CS315: Lab Assignment 7

B Siddharth Prabhu
200010003@iitdh.ac.in

14 February 2023

0 Introduction

First, let's send two traceroute datagrams of the given lengths, of 56 and 3000 bytes respectively. Below are screenshots of the packet traces obtained for the same:

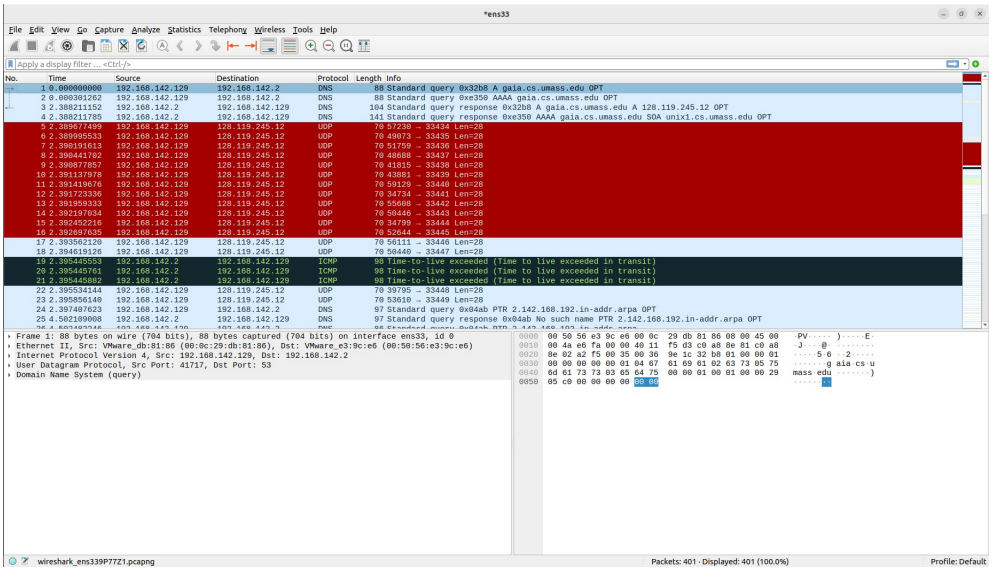

Figure 1: Packet trace for first traceroute

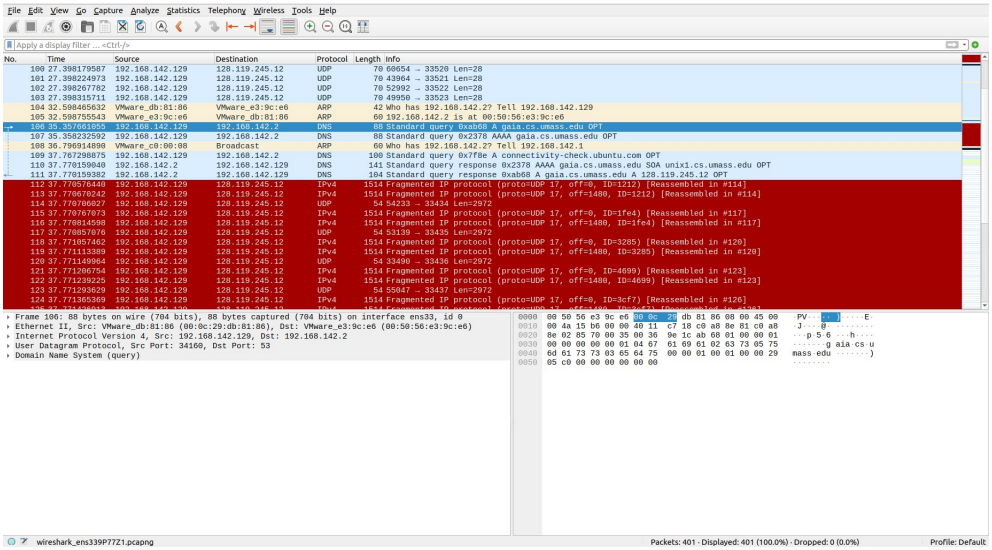

Figure 2: Packet trace for second traceroute (from packet no. 106)

1 Answers for Part 1: Basic IPv4

(1) Select the first UDP segment sent by your computer via the traceroute command to gaia.cs.umass.edu. Expand the Internet Protocol part of the packet in the packet details window. What is the IP address of your computer?

