CS 315: Computer Networks Lab
Spring 2024-25, IIT Dharwad

Chidurala Tejaswini
(220010012 / CS22BT012)

Assignment-7

Wireshark Lab: IP

February 17, 2025

Part 0: Paste a screenshot of your system IP address, using `ipconfig` (on Windows) or `ifconfig` (on Mac and Linux), and fill out this Google form to submit the details of your system. The same system must be used to attempt all exercises of this lab.

```
user@sysad-HP-Elite-Tower-600-G9-Desktop-PC:~$ ifconfig
eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.240.118.97  netmask 255.255.248.0  broadcast 10.240.119.255
        inet6 fe80::1d6b:1bfb:2bd6:ef0d  prefixlen 64  scopeid 0x20<link>
        ether e0:73:e7:0a:99:9a  txqueuelen 1000  (Ethernet)
        RX packets 1101835  bytes 357717957 (357.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 82394  bytes 11054313 (11.0 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 19  memory 0x80900000-80920000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 9790  bytes 1169259 (1.1 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9790  bytes 1169259 (1.1 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Part 1: Basic IPv4

In this part, we'll analyze packets in a trace of IPv4 datagrams sent and received by the `Ping`. Use the following to capture and analyze an IPv4 trace in Wireshark, open a terminal and follow these steps:

On Linux/macOS:

```
ping google.com -c 5
```

On Windows:

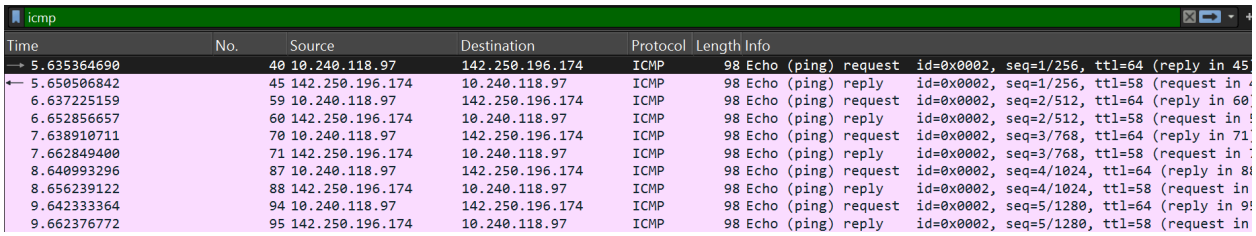
```
ping -n 5 google.com
```

```
user@sysad-HP-Elite-Tower-600-G9-Desktop-PC:~$ ping google.com -c 5
PING google.com (142.250.193.110) 56(84) bytes of data.
64 bytes from maa05s24-in-f14.1e100.net (142.250.193.110): icmp_seq=1 ttl=58 time=15.2 ms
64 bytes from maa05s24-in-f14.1e100.net (142.250.193.110): icmp_seq=2 ttl=58 time=15.2 ms
64 bytes from maa05s24-in-f14.1e100.net (142.250.193.110): icmp_seq=3 ttl=58 time=15.6 ms
64 bytes from maa05s24-in-f14.1e100.net (142.250.193.110): icmp_seq=4 ttl=58 time=15.5 ms
64 bytes from maa05s24-in-f14.1e100.net (142.250.193.110): icmp_seq=5 ttl=58 time=15.3 ms

--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 15.229/15.382/15.614/0.155 ms
```

Answer the following questions.

1. **What is the source and destination IP address for the above ping request you observe in your trace?**



   - **Source IP address:** 10.240.110.97
   - **Destination IP address:** 142.250.196.174

2. **Mention the protocol used in the ping request.**

- Internet Control Message Protocol (ICMP) is used for Ping requests and replies.
- ICMP is a network-layer protocol used for diagnostics and error reporting.

3. **State the number of fields in the IPv4 header along with its size.**

   Fields in the IPv4 Header:

```
▾ Internet Protocol Version 4, Src: 10.240.118.97, Dst: 142.250.196.174
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0xc5a5 (50597)
  ▸ 010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: ICMP (1)
    Header Checksum: 0xa009 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.240.118.97
    Destination Address: 142.250.196.174
    [Stream index: 6]
▸ Internet Control Message Protocol
```

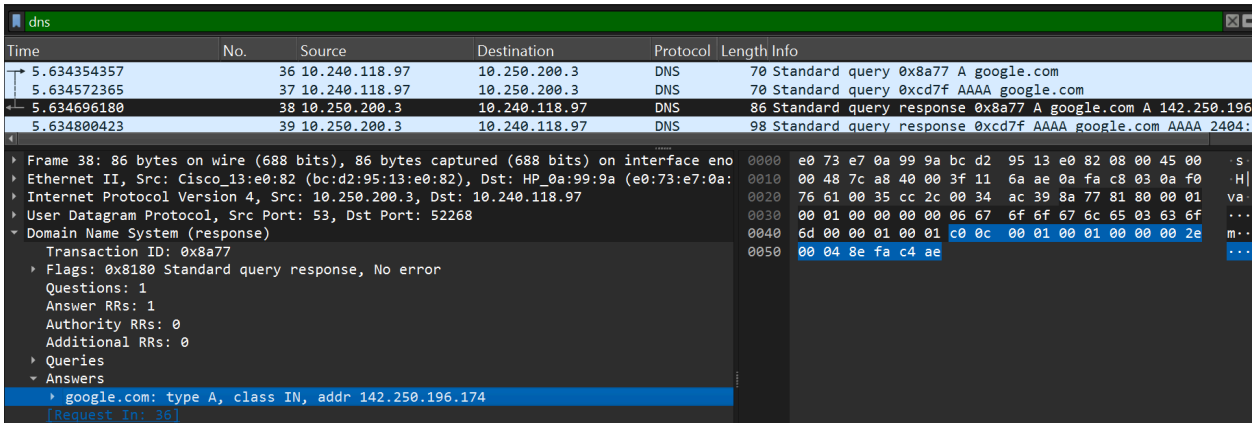
Sno	Field in the IPV4 Header	Size of the Field
1	Version	4 bits
2	Header Length in 32 -bit words	4 bits
3	Differentiated Services Field	1 byte (or) 8 bits
4	Total Length	2 bytes (or) 16 bits
5	Identification	2 bytes (or) 16 bits
6	Flags	3 bits
7	Fragment Offset	13 bits
8	Time to Live	1 byte (or) 8 bits
9	Protocol	1 byte (or) 8 bits
10	Header Checksum	2 bytes (or) 16 bits
11	Source Address	4 bytes (or) 32 bits
12	Destination Address	4 bytes (or) 32 bits

Select the first UDP segment sent by your computer via the `Ping` command .

