

CENG312

Assignment#3

IP, ICMP, NAT, DHCP



Name Surname: Sude Nur Çevik

ID: 270201041

Department: Computer Engineering

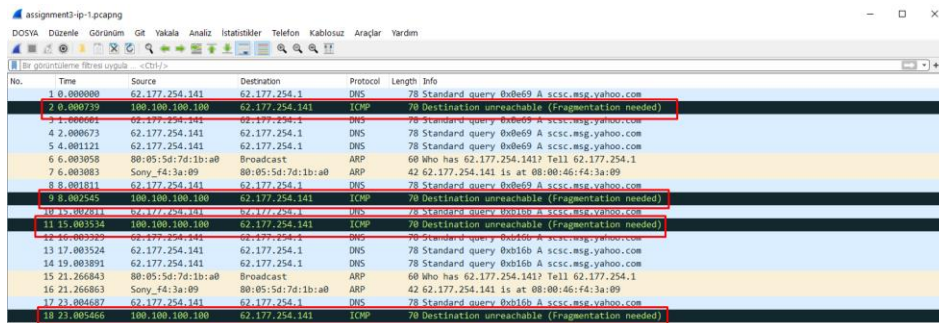
Lecture: Computer Networks - Spring 2023

Instructor: Prof. Cüneyt Fehmi Bazlamaçcı

IP and ICMP Section

1.

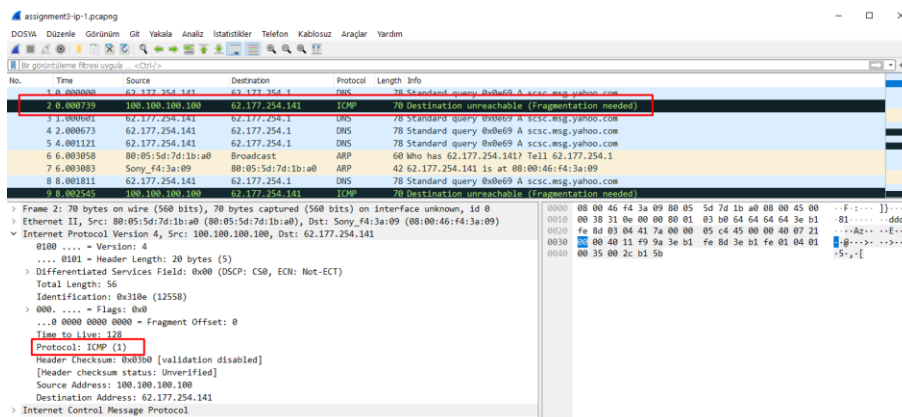
- a. The packets that trigger the "Fragmentation Needed" ICMP message are packet 2, packet 9, packet 11, and packet 18 (see fig1.1). These packets share the following characteristics that necessitate fragmentation:
 - i. **Protocol:** The protocol field in the IP header of these packets is set to ICMP (1) (see fig 1.2). A fragmentation problem is being indicated via ICMP, a network-layer protocol used for diagnostic and control reasons.
 - ii. **Type and Code:** The ICMP message type is 3, indicating a "Destination Unreachable" message, and the code is 4, specifically indicating "Fragmentation Needed." (see fig 1.3). The packet is too big to be sent without fragmentation, according to this type and code combination.
 - iii. **MTU of Next Hop:** The "MTU of next hop" ICMP message field in each of these packets has the value 1476. The MTU (Maximum Transmission Unit) designates the largest packet size that may be transferred over a network without being fragmented. (see fig1.4). This field's existence indicates that fragmentation is necessary since the packet's size exceeds the next hop network's MTU.
 - iv. **Total Length:** These packets' IP headers have a "Total Length" field that reveals the complete size of the packet, including the header and contents. These packets are 56 bytes long overall, which is longer than the next hop network's MTU. (see fig 1.5).



The screenshot shows a Wireshark packet capture of an ICMP flood attack. The packet list on the left shows several ICMP 'Destination unreachable (Fragmentation needed)' messages (type 3, code 4) from 100.100.100.100 to 62.177.254.141. The packet details pane on the right shows the ICMP header for packet 2, which is a 'Destination unreachable (Fragmentation needed)' message. The 'Fragmentation needed' field is highlighted, showing a value of 1476. The packet length is 56 bytes.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	62.177.254.141	62.177.254.141	DNS	78	Standard query 0x0be69 A scsc.msg.yahoo.com
2	0.000739	100.100.100.100	62.177.254.141	ICMP	78	Destination unreachable (Fragmentation needed)
3	1.000001	62.177.254.141	62.177.254.141	DNS	78	Standard query 0x0be69 A scsc.msg.yahoo.com
4	2.000673	62.177.254.141	62.177.254.141	DNS	78	Standard query 0x0be69 A scsc.msg.yahoo.com
5	4.001121	62.177.254.141	62.177.254.141	DNS	78	Standard query 0x0be69 A scsc.msg.yahoo.com
6	6.003058	80:05:5d:7d:1b:a0	Broadcast	ARP	60	Who has 62.177.254.141? Tell 62.177.254.1
7	6.003083	Sony_f4:3a:09	80:05:5d:7d:1b:a0	ARP	42	62.177.254.141 is at 00:00:46:f4:3a:09
8	8.001811	62.177.254.141	62.177.254.141	DNS	78	Standard query 0x0be69 A scsc.msg.yahoo.com
9	8.002545	100.100.100.100	62.177.254.141	ICMP	78	Destination unreachable (Fragmentation needed)
10	8.003073	62.177.254.141	62.177.254.141	DNS	78	Standard query 0x0be69 A scsc.msg.yahoo.com
11	15.003534	100.100.100.100	62.177.254.141	ICMP	78	Destination unreachable (Fragmentation needed)
12	16.003320	62.177.254.141	62.177.254.141	DNS	78	Standard query 0x0be69 A scsc.msg.yahoo.com
13	17.003524	62.177.254.141	62.177.254.141	DNS	78	Standard query 0x0be69 A scsc.msg.yahoo.com
14	19.003891	62.177.254.141	62.177.254.141	DNS	78	Standard query 0x0be69 A scsc.msg.yahoo.com
15	21.266843	80:05:5d:7d:1b:a0	Broadcast	ARP	60	Who has 62.177.254.141? Tell 62.177.254.1
16	21.266863	Sony_f4:3a:09	80:05:5d:7d:1b:a0	ARP	42	62.177.254.141 is at 00:00:46:f4:3a:09
17	23.004687	62.177.254.141	62.177.254.141	DNS	78	Standard query 0x0be69 A scsc.msg.yahoo.com
18	23.005466	100.100.100.100	62.177.254.141	ICMP	78	Destination unreachable (Fragmentation needed)

Fig1.1 "Fragmentation Needed" packets



The screenshot shows the packet details pane for packet 2, which is an ICMP 'Destination unreachable (Fragmentation needed)' message. The 'Protocol' field is highlighted as 'ICMP (1)'. The 'Fragmentation needed' field is also highlighted, showing a value of 1476. The packet length is 56 bytes.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	62.177.254.141	62.177.254.141	DNS	78	Standard query 0x0be69 A scsc.msg.yahoo.com
2	0.000739	100.100.100.100	62.177.254.141	ICMP	78	Destination unreachable (Fragmentation needed)
3	1.000001	62.177.254.141	62.177.254.141	DNS	78	Standard query 0x0be69 A scsc.msg.yahoo.com
4	2.000673	62.177.254.141	62.177.254.141	DNS	78	Standard query 0x0be69 A scsc.msg.yahoo.com
5	4.001121	62.177.254.141	62.177.254.141	DNS	78	Standard query 0x0be69 A scsc.msg.yahoo.com
6	6.003058	80:05:5d:7d:1b:a0	Broadcast	ARP	60	Who has 62.177.254.141? Tell 62.177.254.1
7	6.003083	Sony_f4:3a:09	80:05:5d:7d:1b:a0	ARP	42	62.177.254.141 is at 00:00:46:f4:3a:09
8	8.001811	62.177.254.141	62.177.254.141	DNS	78	Standard query 0x0be69 A scsc.msg.yahoo.com
9	8.002545	100.100.100.100	62.177.254.141	ICMP	78	Destination unreachable (Fragmentation needed)

