seen that there are 20 bytes in the IP header, while having 56 bytes in total. From that point, there are 36 bytes for the payload of the IP datagram.

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

The first echo request is examined for this question and this IP datagram is not been fragmented. In order to reach that idea, Fragment offset field is analyzed. From the Figure 4, it can be seen that it is 0.

7. If ICMP sent UDP packets instead (as in Unix/Linux), would the IP protocol number still be 01 for the probe packets? If not, what would it be?

If ICMP sent UDP packets, IP protocol number would not be 01 but 17, indicating that the protocol is UDP.

8. Which fields in the IP datagram always change from one datagram to the next within this series of ICMP messages sent by your computer? Which fields stay constant? Explain the reasons. 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.

The IP datagram holds version, header, differentiated services field, length, identification, flags, fragment offset, ttl, protocol, checksum, source and destination. Among those fields, identification, checksum and ttl changes in every packet. However, as the destination changes, a hop is observed, the total length also changes. While identification, checksum and ttl is packet specific and total length, destination is changeable, IP version, header, fragment offset, protocol and checksum is constant. As it can be seen, the main properties of the transaction stay constant through the exchange.

From the figures above, one can see the first TTL exceeded message from the IP address of 195.87.128.38, the nearest router and the rest of the messages.

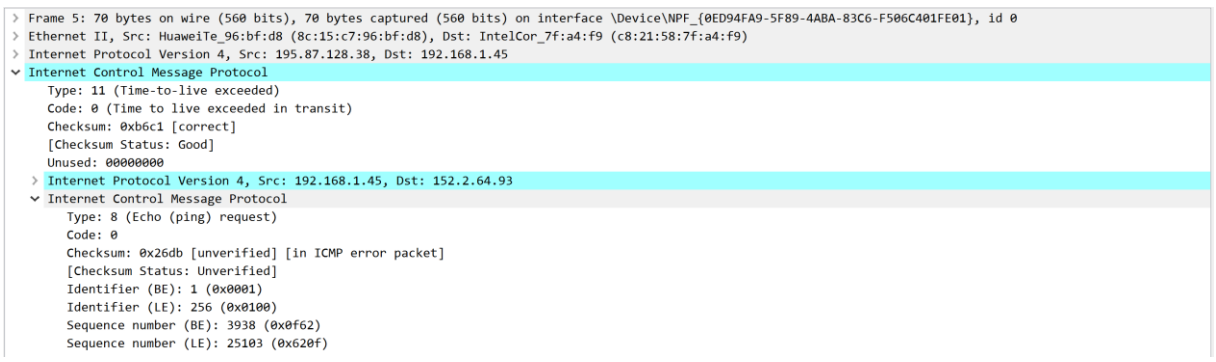
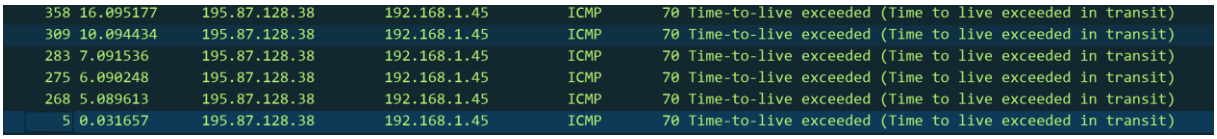


Figure 6

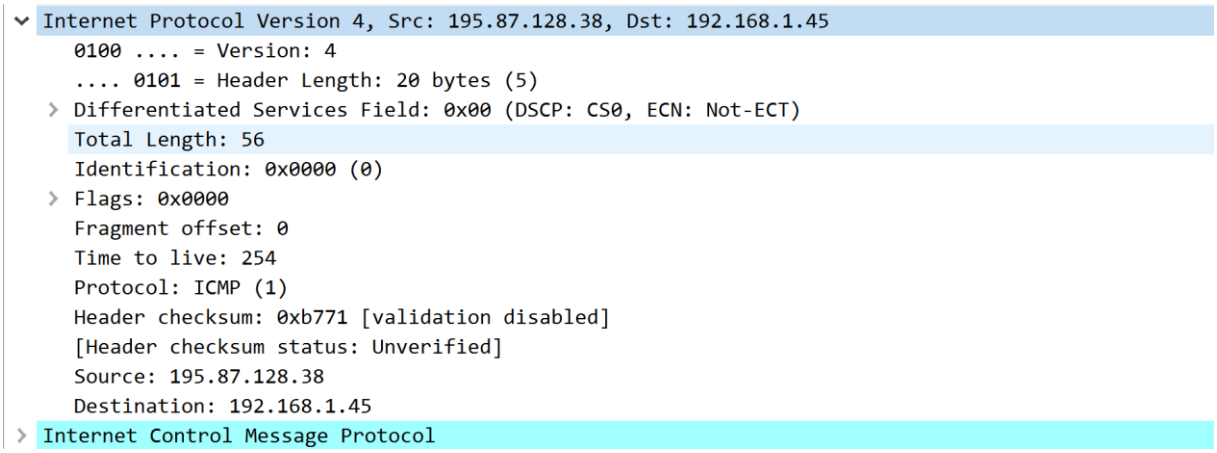


358	16.095177	195.87.128.38	192.168.1.45	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
309	10.094434	195.87.128.38	192.168.1.45	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
283	7.091536	195.87.128.38	192.168.1.45	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
275	6.090248	195.87.128.38	192.168.1.45	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
268	5.089613	195.87.128.38	192.168.1.45	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
5	0.031657	195.87.128.38	192.168.1.45	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)

