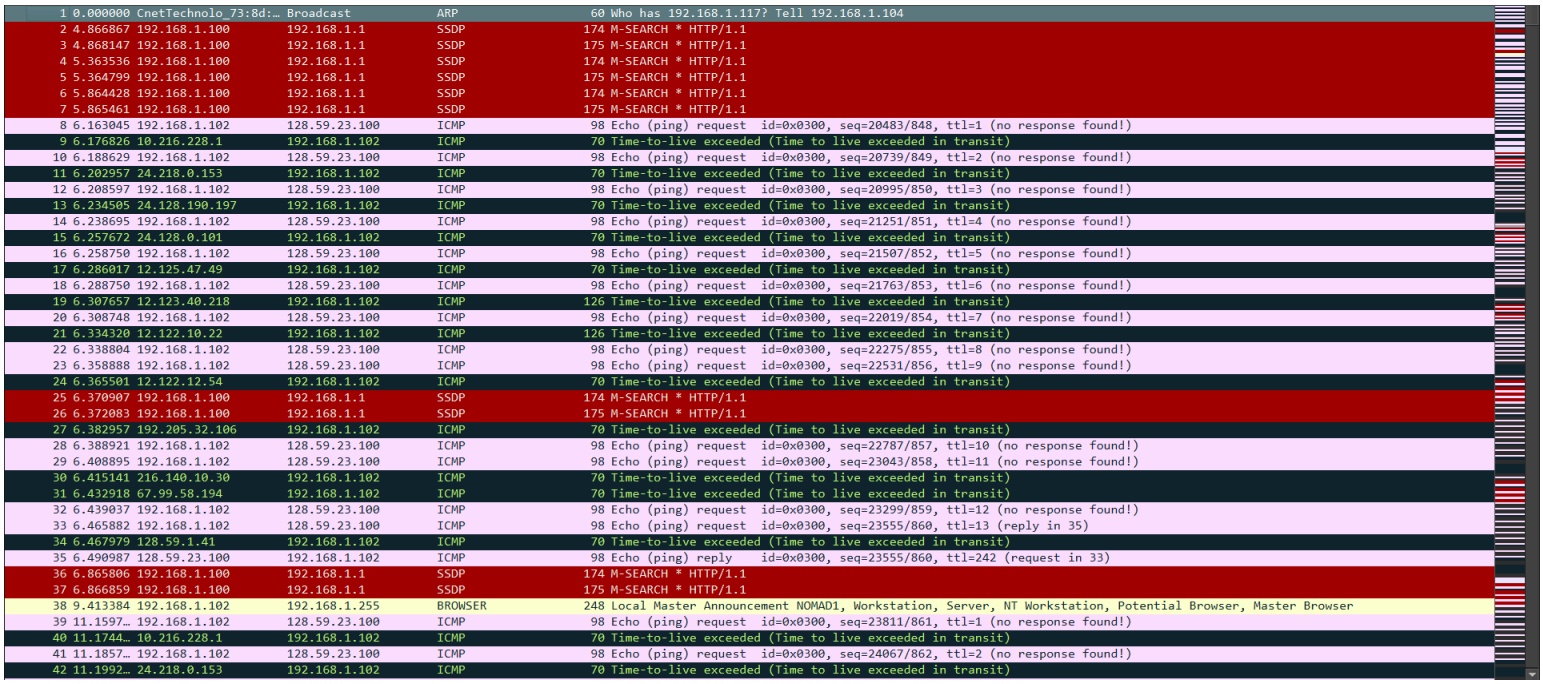
Lab 04 - Lauren Escobedo - CS 6330 A

Setup: Using pingplotter or traceroute, with Wireshark capturing packets, run the following commands: 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.

a. Since I have a Linux virtual machine, I will not bother with pingplotter on Windows, I will be using traceroute. I run the specified traceroute commands while capturing packets with Wireshark, but for some reason I end up with 36,000+ packets captured. I'm on my mobile hotspot so maybe this is the cause, but either way I will instead be using the packet capture provided by the book for the next questions.



1	0.000000	CnetTechnolo_73:8d:…	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	SSDP	174 M-SEARCH * HTTP/1.1
3	4.868147	192.168.1.100	192.168.1.1	SSDP	175 M-SEARCH * HTTP/1.1
4	5.363536	192.168.1.100	192.168.1.1	SSDP	174 M-SEARCH * HTTP/1.1
5	5.364799	192.168.1.100	192.168.1.1	SSDP	175 M-SEARCH * HTTP/1.1
6	5.864428	192.168.1.100	192.168.1.1	SSDP	174 M-SEARCH * HTTP/1.1
7	5.865461	192.168.1.100	192.168.1.1	SSDP	175 M-SEARCH * HTTP/1.1
8	6.163045	192.168.1.102	128.59.23.100	ICMP	98 Echo (ping) request id=0x0300, seq=20483/848, ttl=1 (no response found!)
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 (no response found!)
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 (no response found!)
13	6.234505	24.128.190.197	192.168.1.102	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
14	6.238695	192.168.1.102	128.59.23.100	ICMP	98 Echo (ping) request id=0x0300, seq=21251/851, ttl=4 (no response found!)
15	6.257672	24.128.0.101	192.168.1.102	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
16	6.258750	192.168.1.102	128.59.23.100	ICMP	98 Echo (ping) request id=0x0300, seq=21507/852, ttl=5 (no response found!)
17	6.286017	12.125.47.49	192.168.1.102	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
18	6.288750	192.168.1.102	128.59.23.100	ICMP	98 Echo (ping) request id=0x0300, seq=21763/853, ttl=6 (no response found!)
19	6.307657	12.123.40.218	192.168.1.102	ICMP	126 Time-to-live exceeded (Time to live exceeded in transit)
20	6.308748	192.168.1.102	128.59.23.100	ICMP	98 Echo (ping) request id=0x0300, seq=22019/854, ttl=7 (no response found!)
21	6.334320	12.122.10.22	192.168.1.102	ICMP	126 Time-to-live exceeded (Time to live exceeded in transit)
22	6.338804	192.168.1.102	128.59.23.100	ICMP	98 Echo (ping) request id=0x0300, seq=22275/855, ttl=8 (no response found!)
23	6.358888	192.168.1.102	128.59.23.100	ICMP	98 Echo (ping) request id=0x0300, seq=22531/856, ttl=9 (no response found!)
24	6.365501	12.122.12.54	192.168.1.102	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
25	6.370907	192.168.1.100	192.168.1.1	SSDP	174 M-SEARCH * HTTP/1.1
26	6.372083	192.168.1.100	192.168.1.1	SSDP	175 M-SEARCH * HTTP/1.1
27	6.382957	192.205.32.106	192.168.1.102	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
28	6.388921	192.168.1.102	128.59.23.100	ICMP	98 Echo (ping) request id=0x0300, seq=22787/857, ttl=10 (no response found!)
29	6.408895	192.168.1.102	128.59.23.100	ICMP	98 Echo (ping) request id=0x0300, seq=23043/858, ttl=11 (no response found!)
30	6.415141	216.140.10.30	192.168.1.102	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
31	6.432918	67.99.58.194	192.168.1.102	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
32	6.439037	192.168.1.102	128.59.23.100	ICMP	98 Echo (ping) request id=0x0300, seq=23299/859, ttl=12 (no response found!)
33	6.465882	192.168.1.102	128.59.23.100	ICMP	98 Echo (ping) request id=0x0300, seq=23555/860, ttl=13 (reply in 35)
34	6.467979	128.59.1.41	192.168.1.102	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
35	6.490987	128.59.23.100	192.168.1.102	ICMP	98 Echo (ping) reply id=0x0300, seq=23555/860, ttl=242 (request in 33)
36	6.865806	192.168.1.100	192.168.1.1	SSDP	174 M-SEARCH * HTTP/1.1
37	6.866859	192.168.1.100	192.168.1.1	SSDP	175 M-SEARCH * HTTP/1.1
38	9.413384	192.168.1.102	192.168.1.255	BROWSER	248 Local Master Announcement NOMAD1, Workstation, Server, NT Workstation, Potential Browser, Master Browser
39	11.1597…	192.168.1.102	128.59.23.100	ICMP	98 Echo (ping) request id=0x0300, seq=23811/861, ttl=1 (no response found!)
40	11.1744…	10.216.228.1	192.168.1.102	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
41	11.1857…	192.168.1.102	128.59.23.100	ICMP	98 Echo (ping) request id=0x0300, seq=24067/862, ttl=2 (no response found!)
42	11.1992…	24.218.0.153	192.168.1.102	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)