As visible in Figure (1), the IP address of the system is **192.168.142.129**.

(**Note:** The reason for the IP address above is that I'm running the browser (and Wireshark) on an Ubuntu VM with Network settings set to NAT (Network Address Translation). IP addresses beginning with 192.168 are for private networks, and in this case the private network is formed by the VM and the host system, via a network adapter. Hence, to communicate with the internet, this would take 1 hop more, than if this was run on the host.)

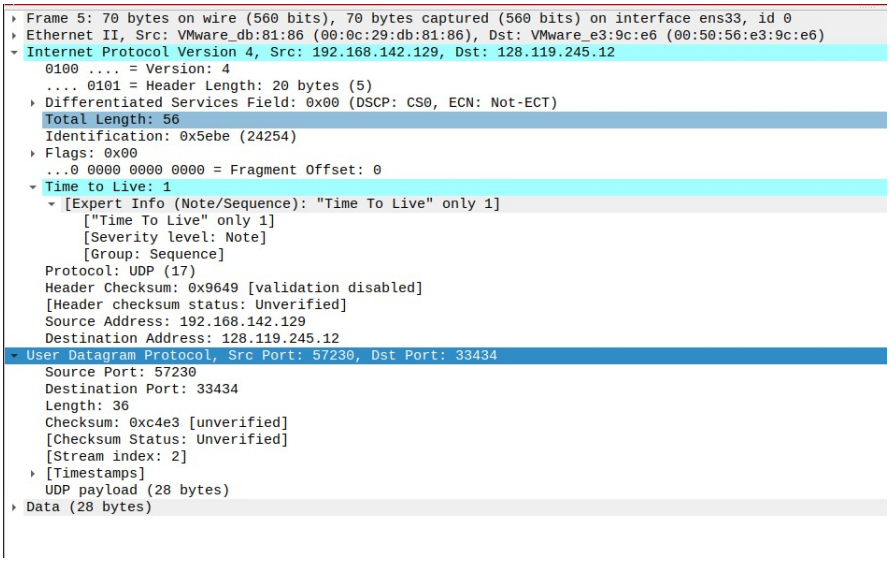

Figure 3: Packet details of first UDP segment

(2) 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 `1` .

(3) 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**. This indicates that IPv4 is being used as a service by the transport layer's User Datagram Protocol.

(4) How many bytes are in the IP header?

As visible in Figure (3), the IP header has **20 Bytes** (as per Header Length field).

(5) How many bytes are in the payload of the IP datagram? Explain how you determined the number of payload bytes.

As visible in Figure (3), the IP datagram payload has $56 - 20 =$ **36 Bytes**.
(This is obtained by subtracting header length from total length; this is also verifiable as it is the 'Length' field in the UDP details.)

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

As visible in Figure (3), all of the flag bits are zero. Hence, the 'more fragments' field is zero (Further verifiable on expanding the flags, in Figure (4)). Thus, this IP datagram **has not been fragmented**.

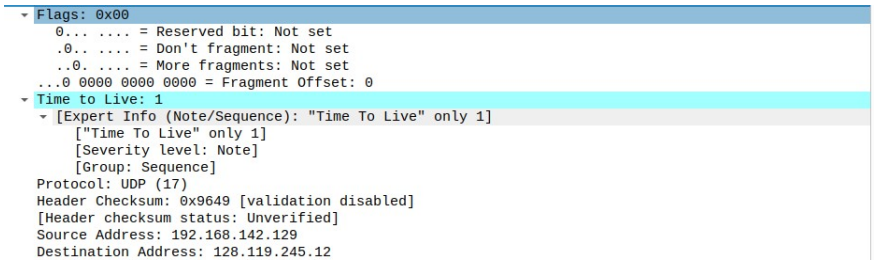

Figure 4: More Packet details

Next, we shall filter out the packet records using the filter command `ip.src==192.168.142.129 and ip.dst== 128.119.245.12 and udp and !icmp` . Below is a screenshot of the same, after which we will answer some questions about it:

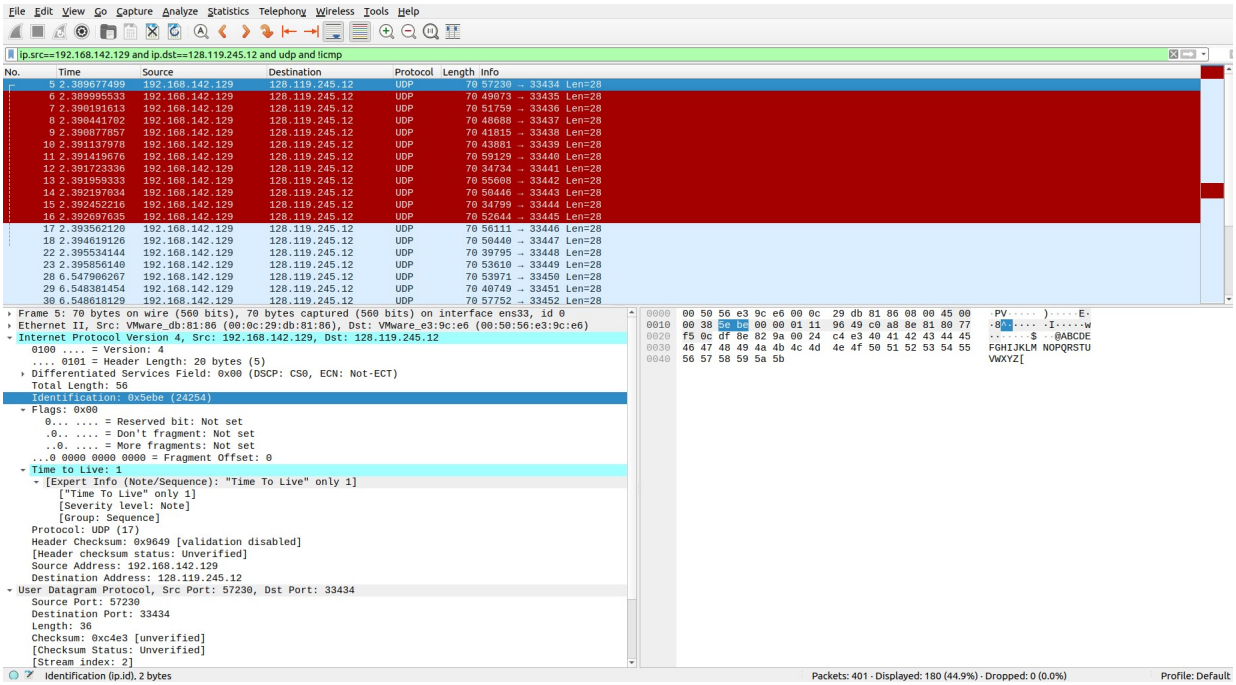

Figure 5: Filtered records

(7) Which fields in the IP datagram always change from one datagram to the next within this series of UDP segments sent by your computer destined to 128.119.245.12, via traceroute? Why?

We observe the following changes from one datagram to the next:

- `Identification` : This is a 16-bit value that is unique for every datagram for a given source IP address, destination IP address, and protocol, such that it does not repeat within the maximum datagram lifetime.
- `Header Checksum` : This changes if even one field changes in the IP datagram. Hence, due to the other changes listed here, this field also changes.
- `Time To Live (TTL)` : (This is changing once every 3 records.) The traceroute command starts by sending a UDP datagram to the destination host, with the TTL field set to 1. If a router along the way detects a TTL value of 1 or 0, it drops the datagram and sends back a "time exceeded" message to the sender. Hence, traceroute determines the address of the first hop by examining the source address field of the ICMP time-exceeded message. To iteratively find the routers along the way, the TTL field is increase by the source at regular intervals.

(8) Which fields in this sequence of IP datagrams (containing UDP segments) stay constant? Why?

