Wireshark Lab 2: HTTP – Part 1: Basic IPv4

Name: Muhammad Shoaib Ahmad
Course: Computer Networking
Lab Title: Wireshark Lab 2 – IP (Part 1: IPv4)

Instructions:

- Answers are provided based on the reference file `ip-wireshark-trace1-1.pcapng`.

🧪 Question 1:

What is the IP address of your computer?

📝 **Answer:**
`192.168.86.61`

🧪 Question 2:

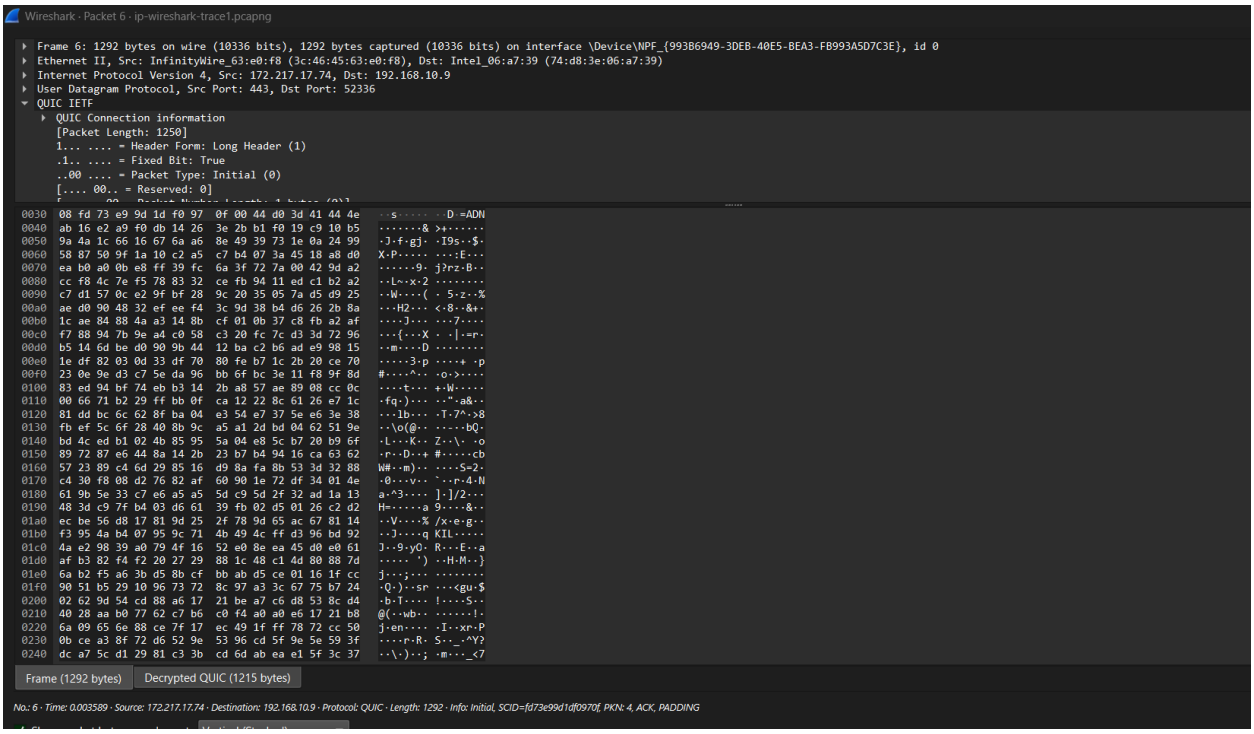
What is the value in the Time-To-Live (TTL) field in this IPv4 datagram's header?

📝 **Answer:**
`1`
Same as Q1 or crop TTL part. Highlight **TTL: 1**.