Frame 2: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface unknown, id 0
Ethernet II, Src: 80:05:5d:7d:1b:a0 (00:05:5d:7d:1b:a0), Dst: Sony_f4:3a:09 (08:00:46:f4:3a:09)
Internet Protocol Version 4, Src: 100.100.100.100, Dst: 62.177.254.141
ICMP, Version: 4
... 0101 = Header Length: 20 bytes (5)
... 0101 = Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 56
Identification: 0x318e (12558)
... 0000 = Flags: 0x0
... 0000 0000 0000 = Fragment Offset: 0
Time to live: 128
Protocol: ICMP (1)
Header checksum: 0x0300 [validation disabled]
[Header checksum status: Unverified]
Source Address: 100.100.100.100
Destination Address: 62.177.254.141
Internet Control Message Protocol

Fig1.2 Protocol "ICMP (1)" Example

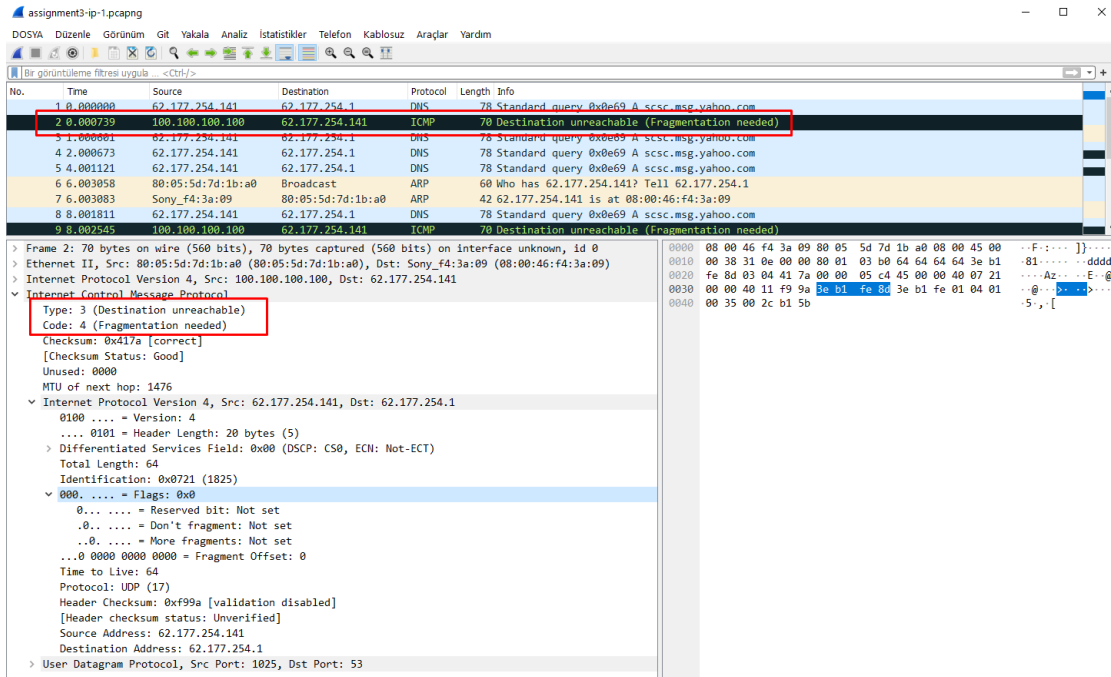


Fig1.3 Type and Code Example

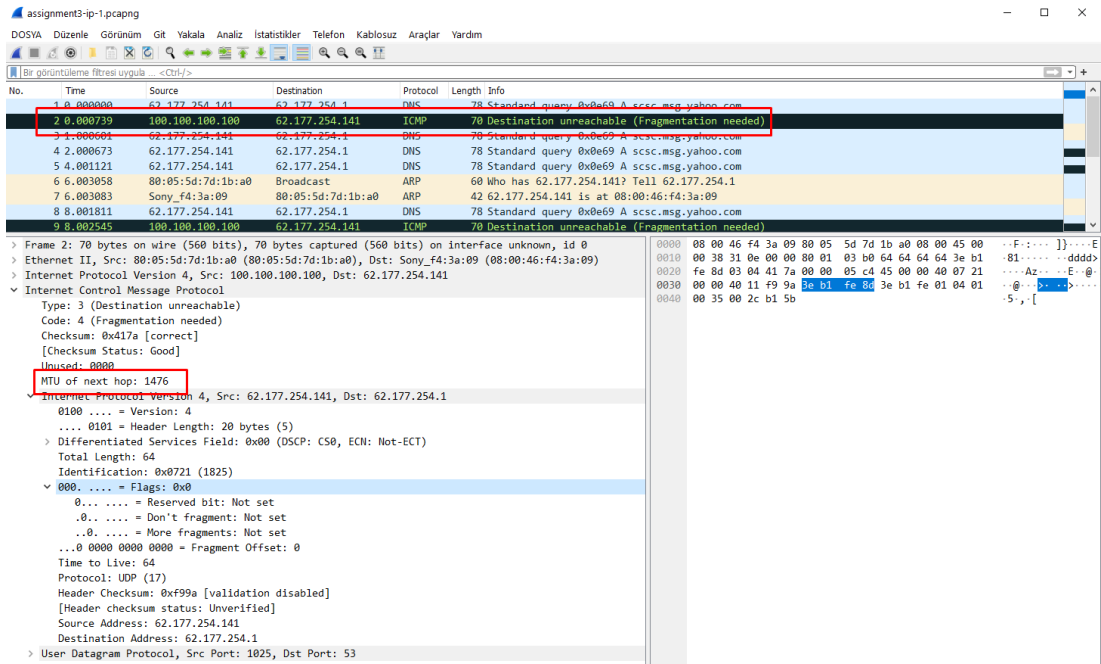


Fig1.4 MTU of Next Hop Example

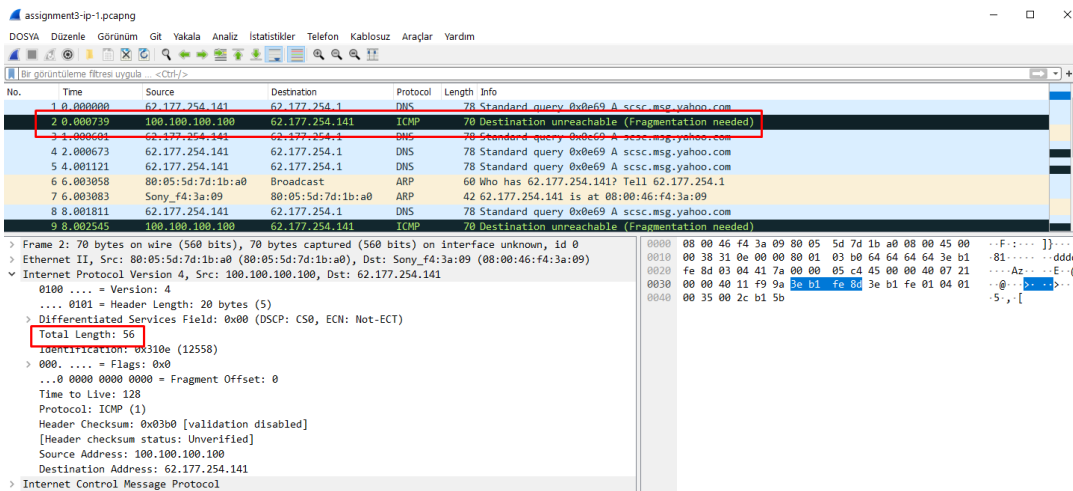


Fig1.5 Total Length Example

- b. When endpoints receive a packet with the "Fragmentation Needed" flag, it indicates that the packet is too large to be transmitted over the network without fragmentation. To process these packets and ensure successful delivery, the following steps can be taken:
 - i. Upon receiving the "Fragmentation Needed" ICMP message, the endpoint analyzes the message and extracts relevant information, such as the original packet causing the fragmentation and the recommended MTU size.
 - ii. The endpoint then checks its local MTU configuration to determine the maximum size of packets it can transmit without fragmentation. This value is typically based on the network interface or the network path to the destination.
 - iii. If the local MTU is smaller than the recommended MTU in the ICMP message, the endpoint needs to fragment the original packet into smaller fragments that fit within its local MTU. This process involves breaking the packet into smaller pieces and adjusting the necessary IP header fields.
 - iv. The endpoint creates multiple fragments, each with a smaller size that fits within the local MTU. The IP header of each fragment is modified to indicate its position within the original packet, using the "Fragment Offset" and "More Fragments" fields.
 - v. The endpoint then transmits each fragment individually, ensuring that they are sent in the correct order. The receiving endpoint or intermediate routers will use the information in the IP headers to reassemble the fragments into the original packet.
 - vi. Upon receiving the fragments, the receiving endpoint or intermediate routers reassemble them based on the information in the IP headers. This involves combining the fragments in the correct order and adjusting the IP header fields of the reassembled packet.

