

Computer Networks Lab, Assignment 7

Suyash Gaurav

210010054

1 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?

The IP address of my computer is: 10.200.95.40

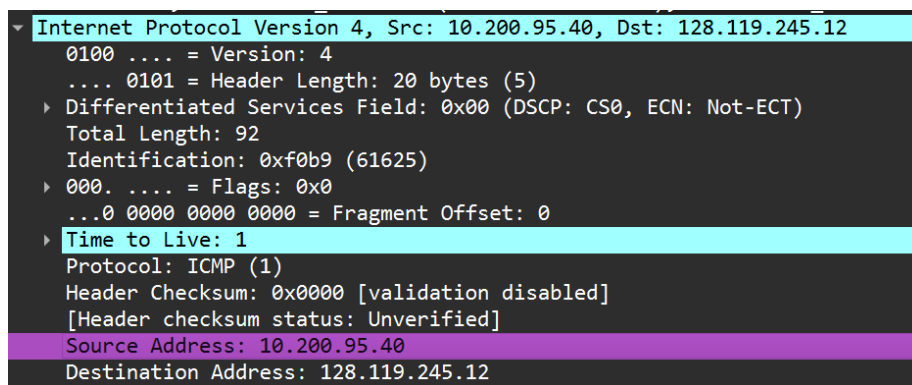


Figure 1: Q1: IPv4 of my computer

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

The time-to-live (TTL) is 1.

3. What is the value in the upper layer protocol field in this IPv4 datagram's header? [Note: the

```

> 000. .... = Flags: 0x0
...0 0000 0000 0000 = Fragment Offset: 0
> Time to Live: 1
Protocol: ICMP (1)
Header Checksum: 0x0000 [validation disabled]
[Header checksum status: Unverified]

```

Figure 2: Q2. TTL

answers for Linux/MacOS differ from Windows here].

The upper layer protocol field in this IPv4 datagram's header is UDP(17).

4. How many bytes are in the IP header?

There are 20 bytes in the IP header.

```

▼ Internet Protocol Version 4, Src: 10.200.95.40, Dst: 128.119.245.12
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ► Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

```

Figure 3: Q4: IP header

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

The no of bytes in the payload of the IP datagram is $TotalLength - HeaderLength = 92 - 20 = 72$

```

  .... 0101 = Header Length: 20 bytes (5)
  ► Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 92

```

Figure 4: Q5: Payload = Total length - header Length

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

The IP datagram is not fragmented as the Fragment Offset is 0.

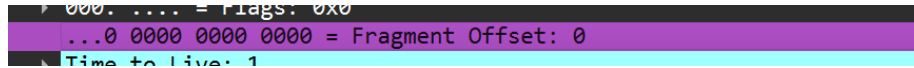


Figure 5: Q6: Fragment offset

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?

Fields in the IP datagram that always change from one datagram to the next are:

- **Identification:** This is like a serial number for each packet. It helps the receiving end put the packets back together in the right order. Since each packet is separate, they need different IDs.
- **Time To Live:** This is like a countdown timer for each packet. It starts with a certain number and decreases as the packet hops from one router to another.
- **Header Checksum:** It helps ensure that the packet hasn't been corrupted during transmission. Since the header changes with each new packet, the checksum also needs to change to match the new header.

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

Fields in the IP datagram (containing UDP segments) that stay constant:

- **Version:** This remains constant because all the packets are using IPv4.
- **Source IP:** This stays the same because the packets are all sent from the same source, likely your computer.
- **Differentiated Services:** Since all packets are ICMP, they would typically use the same type of service class, so this stays constant.
- **Header Length:** Since these are ICMP packets, the header length remains constant because ICMP headers have a fixed length.
- **Upper Layer Protocol:** Since these are ICMP packets, the upper layer protocol remains constant. ICMP is the protocol being used for these packets.

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 sent by your computer shows a pattern of increasing value with each ICMP Echo (ping) request.

10. What is the upper layer protocol specified in the IP datagrams returned from the routers? [Note: the answers for Linux/MacOS differ from Windows here].

The upper layer protocol field in this IPv4 datagram's header is: ICMP (1).

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?

No, the values in the Identification fields are not similar to Q9. I can see, it is coming in random order (in some cases, it increases by 1).

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

No values of TTL fields are not similar across all of ICMP packets from all of the routers.