The following fields stay constant in the sequence of IP datagrams (containing UDP segments):

- `Version` (Value = 4) : This stays same since IPv4 is used for all the packets in the trace.
- `Header Length` (Value = 20) : This stays same since the format of header is fixed; all the IP datagrams contain UDP segments.
- `Differentiated Services Field` (Value = 0x00) : All of the packets use the same service class, so this field stays same.
- `Protocol field` (Value = UDP) : This is same for all of the packets since they all contain UDP segments.
- `Source IP Address` (Value = 192.168.142.129) : This stays same since all of these packets are sent from the same source.
- `Destination IP Address` (Value = 128.119.245.12) : This stays same since all of these packets are sent to the same destination.

(9) Describe the pattern you see in the values in the Identification field of the IP datagrams being sent by your computer.

The Identification field of the IP datagrams seems to have a fairly irregular pattern. Note that, for a series of ICMP echo requests, this field increments by 1 each time. However, in the current case, traceroute (Linux) uses UDP segments instead of ICMP requests. Hence, this may be the reason the pattern appears irregular in the packet trace observed.

Now, let's take a look at the ICMP packets being returned to the system by the intervening routers where the TTL value was decremented to zero (and hence caused the ICMP error message to be returned to the host computer). The display filter that we will use here is `ip.dst==192.168.142.129 and icmp` . A screenshot of the same is attached, after which we shall answer some questions related to the same:

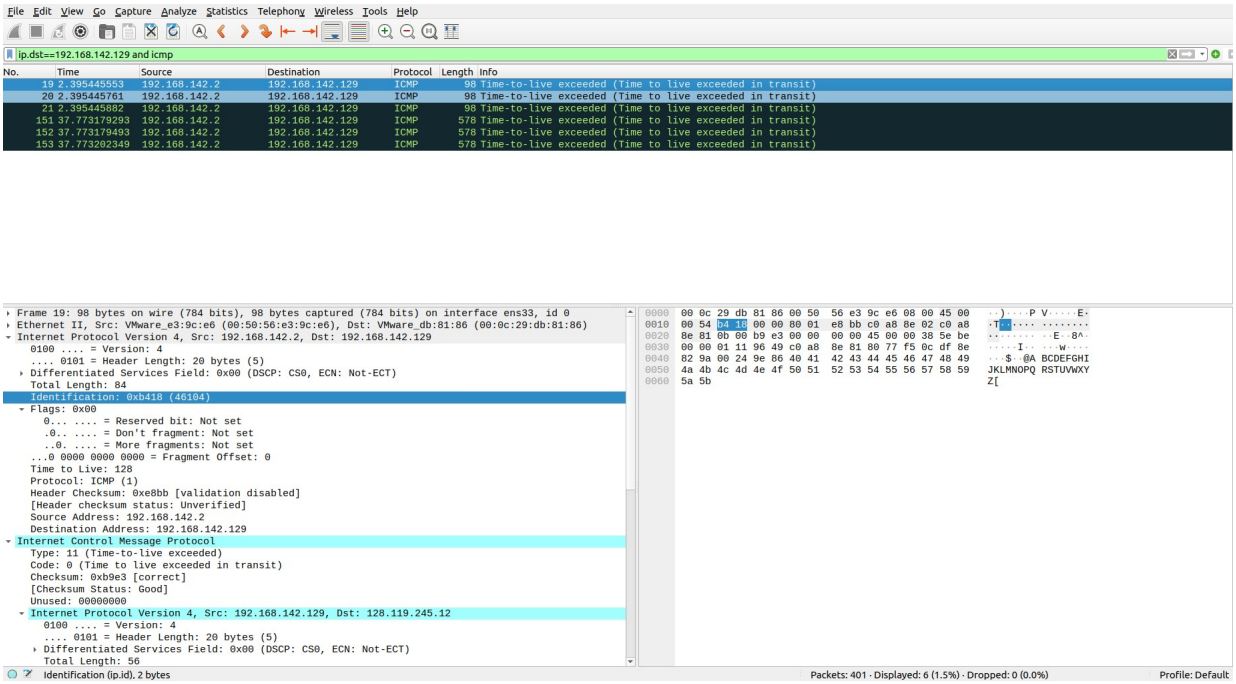

Figure 6: Filtered records, ICMP responses

(10) What is the upper layer protocol specified in the IP datagrams returned from the routers?