Figure 7

9. What is the value in the Identification field and the TTL field? Do these values remain unchanged for all of the ICMP TTL-exceeded replies sent to your computer by the nearest (first hop) router? Explain the reason.

From the figure 8, it can be seen that identification field is 0 and ttl field is 254. Those values stays the same in all the TTL exceeded message from that router. TTL is same since the TTL for that hop router is same. However, I observed that some of Identification field are different in other hops. In my opinion, identification field can be source-destination tuple specific or can be indicating some type, which is not a high probability for me since it is used for unique packets in other packets.



```
Internet Protocol Version 4, Src: 195.87.128.38, Dst: 192.168.1.45
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 56
    Identification: 0x0000 (0)
  > Flags: 0x0000
    Fragment offset: 0
    Time to live: 254
    Protocol: ICMP (1)
    Header checksum: 0xb771 [validation disabled]
    [Header checksum status: Unverified]
    Source: 195.87.128.38
    Destination: 192.168.1.45
> Internet Control Message Protocol
```

Figure 8

Fragmentation:

For this part of the project, the packet size arranged to 2400 and 3450. General setting other than the packet size is same with the first part of the project.

10. Find the first ICMP Echo Request message that was sent by your computer after you changed the Packet Size (in pingplotter) to be 2400. Has that message been fragmented across more than one IP datagram?

The figure 9 shows the first echo request with the packet size of 2400. The main difference in such packet size the fragment offset. In that packet, fragment offset is specified as 1480. From that point, it is evidence that the message has been fragment across more than one IP datagram. However, from my observation, I saw that this is cont'd of the IPv4 protocol packet. And when I examined that packet, I see that it is also specified as ICMP as protocol in the IP datagram.