Figure 1: Wireshark capture provided by the lab website

1. Select the first ICMP Echo Request message sent by your computer, and expand the IP part of the packet.

a. The IP address of the host computer is 192.168.1.102

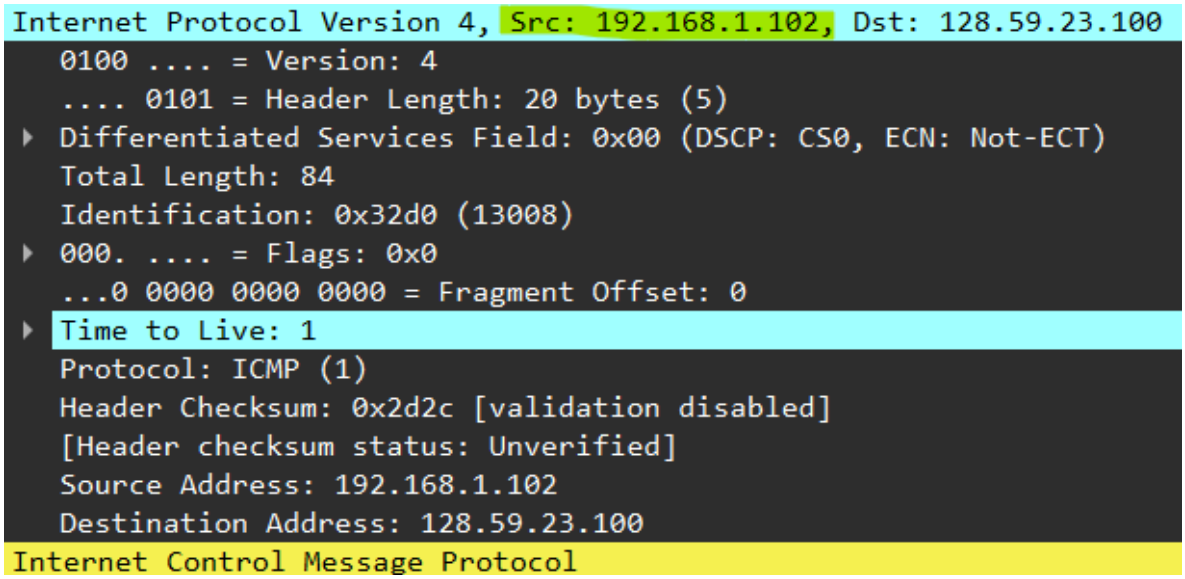


Figure 2: IP header of the first ICMP Echo Request packet with host IP highlighted

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

a. The protocol in the upper layer protocol field is ICMP.

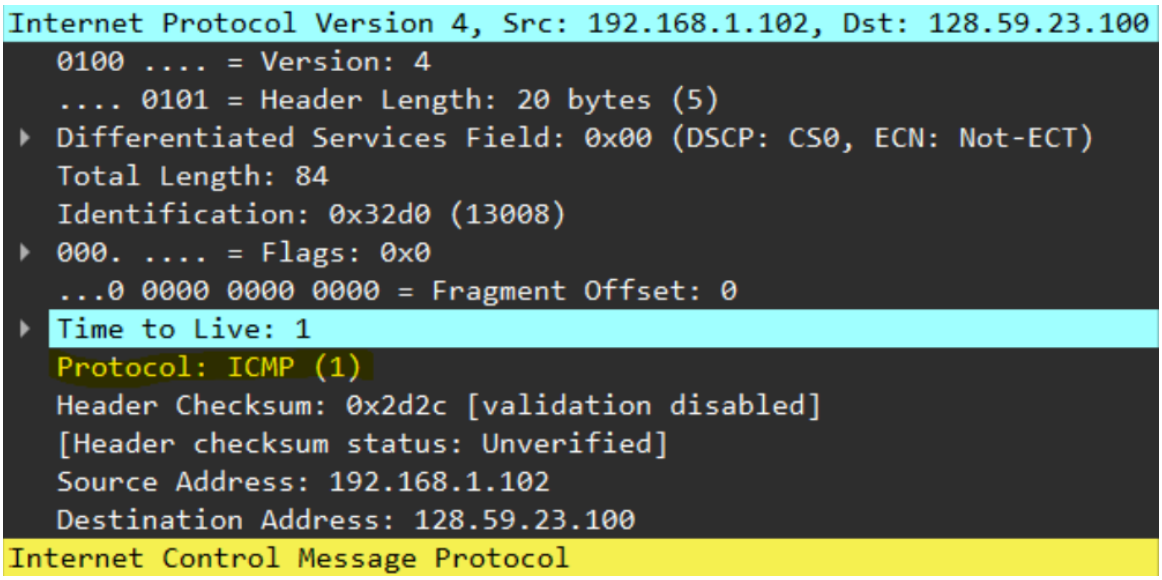


Figure 3: IP header of the first ICMP Echo Request packet with protocol highlighted

3. How many bytes are in the IP header? How many bytes are in the payload of the IP datagram? Explain.

a. The IP header is 20 bytes, the payload is the total bytes of the packet - 20 header bytes, which is 78 bytes.

