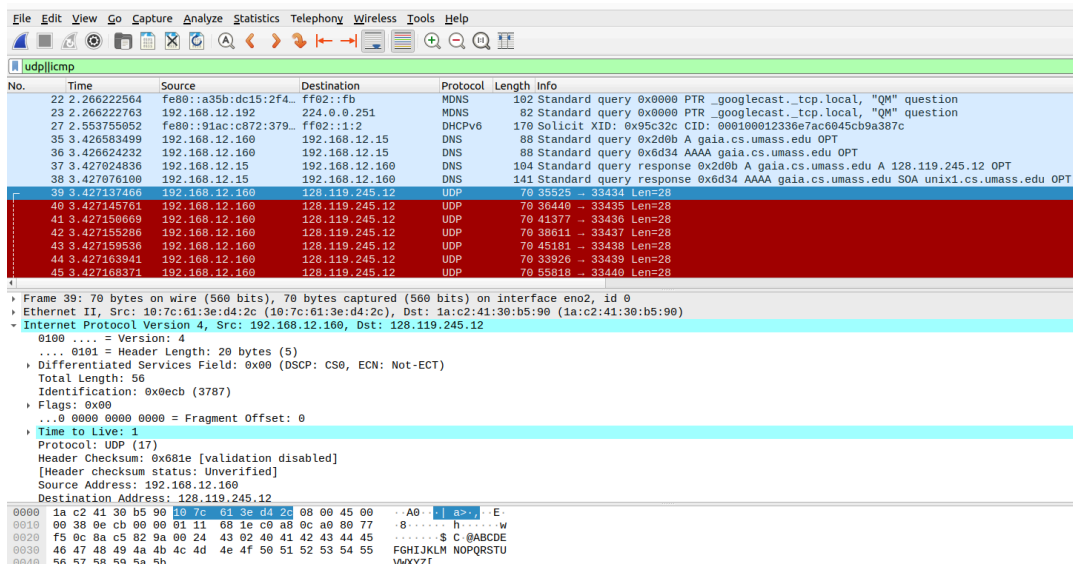


Sarwar, Andre, and Mohammed

Lab 6

November 1, 2024

1. Select the first UDP segment sent by your computer via the traceroute command to gaia.cs.umass.edu. (Hint: this is 44th packet in the trace file in the ipwireshark-trace1-1.pcapng file in footnote 2). Expand the Internet Protocol part of the packet in the packet details window. What is the IP address of your computer?



- The first UDP segment is packet number 39. The IP address of our computer is 192.168.12.160.
2. What is the value in the time-to-live (TTL) field in this IPv4 datagram's header?
 - The value is 1.
3. What is the value in the upper layer protocol field in this IPv4 datagram's header? [Note: the answers for Linux/MacOS differ from Windows here].
 - The upper layer protocol is UDP (17).
4. How many bytes are in the IP header?
 - The IP header is 20 bytes.
5. How many bytes are in the payload of the IP datagram? Explain how you determined the number of payload bytes.
 - The total length is 56 bytes. We know that the IP header is 20 bytes, which means there are 36 bytes remaining. This difference, 36, is the amount of Payload bytes.
6. Has this IP datagram been fragmented? Explain how you determined whether or not the datagram has been fragmented.
 - The IP datagram has not been fragmented. We determined this by examining the fragment offset. The fragment offset is 0.

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?

- The fields that always change are time to live and identification. These change because there are different datagrams being sent.

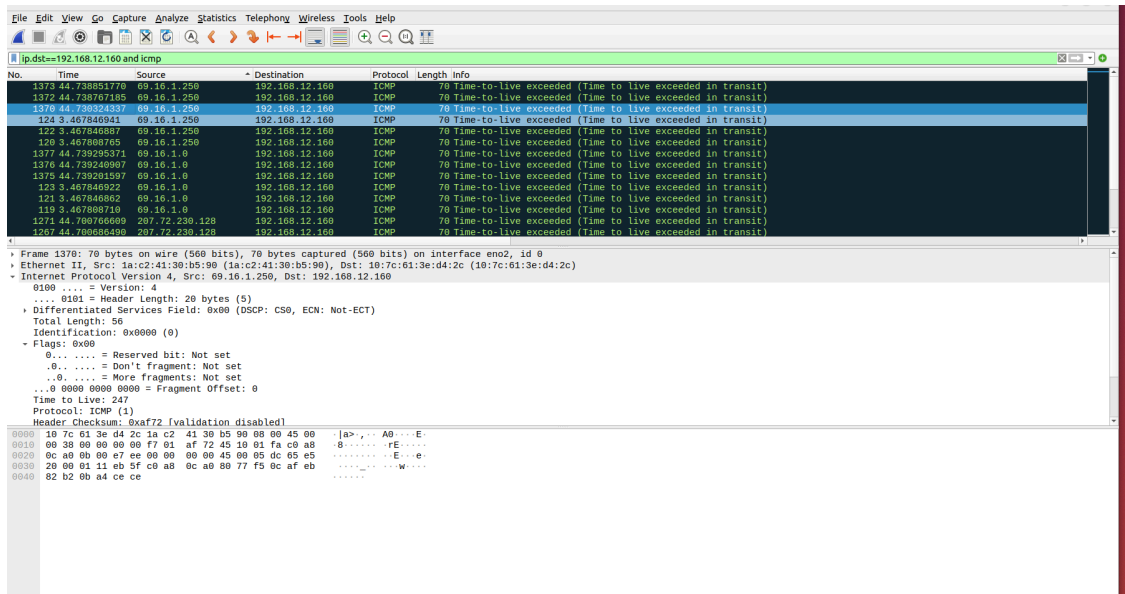
8. Which fields in this sequence of IP datagrams (containing UDP segments) stay constant? Why?