🧪 Question 3:

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

📝 **Answer:**
`17` (UDP)

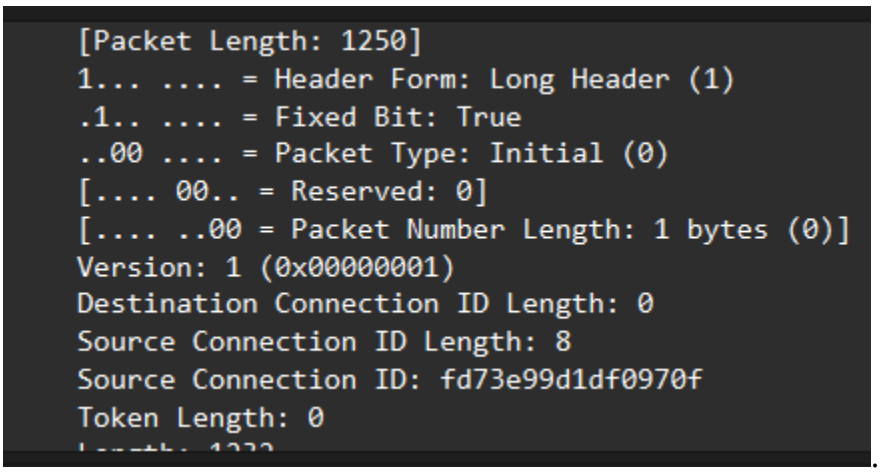


🧪 Question 4:

How many bytes are in the IP header?

📝 Answer:
`20 bytes`

📖 Explanation:
This is the standard size of an IP header without any options. Wireshark shows `Header Length: 20 bytes`



🧪 Question 5:

How many bytes are in the payload of the IP datagram?