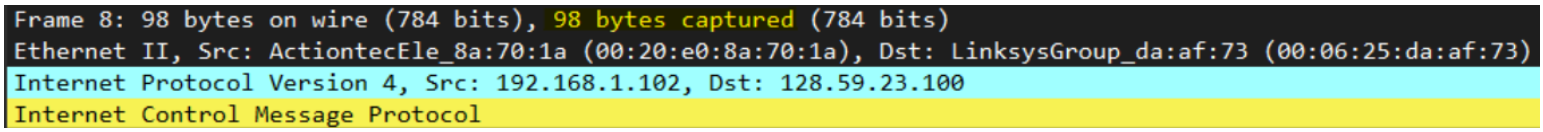


Figure 3: IP header of the first ICMP Echo Request packet with total size in bytes highlighted

4. Has this IP datagram been fragmented? Explain.

a. This datagram has not been fragmented, as seen by the fragmentation flags not being set.

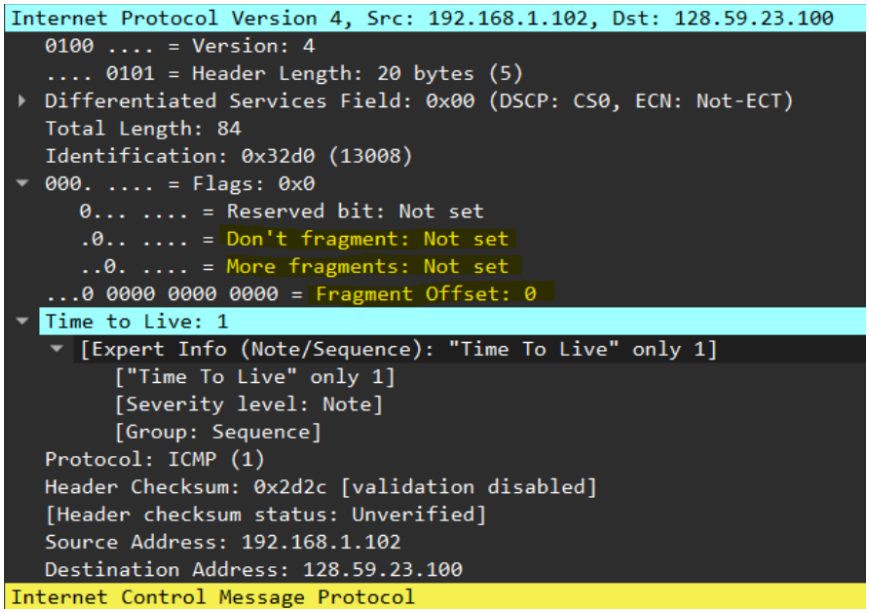


Figure 4: IP header of the first ICMP Echo Request packet with fragmentation flags highlighted

5. Sort the packets by source IP address, find the first ICMP Echo Request sent by the host, observe all of the subsequent ICMP messages. Which fields in the IP datagram always change from one datagram to the next within this series of ICMP messages?

a. Identification, Time to live, Checksum

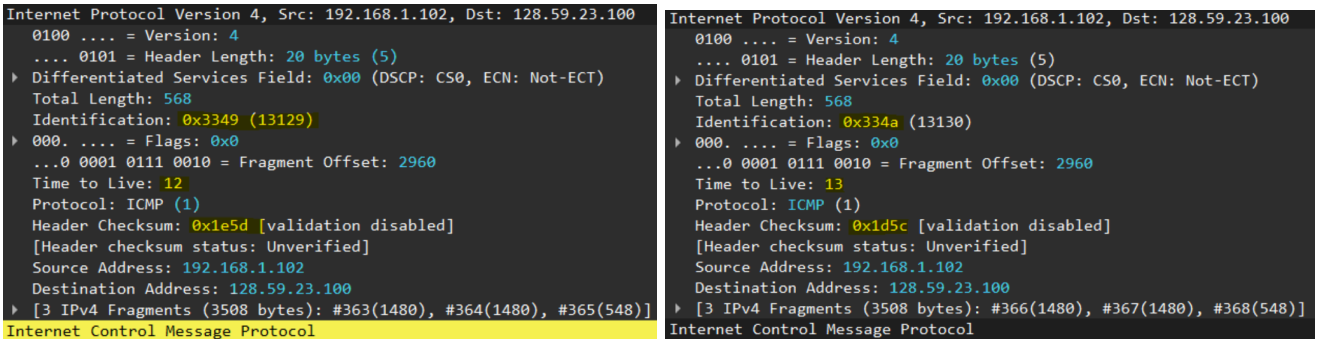


Figure 5: IP headers of packets in series with changing fields highlighted blue, and constant header fields highlighted yellow

6. Which fields stay constant? Which *must* stay constant? Which *must* change? Explain.

a. In this case, Version, Header length, Differentiated Services Field, Total length, Flags, Fragment offset, Protocol, Source IP, and Destination IP stayed the same.
b. Version, Protocol, and Source/Dest. IP must not change since these are from the same host to the same destination..
c. TTL, Header checksum, and Identification must change because they are calculated or in the case of TTL set specifically incremented to trace the route.

7. Describe the pattern observed in the values in the Identification field of the IP datagram.

a. The Identification value decreases by 1 each packet.