- vii. Once the reassembly is complete, the receiving endpoint can process the packet as a complete unit. This may involve further processing at higher network layers, such as forwarding the packet or passing it to the relevant application or service.

Fig1.6 Fragmented Packets

2.

- a. We can determine fragmented packets by looking at the more fragment flag whether it is set and recognizing the fragment offset. The fragment offset is a 13-bit field in the IP header that identifies the position of a fragment relative to the beginning of the original unfragmented datagram. It indicates the number of data bytes preceding or ahead of the fragment. To see the details of fragmented packet, look at the fig2.1.

Assignment 3-İp Zıncır

DOSYA DÜZENİ Gözümleme Git Yıkala Analiz İstatistikler Telefon Kablosuz Araçlar Yardım

İki görüntülenmiş filtre uygula ->Ctrl+Z</p>
 </div>

Fig.2.1 Fragmented Packet

- b. IP fragmentation is the process of breaking up a large IP packet into smaller packets that can be transmitted over the network. This is necessary because different networks have different maximum transmission unit (MTU) sizes, which is the largest size of packet that can be transmitted over the network without being fragmented. If a packet is too large to be transmitted over the network without being fragmented, it will be broken up into smaller packets that can be transmitted over the network and then reassembled at the destination.
- c. There are four fragmented packet for all purple colored packets(see fig2.2) .To identify and reassemble fragmented IP packets in Wireshark, you can use the “Follow TCP Stream” feature. This feature will automatically reassemble all the fragments of a TCP stream into a single packet. You can also use the “IP Fragmentation” filter to display only fragmented packets.

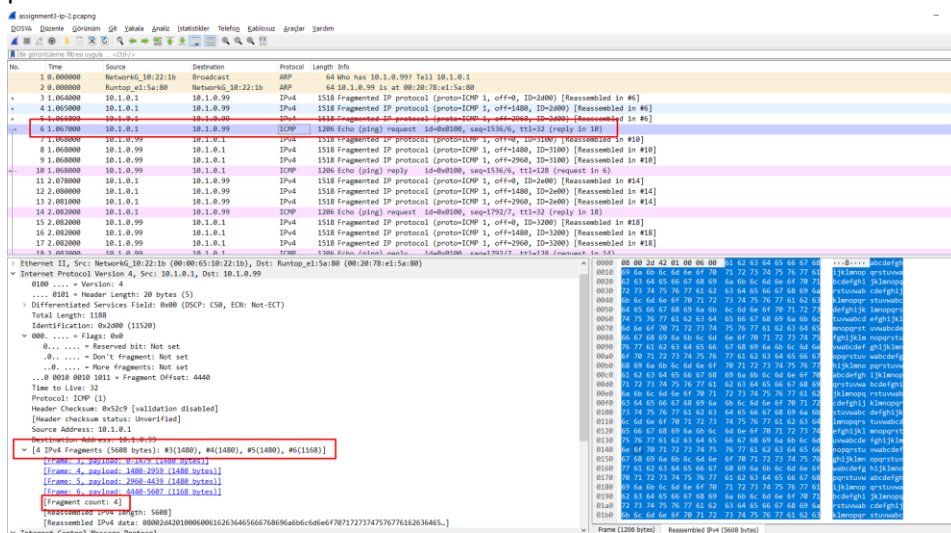


Fig.2.2 Fragmented Packet

- d. Total Size = (Last Fragment Offset + Last Fragment Length),
Total size = 4400 + 1168 = **5608** (see fig2.3)

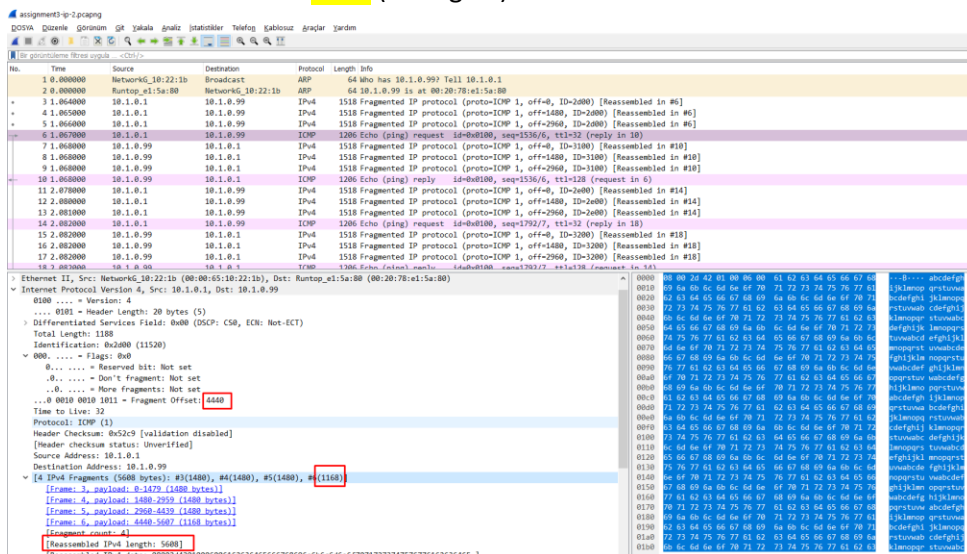


Fig2.3 Total Size of IP Packet

- e. The time to live field is not changed (see fig2.5 & fig2.6 & fig2.7). In this case, the TTL value remains the same (32) for all fragments. The TTL field in the IP header is initially set by the source host and is decremented by one by each router that forwards the packet. The purpose of the TTL field is to limit the lifetime of IP packets and prevent them from indefinitely circulating in the network. If the TTL value were to change in the fragments, it could indicate that the fragments took different paths in the network, encountering routers with different default TTL settings or routers that modified the TTL value. However, in this case, the TTL value remains constant across all fragments, suggesting that they likely followed the same path and encountered routers that did not modify the TTL field.

assigments-ip-2.pcapng

DOSYA Düzenle Görünüm Filtre Yaka Analiz İstatistikler Telefon Kabloşuz Araçlar Fihrist

İbr görüntüleme filtresi uygula <Ctrl>F>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Network6_10:22:1b	Broadcast	ARP	64	Who has 10.1.0.99? Tell 10.1.0.1
2	0.000000	Runtime_e1:5a:80	Network6_10:22:1b	ARP	64	10.1.0.99 is at 00:20:78:e1:5a:80
3	1.064000	10.1.0.1	10.1.0.99	IPv4	1518	Fragmented IP protocol (proto=ICMP 1, off=0, ID=2d00) [Reassembled in #6]
4	1.065000	10.1.0.1	10.1.0.99	IPv4	1518	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=2d00) [Reassembled in #6]
5	1.066000	10.1.0.1	10.1.0.99	IPv4	1518	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=2d00) [Reassembled in #6]
6	1.067000	10.1.0.1	10.1.0.99	ICMP	1206	Echo (ping) request id=0x0100, seq=1536/6, ttl=32 (reply in 10)
7	1.068000	10.1.0.99	10.1.0.1	IPv4	1518	Fragmented IP protocol (proto=ICMP 1, off=0, ID=3100) [Reassembled in #10]
8	1.068000	10.1.0.99	10.1.0.1	IPv4	1518	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=3100) [Reassembled in #10]
9	1.068000	10.1.0.99	10.1.0.1	IPv4	1518	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=3100) [Reassembled in #10]
10	1.068000	10.1.0.99	10.1.0.1	ICMP	1206	Echo (ping) reply id=0x0100, seq=1536/6, ttl=128 (request in 6)
11	2.078000	10.1.0.1	10.1.0.99	IPv4	1518	Fragmented IP protocol (proto=ICMP 1, off=0, ID=2e00) [Reassembled in #14]
12	2.080000	10.1.0.1	10.1.0.99	IPv4	1518	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=2e00) [Reassembled in #14]
13	2.081000	10.1.0.1	10.1.0.99	IPv4	1518	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=2e00) [Reassembled in #14]
14	2.082000	10.1.0.1	10.1.0.99	ICMP	1206	Echo (ping) request id=0x0100, seq=1792/7, ttl=32 (reply in 18)
15	2.082000	10.1.0.99	10.1.0.1	IPv4	1518	Fragmented IP protocol (proto=ICMP 1, off=0, ID=3200) [Reassembled in #18]
16	2.082000	10.1.0.99	10.1.0.1	IPv4	1518	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=3200) [Reassembled in #18]
17	2.082000	10.1.0.99	10.1.0.1	IPv4	1518	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=3200) [Reassembled in #18]

Frame 3: 1518 bytes on wire (12144 bits), 1518 bytes captured (12144 bits) on interface unknown, id 0

Ethernet II, Src: Network6_10:22:1b (00:00:65:10:22:1b), Dst: Runtime_e1:5a:80 (00:20:78:e1:5a:80)

Internet Protocol Version 4, Src: 10.1.0.1, Dst: 10.1.0.99

0100 = Version: 4

...0101 = Header Length: 20 bytes (5)

Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 1500

Identification: 0x2d00 (11520)

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: 32

Protocol: ICMP (1)

Header Checksum: 0x33bc [validation disabled]

[Header checksum status: Unverified]

Source Address: 10.1.0.1

Destination Address: 10.1.0.99

[Reassembled IPv4 in frame: 6]

Data (1480 bytes)

Data: 08002d42010000006126546566768696a06c6d6e6f70717273747576776162636465...