2 Part 2: Fragmentation

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. (Hint: This is packet 179 in the ip-wireshark-trace1-1.pcapng trace file in footnote 2. Packets 179, 180, and 181 are three IP datagrams created by fragmenting the first single 3000-byte UDP segment sent to 128.119.145.12). Has that segment been fragmented across more than one IP datagram? (Hint: the answer is yes!)

Yes, The segment has been fragmented across more than one IP datagram. In the screenshot, the More frag-

ments flag is set for 1st IP Datagram.

```
IDENTIFICATION: 0x5100 (70040)
▼ 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
```

Figure 6: Q1: Fragmentation

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

When a datagram is fragmented, it's split into smaller packets because it's too large to fit through the network without being broken down. The "More Fragments" flag is set to indicate whether there are more fragments following the current one.

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

If the "Fragment Offset" is zero, it indicates that it is the first fragment because there is no offset from the beginning of the original datagram. If the "Fragment Offset" is greater than zero, it indicates that it is a later fragment.

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

There are 1500 bytes (Header Length + Payload) in this IP datagram.

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

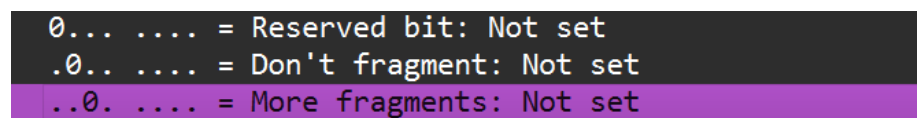
The fields that typically change between the first and

subsequent fragments are:

- **Fragment Offset:** This field indicates the position of the fragment in the original packet. For the first fragment, the offset is typically 0, as it contains the beginning portion of the original packet.
- **Header Checksum:** The checksum is a value used for error-checking purposes. It ensures the integrity of the IP header. Since the fragments have different positions within the original packet, the checksum needs to be recalculated for each fragment.

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?

The information in the IP header that indicates that the third fragment is the last fragment of the original UDP segment is the absence of the "More Fragments" flag being set.



```
0... .... = Reserved bit: Not set
.0.. .... = Don't fragment: Not set
..0. .... = More fragments: Not set
```

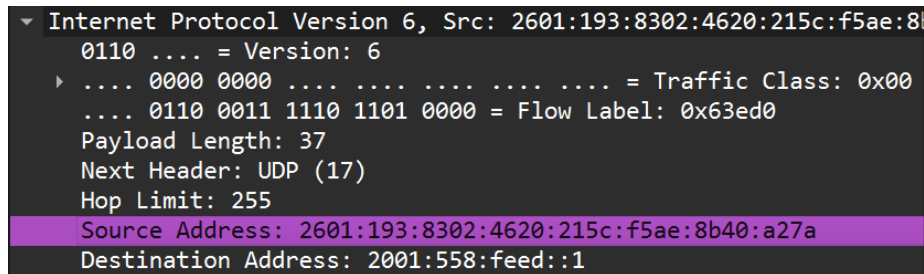
Figure 7: Q6: More Fragment: not set

3 Part 3: IPv6

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

source address of the 20th packet in the trace. Give the IPv6 source address for this datagram in the exact same form as displayed in the Wireshark window1?

Source Address: 2601:193:8302:4620:215c:f5ae:8b40:a27a

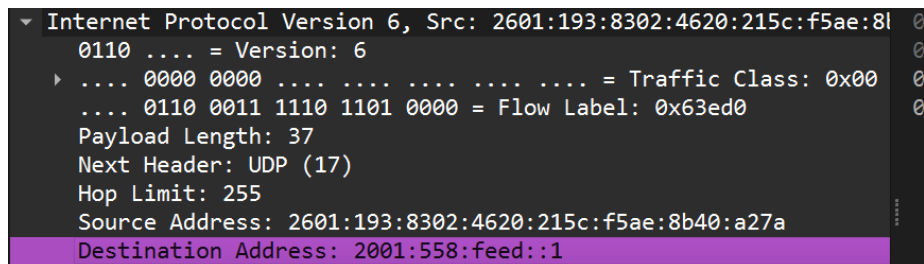
A screenshot of the Wireshark packet details pane for an Internet Protocol Version 6 packet. The packet is expanded, showing fields like Version, Traffic Class, Flow Label, Payload Length, Next Header, Hop Limit, Source Address, and Destination Address. The Source Address field is highlighted in purple.

```
▼ Internet Protocol Version 6, Src: 2601:193:8302:4620:215c:f5ae:8b40:a27a
  0110 .... = Version: 6
  ▶ .... 0000 0000 .... = Traffic Class: 0x00
    .... 0110 0011 1110 1101 0000 = Flow Label: 0x63ed0
    Payload Length: 37
    Next Header: UDP (17)
    Hop Limit: 255
    Source Address: 2601:193:8302:4620:215c:f5ae:8b40:a27a
    Destination Address: 2001:558:feed::1
```

Figure 8: Q1. Source IPv6

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.