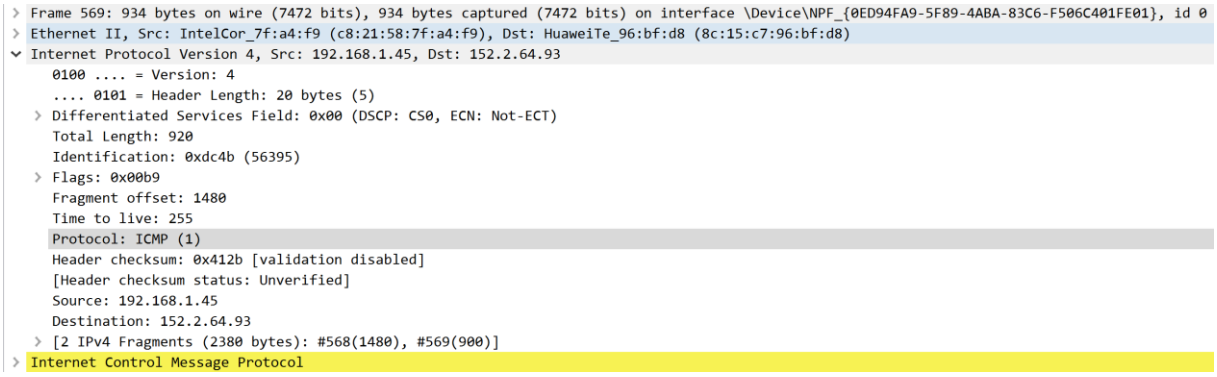


Figure 9

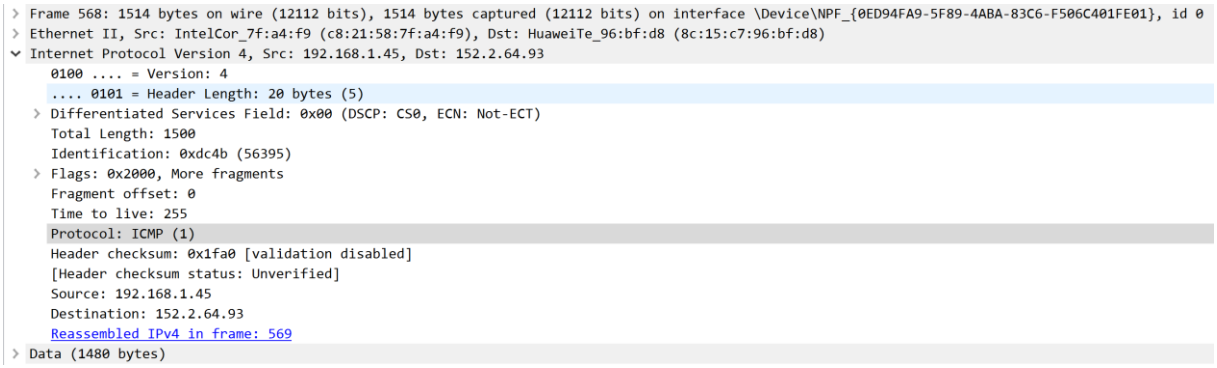


Figure 10

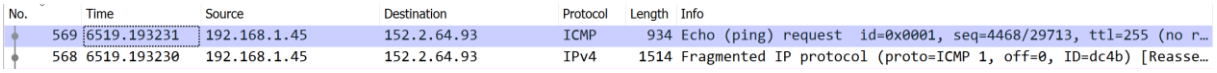


Figure 11

11. 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?

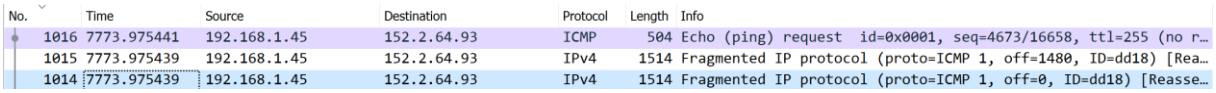
Figure 10 shows the first of the fragmented IP datagram. In the flags parts, it is indicated that more fragments are there. And as the fragment offset is 0, it basically tells that it is the first of the fragments. As it can be seen, this IP datagram has total length of 1500 and 1480 of it is the data wanted to be transferred.

12. 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?

Figure 10, which is also used in question 10, is the second fragment of the IP datagram. Since no more fragments are specified in the flag field, it can be said that it is the last fragment. Moreover, as the fragment offset is 1480, it can not be the first of the fragments. Also, the ICMP message is contained in that fragment. It is another evidence that it is the last fragment.

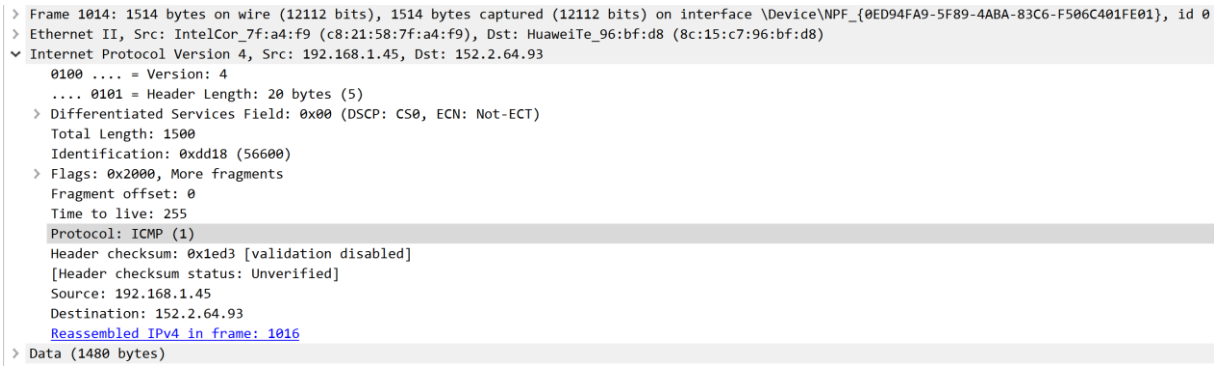
13. Find the first ICMP Echo Request message that was sent by your computer after you changed the Packet Size to 3450. How many fragments were created from the original datagram? What fields change in the IP header among the fragments?

The message is fragmented into 3 while having a packet size of 3450, figure 12. The first two fragments have flag of “more fragments”, while the last one does not have any flag. In addition, as the payload is 1480, offset values are increasing by 1480, while the first fragment has it as 0. Also, only the last fragment has the ICMP message, while all of them are specified as ICMP as protocol. Moreover, first two fragment has indicator of reassembling frame in Wireshark, which can be helpful in more packet size.