4. **List the type of queries used for the above request. Expand the Internet Protocol part of the packet in the packet details window. What is the version of the IP address used for the above request?**

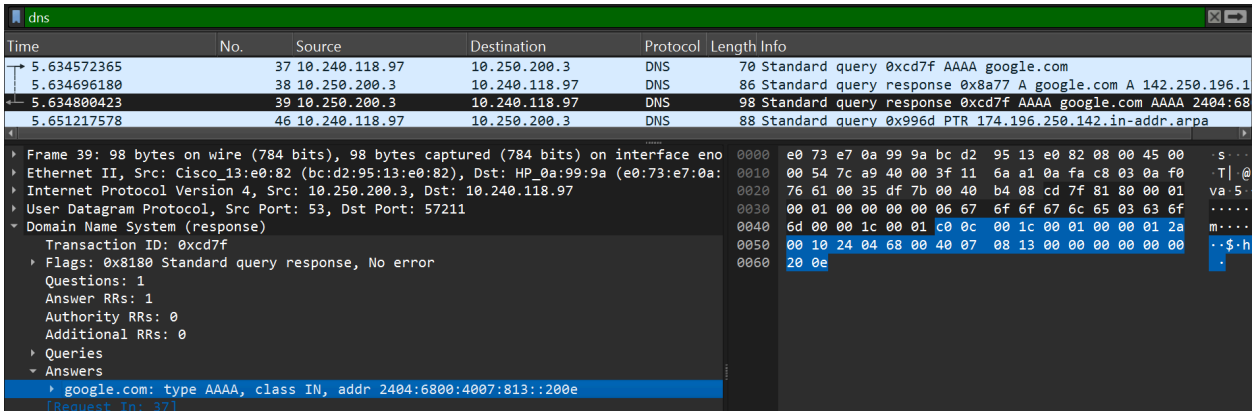
Types of Queries Used for the Request: From the packet capture, we can see that the system performs **DNS** queries before sending an ICMP Echo Request (Ping) to `google.com`:

1. **A Record Query (IPv4)**



- The system sent a **DNS A record query** to resolve `google.com` to an IPv4 address.
- The response returned an IPv4 address: **142.250.196.174**.

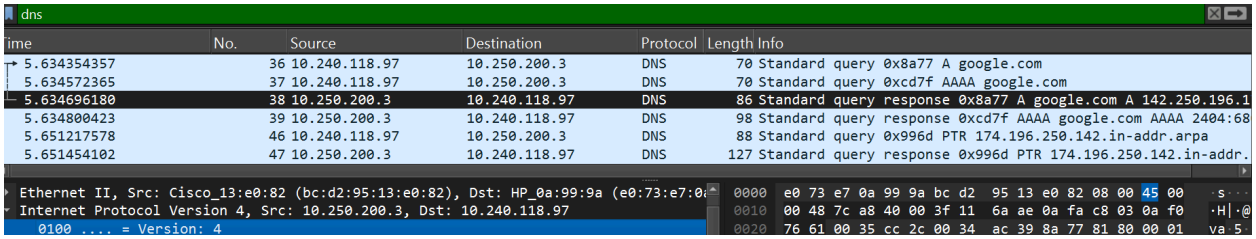
2. **AAAA Record Query (IPv6)**



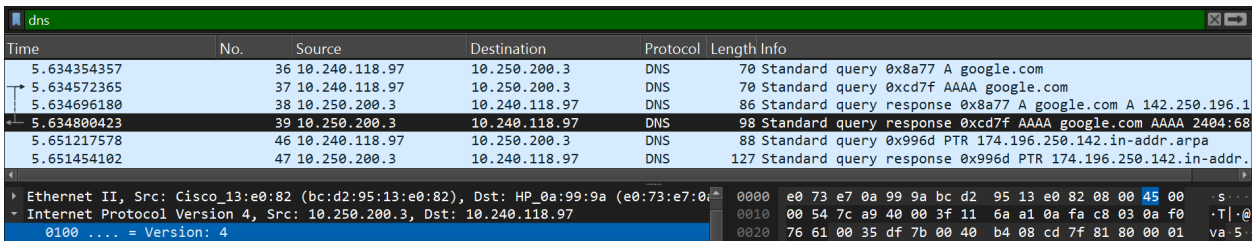
- The system also sent a **DNS AAAA record query** to resolve `google.com` to an IPv6 address.
- The response returned an IPv6 address: **2404:6800:4007:813::200e**.

These queries confirm that the system attempted to resolve both **IPv4 and IPv6 addresses** for `google.com`.