[Length: 1480]

0000 00 20 78 e1 5a 80 00 00 00 65 10
0010 05 cd 2d 00 2d 00 20 01 33 bc
0020 00 63 08 00 2d 42 01 00 06 00
0030 67 68 69 6a 6b 6c 6d 6e 6f 70
0040 71 62 63 64 65 66 67 68 69
0050 70 71 72 73 74 75 76 77 61 62
0060 63 64 65 66 67 68 69 6a 6b
0070 63 64 65 66 67 68 69 6a 6b
0080 72 73 74 75 76 77 61 62 63 64
0090 6b 6c 6d 6e 6f 70 71 72 73 74
00a0 64 65 66 67 68 69 6a 6b 6c 6d
00b0 74 75 76 77 61 62 63 64 65 66
00c0 6d 6e 6f 70 71 72 73 74 75 76
00d0 66 67 68 69 6a 6b 6c 6d 6e 6f
00e0 70 71 61 62 63 64 65 66 67 68
00f0 64 70 71 72 73 74 75 76 77 63
0100 08 69 6a 6b 6c 6d 6e 6f 70 71
0110 61 62 63 64 65 66 67 68 69 6a
0120 71 72 73 74 75 76 77 61 62 63
0130 6a 6b 6c 6d 6e 6f 70 71 72 73
0140 63 64 65 66 67 68 69 6a 6b 6c
0150 73 74 75 76 61 62 63 64 65
0160 6a 6b 6c 6d 6e 71 72 73 74 75
0170 65 66 67 68 69 6a 6b 6c 6d 6e
0180 75 76 77 61 62 63 64 65 66 67
0190 6e 6f 70 71 72 73 74 75 76 77

Fig2.5 First Fragmented Packet

3.

It appears that there is no direct IPv6 connectivity between the two networks. To send IP datagrams between these networks, a technique called IPv6 over IPv4 tunneling is used. In this scenario, IPv6 packets are encapsulated within IPv4 packets to traverse an IPv4 network and reach the destination network.

To identify and understand this action within Wireshark, you can look for the following steps or indicators:

Source and Destination IP Addresses: Check the source and destination IP addresses in the captured packets. If you see IPv6 addresses that start with "2002:" and are followed by an IPv4 address, it indicates the presence of IPv6 over IPv4 tunneling. For example, in the provided packet capture, you can see addresses like "2002:1806:adcc::1806:adcc" and "2607:f0d0:2001:e:1::120," where the IPv6 address is encapsulated within the IPv4 address.(fig 3.1)

Protocol Field: In the Ethernet II header, the protocol field should indicate "IPv4" (0x0800) if the encapsulation is being used. This indicates that the Ethernet frame is carrying IPv4 packets with encapsulated IPv6 packets. (fig 3.2)

TTL Values: Compare the TTL values of the IPv4 and IPv6 packets. If the TTL values of the encapsulating IPv4 packets are lower than the TTL values of the encapsulated IPv6 packets, it suggests that the encapsulation is taking place, as the IPv4 TTL is decremented by each router it passes through.

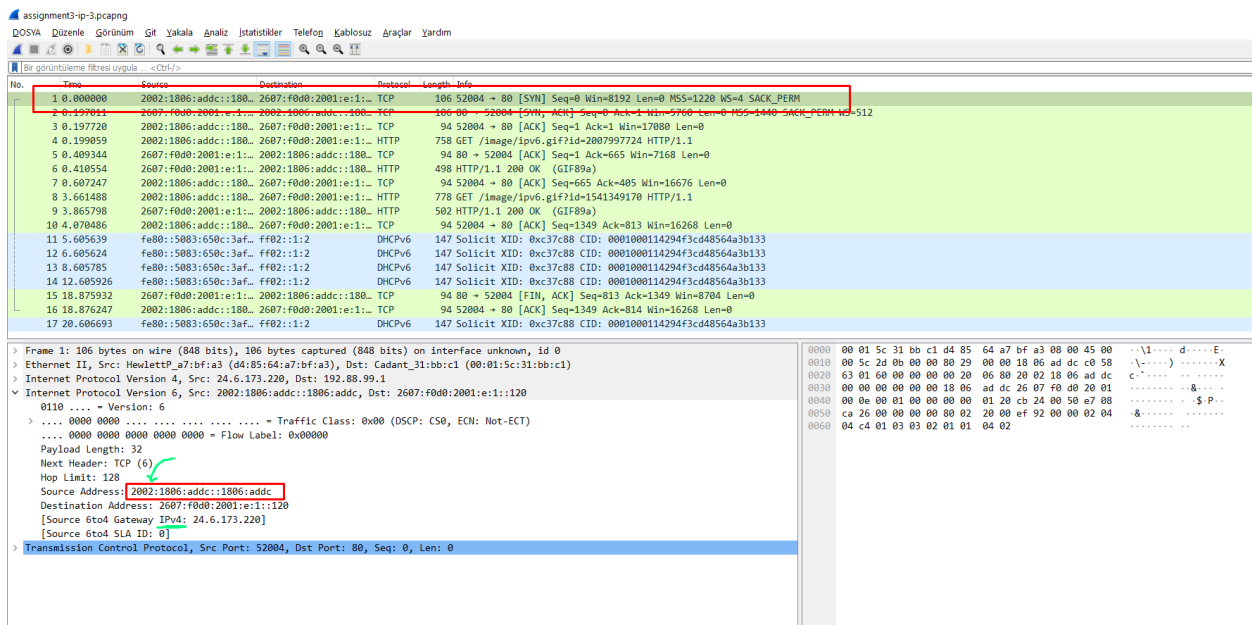


Fig3.1 IPv6 Address is Encapsulated Within the IPv4 Address

assignment3-ip-3.pcapng

DOSYA Düzenle Görünüm Git Yakala Analiz İstatistikler Telefon Kabloşuz Araçlar Yardım

Bir görüntüleme filtresi uygula... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	2002:1806:adcc::180...	2607:f0d0:2001:e:1:1...	TCP	106	52004 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1220 WS=4 SACK_PERM
2	0.197011	2607:f0d0:2001:e:1:1...	2002:1806:adcc::180...	TCP	106	80 → 52004 [SYN, ACK] Seq=0 Ack=1 Win=5760 Len=0 MSS=1440 SACK_PERM WS=512
3	0.197720	2002:1806:adcc::180...	2607:f0d0:2001:e:1:1...	TCP	94	52004 → 80 [ACK] Seq=1 Ack=1 Win=17080 Len=0
4	0.199059	2002:1806:adcc::180...	2607:f0d0:2001:e:1:1...	HTTP	758	GET /image/ipv6.gif?id=2007997724 HTTP/1.1
5	0.409344	2607:f0d0:2001:e:1:1...	2002:1806:adcc::180...	TCP	94	80 → 52004 [ACK] Seq=1 Ack=665 Win=7168 Len=0
6	0.410554	2607:f0d0:2001:e:1:1...	2002:1806:adcc::180...	HTTP	498	HTTP/1.1 200 OK (GIF89a)
7	0.607247	2002:1806:adcc::180...	2607:f0d0:2001:e:1:1...	TCP	94	52004 → 80 [ACK] Seq=665 Ack=405 Win=16676 Len=0
8	3.661488	2002:1806:adcc::180...	2607:f0d0:2001:e:1:1...	HTTP	778	GET /image/ipv6.gif?id=1541349170 HTTP/1.1
9	3.865798	2607:f0d0:2001:e:1:1...	2002:1806:adcc::180...	HTTP	502	HTTP/1.1 200 OK (GIF89a)
10	4.070486	2002:1806:adcc::180...	2607:f0d0:2001:e:1:1...	TCP	94	52004 → 80 [ACK] Seq=1349 Ack=813 Win=16268 Len=0
11	5.605639	fe80::5083:650c:3af...	ff02::1:2	DHCPv6	147	Solicit XID: 0xc37c88 CID: 0001000114294f3cd48564a3b133
12	6.605624	fe80::5083:650c:3af...	ff02::1:2	DHCPv6	147	Solicit XID: 0xc37c88 CID: 0001000114294f3cd48564a3b133
13	8.605785	fe80::5083:650c:3af...	ff02::1:2	DHCPv6	147	Solicit XID: 0xc37c88 CID: 0001000114294f3cd48564a3b133
14	12.605926	fe80::5083:650c:3af...	ff02::1:2	DHCPv6	147	Solicit XID: 0xc37c88 CID: 0001000114294f3cd48564a3b133
15	18.875932	2607:f0d0:2001:e:1:1...	2002:1806:adcc::180...	TCP	94	80 → 52004 [FIN, ACK] Seq=813 Ack=1349 Win=8704 Len=0
16	18.876247	2002:1806:adcc::180...	2607:f0d0:2001:e:1:1...	TCP	94	52004 → 80 [ACK] Seq=1349 Ack=814 Win=16268 Len=0
17	20.606693	fe80::5083:650c:3af...	ff02::1:2	DHCPv6	147	Solicit XID: 0xc37c88 CID: 0001000114294f3cd48564a3b133