As visible in Figure (6), the upper layer protocol specified in the IP datagrams returned from the routers is **ICMP**. This is Internet Control Message Protocol.

(11) Are the values in the Identification fields (across the sequence of all of ICMP packets from all of the routers) similar in behavior to your answer to question 9 above?

In the answer to question 9, we were **NOT** dealing with ICMP packets; the traceroute command used UDP segments. Here, the ICMP packets from the routers have sequential identification fields, such that it increases by 1 from one packet to the next.

(12) Are the values of the TTL fields similar, across all of ICMP packets from all of the routers?

Across all of the ICMP packets, we observe that the values of the TTL fields is always `128` .

2 Answers to Part 2: Fragmentation

In this part, we will analyze the packet trace after the traceroute packet length was specified to be 3000.

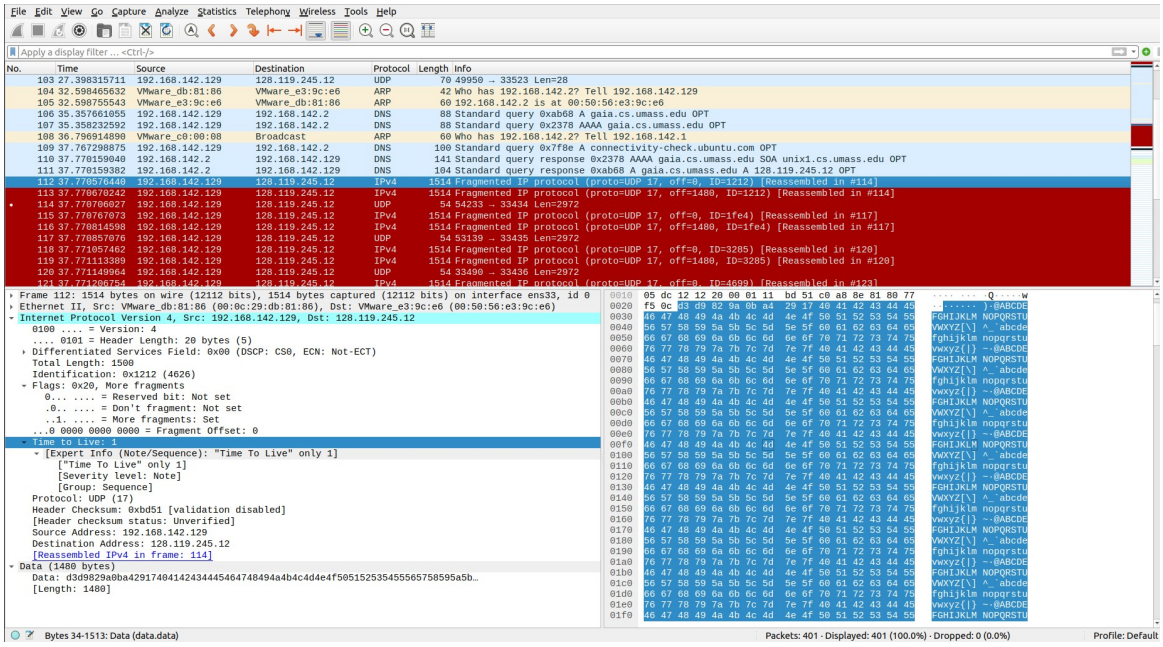

Figure 7: Packet Trace

(1) Find the first IP datagram containing the first part of the segment sent to 128.119.245.12 sent by your computer via the traceroute command to gaia.cs.umass.edu, after you specified that the traceroute packet length should be 3000. Has that segment been fragmented across more than one IP datagram?

It is visible from Figure (7), that the segment has been fragmented across more than one IP datagram. Hence, the answer is **YES**.

(2) What information in the IP header indicates that this datagram has been fragmented?

It is visible from Figure (7), that the segment has been fragmented across more than one IP datagram. This is observable from the **more fragments** flag in the packet details.

(3) What information in the IP header for this packet indicates whether this is the first fragment versus a latter fragment?

We observe in Figure (7), that fragment offset field has a value of **ZERO**. Thus indicates that this is the first fragment.