Destination Address: 2001:558:feed::1

A screenshot of the Wireshark packet details pane for an Internet Protocol Version 6 packet, similar to Figure 8. The Destination Address field is highlighted in purple.

```
▼ Internet Protocol Version 6, Src: 2601:193:8302:4620:215c:f5ae:8b40:a27a
  0110 .... = Version: 6
  ▶ .... 0000 0000 .... = Traffic Class: 0x00
    .... 0110 0011 1110 1101 0000 = Flow Label: 0x63ed0
    Payload Length: 37
    Next Header: UDP (17)
    Hop Limit: 255
    Source Address: 2601:193:8302:4620:215c:f5ae:8b40:a27a
    Destination Address: 2001:558:feed::1
```

Figure 9: Q2: Destination IPv6

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

.... 0110 0011 1110 1101 0000 = Flow Label: 0x63ed0

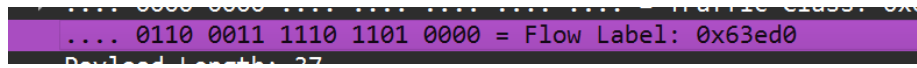


Figure 10: Q3: Flow label

4. How much payload data is carried in this datagram?

Payload Length: 37 bytes



Figure 11: Q4: Payload

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

The upper layer protocol to which the datagram's payload will be delivered is: UDP (17)

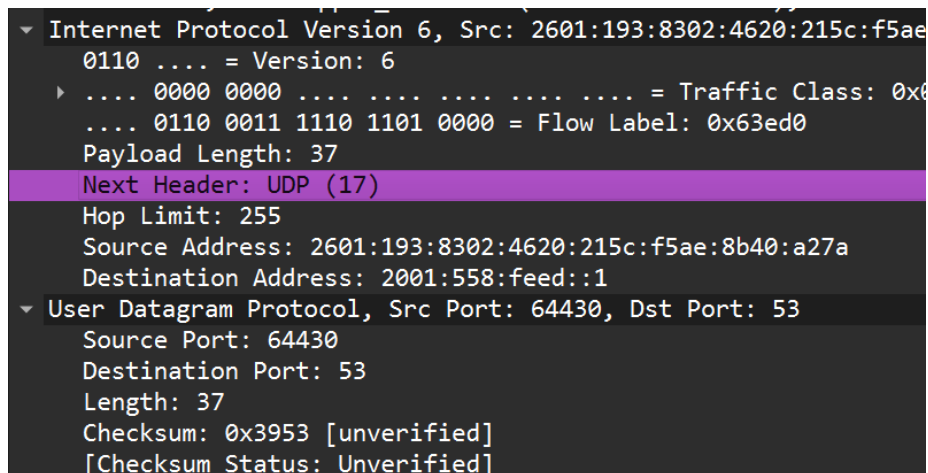
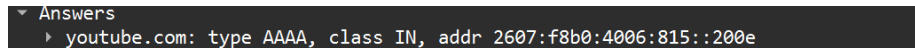


Figure 12: Q5: Upper layer protocol: UDP(17)

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

1 (Four) IPv6 addresses are returned in the response to this AAAA request. (Screenshot attached)



```
▼ Answers  
▶ youtube.com: type AAAA, class IN, addr 2607:f8b0:4006:815::200e
```

Figure 13: Q6: Response to this AAAA request

7. What is the first of the IPv6 addresses returned by the DNS for youtube.com (in the ip-wireshark-trace2-1.pcapng trace file, this is also the address that is numerically the smallest)? Give this IPv6 address in the exact same shorthand form as displayed in the Wireshark window.

The first of the IPv6 addresses returned by the DNS for youtube.com is: 2607:f8b0:4006:815::200e