> Frame 1: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface unknown, id 0

▼ Ethernet II, Src: HewlettP_a7:bf:a3 (d4:85:64:a7:bf:a3), Dst: Cadant_31:bb:c1 (00:01:5c:31:bb:c1)

> Destination: Cadant_31:bb:c1 (00:01:5c:31:bb:c1)

> Source: HewlettP_a7:bf:a3 (d4:85:64:a7:bf:a3)

Type: IPv4 (0x0800)

> Internet Protocol Version 4, Src: 24.6.173.220, Dst: 192.88.99.1

> Internet Protocol Version 6, Src: 2002:1806:adcc::1806:adcc, Dst: 2607:f0d0:2001:e:1:120

> Transmission Control Protocol, Src Port: 52004, Dst Port: 80, Seq: 0, Len: 0

0000 00 01 5c 31 bb c1 d4
0010 00 5c 2d 0b 00 00 86
0020 63 01 60 00 00 00 06
0030 00 00 00 00 00 00 1f
0040 00 0e 00 01 00 00 06
0050 ca 26 00 00 00 00 86
0060 04 c4 01 03 03 02 01

Fig3.2 "IPv4" (0x0800)

4.

a. These are the ICMP types and codes below (see fig4.1):

- i. ICMP Type: 8 (Echo (ping) request)
ICMP Code: 0 Explanation: This ICMP type and code combination indicates an echo request message. It is commonly known as a "ping" request. In this case, a host is requesting a response from another host to check if it is reachable and measure the round-trip time.
- ii. ICMP Type: 0 (Echo (ping) reply)
ICMP Code: 0 Explanation: This ICMP type and code combination indicates an echo reply message. It is the response to an echo request (ping) message. It confirms the reachability of a host and provides the round-trip time.
- iii. ICMP Type: 3 (Destination Unreachable)
ICMP Code: 3 (Port Unreachable) Explanation: This ICMP type and code combination indicates a destination unreachable message due to a specific port being unreachable. It is used to inform the sender that the destination host is reachable, but the port requested is not open or available on that host.
- iv. ICMP Type: 11 (Time Exceeded)
ICMP Code: 0 (Time to Live (TTL) exceeded in transit) Explanation: This ICMP type and code combination indicates a time exceeded message. It is sent by a router when the TTL field in an IP packet reaches zero. The purpose is to prevent packets from circulating indefinitely in the network, and it helps identify routing problems.

- b. Yes, they are the same, however there are some differences. The intermediate network devices identified in the traceroute results and Wireshark analysis may not always be the same. Traceroute might miss some devices that Wireshark can detect, especially if ICMP messages are blocked or filtered along the network path. Additionally, Wireshark can provide more granular information about the network traffic, including devices that might not participate in the ICMP Time Exceeded message exchange.
- c. Three packets are sent (see fig 4.2)

Wi-Fi

DOSYA Düzenle Görünüm Git Yakala Analiz İstatistikler Telefon Kablosuz Araçlar Yardım

No.	Time	Source	Destination	Protocol	Length	Type	Code	Info
307	12.487361	10.8.37.167	93.184.216.34	ICMP	106	8	0	Echo (ping) request id=0x0001, seq=64/16384, ttl=1 (no response found!)
308	12.490381	10.8.32.1	10.8.37.167	ICMP	106	11,8	0,0	Time-to-live exceeded (Time to live exceeded in transit)
309	12.490842	10.8.37.167	93.184.216.34	ICMP	106	8	0	Echo (ping) request id=0x0001, seq=65/16640, ttl=1 (no response found!)
311	12.496398	10.8.32.1	10.8.37.167	ICMP	106	11,8	0,0	Time-to-live exceeded (Time to live exceeded in transit)
312	12.496827	10.8.37.167	93.184.216.34	ICMP	106	8	0	Echo (ping) request id=0x0001, seq=66/16896, ttl=1 (no response found!)
313	12.498955	10.8.32.1	10.8.37.167	ICMP	106	11,8	0,0	Time-to-live exceeded (Time to live exceeded in transit)
435	18.031648	10.8.37.167	93.184.216.34	ICMP	106	8	0	Echo (ping) request id=0x0001, seq=67/17152, ttl=2 (no response found!)
436	18.041057	194.27.0.33	10.8.37.167	ICMP	70	11,8	0,0	Time-to-live exceeded (Time to live exceeded in transit)
437	18.042755	10.8.37.167	93.184.216.34	ICMP	106	8	0	Echo (ping) request id=0x0001, seq=68/17408, ttl=2 (no response found!)
438	18.057570	194.27.0.33	10.8.37.167	ICMP	70	11,8	0,0	Time-to-live exceeded (Time to live exceeded in transit)
439	18.059047	10.8.37.167	93.184.216.34	ICMP	106	8	0	Echo (ping) request id=0x0001, seq=69/17664, ttl=2 (no response found!)
440	18.096983	194.27.0.33	10.8.37.167	ICMP	70	11,8	0,0	Time-to-live exceeded (Time to live exceeded in transit)
445	18.111623	194.27.0.33	10.8.37.167	ICMP	70	3	3	Destination unreachable (Port unreachable)
518	19.639333	194.27.0.33	10.8.37.167	ICMP	70	3	3	Destination unreachable (Port unreachable)
552	21.123710	194.27.0.33	10.8.37.167	ICMP	70	3	3	Destination unreachable (Port unreachable)
593	23.583998	10.8.37.167	93.184.216.34	ICMP	106	8	0	Echo (ping) request id=0x0001, seq=70/17920, ttl=3 (no response found!)
594	23.626685	194.27.0.249	10.8.37.167	ICMP	70	11,8	0,0	Time-to-live exceeded (Time to live exceeded in transit)
595	23.628371	10.8.37.167	93.184.216.34	ICMP	106	8	0	Echo (ping) request id=0x0001, seq=71/18176, ttl=3 (no response found!)
596	23.645305	194.27.0.249	10.8.37.167	ICMP	70	11,8	0,0	Time-to-live exceeded (Time to live exceeded in transit)
597	23.647283	10.8.37.167	93.184.216.34	ICMP	106	8	0	Echo (ping) request id=0x0001, seq=72/18432, ttl=3 (no response found!)
677	27.312747	194.27.0.249	10.8.37.167	ICMP	70	3	3	Destination unreachable (Port unreachable)
704	28.807842	194.27.0.249	10.8.37.167	ICMP	70	3	3	Destination unreachable (Port unreachable)
727	30.317387	194.27.0.249	10.8.37.167	ICMP	70	3	3	Destination unreachable (Port unreachable)
741	31.806233	10.8.37.167	93.184.216.34	ICMP	106	8	0	Echo (ping) request id=0x0001, seq=73/18688, ttl=4 (no response found!)
742	31.834167	212.154.96.69	10.8.37.167	ICMP	134	11,8	0,0	Time-to-live exceeded (Time to live exceeded in transit)

...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 249
Protocol: ICMP (1)
Header Checksum: 0xb8d1 [validation disabled]
[Header checksum status: Unverified]
Source Address: 152.195.64.129
Destination Address: 10.8.37.167

Internet Control Message Protocol

Type: 11 (Time-to-live exceeded)
Code: 0 (Time to live exceeded in transit)
Checksum: 0xf4ff [correct]
[Checksum Status: Good]
Unused: 00000000

Internet Protocol Version 4, Src: 10.8.37.167, Dst: 93.184.216.34

0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 92
Identification: 0x17da (6110)