8. In the TTL-exceeded replies sent to your computer from the nearest (first-hop) router, what is the value of the Identification field and the TTL field?

a. 255

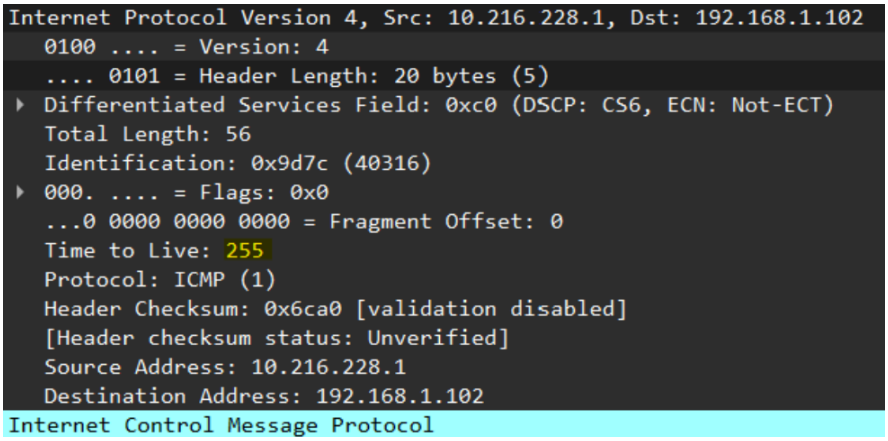


Figure 6: IP header of first TTL exceeded return packet

9. In the same TTL-exceeded replies, do these values remain unchanged for all the TTL-exceeded replies? Why?

a. These values decrease by 1 each time, due to the change in TTL in the sent packet for the trace.

10. In the first ICMP Echo Request packet sent by your computer after changing the packet size to 2000, has this message been fragmented over more than one IP datagram?

a. Yes, they were fragmented into 3 parts, of lengths 1500, 1500, 568.

```
192.168.1.102   128.59.23.100   IPv4   1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=3323) [Reassembled in #218]
192.168.1.102   128.59.23.100   IPv4   1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=3323) [Reassembled in #218]
192.168.1.102   128.59.23.100   ICMP    582 Echo (ping) request  id=0x0300, seq=40451/926, ttl=1 (no response found!)
```

Figure 7: Fragmented IP datagram

11. Print out the first fragment of the fragmented IP datagram. What information in the IP header indicates that the datagram has 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?

a. The “More fragments” flag is set.
b. The “Fragment offset” will indicate where this fragment should be placed in the reassembled packet.
c. This datagram is 1500 bytes.

```
Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.59.23.100
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1500
    Identification: 0x3323 (13091)
  ▾ 001. .... = Flags: 0x1, More fragments
      0... .... = Reserved bit: Not set
      .0.. .... = Don't fragment: Not set
      ..1. .... = More fragments: Set
    ...0 0000 0000 0000 = Fragment Offset: 0
  ▸ Time to Live: 1
    Protocol: ICMP (1)
    Header Checksum: 0x0751 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.1.102
    Destination Address: 128.59.23.100
    [Reassembled IPv4 in frame: 218]
```

Figure 8: “More fragments” flag is set

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?

a. “Fragment offset” flag indicates this is not the first fragment.
b. “More fragments” flag indicates there are more fragments to come.

```
Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.59.23.100
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1500
    Identification: 0x3323 (13091)
  ▾ 001. .... = Flags: 0x1, More fragments
      0... .... = Reserved bit: Not set
      .0.. .... = Don't fragment: Not set
      ..1. .... = More fragments: Set
    ...0 0000 1011 1001 = Fragment Offset: 1480
  ▸ Time to Live: 1
    Protocol: ICMP (1)
    Header Checksum: 0x0698 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.1.102
    Destination Address: 128.59.23.100
    [Reassembled IPv4 in frame: 218]
```

Figure 9: “Fragment offset” and “More fragments” flags are set

13. What fields change in the IP header between the first and second fragment?

a. In Figure 8 and Figure 9 we can see that the Identification, TTL, Checksum, and Fragment offset change from one fragment to the next.

14. In the first ICMP Echo Request packet sent by your computer after changing the packet size to 3500, how many fragments were created from the original datagram?

a. Three fragments were created as noted in Figure 7.

15. What fields change in the IP header among the fragments?

a. In Figure 8 and Figure 9 we can see that the Identification, TTL, Checksum, and Fragment offset change from one fragment to the next.

PART II

Setup: Open client side NAT trace file.

1. What is the IP address of the client?

a. 192.168.1.100, as seen below in this POST message, since POST is initiated by client.

Figure 1: Client IP address

2. Filter packets by IP address 64.233.169.104

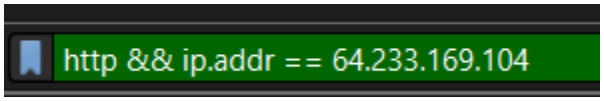


Figure 2: Filtering by HTTP and IP address

3. In reference to the packet sent from the client to the previous IP address, what are the source and destination IP addresses and TCP source and destination ports on the datagram?

a. 192.168.1.100, 64.233.169.104
b. 4335, 80

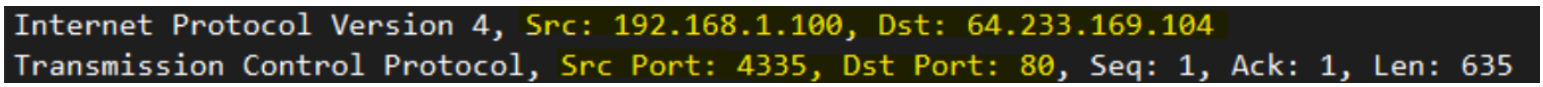


Figure 3: Source and destination socket

4. At what time is the corresponding 200 OK message received from the Google server (previous IP address mentioned)? What are the source and destination IP addresses and TCP source and destination ports on this datagram?

a. 7.158797
b. 64.233.169.104, 192.168.1.100
c. 80, 4335

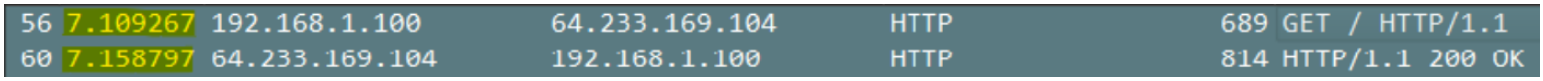


Figure 4: Time GET was sent and time OK was received

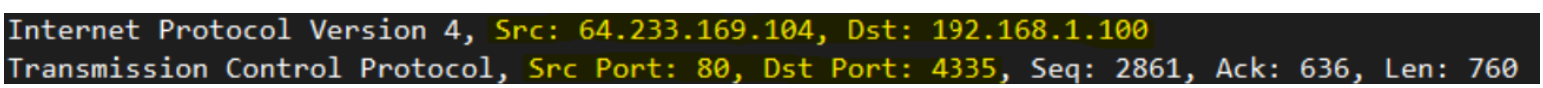


Figure 5: Source and destination socket on returned packet

5. At what time is the client-to-server TCP SYN segment sent to establish the connection? What are the source and destination IP addresses and source and destination ports of this SYN segment? What are the source and destination IP addresses and source and destination ports of the ACK sent in response to the SYN? What time is the ACK received by the client?