(4) How many bytes are there in this IP datagram (header plus payload)?

We observe in Figure (7), according to the 'Total Length' field of the IP datagram header, the size of the IP datagram is **1500 Bytes** (header plus payload).

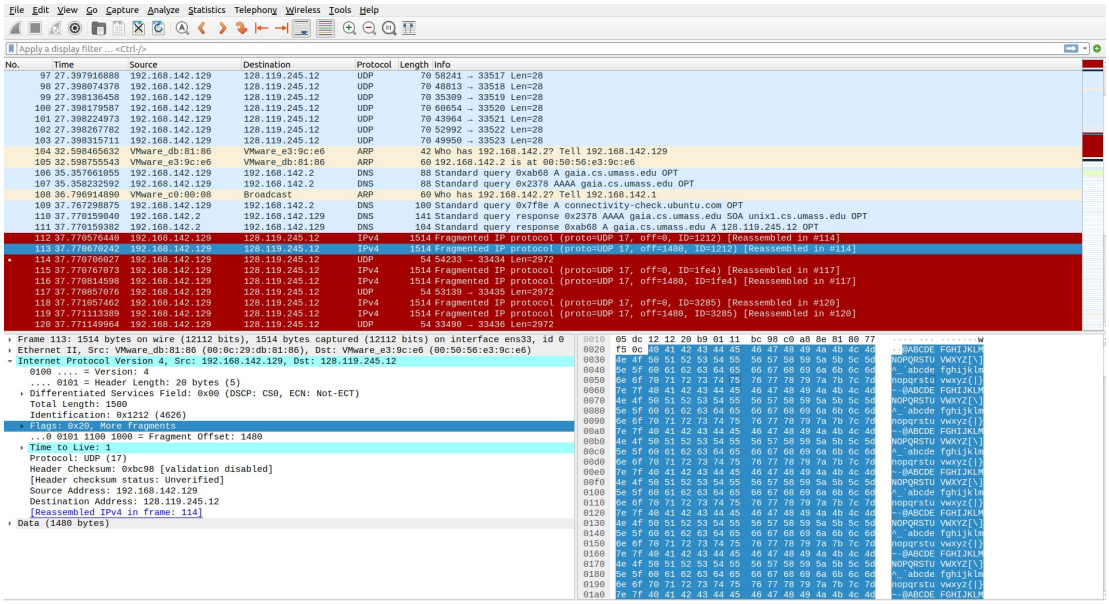

Figure 8: Details of Second Fragment

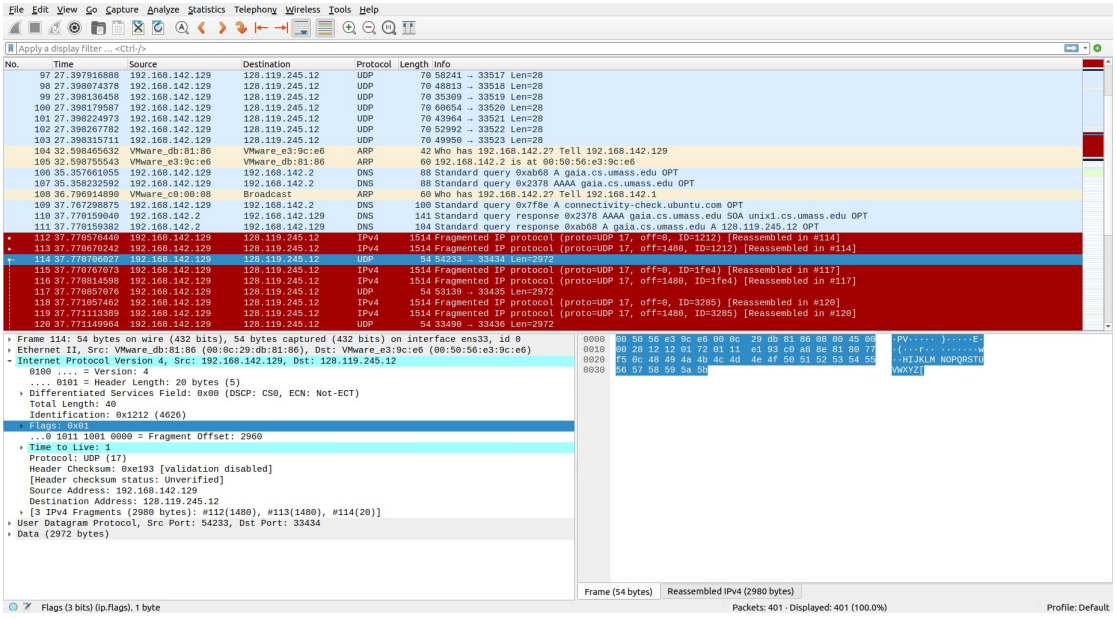

Figure 9: Details of Third Fragment

(5) What fields change in the IP header between the first and second fragment?

The following fields change in the IP header between the first and second fragment:

- Total length
- Flags
- Fragment offset
- Checksum

This can be verified based on Figure (7) and Figure (8).

(6) Now find the IP datagram containing the third fragment of the original UDP segment. What information in the IP header indicates that this is the last fragment of that segment?