Fig4.1 ICMP Type & Code Column

```
Komut İstemi
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\sude>tracert www.example.com

Tracing route to www.example.com [93.184.216.34]
over a maximum of 30 hops:

  0  3 ms  5 ms  2 ms  10.8.32.1
  1  9 ms  14 ms  37 ms  194.27.0.33
  2  42 ms  16 ms  *  194.27.0.249
  3  27 ms  17 ms  30 ms  69.96.154.212.static.turk.net [212.154.96.69]
  4  17 ms  44 ms  18 ms  212.156.64.45.static.turktelekom.com.tr [212.156.64.45]
  5  17 ms  46 ms  19 ms  06-ulus-sr14s-t2-2---06-ulus-sr12e-t3-1.statik.turktelekom.com.tr [81.212.222.209]
  6  *  *  *  Request timed out.
  7  *  *  *  Request timed out.
  8  44 ms  34 ms  61 ms  308-buk-col-2---06-ebgp-ulus-sr12e-k.statik.turktelekom.com.tr [212.156.139.6]
  9  47 ms  *  33 ms  buca-b3-link.ip.twelve99.net [62.115.37.72]
 10  *  *  *  Request timed out.
 11  60 ms  87 ms  59 ms  ffm-bb2-link.ip.twelve99.net [62.115.138.22]
 12  94 ms  98 ms  68 ms  prs-bb2-link.ip.twelve99.net [62.115.122.138]
 13  183 ms  154 ms  153 ms  rest-bb1-link.ip.twelve99.net [62.115.122.159]
 14  174 ms  157 ms  151 ms  ash-b2-link.ip.twelve99.net [62.115.123.123]
 15  171 ms  152 ms  149 ms  62.115.175.71
 16  164 ms  155 ms  155 ms  ae-65.core1.dcb.edgecastcdn.net [152.195.64.129]
 17  153 ms  185 ms  154 ms  93.184.216.34

Trace complete.
```

Fig4.2 Traceroute

NAT Section:

1. Network Address Translation (NAT) is a networking protocol that enables multiple devices on a private network to share a single public IP address when accessing the internet. NAT is typically implemented on routers, firewalls, or other network devices that act as a gateway between a private network and the public internet. NAT is a process in which one or more local IP addresses are translated into one or more Global IP address and vice versa in order to provide Internet access to the local hosts. NAT is often implemented at the WAN edge router to enable internet access in core, campus, branch, and colocation sites.
2. Packets are created as seen in *fig5.1*.
 - a. If these packets were collected from a real environment, they could be collected at various points within the network infrastructure. Here are some possible locations where these packets might be captured:

Endpoints: The LAN packet capture could be collected from an endpoint device within a local network, such as a desktop computer, laptop, or server. Similarly, the WAN packet capture

could be collected from an endpoint device in a different network, such as a server or gateway.

Network Taps: Packet captures can be obtained by using network taps, which are physical devices connected to a network link to capture and monitor network traffic. Network taps can be placed at various points in the network infrastructure, such as between routers, switches, or firewall devices.

Network Monitoring Systems: Organizations often deploy network monitoring systems or network analyzers to capture and analyze network traffic. These systems can be set up to capture packets at specific network segments or interfaces, providing visibility into the network traffic passing through those points.

Network Security Appliances: Network security appliances, such as intrusion detection systems (IDS) or intrusion prevention systems (IPS), often capture packets for security analysis and threat detection. These appliances are typically placed at strategic points in the network to monitor and analyze traffic.

Internet Service Provider (ISP): Packet captures could also be collected by an Internet Service Provider (ISP) as part of their network monitoring and troubleshooting activities. ISPs may capture packets at their network gateways or points of presence (PoPs) to monitor and ensure the quality and reliability of their services.

It's important to note that the specific location where these packets are collected would depend on the network infrastructure and the purpose of the packet capture. Different organizations and network administrators may choose different points for capturing network traffic based on their requirements and objectives, such as troubleshooting, performance monitoring, or security analysis.

- b. In the LAN packet capture (see fig5.2), we see communication between a device with the IP address 10.17.2.243 (source) and a server with the IP address 8.8.8.8 (destination). The communication is using both TCP and UDP protocols. Similarly, in the WAN packet capture (see fig5.3), we see communication between a device with the public IP address 102.37.24.187 (source) and the same server with the IP address 8.8.8.8 (destination).

The presence of a public IP address (102.37.24.187) in the WAN packet capture suggests that NAT is involved. When the LAN device communicates with the server on the WAN, the NAT protocol translates the private IP address (10.17.2.243) to a public IP address (102.37.24.187) before the packets are sent over the internet. This translation allows the LAN device to communicate with the server using a public IP address that can be routed on the WAN.

The LAN packet capture shows the communication initiated by the LAN device with a source port of 5000 and a destination port of 80. The WAN packet capture shows the translated

packets with the source IP address of 102.37.24.187 and the same source and destination ports.

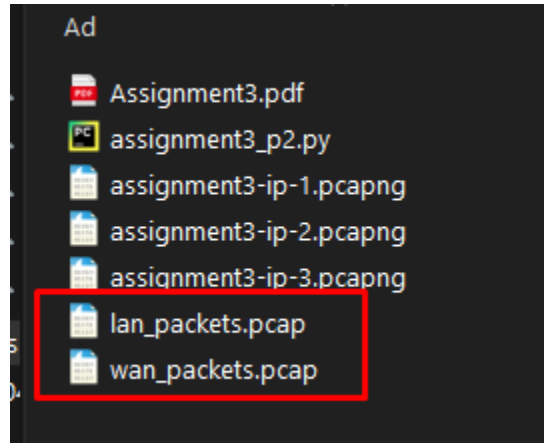


Fig5.1 Packets Created

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.17.2.243	8.8.8.8	TCP	54	5000 → 80 [SYN] Seq=0 Win=8192 Len=0
2	0.007000	8.8.8.8	10.17.2.243	TCP	54	80 → 5000 [SYN, ACK] Seq=0 Ack=0 Win=8192 Len=0
3	0.012000	10.17.2.243	8.8.8.8	TCP	54	[TCP Window Update] 5001 → 443 [ACK] Seq=1 Ack=1 Win=8192 Len=0
4	0.012000	8.8.8.8	10.17.2.243	TCP	54	[TCP Window Update] 443 → 5001 [PSH, ACK] Seq=1 Ack=1 Win=8192 Len=0
5	0.022000	10.17.2.243	8.8.8.8	UDP	42	3000 → 53 Len=0

Fig5.2 Packets Created

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	102.37.24.187	8.8.8.8	TCP	54	5000 → 80 [SYN] Seq=0 Win=8192 Len=0
2	0.007000	8.8.8.8	102.37.24.187	TCP	54	80 → 5000 [SYN, ACK] Seq=0 Ack=0 Win=8192 Len=0
3	0.012000	102.37.24.187	8.8.8.8	TCP	54	[TCP Window Update] 5001 → 443 [ACK] Seq=1 Ack=1 Win=8192 Len=0
4	0.012000	8.8.8.8	102.37.24.187	TCP	54	[TCP Window Update] 443 → 5001 [PSH, ACK] Seq=1 Ack=1 Win=8192 Len=0
5	0.022000	102.37.24.187	8.8.8.8	UDP	42	3000 → 53 Len=0

Fig5.3 Packets Created

DHCP Section:

- i. The "ipconfig /release" and "ipconfig /renew" commands are related to the Dynamic Host Configuration Protocol (DHCP). DHCP is a network protocol used to automatically assign IP addresses and network configuration settings to devices on a network.

"ipconfig /release": This command is used to release the currently assigned IP address lease obtained through DHCP. When executed, it sends a request to the DHCP server to release the IP address associated with the network interface of the device. By releasing the IP address, the device informs the DHCP server that it no longer requires the IP address lease.

"ipconfig /renew": This command is used to request a new IP address lease from the DHCP server. When executed, it sends a DHCP request to the server, asking for a new IP address assignment. The DHCP server responds by providing the device with a new IP address lease, along with other network configuration settings such as subnet mask, default gateway, and DNS servers.