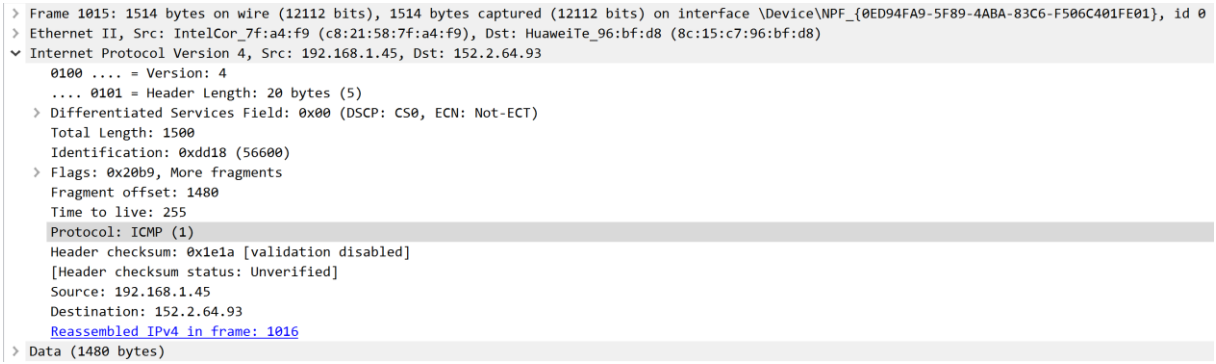
No.	Time	Source	Destination	Protocol	Length	Info
1016	7773.975441	192.168.1.45	152.2.64.93	ICMP	504	Echo (ping) request id=0x0001, seq=4673/16658, ttl=255 (no r…
1015	7773.975439	192.168.1.45	152.2.64.93	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=dd18) [Rea…
1014	7773.975439	192.168.1.45	152.2.64.93	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=dd18) [Reasse…

Figure 12



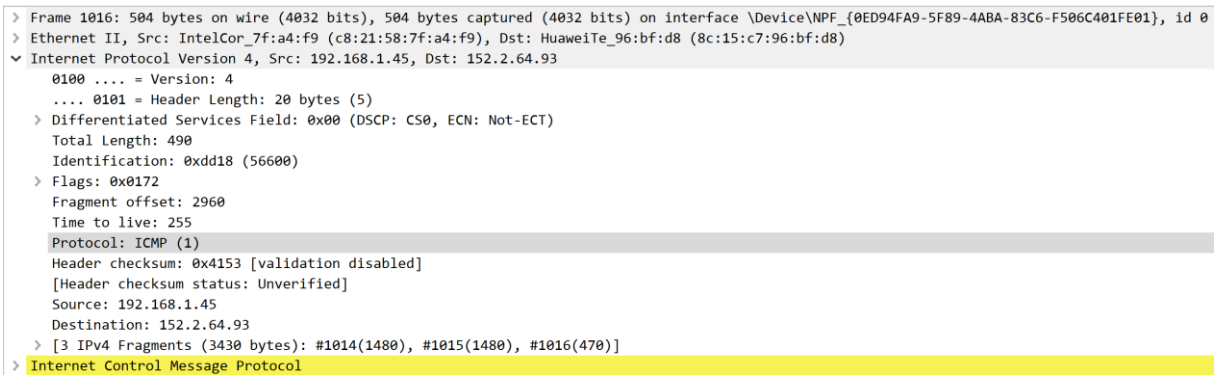
```
> Frame 1014: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface \Device\NPF_{0ED94FA9-5F89-4ABA-83C6-F506C401FE01}, id 0
> Ethernet II, Src: IntelCor_7f:a4:f9 (c8:21:58:7f:a4:f9), Dst: HuaweiTe_96:bf:d8 (8c:15:c7:96:bf:d8)
v Internet Protocol Version 4, Src: 192.168.1.45, Dst: 152.2.64.93
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1500
    Identification: 0xdd18 (56600)
  > Flags: 0x2000, More fragments
    Fragment offset: 0
    Time to live: 255
    Protocol: ICMP (1)
    Header checksum: 0x1ed3 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.45
    Destination: 152.2.64.93
    Reassembled IPv4 in frame: 1016
> Data (1480 bytes)
```

Figure 13



```
> Frame 1015: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface \Device\NPF_{0ED94FA9-5F89-4ABA-83C6-F506C401FE01}, id 0
> Ethernet II, Src: IntelCor_7f:a4:f9 (c8:21:58:7f:a4:f9), Dst: HuaweiTe_96:bf:d8 (8c:15:c7:96:bf:d8)
v Internet Protocol Version 4, Src: 192.168.1.45, Dst: 152.2.64.93
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1500
    Identification: 0xdd18 (56600)
  > Flags: 0x20b9, More fragments
    Fragment offset: 1480
    Time to live: 255
    Protocol: ICMP (1)
    Header checksum: 0x1e1a [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.45
    Destination: 152.2.64.93
    Reassembled IPv4 in frame: 1016
> Data (1480 bytes)
```

Figure 14



```
> Frame 1016: 504 bytes on wire (4032 bits), 504 bytes captured (4032 bits) on interface \Device\NPF_{0ED94FA9-5F89-4ABA-83C6-F506C401FE01}, id 0
> Ethernet II, Src: IntelCor_7f:a4:f9 (c8:21:58:7f:a4:f9), Dst: HuaweiTe_96:bf:d8 (8c:15:c7:96:bf:d8)
v Internet Protocol Version 4, Src: 192.168.1.45, Dst: 152.2.64.93
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 490
    Identification: 0xdd18 (56600)
  > Flags: 0x0172
    Fragment offset: 2960
    Time to live: 255
    Protocol: ICMP (1)
    Header checksum: 0x4153 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.45
    Destination: 152.2.64.93
  > [3 IPv4 Fragments (3430 bytes): #1014(1480), #1015(1480), #1016(470)]
> Internet Control Message Protocol
```

Figure 15

Question:

During this experiment, I see that the route to destination gets longer. What is man reason behind? Is it because of the control plane issues, the shortest path?