Expanding the Internet Protocol in the Packet Details Window:



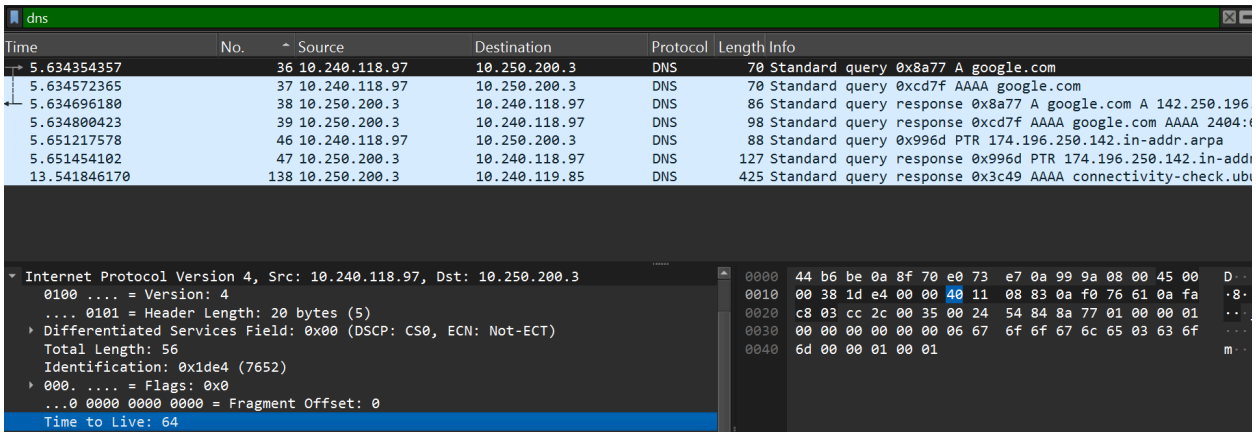
The version of the IP address used for the above request=IPV4



From the expanded packet details:-The packet capture shows an **IPv4 (Version: 4)** packet being used in this request.However, since the DNS response also returned an **IPv6 address**, the system may use **IPv6** for further communication.

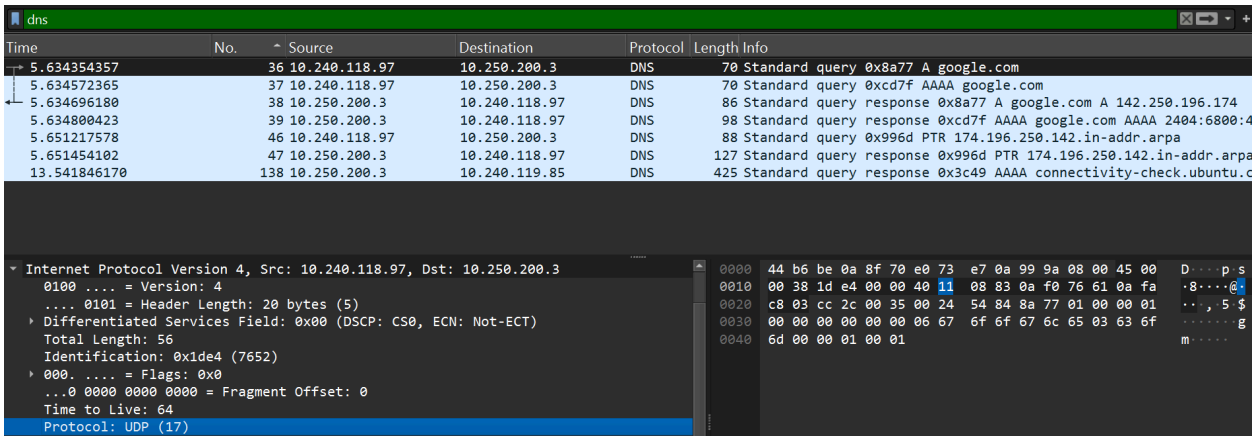
- This confirms that the system performed **both A and AAAA DNS queries**, but the packet captured is using **IPv4** for this particular request.

5. **What is the value in the time-to-live (TTL) field in this IPv4 datagram's header?**



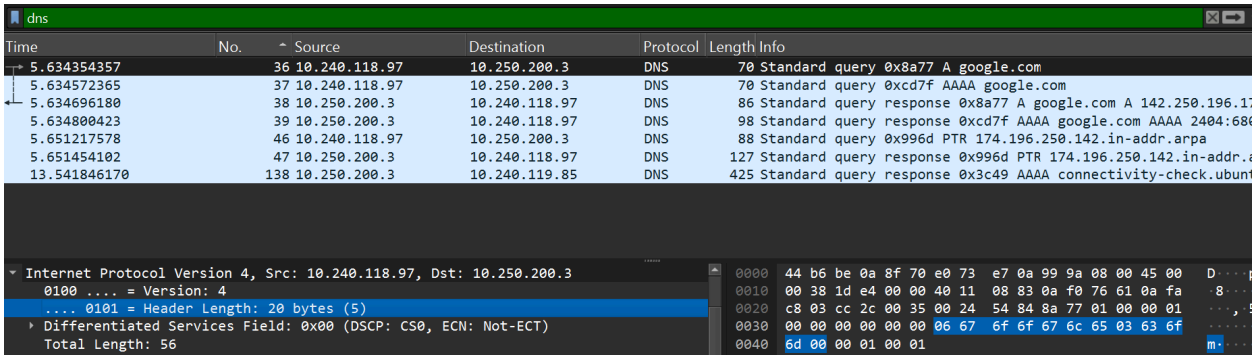
The value in the time-to-live (TTL) field in this IPv4 datagram's header is=64

6. **What is the value in the upper layer protocol field in this IPv4 datagram's header?**



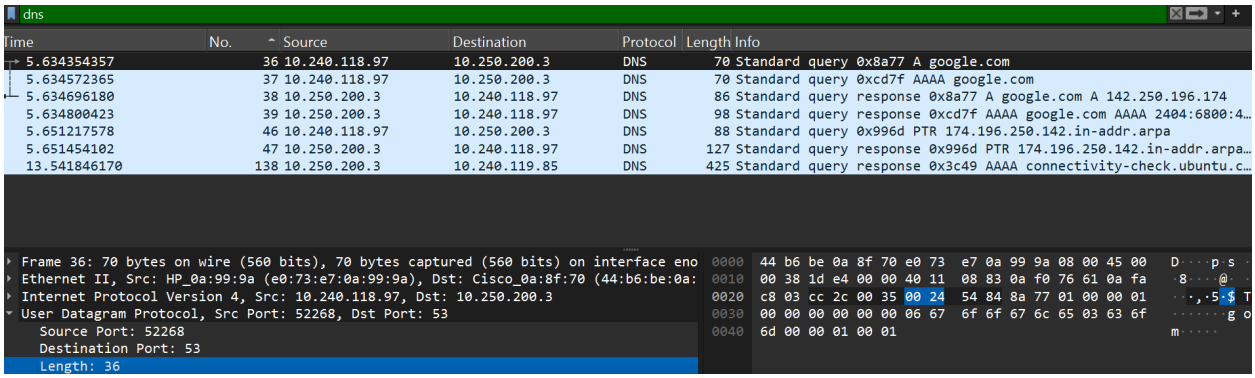
The value in the upper layer protocol field in this IPv4 datagram's header is **UDP(17)**.This indicates that IPv4 is being used as a service by the transport layer's User Datagram Protocol.