- ii. One packet is created for ipconfig/release command (see fig6.2)
- Four packet is created for first ipconfig/renew command (see fig6.3)
- Two packet is created for first ipconfig/renew command (see fig6.4)

No.	Time	Source	Destination	Protocol	Length	Info
138	15.852792	10.8.37.167	10.8.32.9	DHCP	342	DHCP Release - Transaction ID 0x7a9d1518
306	26.994831	0.0.0.0	255.255.255.255	DHCP	344	DHCP Discover - Transaction ID 0xd5ef335a
307	27.509992	0.0.0.0	255.255.255.255	DHCP	344	DHCP Discover - Transaction ID 0xaf3c595
336	27.695499	10.8.32.9	255.255.255.255	DHCP	352	DHCP Offer - Transaction ID 0xaf3c595
337	27.695962	0.0.0.0	255.255.255.255	DHCP	370	DHCP Request - Transaction ID 0xaf3c595
345	27.799188	10.8.32.9	255.255.255.255	DHCP	357	DHCP ACK - Transaction ID 0xaf3c595
1492	38.228566	10.8.37.167	10.8.32.9	DHCP	358	DHCP Request - Transaction ID 0xae27197a
1500	38.326919	10.8.32.9	10.8.37.167	DHCP	357	DHCP ACK - Transaction ID 0xae27197a

> Frame 138: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface \Device\NPF_{00900E90-5F1E-40B} 0000 00 50 56 60
 > Ethernet II, Src: IntelCor_04:b0:3c (84:1b:77:04:b0:3c), Dst: VMware_83:3a:e1 (00:50:56:83:3a:e1) 0010 01 48 7c 10
 > Internet Protocol Version 4, Src: 10.8.37.167, Dst: 10.8.32.9 0020 20 09 00 40
 > User Datagram Protocol, Src Port: 68, Dst Port: 67 0030 15 18 00 60
 > Dynamic Host Configuration Protocol (Release) 0040 00 00 00 60
 0050 00 00 00 60
 0060 00 00 00 60
 0070 00 00 00 60

Fig6.1 Packets with DHCP Protocol

No.	Time	Source	Destination	Protocol	Length	Info
399	28.260512	10.8.32.9	255.255.255.255	DHCP	352	DHCP Offer - Transaction ID 0xf60c163b
402	28.362841	10.8.32.9	255.255.255.255	DHCP	357	DHCP ACK - Transaction ID 0xf60c163b
524	42.259716	10.8.37.167	10.8.32.9	DHCP	342	DHCP Release - Transaction ID 0x55a107dd

Fig6.2 One Packet Created for "ipconfig/release" Command

No.	Time	Source	Destination	Protocol	Length	Info
399	28.260512	10.8.32.9	255.255.255.255	DHCP	352	DHCP Offer - Transaction ID 0xf60c163b
402	28.362841	10.8.32.9	255.255.255.255	DHCP	357	DHCP ACK - Transaction ID 0xf60c163b
524	42.259716	10.8.37.167	10.8.32.9	DHCP	342	DHCP Release - Transaction ID 0x55a107dd
593	47.308360	0.0.0.0	255.255.255.255	DHCP	344	DHCP Discover - Transaction ID 0x54556882
594	47.409280	10.8.32.9	255.255.255.255	DHCP	352	DHCP Offer - Transaction ID 0x54556882
595	47.409799	0.0.0.0	255.255.255.255	DHCP	370	DHCP Request - Transaction ID 0x54556882
601	47.512337	10.8.32.9	255.255.255.255	DHCP	357	DHCP ACK - Transaction ID 0x54556882

Fig6.3 Four Packets Created for first "ipconfig/renew" Command

No.	Time	Source	Destination	Protocol	Length	Info
399	28.260512	10.8.32.9	255.255.255.255	DHCP	352	DHCP Offer - Transaction ID 0xf60c163b
402	28.362841	10.8.32.9	255.255.255.255	DHCP	357	DHCP ACK - Transaction ID 0xf60c163b
524	42.259716	10.8.37.167	10.8.32.9	DHCP	342	DHCP Release - Transaction ID 0x55a107dd
593	47.308360	0.0.0.0	255.255.255.255	DHCP	344	DHCP Discover - Transaction ID 0x54556882
594	47.409280	10.8.32.9	255.255.255.255	DHCP	352	DHCP Offer - Transaction ID 0x54556882
595	47.409799	0.0.0.0	255.255.255.255	DHCP	370	DHCP Request - Transaction ID 0x54556882
601	47.512337	10.8.32.9	255.255.255.255	DHCP	357	DHCP ACK - Transaction ID 0x54556882
1083	53.143840	10.8.37.167	10.8.32.9	DHCP	358	DHCP Request - Transaction ID 0x1133b75f
1086	53.150326	10.8.32.9	10.8.37.167	DHCP	357	DHCP ACK - Transaction ID 0x1133b75f

Fig6.4 Two Packets Created for second "ipconfig/renew" Command

iii. First "ipconfig /renew":

Packet 593: DHCP Discover - This packet is sent by the device to discover available DHCP servers on the network.

Packet 594: DHCP Offer - This packet is sent by the DHCP server in response to the DHCP Discover. It offers an IP address lease to the device.

Packet 595: DHCP Request - This packet is sent by the device to request the offered IP address lease from the DHCP server.

Packet 601: DHCP ACK - This packet is sent by the DHCP server in response to the DHCP Request. It acknowledges the request and provides the device with the IP address lease.

Second "ipconfig /renew":

Packet 1083: DHCP Request - This packet is sent by the device to request a new IP address lease from the DHCP server.

Packet 1086: DHCP ACK - This packet is sent by the DHCP server in response to the DHCP Request. It acknowledges the request and provides the device with the new IP address lease.

From the packet captures, we can see that the first "ipconfig /renew" command goes through the complete DHCP process, starting with a DHCP Discover, receiving a DHCP Offer, sending a DHCP Request, and receiving a DHCP ACK. This indicates that the device had no previous IP address lease or it expired, and it is obtaining a new IP address from the DHCP server.

On the other hand, the second "ipconfig /renew" command only involves a DHCP Request and a DHCP ACK. This suggests that the device already had an existing IP address lease, and it is simply requesting a renewal of the same IP address from the DHCP server.

Therefore, the differences in the captured packets reflect the different stages of the DHCP process and the specific actions taken by the device depending on whether it needs to obtain a new IP address or renew an existing IP address lease.

iv. **DHCP packets are observed as the result of the first "ipconfig /renew" command:**

DHCP Discover (Packet 593): This packet is sent by the device with the source IP address of 0.0.0.0 and the broadcast destination address (255.255.255.255). The DHCP Discover packet is used by the device to request network configuration information from DHCP servers on the network. The Transaction ID (0x54556882) helps identify this specific DHCP transaction.

DHCP Offer (Packet 594): This packet is sent by a DHCP server in response to the DHCP Discover. It has the DHCP server's IP address (10.8.32.9) as the source and a broadcast destination address. The DHCP Offer packet provides an IP address lease offer to the device, along with other network configuration parameters. The Transaction ID (0x54556882) matches the one in the DHCP Discover packet.

DHCP Request (Packet 595): This packet is sent by the device to accept the offered IP address lease from the DHCP server. It has the source IP address of 0.0.0.0 and the broadcast destination address. The DHCP Request packet indicates the device's acceptance of the DHCP server's offer. The Transaction ID (0x54556882) matches the one in the DHCP Discover packet.

DHCP ACK (Packet 601): This packet is sent by the DHCP server in response to the DHCP Request. It acknowledges the device's acceptance of the IP address lease. The DHCP ACK packet provides the device with the confirmed IP address lease and additional network configuration details. The Transaction ID (0x54556882) matches the one in the DHCP Discover packet.

Therefore, the captured packets for the first "ipconfig /renew" command include the DHCP Discover, DHCP Offer, DHCP Request, and DHCP ACK packets. These DHCP packet types are fundamental to the DHCP process, allowing devices to obtain IP addresses and network settings dynamically from DHCP servers.

v. Yes, by inspecting the option fields in the captured DHCP packets as the result of the first "ipconfig /renew" command, there are common options present in all packets (see fig6.5 & fig6.6 & fig6.7 & fig6.7). These common options are as follows:

DHCP Message Type (Option 53): This option specifies the type of DHCP message being transmitted. In all captured packets, this option is present and indicates the message type for each DHCP packet (Discover, Offer, Request, ACK).

End Option (Option 255): The End option is a standard DHCP option and is mandatory to signify the completion of the options section within the DHCP packet. It ensures that the DHCP client or server parsing the packet knows where the options field ends and where other fields or sections of the packet begin.