a. 7.075657
b. 192.168.1.100, 64.233.169.104
c. 64.233.169.104, 192.168.1.100
d. 7.108986

Time	Source	Destination	Protocol	Length	Info
7.075657	192.168.1.100	64.233.169.104	TCP	66	4335 → 80 [SYN] Seq=0
7.108986	64.233.169.104	192.168.1.100	TCP	66	80 → 4335 [SYN, ACK]
7.109053	192.168.1.100	64.233.169.104	TCP	54	4335 → 80 [ACK] Seq=1

Figure 6: TCP 3-way handshake with timing highlighted

```
Internet Protocol Version 4, Src: 192.168.1.100, Dst: 64.233.169.104
Transmission Control Protocol, Src Port: 4335, Dst Port: 80, Seq: 0, Len: 0
```

Figure 7: Client SYN segment source and destination socket

```
Internet Protocol Version 4, Src: 64.233.169.104, Dst: 192.168.1.100
Transmission Control Protocol, Src Port: 80, Dst Port: 4335, Seq: 0, Ack: 1, Len: 0
```

Figure 8: Server SYN, ACK segment source and destination socket

Setup: Open ISP side NAT trace file.

6. What time does the HTTP GET message sent from the client at time 7.109267 in the client side file appear in the ISP side file? What are the source and destination IP addresses and source and destination ports on this datagram? Which of these field are the same, and which are different, than in your answer to question 3?

a. 6.069168
b. 71.192.34.104, 64.233.169.104
c. 4335, 80
d. Source IP changed since we now see the public IP address

Time	Source	Destination	Protocol	Length	Info
6.069168	71.192.34.104	64.233.169.104	HTTP	689	GET / HTTP/1.1
6.117570	64.233.169.104	71.192.34.104	HTTP	814	HTTP/1.1 200 OK (text/html)

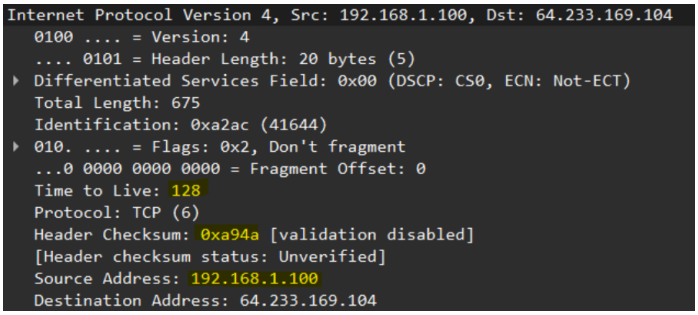
Figure 9: GET and OK time sent and received

```
Internet Protocol Version 4, Src: 71.192.34.104, Dst: 64.233.169.104
Transmission Control Protocol, Src Port: 4335, Dst Port: 80, Seq: 1, Ack: 1, Len: 635
```

Figure 10: GET source and destination socket

7. Are any fields in the HTTP GET message changed? Which of the following fields in the datagram are changed: Version, Header length, Flags, Checksum. If any are changed, explain.

a. TTL, Checksum, and Source IP changed
b. TTL changed as time passed, checksum changed as source IP changed, source IP changed due to NAT translation



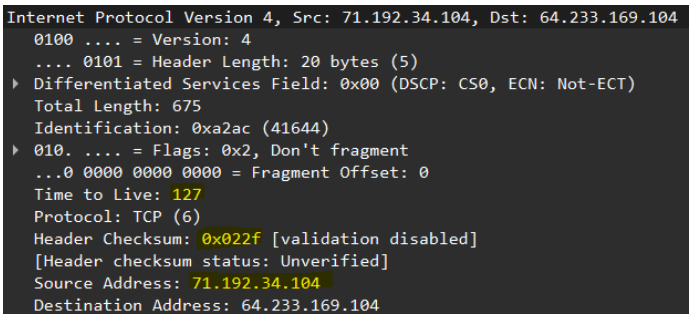


Figure 11, 12: Client side and ISP side GET datagram, respectively

8. At what time is the first 200 OK HTTP message received from the Google server? What are the source and destination IP addresses and ports on this datagram? Which fields are the same and which are different than on the answer to question 4?

a. 6.117570 as seen in Figure 9
b. 64.233.168.104, 71.192.34.104
c. 80, 4335
d. Destination IP address has changed due to public IP being used for client

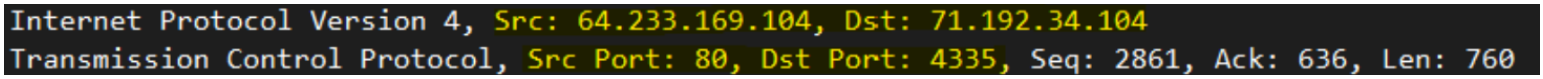


Figure 13: ISP side OK datagram socket

9. At what time were the client-to-server TCP SYN segment and the server-to-client TCP ACK segment corresponding the the segments in question 5 captured? What are source and destination IP addresses and ports? Which of the fields are the same and which are different from the segment observed in question 5?

a. 6.035475, 6.067775
b. 71.192.34.104, 64.233.169.104
c. 4335, 80
d. 64.233.168.104, 71.192.34.104
e. 80, 4335
f. The clients IP address changed to the public IP address

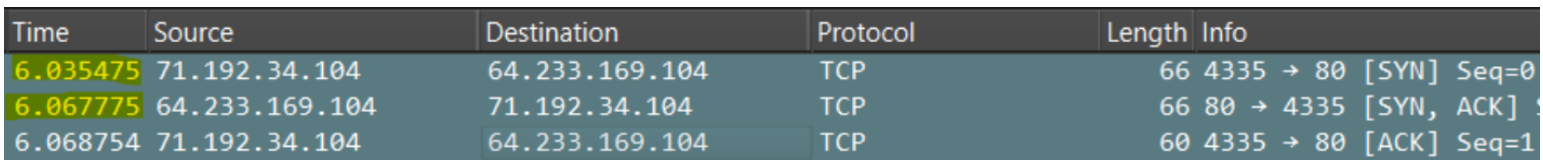


Figure 14: ISP side TCP handshake timing

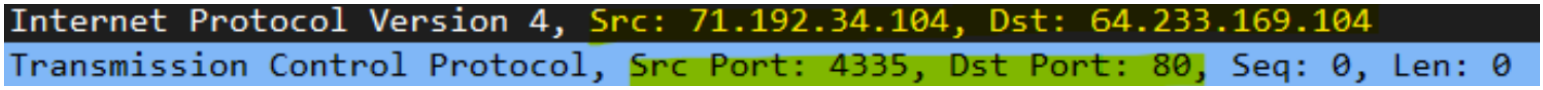


Figure 15: ISP side client SYN segment socket

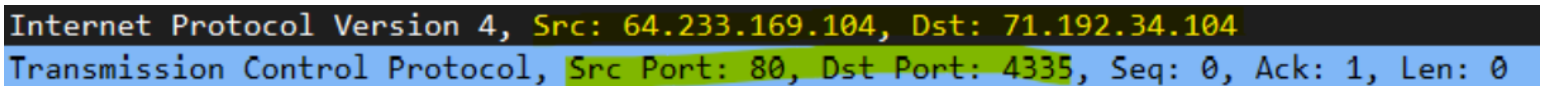


Figure 16: ISP side server ACK, SYN segment socket

10. Fill in the NAT translation table entries for the HTTP connection considered above.

Private Socket	Public Socket	Link Address
192.168.1.100, 4335	71.192.34.104, 4335	bc:54:2f:cf:a1:70