7. **How many bytes are in the payload of the IP datagram? Explain how you determined the number of payload bytes.**



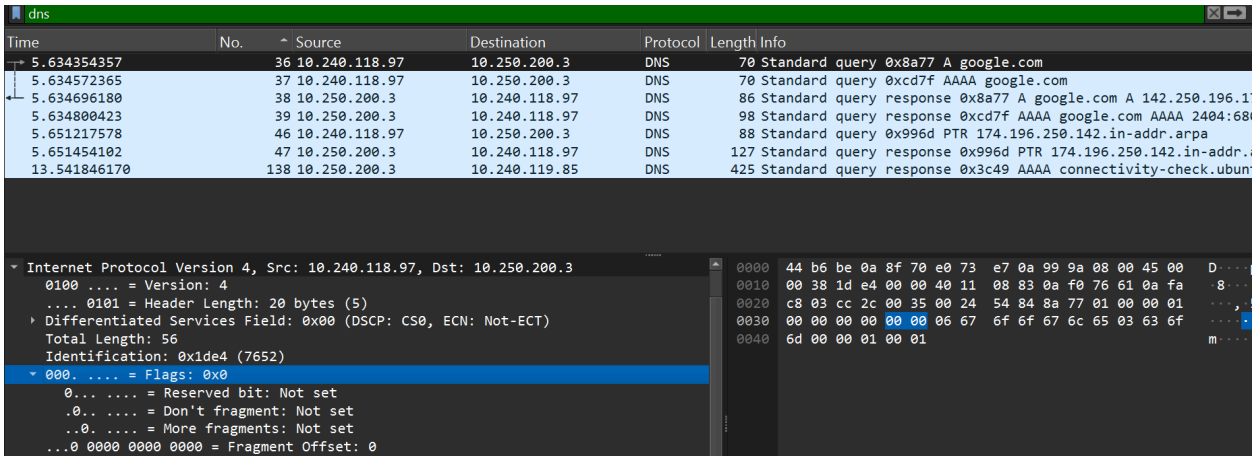
Payload Size = Total Length - IP Header Size

The IP datagram payload has 56 − 20 = **36 Bytes**.



This is also verifiable as it is the 'Length' field in the UDP details.

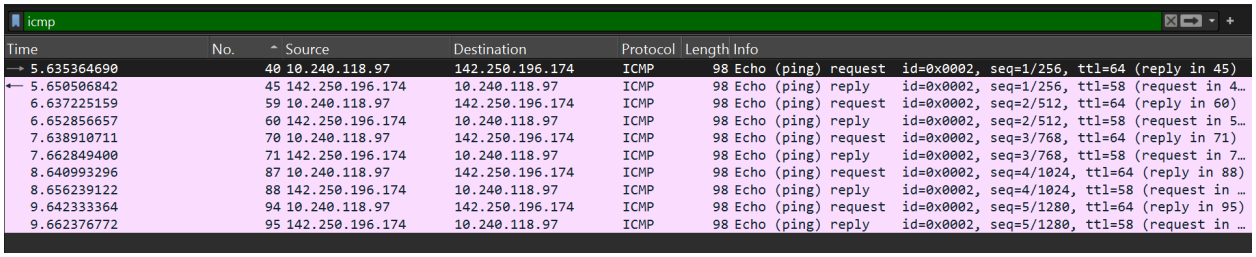
8. **Has this IP datagram been fragmented? Explain how you determined whether or not the datagram has been fragmented.**



All of the flag bits are zero. Hence, the 'more fragments' field is zero. Thus, this IP datagram **has not been fragmented**.

Next, let's look at the ICMP packets being sent from your computer and returned to your computer. The display filter that you can use to show just these packets is "icmp".