DHCP Experiment:

For this part of the experiment, already assigned IP address are released from my machine and several renew command have given. Followingly, after initializing the IP address, capturing via WireShark is done to observe the total DHCP process. In addition to the IP assignment, address release request is also tested and observed.

14. Are DHCP messages sent over UDP or TCP? What is the link-layer (e.g., Ethernet) address of your host?

Figure 16 shows all the DHCP packets. An the following figure 17 is the packet details of the first DHCP packet. From that figure, one can see that the IP datagram has 17 as protocol indicator, which is UDP. In that sense, it can be conclude that the DHCP messages are sent over UDP. Followingly, figure 18 is indicating the link layer protocol details. To have a comparison on the link layer, another screenshot is taken from the previous experiment of ICMP which is figure 19. As it can be expected, there is no destination in DHCP packet, but broadcast. In addition, the ethernet address can be seen as c8:21:58:7f:a4:f9 or IntelCor_7f:a4:f9. As it assigned to the machine, it was the same in the previous experiment.

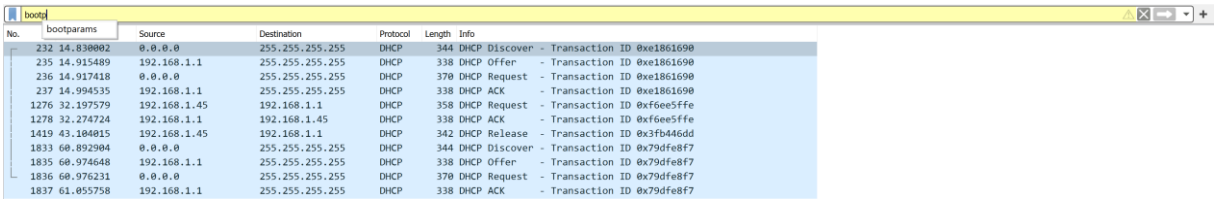


Figure 16

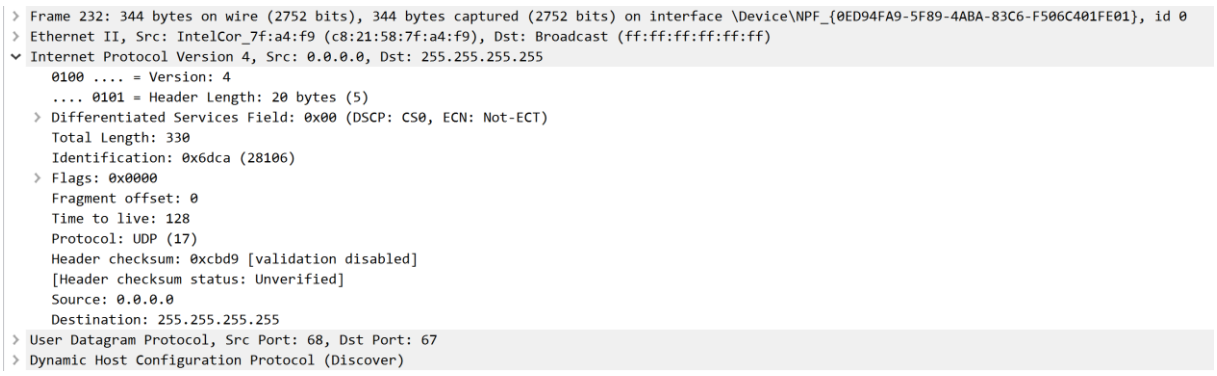


Figure 17

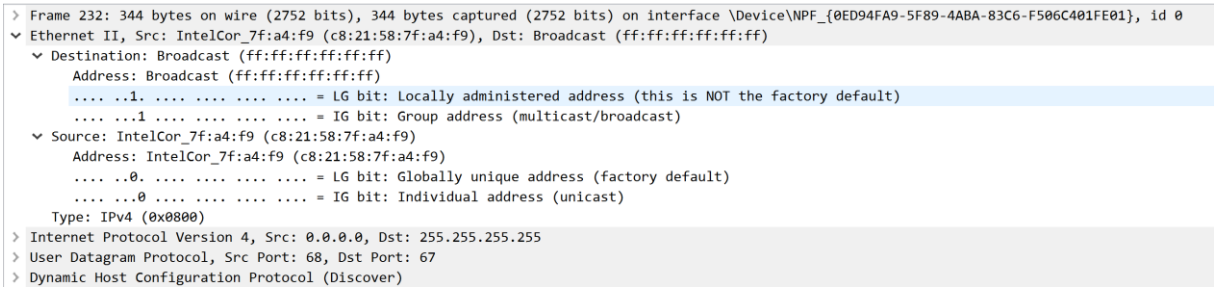


Figure 18

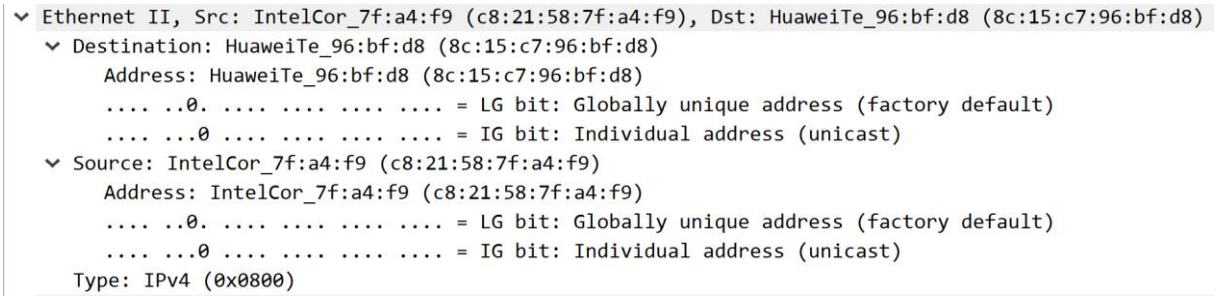


Figure 19

15. What is the value of the Transaction-ID in each of the first four (Discover/Offer/Request/ ACK) DHCP messages? What are the values of the TransactionID in the second set (Request/ACK) set of DHCP messages? What is the purpose of the Transaction-ID field?

In the first 4 packets, respectively Discover, Offer, Request, ACK DHCP messages, the Transaction ID's are same and it is 0xe1861690. On the other hand, in the following tuple of messages, Transaction ID is 0xf6ee5ffe. As far as I concern, those IDs are used for grouping the messages of the same requests of user together and assigning the requests and responses.

16. A host uses DHCP to obtain an IP address, among other things. But a host's IP address is not confirmed until the end of the four-message exchange! If the IP address is not set until the end of the four-message exchange, then what values are used in the IP datagrams in the four-message exchange? For each of the four DHCP messages (Discover/Offer/Request/ACK DHCP), indicate the source and destination IP addresses that are carried in the encapsulating IP datagram.