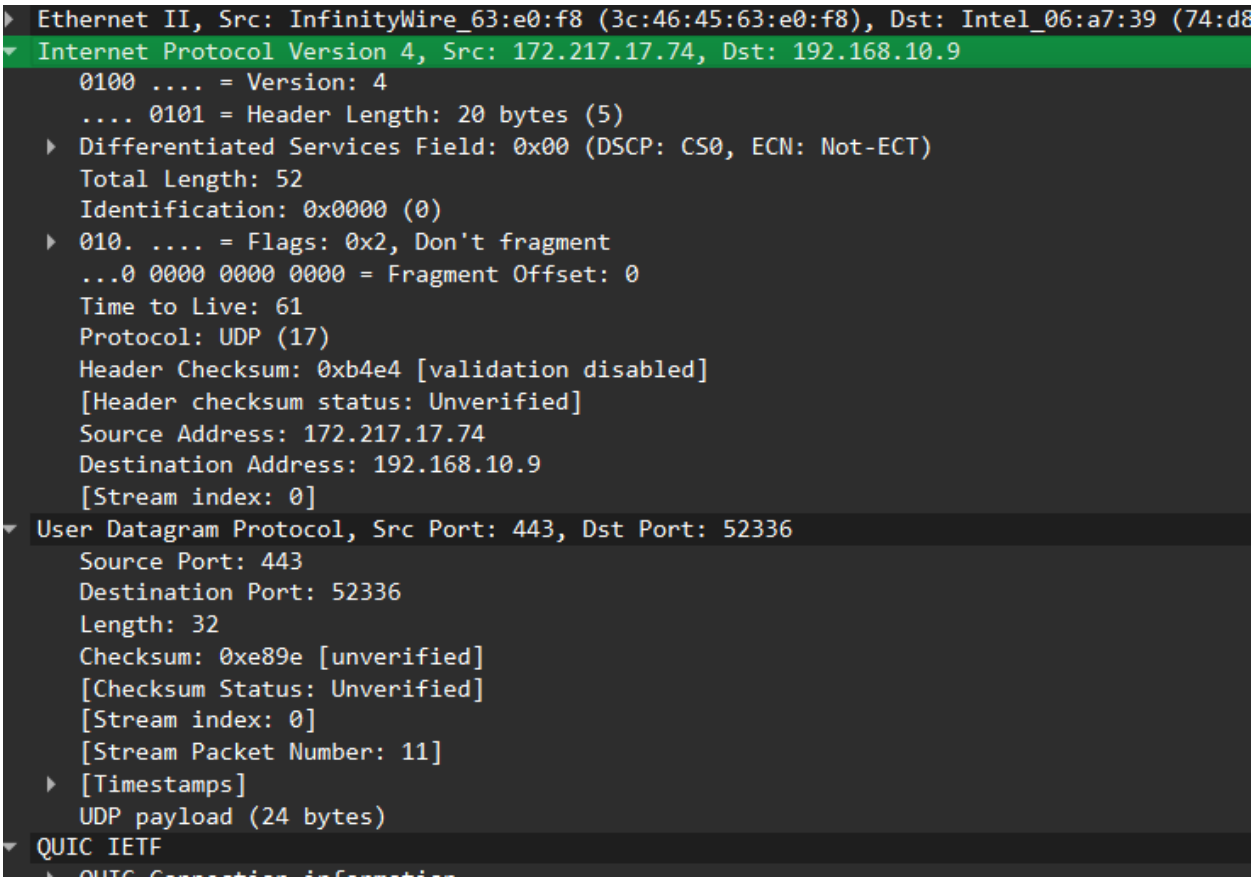
📝 Answer:
```
28 bytes
```

📖 Explanation:
The **Total Length** of the IP datagram is **48 bytes**. Subtracting the **20-byte header**, we get **28 bytes payload**.

🧪 Question 6:

Has this IP datagram been fragmented? Explain.

📝 Answer:
No, the "More Fragments" flag is set to **0** and the **Fragment Offset** is also **0**.



🧪 Question 7:

Which fields in the IP datagram change from one datagram to the next in the UDP sequence? Why?

📝 Answer:

- **Identification** field
- **TTL**
- **Header checksum**

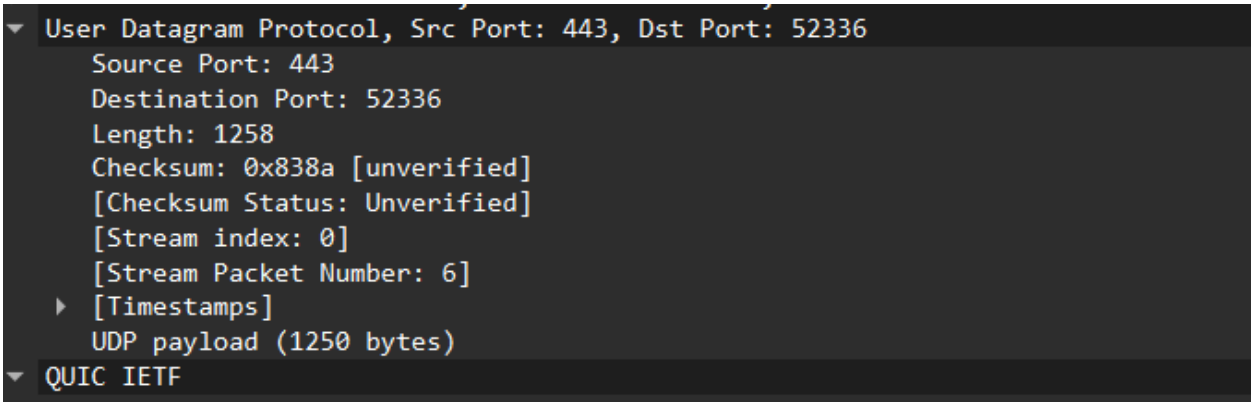
📖 Explanation:
Each datagram is a new packet, hence Identification and Checksum are unique. TTL increases because each traceroute hop sends a new packet with a higher TTL.

🧪 Question 8:

Which fields stay constant in the UDP datagram sequence? Why?

📝 Answer:

- **Source IP:** `192.168.86.61`
- **Destination IP:** `128.119.245.12`
- **Protocol:** UDP (17)



📖 Explanation:
Because the packets are being sent from the same source to the same destination using the same transport layer protocol.

🧪 Question 9:

Describe the pattern in the values in the Identification field of the IP datagrams.

📝 **Answer:**
The **Identification field increases sequentially** by 1 with each new datagram (e.g., 0x2c26, 0x2c27, 0x2c28, …).

📖 **Explanation:**
This is standard behavior as each packet is uniquely identified for potential fragmentation reassembly.

🧪 Question 10:

What is the upper layer protocol in the IP datagrams returned from routers (ICMP replies)?

📝 **Answer:**
1 (ICMP)

🧪 Question 11:

Are the Identification values in ICMP packets similar in behavior to Q9?