9. **Mention the number of requests and replies you observe from your computer to the requested domain name.**

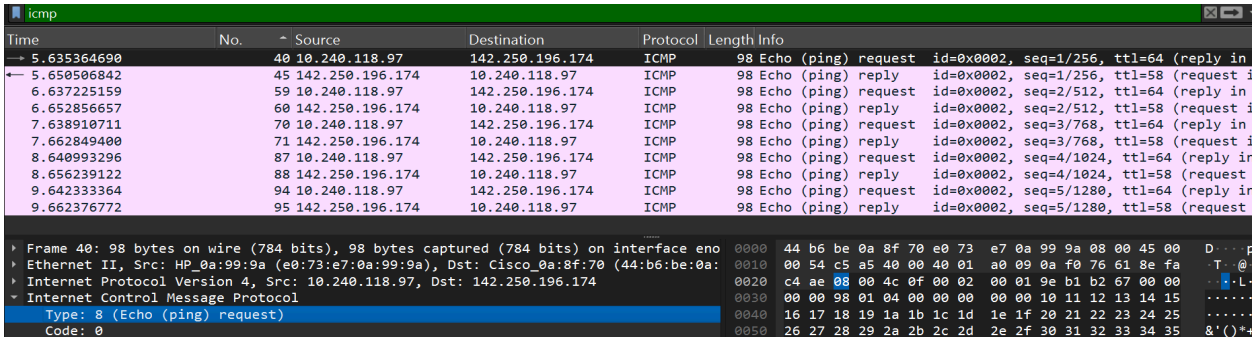


Since the command `ping google.com -c 5` sends **5 requests**, we can see:

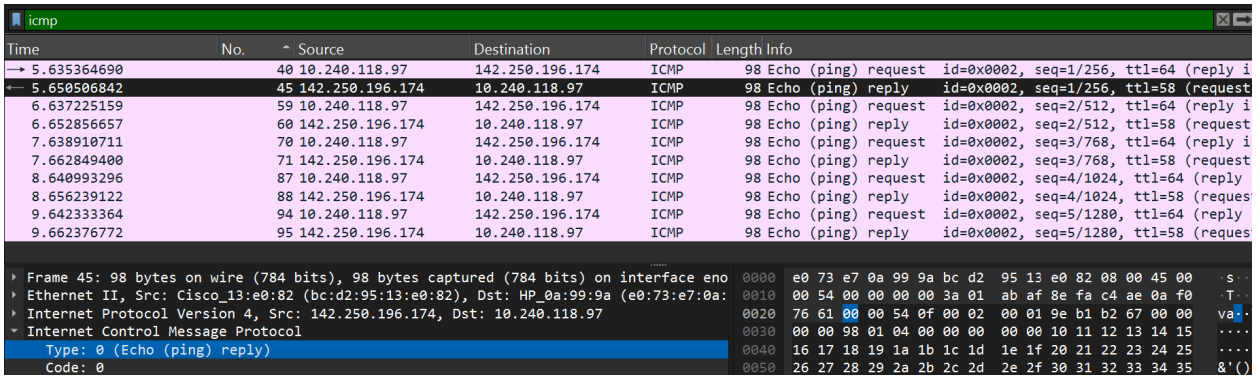
- **5 ICMP Echo Requests** (sent) and **5 ICMP Echo Replies** (received).

10. **State the types of ping requests and replies you observe in the trace for the requested domain name.**

Types of Ping Requests and Replies



ICMP Echo Request (Type **8**, Code **0**)



ICMP Echo Reply (Type **0**, Code **0**)

11. **List in detail the fields that vary as well as remain constant from the ping request and replies in the IP datagrams.**

Fields That Vary:

1. **Identification (16 bits)**
   - A unique identifier assigned to each IP datagram.
   - The **request and reply have different identification numbers**.
2. **Checksum (16 bits)**
   - Used for **error detection** in the IP header.
   - **Recalculated for every packet** because it depends on fields that change (like TTL).
3. **Flags (3 bits)**
   - Controls fragmentation behavior (e.g., **Don't Fragment (DF) flag**).
   - May vary depending on network conditions and whether fragmentation occurs.
4. **Fragment Offset (13 bits)**
   - Specifies the **position of a fragment within the original datagram**.

    - Changes **only if the packet is fragmented**, which is rare in standard ping packets.
5. **Time To Live (TTL) (8 bits)**
    - Determines the **maximum number of hops** a packet can travel.
    - **Decreases by 1 at each router hop** and is **different in request and reply** because the **reply starts from a different system**.
6. **Source IP Address (32 bits)**
    - In the **request**, the source IP is the **sender (client)**.
    - In the **reply**, the source IP is the **destination (pinged host)**.
7. **Destination IP Address (32 bits)**
    - In the **request**, the destination IP is the **host being pinged**.
    - In the **reply**, the destination IP is the **original sender**

<u>Fields that remain constant</u>

1.Version (4 bits)

- Specifies the IP version (IPv4 or IPv6).
- Remains constant as the protocol version does not change.

2.Header Length (4 bits)

- Indicates the length of the IP header in **32-bit words**.
- Stays constant unless **optional fields** are included (rare in ping packets).

3.Differentiated Services (8 bits) (formerly TOS - Type of Service)

- Used for **Quality of Service (QoS)** and priority handling.
- Generally constant unless explicitly modified by the sender.

4.Total Length (16 bits)

- Represents the **entire size of the packet (header + payload)**.
- **For a given request/reply pair, this remains constant** but may vary between different ping sessions.

5.Protocol (8 bits)

- Specifies the **transport-layer protocol**.
- **Always set to 1** for **ICMP (Internet Control Message Protocol)**.

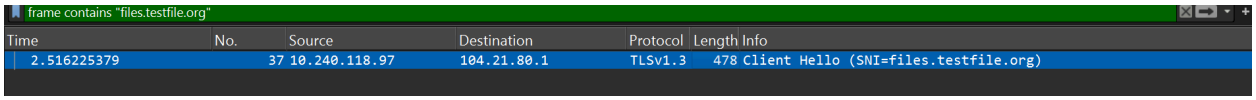
Part 2: Fragmentation

Use the following command in the terminal and capture the trace in Wireshark to answer the following questions.