The following figure are the IPv4 protocol details of the packets Discover, Offer, Request and ACK messages. As it can be seen, the source IP addresses of packets sent from my host are 0.0.0.0. And the IP address of 192.168.1.1 is specified in the server messages, Offer and ACK. Both messages have the destination address of 255.255.255.255 which is broadcast address. After communicating over broadcasting, IP address assignment is done and they have became to use the direct IP addresses, Figure 24: second party of DHCP messages.

```
> Frame 232: 344 bytes on wire (2752 bits), 344 bytes captured (2752 bits) on interface \Device\NPF_{0ED94FA9-5F89-4ABA-83C6-F506C401FE01}, id 0
> Ethernet II, Src: IntelCor_7f:a4:f9 (c8:21:58:7f:a4:f9), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
v Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 330
    Identification: 0x6dca (28106)
  > Flags: 0x0000
    Fragment offset: 0
    Time to live: 128
    Protocol: UDP (17)
    Header checksum: 0xcbd9 [validation disabled]
    [Header checksum status: Unverified]
    Source: 0.0.0.0
    Destination: 255.255.255.255
> User Datagram Protocol, Src Port: 68, Dst Port: 67
> Dynamic Host Configuration Protocol (Discover)
```

Figure 20

```
> Frame 235: 338 bytes on wire (2704 bits), 338 bytes captured (2704 bits) on interface \Device\NPF_{0ED94FA9-5F89-4ABA-83C6-F506C401FE01}, id 0
> Ethernet II, Src: HuaweiTe_96:bf:d8 (8c:15:c7:96:bf:d8), Dst: IntelCor_7f:a4:f9 (c8:21:58:7f:a4:f9)
v Internet Protocol Version 4, Src: 192.168.1.1, Dst: 255.255.255.255
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 324
    Identification: 0x0000 (0)
  > Flags: 0x0000
    Fragment offset: 0
    Time to live: 64
    Protocol: UDP (17)
    Header checksum: 0xb800 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.1
    Destination: 255.255.255.255
> User Datagram Protocol, Src Port: 67, Dst Port: 68
> Dynamic Host Configuration Protocol (Offer)
```

Figure 21

```
> Frame 236: 370 bytes on wire (2960 bits), 370 bytes captured (2960 bits) on interface \Device\NPF_{0ED94FA9-5F89-4ABA-83C6-F506C401FE01}, id 0
> Ethernet II, Src: IntelCor_7f:a4:f9 (c8:21:58:7f:a4:f9), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
v Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 356
    Identification: 0x6dcb (28107)
  > Flags: 0x0000
    Fragment offset: 0
    Time to live: 128
    Protocol: UDP (17)
    Header checksum: 0xcbbe [validation disabled]
    [Header checksum status: Unverified]
    Source: 0.0.0.0
    Destination: 255.255.255.255
> User Datagram Protocol, Src Port: 68, Dst Port: 67
> Dynamic Host Configuration Protocol (Request)
```