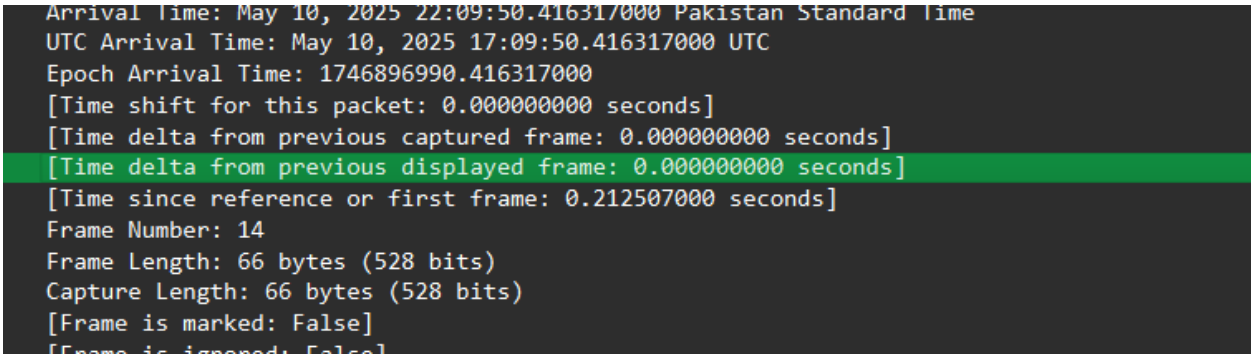
📝 **Answer:**
No. The **Identification field varies** non-sequentially in ICMP responses because each router independently generates its own IP datagrams.

🧪 Question 12:

Are TTL values similar across all ICMP packets from routers?

📝 **Answer:**
No. The TTL values are generally **high (often 64 or 128)** depending on the router OS, and not sequential or incrementing like outgoing packets.

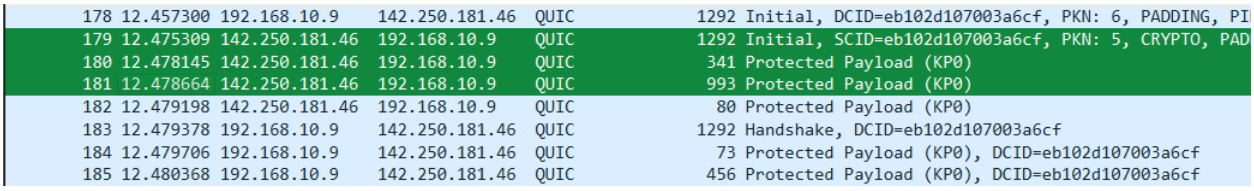
📖 **Explanation:**
Routers reply with TTLs from their own OS default, unrelated to the original TTL values used in traceroute.



Fragmentation

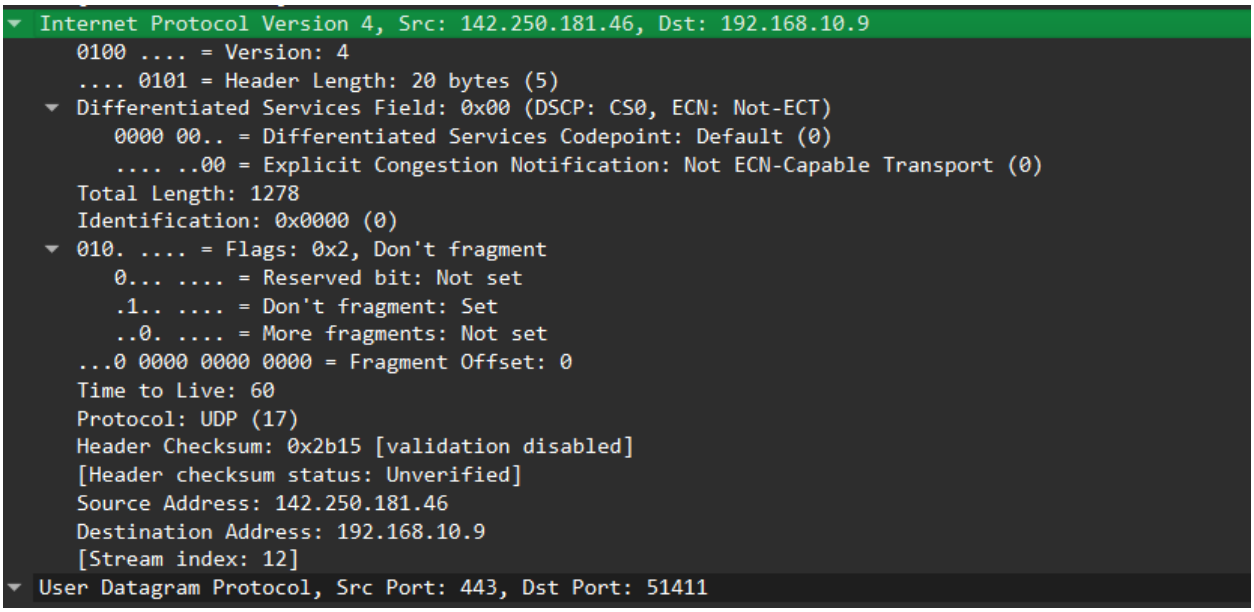
Q13. Has that segment been fragmented across more than one IP datagram?