PART III

Steps:

1. **Run “ipconfig”.**

```
PS C:\Users\laesc> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet 3:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::91d3:baa6:7f08:8744%21
   IPv4 Address. . . . . . . . . . . : 192.168.56.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Unknown adapter Local Area Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2600:1700:10dd:5c1f:41c4:d1c1:b0a7:1bb2
   Temporary IPv6 Address. . . . . . : 2600:1700:10dd:5c1f:953c:7f5d:73da:e1ea
   Link-local IPv6 Address . . . . . : fe80::686d:25b4:583:5a07%17
   IPv4 Address. . . . . . . . . . . : 192.168.0.80
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::4a22:54ff:feda:7820%17
                                       192.168.0.1
PS C:\Users\laesc>
```

Figure 1: First ipconfig output

2. **Begin a packet capture in Wireshark.**

3. **Run “ipconfig /renew”**

4. **After the previous command completes and terminates, run “ipconfig /renew” again**

5. **After the previous command completes and terminates, run “ipconfig /release”**

6. **Enter again “ipconfig /renew” to once more be allocated an IP address.**

7. **Stop the packet capture.**

478	8.211696	192.168.0.80	192.168.0.1	DHCP	342 DHCP Request - Transaction ID 0xb84b5bb9
479	8.221132	192.168.0.1	192.168.0.80	DHCP	346 DHCP ACK - Transaction ID 0xb84b5bb9
607	20.6491…	192.168.0.80	192.168.0.1	DHCP	342 DHCP Request - Transaction ID 0x607446c6
608	20.6581…	192.168.0.1	192.168.0.80	DHCP	346 DHCP ACK - Transaction ID 0x607446c6
1129	32.6627…	192.168.0.80	192.168.0.1	DHCP	342 DHCP Release - Transaction ID 0xef9452ad
1200	36.0974…	0.0.0.0	255.255.255.255	DHCP	342 DHCP Discover - Transaction ID 0x4d5fe03f
1398	39.6018…	0.0.0.0	255.255.255.255	DHCP	342 DHCP Discover - Transaction ID 0x4d5fe03f
1413	40.1329…	192.168.0.1	192.168.0.80	DHCP	342 DHCP Offer - Transaction ID 0x4d5fe03f
1414	40.1335…	192.168.0.1	192.168.0.80	DHCP	342 DHCP Offer - Transaction ID 0x4d5fe03f
1415	40.1349…	0.0.0.0	255.255.255.255	DHCP	354 DHCP Request - Transaction ID 0x4d5fe03f
1417	40.1412…	192.168.0.1	192.168.0.80	DHCP	346 DHCP ACK - Transaction ID 0x4d5fe03f
2080	51.5075…	0.0.0.0	255.255.255.255	DHCP	590 DHCP Request - Transaction ID 0xaa5b87b2
2084	51.6111…	0.0.0.0	255.255.255.255	DHCP	590 DHCP Request - Transaction ID 0xaa5b87b2
2127	55.0612…	0.0.0.0	255.255.255.255	DHCP	348 DHCP Request - Transaction ID 0xef35c6f1
2128	55.0652…	192.168.0.1	192.168.0.80	DHCP	346 DHCP ACK - Transaction ID 0xef35c6f1

Figure 2: Corresponding packet capture

Questions:

1. Are DHCP messages sent over TCP or UDP?

a. UDP

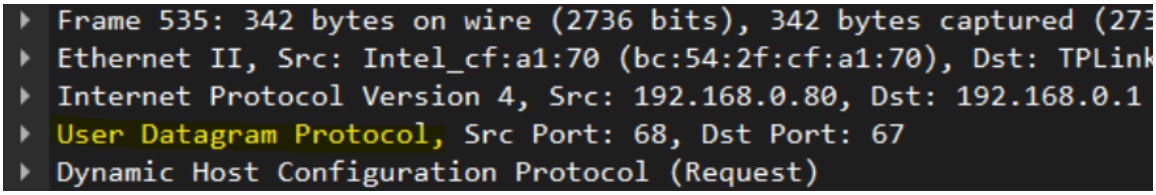


Figure 3: DHCP packet using UDP transport layer protocol

2. Draw timing diagram for sequence numbers of the first set of four packet Discover/Offer/Request/ACK exchange. Indicate source and destination port numbers. Are these port numbers the same as the example?

Seq. No	DHCP Packet Type	Source Port	Destination Port
1398	Discover	68	67
1413	Offer	67	68
1415	Request	68	67
1417	ACK	67	68

3. What is the link layer address of your host?

a. bc:54:2f:cf:a1:70

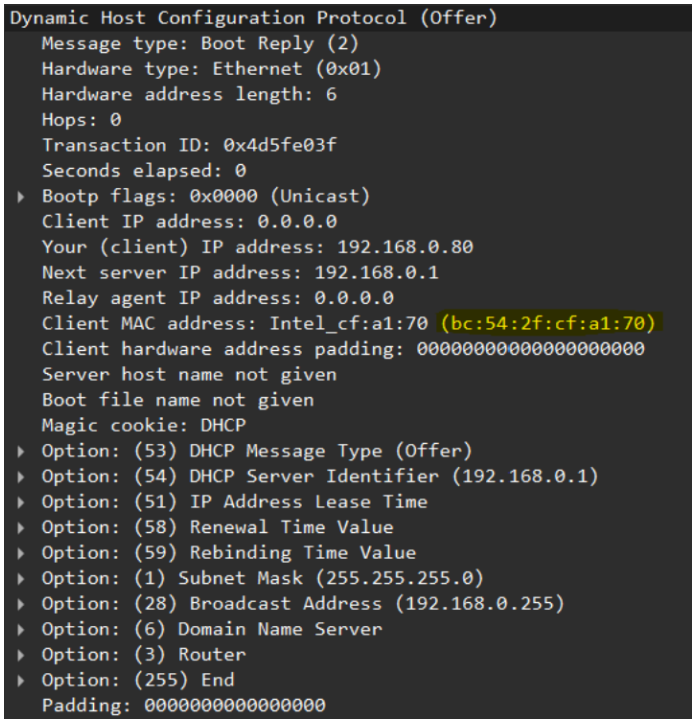


Figure 4: Client MAC address

4. What values in the DHCP discover message differentiate this message from the DHCP request message?

a. DHCP Message Type, and Request message has additional options 54, 81, and 255.