From Figure (9), we observe that the 'more fragments' flag is set to zero for the third fragment of the original UDP segment. Also, there is a line which says '3 IPv4 Fragments', followed by their lengths and frame numbers. Hence, this is the last fragment of that segment.

3 Answers for Part 3: IPv6

In this part, we analyze the packet trace file `ip-wireshark-trace2-1.pcapng` . Below is a screenshot of the same, after which we shall answer some questions:

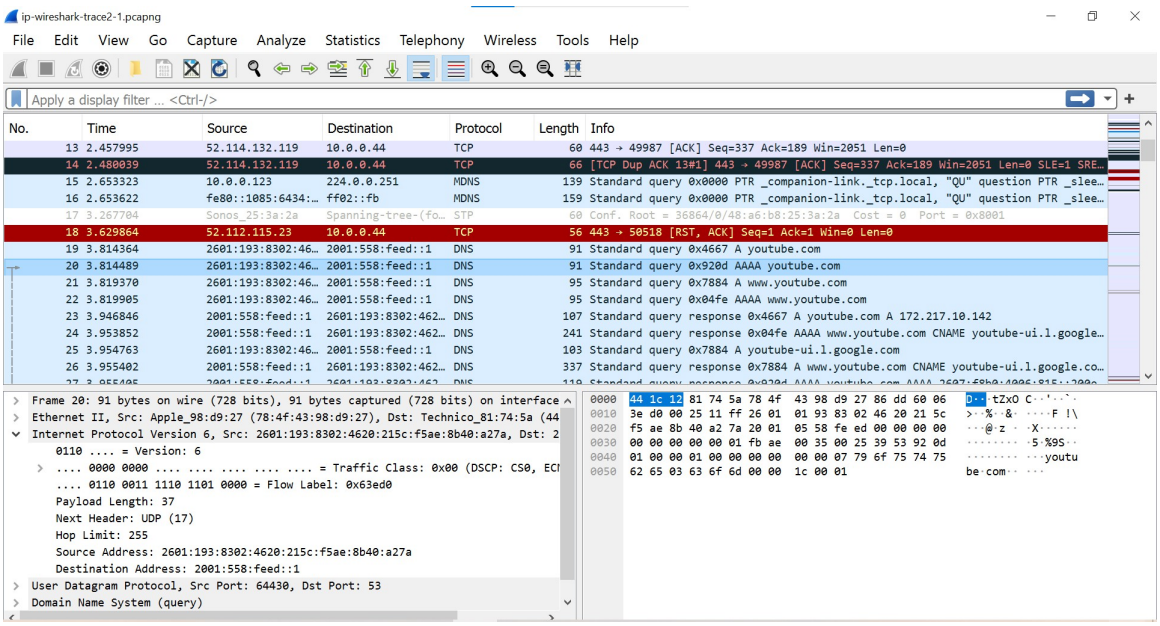

Figure 10: IPv6 Packet Trace

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

IPv6 address of the computer making the DNS AAAA request is below:
`2601:193:8302:4620:215c:f5ae:8b40:a27a` .