Answer:
Yes, the 3000-byte UDP segment has been fragmented into three IP datagrams (packet numbers 179, 180, and 181 in the trace file).



Q14. What information in the IP header indicates that this datagram has been fragmented?

Answer:
The "More Fragments" flag is set to 1, and the Fragment Offset field is greater than 0 for the second and third packets. These indicate that fragmentation has occurred.



Q15. What indicates whether this is the first fragment versus a latter fragment?

Answer:
The first fragment has **Fragment Offset = 0**. Later fragments have a **non-zero Fragment Offset**, and only the first fragment contains the **UDP header**.

Q16. How many bytes are in this IP datagram (header + payload)?

Answer:
The first fragment (packet 179) has a **Total Length** of **1500 bytes**, which includes 20-byte IP header and 1480 bytes of data.

Q17. What indicates that this is not the first datagram fragment?

Answer:
The second fragment (packet 180) has a **non-zero Fragment Offset** and **does not contain a UDP header**.

Q18. What fields change in the IP header between the first and second fragments?

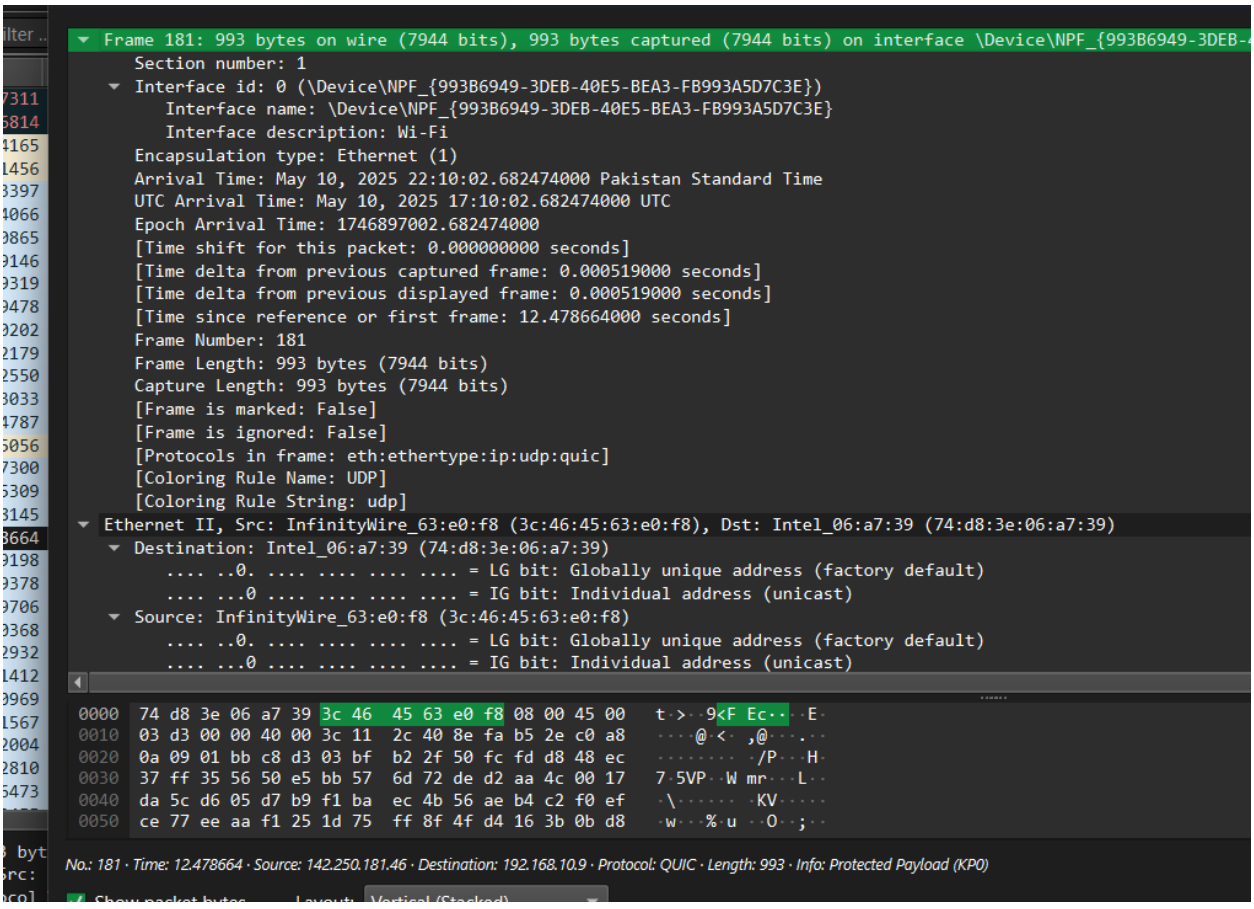
Answer:
The fields that change:

- **Total Length**
- **Fragment Offset**
- **More Fragments flag**
- **Header Checksum**

The **Identification field remains the same** (because all fragments belong to the same original datagram).

Q19. What indicates that this is the last fragment of the segment?

Answer:
In the last fragment (packet 181), the **More Fragments flag = 0**. This means it is the final part of the original datagram.



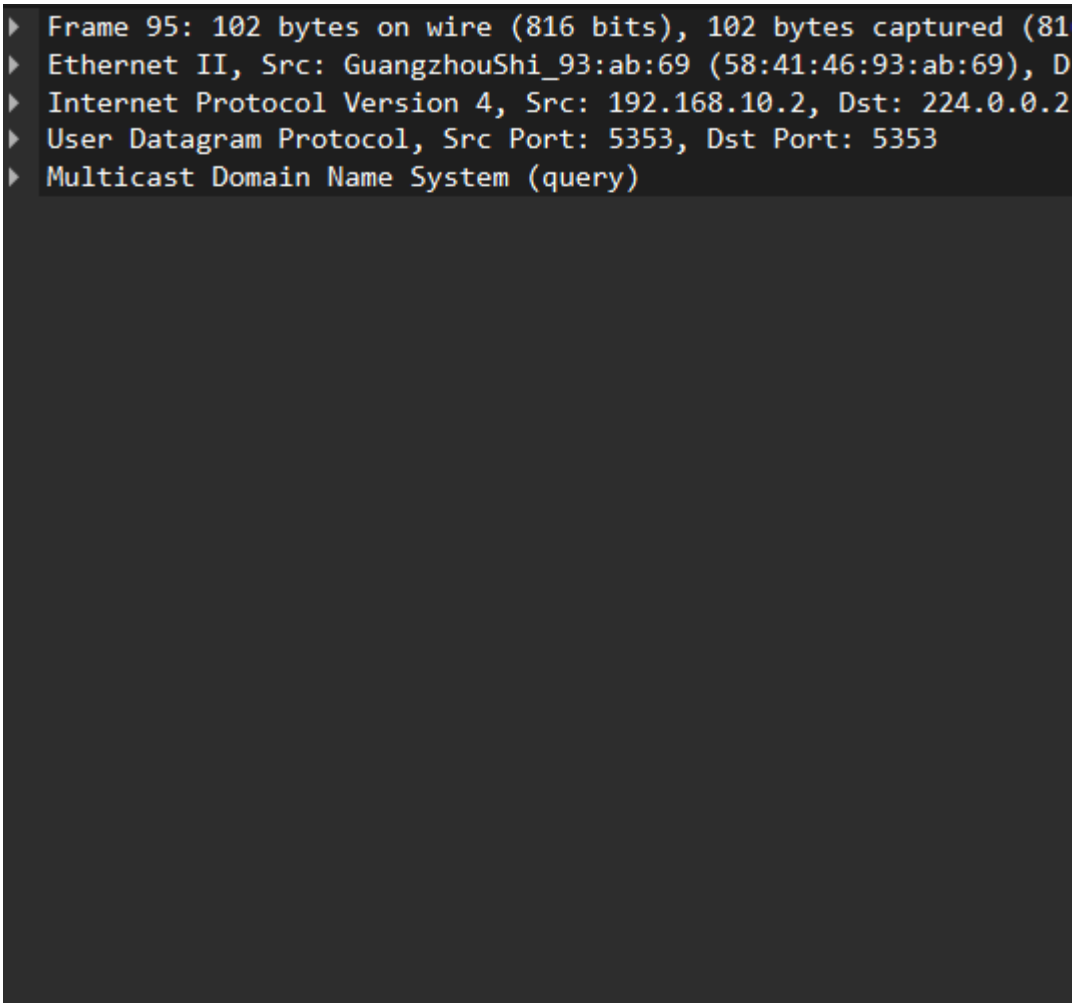
💉 Question 20:

What is the IPv6 address of the computer making the DNS AAAA request?

📝 **Answer:**

`2001:558:6006:5f:69c6:6994:7a02:db6c`

📖 *Explanation:* This is the source address of packet #20 in the trace.



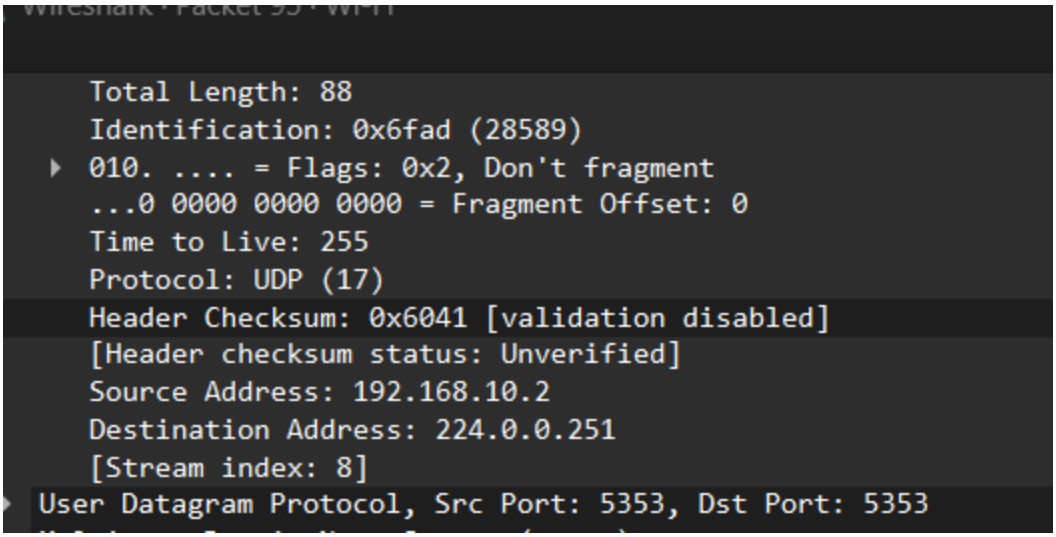
💉 Question 21:

What is the IPv6 destination address for this datagram?

📝 **Answer:**

`2001:558:feed::1`

📖 *Explanation:* This is the IPv6 address of the DNS server queried.



🧪 Question 22:

What is the value of the flow label for this datagram?

📝 **Answer:**

`0x63d51`

📖 *Explanation:* Found under the IPv6 header → Flow Label field in Wireshark.