```
wget "https://files.testfile.org/PDF/50MB-TESTFILE.ORG.pdf"
```

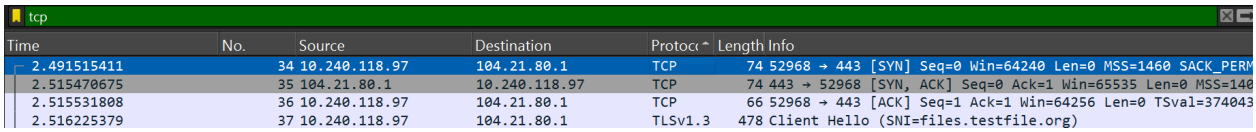
Answer the following:

1. **What are the IP addresses of the client and the above-requested domain?**



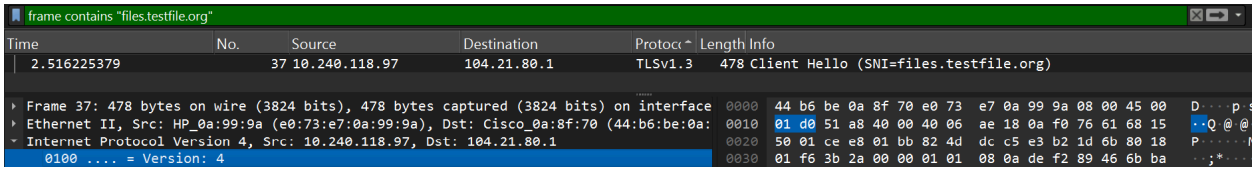
   - **Client IP** → 10.240.118.97
   - **Server IP** → 104.21.80.1

2. **Which transport layer protocol is being used to establish the connection between the client and the requested domain?**
   Since **wget** is used to download files over HTTP or HTTPS, it relies on the TCP (Transmission Control Protocol) at the transport layer.



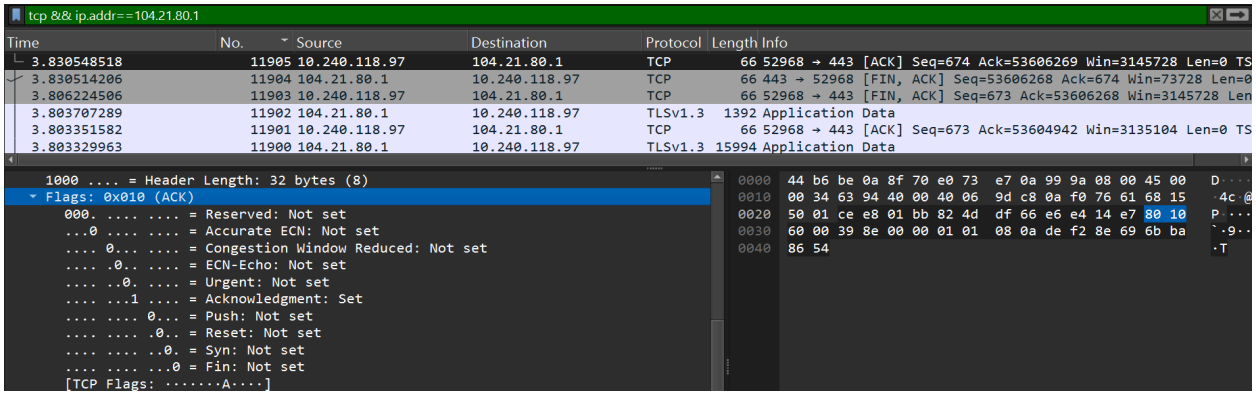
   The transport layer protocol used is: **TCP** (Transmission Control Protocol)

3. **What is the IP version?**



   The IP Version: **4(IPV4)**

4. **In the entire TCP stream for the above request, what is the value of the last <u>Ack number</u> and what does it signify?**



   The **last ACK number** in the TCP stream is **53606269** (found in **packet 11905**).
   By observing the flow graph -

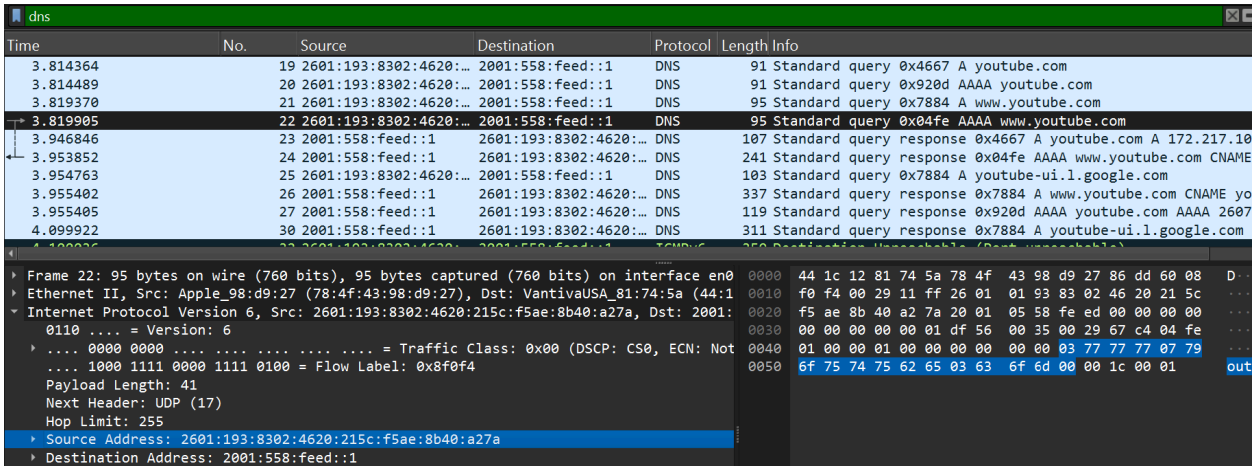
- The **last ACK number(Packet: 11905)** in a TCP stream represents the next expected byte from the sender, confirming that all preceding bytes have been successfully received.
- It signifies the **completion of data transfer**, ensuring that the entire file or message has been acknowledged by the receiver.
- The **last ACK segment** follows the **[FIN, ACK]** packet, which is used to terminate the TCP connection.
- This final ACK marks the **completion of the TCP 4-way handshake**, confirming that the connection has been properly closed.
- This acknowledgment is **crucial for verifying data integrity** and ensuring the proper termination of the TCP session.