(2) What is the IPv6 destination address for this datagram? Give this IPv6 address in the exact same form as displayed in the Wireshark window.

The IPv6 destination address for this datagram, as displayed in the Wireshark window, is `2001:558:feed::1` .

(3) What is the value of the flow label for this datagram?

The value of the flow label is given as: `Flow Label: 0x63ed0` , as visible in Figure (10).

(4) How much payload data is carried in this datagram?

The payload data carried in this datagram is found to be **37 Bytes**.

(5) What is the upper layer protocol to which this datagram's payload will be delivered at the destination?

The IPv6 datagram's payload will be delivered firstly to the `UDP protocol` at the destination. (Note that UDP is a Transport Layer Protocol. After this, the payload of the UDP segment thereof is passed on to the DNS protocol at the application layer.)

Next, we shall find the IPv6 DNS response to the IPv6 DNS AAAA request made in the 20th packet in this trace. This DNS response contains IPv6 addresses for youtube.com. Observe that it is **Packet Number 27** in the trace that is the IPv6 DNS response that we are looking for. Two ways to verify that it indeed is the required response are as follows:

- We observe `[Request In: 20]` in the details under DNS in the packet details pane.
- The arrow marks to the left of the Packet numbers show what response corresponds to what request.

Below is a screenshot of the same, after which we shall answer some questions:

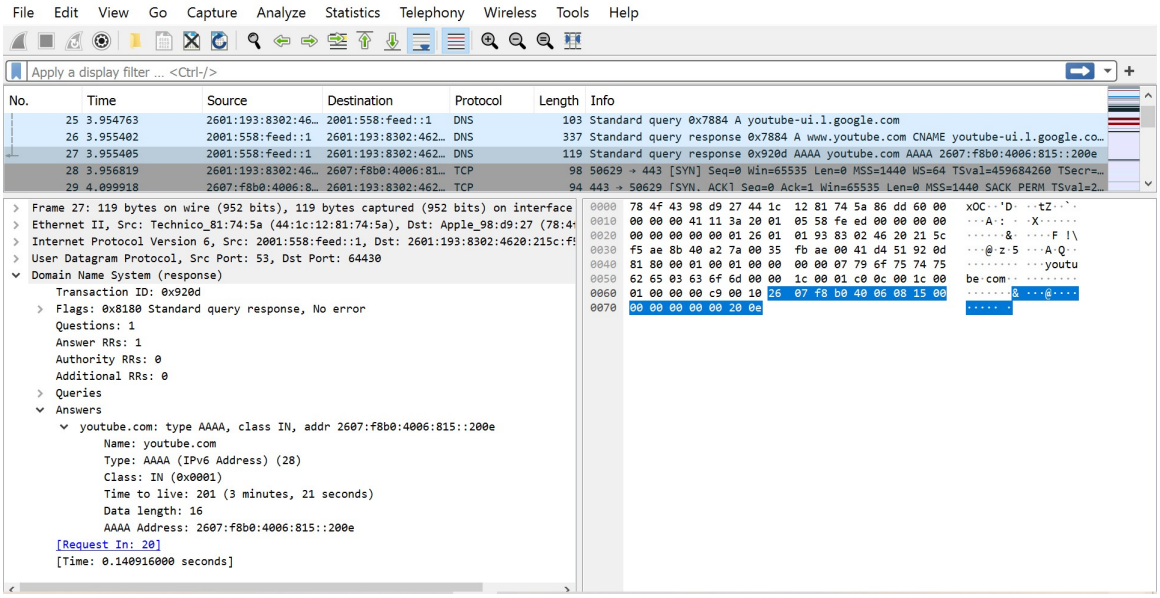

Figure 11: IPv6 DNS Response

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

It is visible in Figure (11) that there is only **ONE** IPv6 address returned in the response to the AAAA request.

(7) What is the first of the IPv6 addresses returned by the DNS for youtube.com? Give this IPv6 address in the exact same shorthand form as displayed in the Wireshark window.

The first of the (ONE) IPv6 address returned by the DNS for youtube.com is visible in Figure (11) as: `AAAA Address: 2607:f8b0:4006:815::200e`