Figure 22

```
> Frame 237: 338 bytes on wire (2704 bits), 338 bytes captured (2704 bits) on interface \Device\NPF_{0ED94FA9-5F89-4ABA-83C6-F506C401FE01}, id 0
> Ethernet II, Src: HuaweiTe_96:bf:d8 (8c:15:c7:96:bf:d8), Dst: IntelCor_7f:a4:f9 (c8:21:58:7f:a4:f9)
v Internet Protocol Version 4, Src: 192.168.1.1, Dst: 255.255.255.255
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 324
    Identification: 0x0000 (0)
  > Flags: 0x0000
    Fragment offset: 0
    Time to live: 64
    Protocol: UDP (17)
    Header checksum: 0xb800 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.1
    Destination: 255.255.255.255
> User Datagram Protocol, Src Port: 67, Dst Port: 68
> Dynamic Host Configuration Protocol (ACK)
```

Figure 23

```
  1276 32.197579    192.168.1.45    192.168.1.1     DHCP    358 DHCP Request  - Transaction ID 0xf6ee5ffe
  1278 32.274724    192.168.1.1     192.168.1.45    DHCP    338 DHCP ACK      - Transaction ID 0xf6ee5ffe
```

Figure 24

17. Explain the purpose of the router and subnet mask lines in the DHCP offer message.

The router tells the user where to send messages, request, ie server's own address. Followingly, Subnet Mask tells the user the assigned subnet mask.

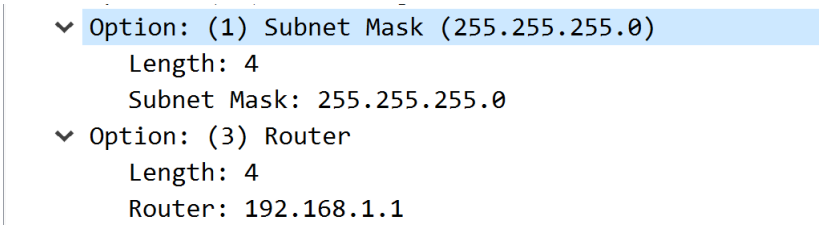


Figure 25

18. Explain the purpose of the lease time. How long is the lease time in your experiment?

Lease time is the time allocated for the use of the IP address to the client. In my experiment, it is 86400 seconds or 1 day in another words.

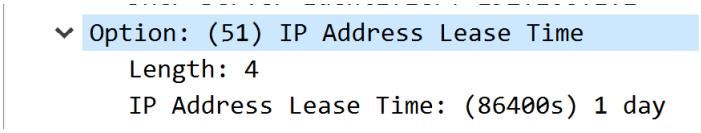


Figure 26

19. What is the purpose of the DHCP release message? Does the DHCP server issue an acknowledgement of receipt of the client's DHCP request? What would happen if the client's DHCP release message is lost?

The main purpose of the DHCP release message is to release the assigned IP address from the server. Details of that packet can be seen in the figure 27. Followingly, since there is no ACK message from the server after the release message, figure 28, it can be said that DHCP server does not issue the acknowledgement of release requests.

If client's message is lost, server could not be informed about the IP address' possession. However, client will be already renounced that IP address, but server will wait for lease time to unassign the IP address.