Part 3: IPv6

In this final section, we'll take a quick look at the IPv6 datagram using Wireshark. The Internet is still primarily at IPv4 network, and your computer or your ISP may not be configured for IPv6, let's look at a trace of already captured packets that contain some IPv6 packets. To generate this trace, our web browser opened the `youtube.com` homepage. YouTube (and Google) provide fairly widespread support for IPv6.Open the file provided `Assignment_7_Part3_IPv6.pcapng`. This is a DNS request (contained in an IPv6 datagram) to an IPv6 DNS server for the IPv6 address of youtube.com. The DNS AAAA request type is used to resolve names to IPv6 IP addresses.

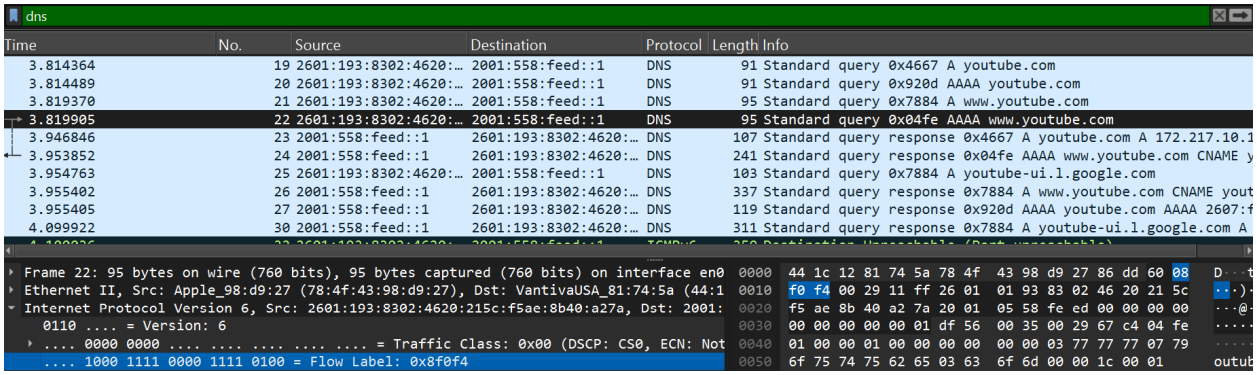
Answer the following questions:

1. **What is the IPv6 source and destination address of the computer making the DNS AAAA request for the above-requested web browser?**



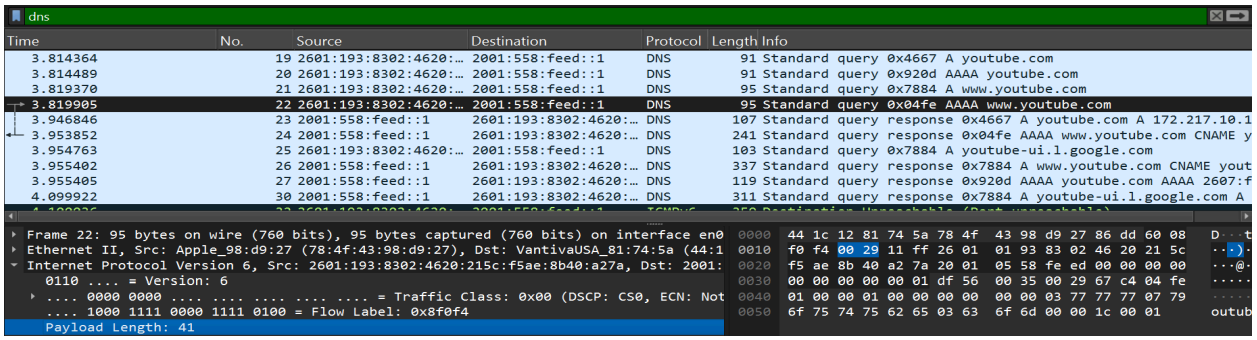
- **Source Address:** 2601:193:8302:4620:215c:f5ae:8b40:a27a
- **Destination Address:** 2001:558:feed::1

2. What are the values of the flow label for these IPv6 datagrams?



- The **Flow Label** is a **20-bit field** in the IPv6 header, used for Quality of Service (QoS) and traffic flow identification.
- The value is typically displayed in **hexadecimal** (e.g., `0x00000` or `0x8f0f4`).
- If no special QoS or flow classification is applied, the value is often `0x00000`.

3. How much payload data are carried for these IPv6 datagrams? What does this signify?



- The payload data carried in an IPv6 datagram is determined by the **Payload Length** field in the IPv6 header, which specifies the size of the payload in bytes.
- The Payload Length for these IPv6 datagrams is 41 bytes, as seen in the IPv6 header.
- The **payload** includes the **DNS AAAA request** sent to resolve `www.youtube.com` into an IPv6 address.

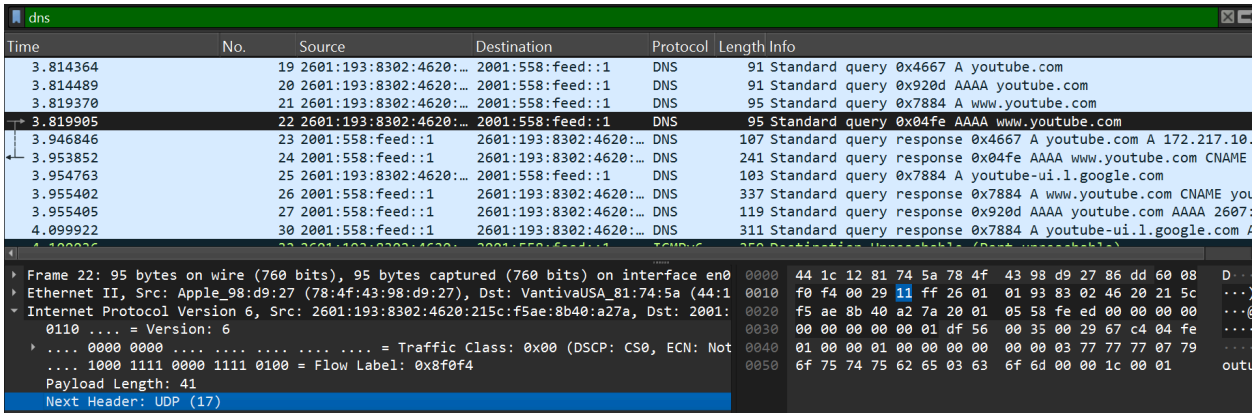
→ **Payload Length = Total Length-Ethernet Header length – IPv6 Header Size**

- Total Length=95
- Ethernet Header Length=14 bytes
- IPv6 Header Size = 40 bytes

Payload Length = 95-14-40=41

Significance: The payload represents the actual data being transmitted. For DNS requests, the payload would typically include the DNS query data (such as the domain name "youtube.com" in the case of a DNS AAAA request).