524	47.259716	10.8.37.167	10.8.32.9	DHCP	342 DHCP Release	- Transaction ID 0x55a107dd
593	47.308360	0.0.0.0	255.255.255.255	DHCP	344 DHCP Discover	- Transaction ID 0x54556882
594	47.409280	10.8.32.9	255.255.255.255	DHCP	352 DHCP Offer	- Transaction ID 0x54556882
595	47.409799	0.0.0.0	255.255.255.255	DHCP	370 DHCP Request	- Transaction ID 0x54556882
601	47.512337	10.8.32.9	255.255.255.255	DHCP	357 DHCP ACK	- Transaction ID 0x54556882
1083	53.143840	10.8.37.167	10.8.32.9	DHCP	358 DHCP Request	- Transaction ID 0x1133b75f
1086	53.150326	10.8.32.9	10.8.37.167	DHCP	357 DHCP ACK	- Transaction ID 0x1133b75f


```

> Frame 593: 344 bytes on wire (2752 bits), 344 bytes captured (2752 bits) on interface \Device\NPF_{00900E90-5F1E-40B7-8F14-11AAC002C5D1}, id 0
> Ethernet II, Src: IntelCor_04:b0:3c (84:1b:77:04:b0:3c), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
> User Datagram Protocol, Src Port: 68, Dst Port: 67
  > Dynamic Host Configuration Protocol (Discover)
    Message type: Boot Request (1)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x54556882
    Seconds elapsed: 0
    > Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 0.0.0.0
    Next server IP address: 0.0.0.0
    Relay agent IP address: 0.0.0.0
    Client MAC address: IntelCor_04:b0:3c (84:1b:77:04:b0:3c)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
    > Option: (53) DHCP Message Type (Discover)
    > Option: (61) Client identifier
    > Option: (50) Requested IP Address (10.8.37.167)
    > Option: (12) Host Name
    > Option: (60) Vendor class identifier
    > Option: (55) Parameter Request List
    > Option: (255) End
  
```

Fig6.5 Options of the first packet of the first ipconfig/release command

593	47.308360	0.0.0.0	255.255.255.255	DHCP	344 DHCP Discover	- Transaction ID 0x54556882
594	47.409280	10.8.32.9	255.255.255.255	DHCP	352 DHCP Offer	- Transaction ID 0x54556882
595	47.409799	0.0.0.0	255.255.255.255	DHCP	370 DHCP Request	- Transaction ID 0x54556882
601	47.512337	10.8.32.9	255.255.255.255	DHCP	357 DHCP ACK	- Transaction ID 0x54556882
1083	53.143840	10.8.37.167	10.8.32.9	DHCP	358 DHCP Request	- Transaction ID 0x1133b75f
1086	53.150326	10.8.32.9	10.8.37.167	DHCP	357 DHCP ACK	- Transaction ID 0x1133b75f

> Frame 594: 352 bytes on wire (2816 bits), 352 bytes captured (2816 bits) on interface \Device\NPF_{00900E90-5F1E-4DB7-8F14-11AAC002C5D1}, id 0
 > Ethernet II, Src: VMware_83:3a:e1 (00:50:56:83:3a:e1), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 > Internet Protocol Version 4, Src: 10.8.32.9, Dst: 255.255.255.255
 > User Datagram Protocol, Src Port: 67, Dst Port: 68
 > Dynamic Host Configuration Protocol (Offer)
 Message type: Boot Reply (2)
 Hardware type: Ethernet (0x01)
 Hardware address length: 6
 Hops: 0
 Transaction ID: 0x54556882
 Seconds elapsed: 0
 > Bootp flags: 0x0000 (Unicast)
 Client IP address: 0.0.0.0
 Your (client) IP address: 10.8.37.167
 Next server IP address: 10.8.32.9
 Relay agent IP address: 0.0.0.0
 Client MAC address: IntelCor_04:b0:3c (84:1b:77:04:b0:3c)
 Client hardware address padding: 00000000000000000000
 Server host name not given
 Boot file name not given
 Magic cookie: DHCP
 > Option: (53) DHCP Message Type (Offer)
 > Option: (1) Subnet Mask (255.255.224.0)
 > Option: (58) Renewal Time Value
 > Option: (59) Rebinding Time Value
 > Option: (51) IP Address Lease Time
 > Option: (54) DHCP Server Identifier (10.8.32.9)
 > Option: (3) Router
 > Option: (6) Domain Name Server
 > Option: (15) Domain Name
 > Option: (255) End

Fig6.6 Options of the second packet of the first ipconfig/release command

593	47.308360	0.0.0.0	255.255.255.255	DHCP	344 DHCP Discover	- Transaction ID 0x54556882
594	47.409280	10.8.32.9	255.255.255.255	DHCP	352 DHCP Offer	- Transaction ID 0x54556882
595	47.409799	0.0.0.0	255.255.255.255	DHCP	370 DHCP Request	- Transaction ID 0x54556882
601	47.512337	10.8.32.9	255.255.255.255	DHCP	357 DHCP ACK	- Transaction ID 0x54556882
1083	53.143840	10.8.37.167	10.8.32.9	DHCP	358 DHCP Request	- Transaction ID 0x1133b75f
1086	53.150326	10.8.32.9	10.8.37.167	DHCP	357 DHCP ACK	- Transaction ID 0x1133b75f

> Frame 595: 370 bytes on wire (2960 bits), 370 bytes captured (2960 bits) on interface \Device\NPF_{00900E90-5F1E-4DB7-8F14-11AAC002C5D1}, id 0
 > Ethernet II, Src: IntelCor_04:b0:3c (84:1b:77:04:b0:3c), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 > Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
 > User Datagram Protocol, Src Port: 68, Dst Port: 67
 > Dynamic Host Configuration Protocol (Request)
 Message type: Boot Request (1)
 Hardware type: Ethernet (0x01)
 Hardware address length: 6
 Hops: 0
 Transaction ID: 0x54556882
 Seconds elapsed: 0
 > Bootp flags: 0x0000 (Unicast)
 Client IP address: 0.0.0.0
 Your (client) IP address: 0.0.0.0
 Next server IP address: 0.0.0.0
 Relay agent IP address: 0.0.0.0
 Client MAC address: IntelCor_04:b0:3c (84:1b:77:04:b0:3c)
 Client hardware address padding: 00000000000000000000
 Server host name not given
 Boot file name not given
 Magic cookie: DHCP
 > Option: (53) DHCP Message Type (Request)
 > Option: (61) Client Identifier
 > Option: (50) Requested IP Address (10.8.37.167)
 > Option: (54) DHCP Server Identifier (10.8.32.9)
 > Option: (12) Host Name
 > Option: (81) Client Fully Qualified Domain Name
 > Option: (60) Vendor class identifier
 > Option: (55) Parameter Request List
 > Option: (255) End

Fig6.7 Options of the third packet of the first ipconfig/release command

593	47.308360	0.0.0.0	255.255.255.255	DHCP	344 DHCP Discover	- Transaction ID 0x54556882
594	47.409280	10.8.32.9	255.255.255.255	DHCP	352 DHCP Offer	- Transaction ID 0x54556882
595	47.409390	0.0.0.0	255.255.255.255	DHCP	370 DHCP Request	- Transaction ID 0x54556882
601	47.512337	10.8.32.9	255.255.255.255	DHCP	357 DHCP ACK	- Transaction ID 0x54556882
1085	53.142840	10.8.37.167	10.8.32.9	DHCP	336 DHCP Request	- Transaction ID 0x1133b75f
1086	53.150326	10.8.32.9	10.8.37.167	DHCP	357 DHCP ACK	- Transaction ID 0x1133b75f

```
> Frame 601: 357 bytes on wire (2856 bits), 357 bytes captured (2856 bits) on interface \Device\NPF_{00900E90-5F1E-4DB7-8F14-11AAC002C5D1}, id 0
> Ethernet II, Src: Vmware_83:3a:e1 (00:50:56:83:3a:e1), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 10.8.32.9, Dst: 255.255.255.255
> User Datagram Protocol, Src Port: 67, Dst Port: 68
> Dynamic Host Configuration Protocol (ACK)
  Message type: Boot Reply (2)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0x54556882
  Seconds elapsed: 0
  > Bootp flags: 0x0000 (Unicast)
  Client IP address: 0.0.0.0
  Your (client) IP address: 10.8.37.167
  Next server IP address: 0.0.0.0
  Relay agent IP address: 0.0.0.0
  Client MAC address: IntelCor_04:b0:3c (84:1b:77:04:b0:3c)
  Client hardware address padding: 00000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
  > Option: (53) DHCP Message Type (ACK)
  > Option: (58) Renewal Time Value
  > Option: (59) Rebinding Time Value
  > Option: (51) IP Address Lease Time
  > Option: (54) DHCP Server Identifier (10.8.32.9)
  > Option: (1) Subnet Mask (255.255.224.0)
  > Option: (81) Client Fully Qualified Domain Name
  > Option: (3) Router
  > Option: (6) Domain Name Server
  > Option: (15) Domain Name
  > Option: (255) End
```

Fig6.8 Options of the fourth packet of the first ipconfig/release command