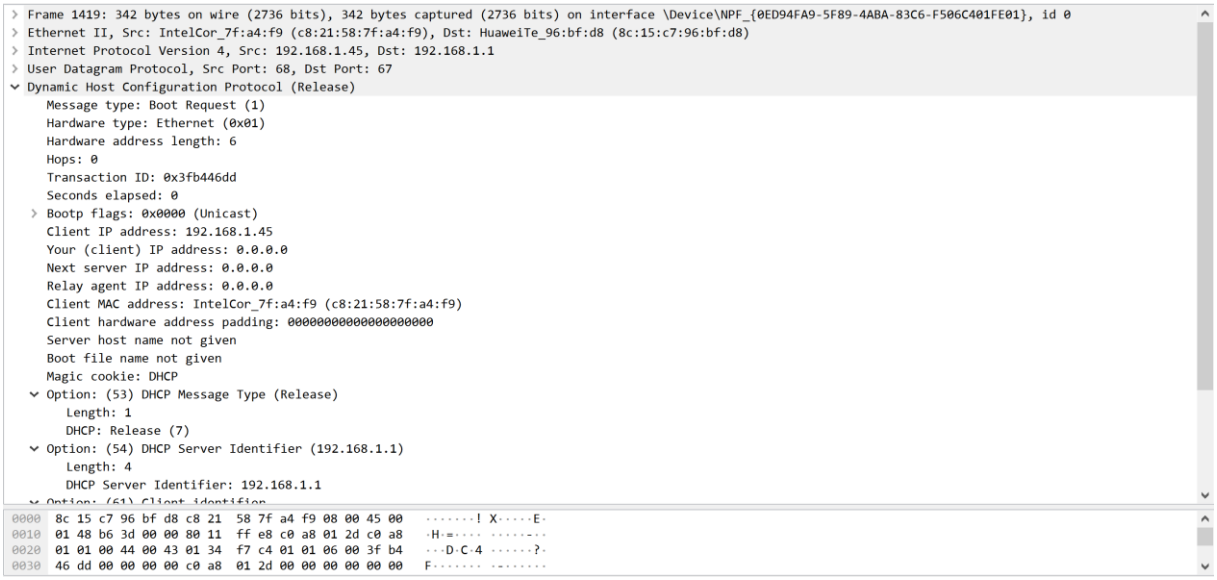


Figure 27

Figure 28

20. Clear the bootp filter from your Wireshark window. Were any ARP packets sent or received during the DHCP packet-exchange period? If so, explain the purpose of those ARP packets.

After dropping the filter, some Address Resolution Protocol (ARP) packets are observed during the exchange period. Packet details can be seen in figure 30. In that figure on can see that t is broadcast message by the HuaweiTe_96:bf:d8 (8c:15:c7:96:bf:d8). From my observations, I see that ARP protocol used for the IP address to Ethernet address conversion. In my opinion, this packet is the broadcast for checking whether the corresponding IP address is occupied or not.

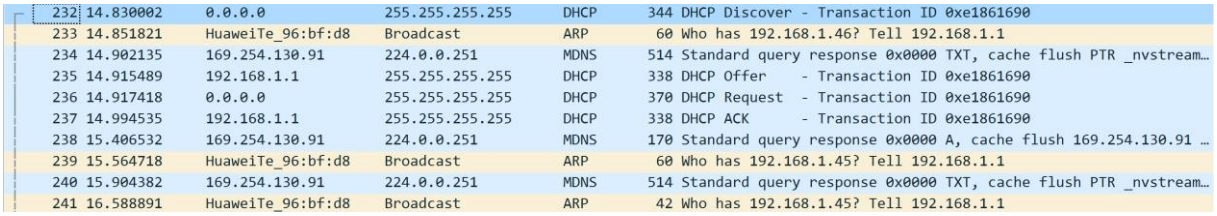


Figure 29

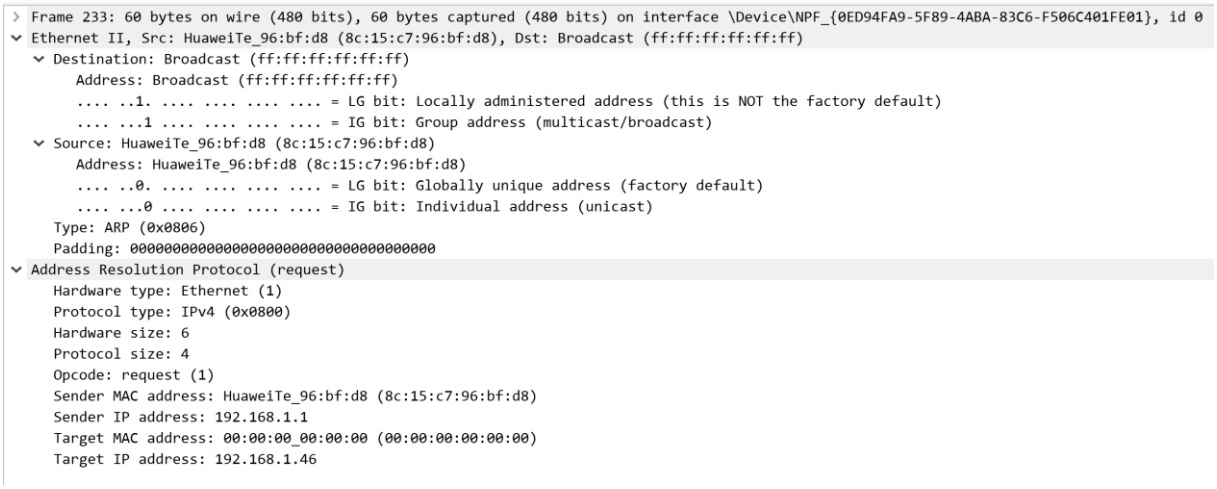


Figure 30

Conclusion:

Network layer concepts are analyzed in this project. For this purpose, the assigned URL target was www.unc.edu and packet exchange traffic between my machine and that domain are observed. In that traffic, ICMP protocol packets is observed at first. Followingly, different packet sizes are tried to be exchanged with the purpose of fragmentation observation. In the last part, DHCP experiment is done to observe the IP address assignment and release requests. As a result of the experiments, network layer is now visualizable for me and network traffic, flow between the layers are observed.

References:

[1]: Compuer Networking A Top-Down Approach 6th Edition; Kurose, Ross