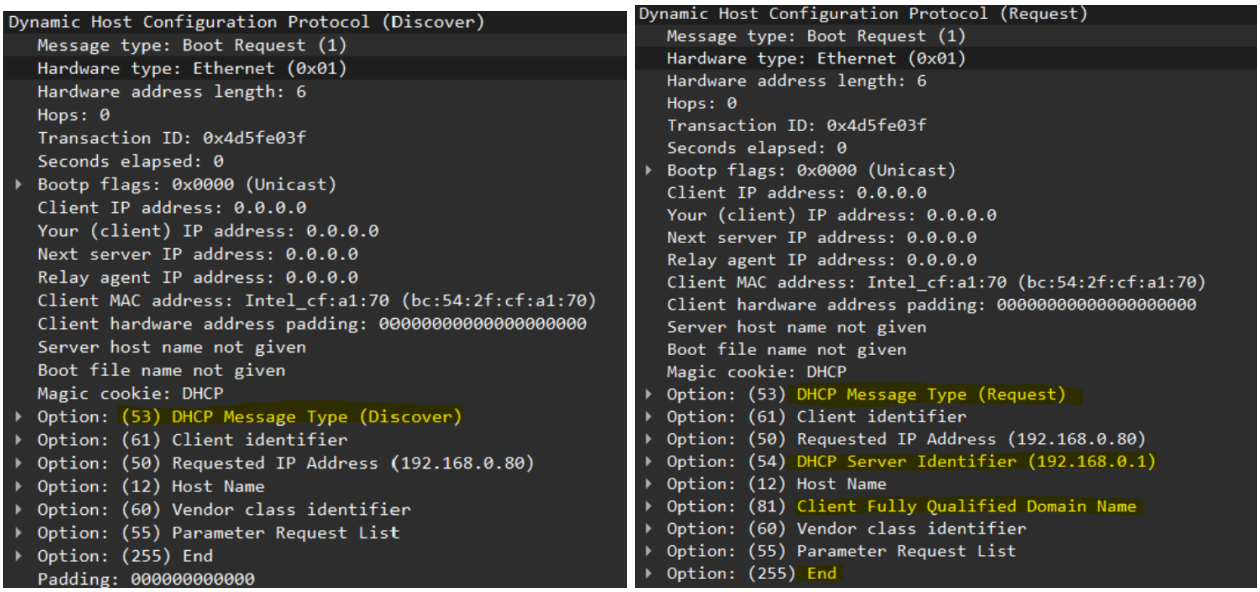


Figure 5: Differences between DHCP Discover and Request packets highlighted

5. What is the value of the Transaction-ID in each of the four messages? Where are the values in the second set of four messages? What is the purpose of the Transaction-ID field?

a. 0x4d5fe03f
b. 0xaa5b87b2
c. To allow client and server to know which DHCP message it is responding to.

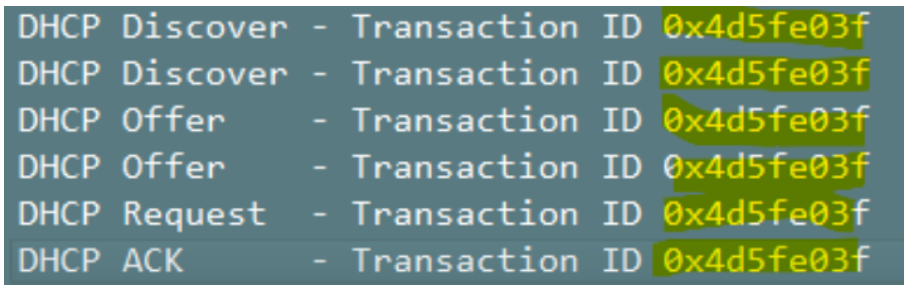


Figure 6: Transaction IDs of the exchange

6. For the first set of indicate the source and destination IP addresses?

a. 0.0.0.0 and 255.255.255.255 from the client
b. 192.168.0.1 and 192.168.0.80 from the DHCP server

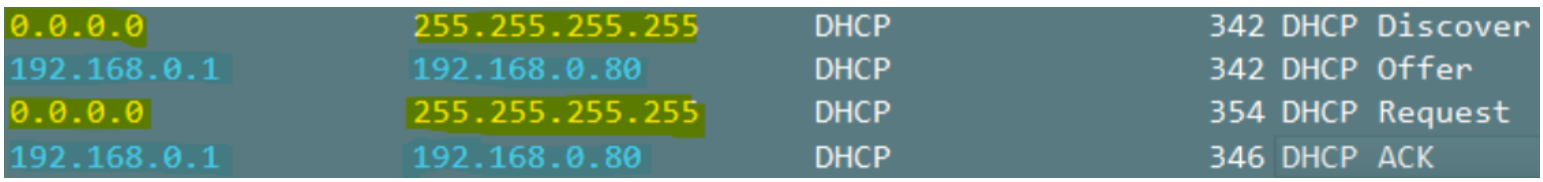


Figure 7: IP addresses of the exchange

7. What is the IP address of your DHCP server?

a. 192.168.0.1, as seen in Figure 7 and 8.

8. What is the IP address of the DHCP server offering to your host in the DHCP offer message?

a. 192.168.0.1, as seen in Figure 7 and 8.

9. In the example, what values in the trace indicate the absence of a relay agent? Is there a relay agent in your capture? What is the IP address of the agent if so?

a. There is no relay agent present in my trace, as highlighted in yellow.