- The fields in this sequence that stay constant are the header length, the Protocol: UDP (17), the source and destination addresses, IPV4, and checksum. These stay the same because they are set within this example IPV4 sequence from the destination to the source.

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

- The pattern that we see in the values in the identification field of the IP datagrams is that the values are increasing.

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 specified is Protocol: 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?

- Yes, the values in the identification fields are similar to the behaviors observed in question 9.

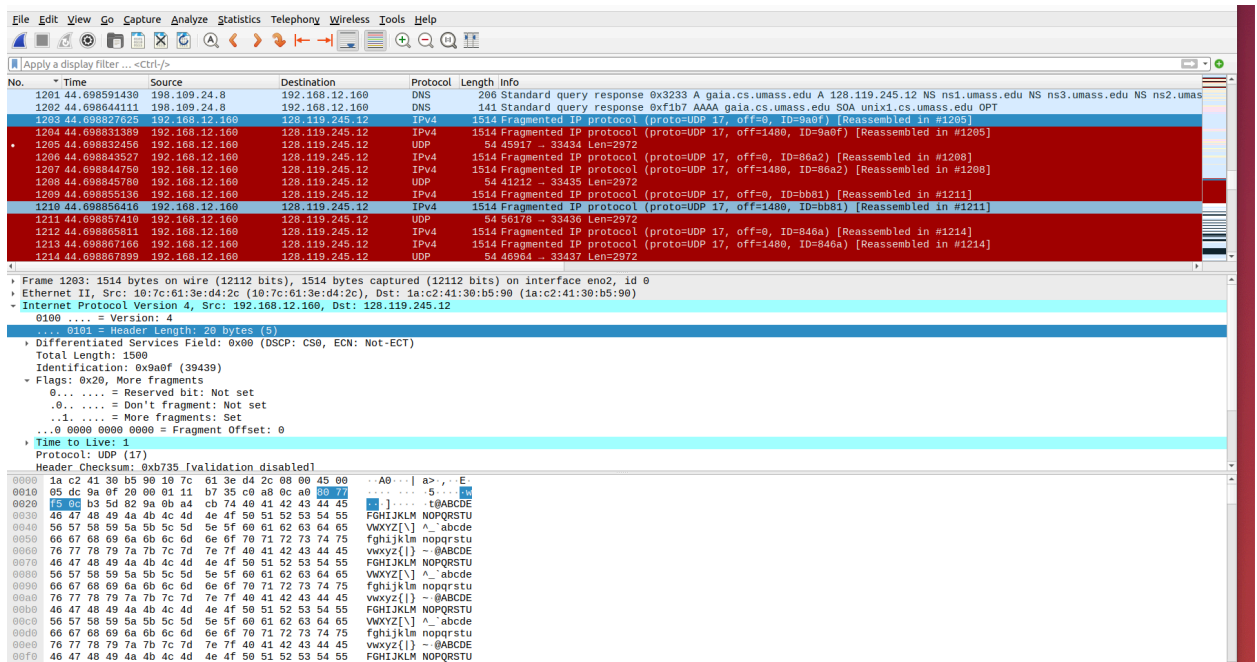
12. Are the values of the TTL fields similar, across all of ICMP packets from all of the routers?

- No, the values are different as they are incrementing as they come from different routers.

13. 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 yes4!)

- The answer is yes.

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



- The More Fragments section states "set". Also, the datagram has "Fragmented IP protocol" in its info.

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

- You can tell this is the first fragment because the fragment offset is 0.

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

- The amount of bytes in this IP datagram is 1500 bytes.

17. Now inspect the datagram containing the second fragment of the fragmented UDP segment. What information in the IP header indicates that this is not the first datagram fragment?

- The fragment offset has a value, in this case 1400.

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

- The checksum values change, the fragment offset changes, and the checksum changes.