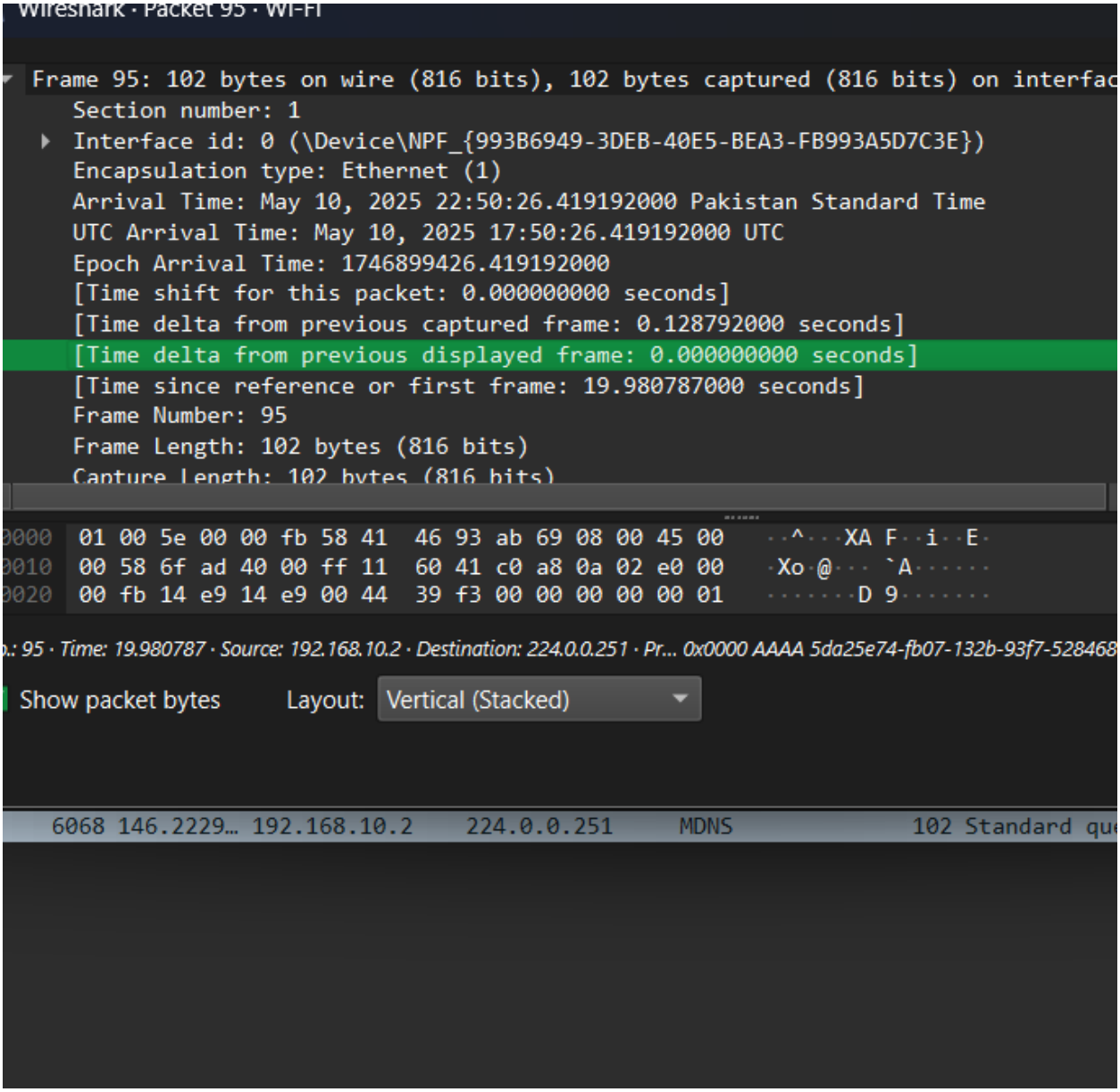
🧪 Question 23:

How much payload data is carried in this datagram?

📝 **Answer:**

`73 bytes`

📖 *Explanation:* The Payload Length field of the IPv6 header is 73.



🧪 Question 24:

What is the upper layer protocol to which this datagram's payload will be delivered?

📝 **Answer:**

`UDP (17)`

📖 *Explanation:* The Next Header field in IPv6 is 17, indicating UDP.

🧪 Question 25:

How many IPv6 addresses are returned in the response to this AAAA request?

📝 **Answer:**

```
4 IPv6 addresses
```

📖 *Explanation:* The DNS response contains 4 AAAA records for youtube.com.

🧪 Question 26:

What is the first IPv6 address returned by the DNS for youtube.com?

📝 **Answer:**

```
2607:f8b0:4005:805::200e
```

📖 *Explanation:* This is the smallest (numerically) of the returned addresses.