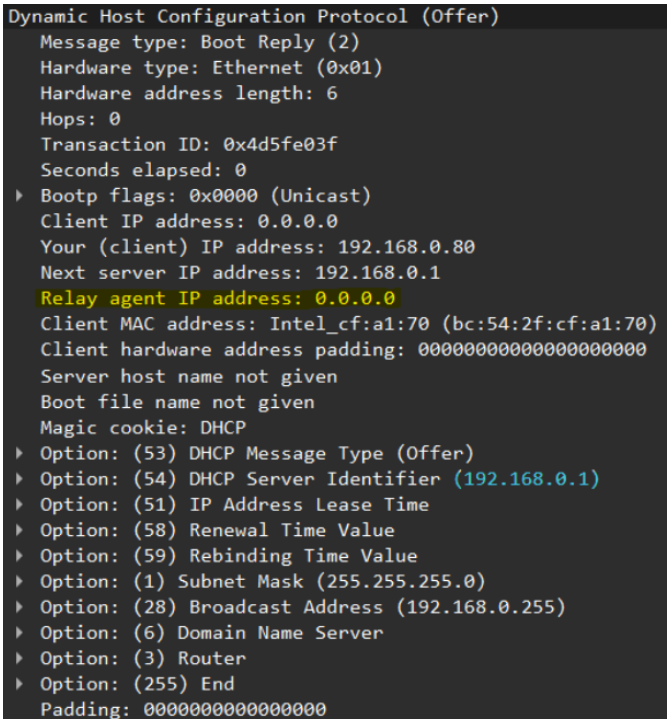


Figure 8: Relay agent address and DHCP server address

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

a. Subnet allows the client to know the mask which allows it to communicate with devices also on the local subnet.
b. Router gives the address of the default gateway.

11. In the client's response to the first offer message in the example, does the client accept the offered IP address? Explain.

a. Yes, because the client replies with a DHCP Request message.

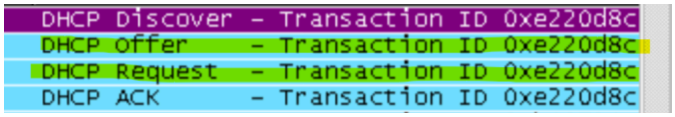


Figure 9: Offer and Request message in same DHCP transaction

12. Explain the purpose of lease time. How long is lease time in your capture?

a. Lease time prevents a client from keeping an address indefinitely. This maintains the network availability.
b. 2 hours.

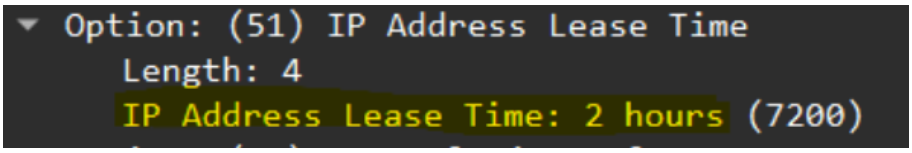


Figure 10: Lease time in my DHCP capture

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

a. The purpose of the release message is to let the DHCP server know that the IP address is available again.
b. No, because the IP has been released and presumably the client is leaving the network.
c. Then the address would simply expire at lease expiration.

14. Were any ARP packets sent or received during the DHCP packet exchange? Explain.

a. There were 176 ARP packets sent during my capture. This is because I captured on a busy home network with many smart devices communicating at once.

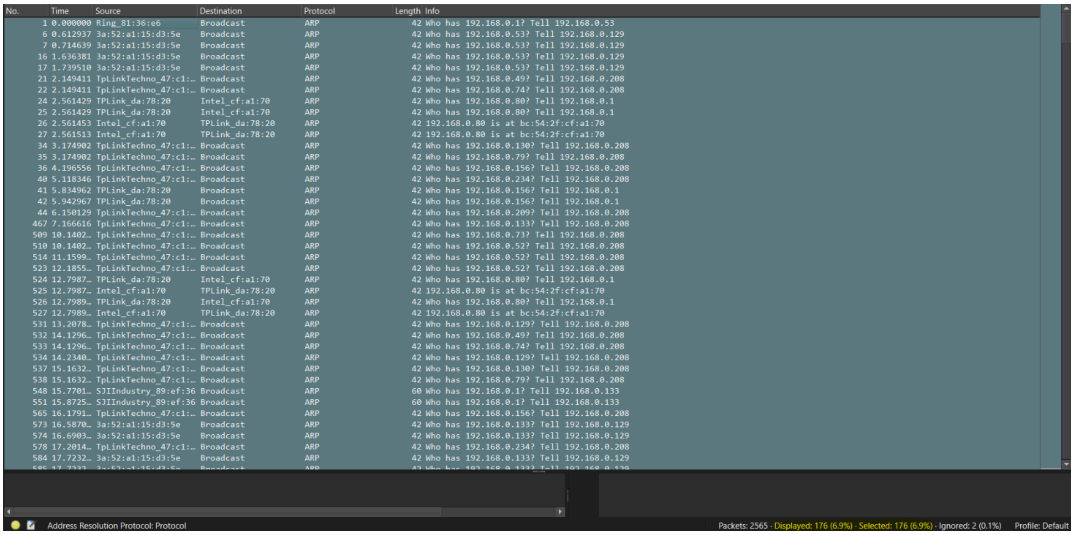


Figure 11: Packets filtered by ARP protocol

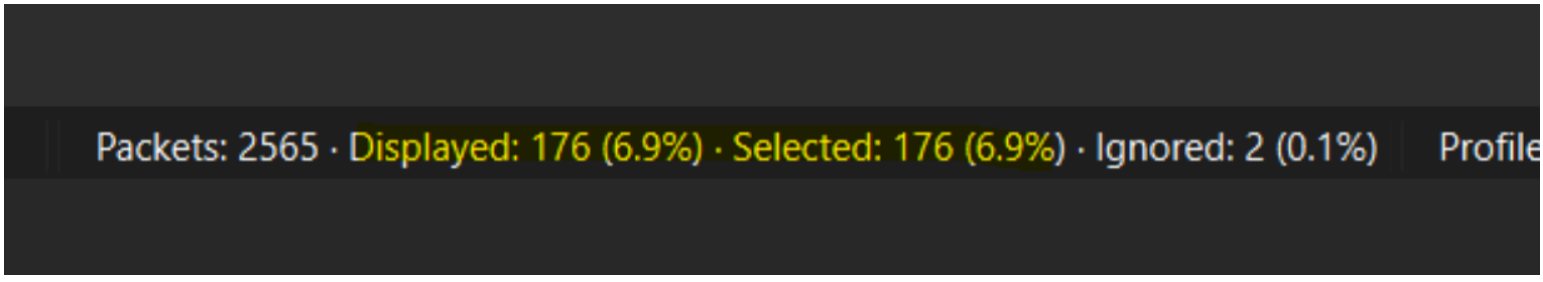


Figure 10: Amount of packets displayed/selected that were ARP