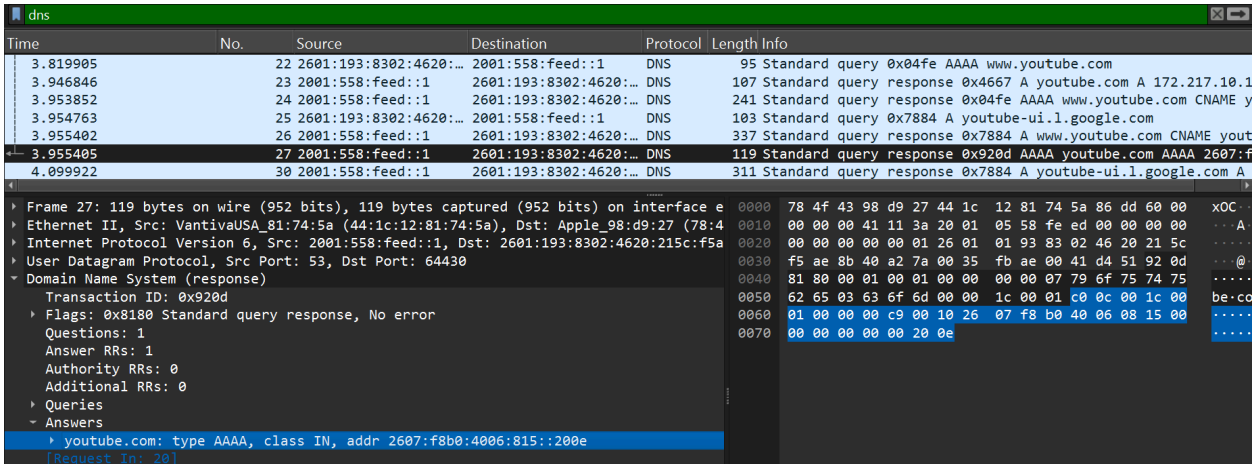
4. **What is the upper layer protocol to which this datagram's payload will be delivered at the destination?**



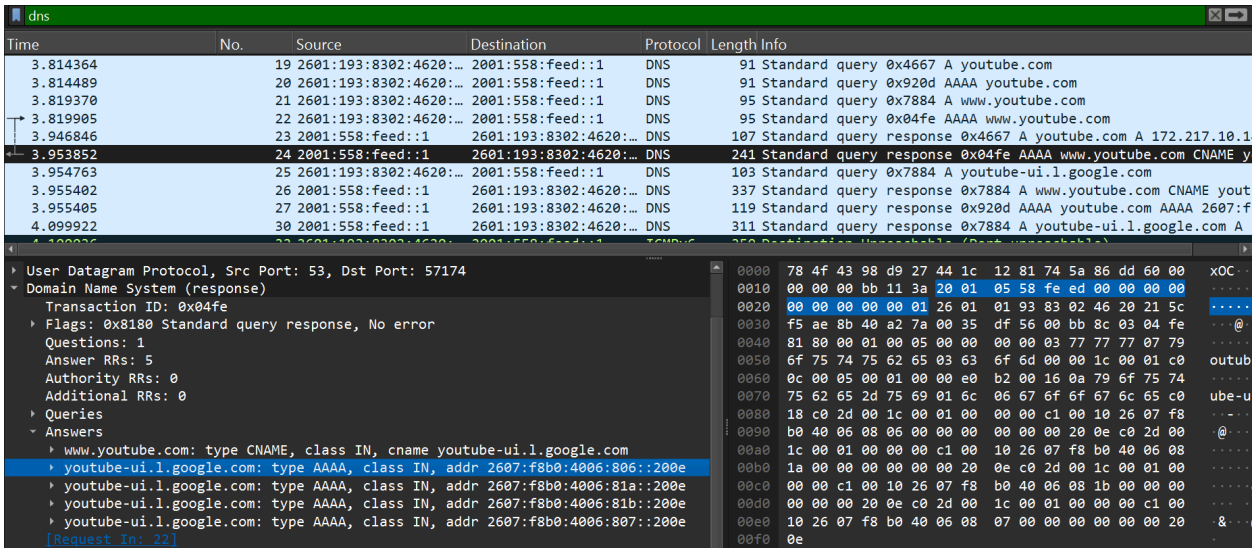
- For DNS, the Next Header will be UDP (17), as DNS typically runs over UDP.
- For DNS requests and responses, the **upper layer protocol** is typically **UDP** (as DNS commonly uses UDP for queries).

Lastly, find the IPv6 DNS response to the IPv6 DNS AAAA requests made in this trace. This DNS response contains IPv6 addresses for youtube.com.

5. **How many IPv6 addresses are returned in response to the AAAA requests?**



For youtube.com, 1 IPv6 address is returned in response to an AAAA request.



For **www.youtube.com**, **4** IPv6 addresses are returned in response to an AAAA request. This is because YouTube uses multiple servers to distribute traffic efficiently, ensuring reliability and reducing load on individual servers.