19. 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 fragment offset matches the first fragment, that being Fragment Offset: 0.

20. 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 window5.

- 2601:193:8302:4620:215c:f5ae:8b40:a27a.

No.	Time	Source	Destination	Protocol	Length	Info
13	2.457995	52.114.132.110	10.0.0.44	TCP	60	443 → 49067 [ACK] Seq=337 Ack=189 Win=2051 Len=0
14	2.458002	52.114.132.110	10.0.0.44	TCP	60	443 → 49067 [ACK] Seq=337 Ack=189 Win=2051 Len=0
15	2.653323	10.0.0.123	224.0.0.251	MDNS	139	Standard query 0x0000 PTR _companion-link_.tcp.local, "QU" question PTR _sleep-proxy.udp.local, "QU" question OPT
16	2.653622	fe80::1085:6434:358::ff02::fb	ff02::fb	MDNS	159	Standard query 0x0000 PTR _companion-link_.tcp.local, "QU" question PTR _sleep-proxy.udp.local, "QU" question OPT
17	3.201704	10.0.0.123	224.0.0.251	MDNS	60	Conf. Root = 0004/6434358::ff02::fb Cost = 0 Port = 0x0001
18	3.020004	52.112.110.82	10.0.0.44	TCP	56	443 → 50510 [RST] Seq=1 Ack=1 Win=0 Len=0
19	3.814364	2601:193:8302:4620::215c:f5ae:8b40:a27a	2001:558:feed::1	DNS	91	Standard query 0x4607 A youtube.com
20	3.814489	2601:193:8302:4620::215c:f5ae:8b40:a27a	2001:558:feed::1	DNS	91	Standard query 0x920d AAAA youtube.com
21	3.815370	2601:193:8302:4620::215c:f5ae:8b40:a27a	2001:558:feed::1	DNS	95	Standard query 0x7804 A www.youtube.com
22	3.815905	2601:193:8302:4620::215c:f5ae:8b40:a27a	2001:558:feed::1	DNS	95	Standard query 0x04fe AAAA www.youtube.com
23	3.946846	2001:558:feed::1	2601:193:8302:4620::215c:f5ae:8b40:a27a	DNS	107	Standard query response 0x4607 A youtube.com A 172.217.10.142
24	3.953852	2001:558:feed::1	2601:193:8302:4620::215c:f5ae:8b40:a27a	DNS	241	Standard query response 0x04fe AAAA www.youtube.com CNAME youtube-ui.l.google.com AAAA 2601:fb0:4006:806::200e AAAA 2601:fb0:4006:806::200e
25	3.954763	2601:193:8302:4620::215c:f5ae:8b40:a27a	2001:558:feed::1	DNS	103	Standard query 0x7804 A youtube-ui.l.google.com
26	3.955402	2001:558:feed::1	2601:193:8302:4620::215c:f5ae:8b40:a27a	DNS	337	Standard query response 0x7804 A www.youtube.com CNAME youtube-ui.l.google.com A 172.217.10.110 A 172.217.10.142 A 172.217.10.144 A 172.217.10.146
27	3.955405	2001:558:feed::1	2601:193:8302:4620::215c:f5ae:8b40:a27a	DNS	119	Standard query response 0x920d AAAA youtube.com AAAA 2601:fb0:4006:815:200e
28	3.956819	2601:193:8302:4620::215c:f5ae:8b40:a27a	2001:558:feed::1	TCP	98	50629 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1440 WS=64 TSval=459684260 TSecr=0 SACK_PERM=1
29	4.099918	2601:193:8302:4620::215c:f5ae:8b40:a27a	2001:558:feed::1	TCP	94	443 → 50629 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1440 SACK_PERM=1 TSval=2729032167 TSecr=459684260 WS=256
30	4.099922	2001:558:feed::1	2601:193:8302:4620::215c:f5ae:8b40:a27a	DNS	311	Standard query response 0x7804 A www.youtube.com A 172.217.10.230 A 172.217.12.174 A 172.217.12.206
31	4.100935	2601:193:8302:4620::215c:f5ae:8b40:a27a	2001:558:feed::1	TCP	86	50629 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0 TSval=459684402 TSecr=2729032167
32	4.100936	2601:193:8302:4620::215c:f5ae:8b40:a27a	2001:558:feed::1	ICMPv6	350	Destination Unreachable (Port unreachable)
33	4.110370	2601:193:8302:4620::215c:f5ae:8b40:a27a	2001:558:feed::1	TLSv1.3	603	Client Hello
34	4.227644	52.70.172.237	10.0.0.44	TLSv1.2	612	Application Data
35	4.227647	52.70.172.237	10.0.0.44	TLSv1.2	97	Encrypted Alert
36	4.227706	10.0.0.44	52.70.172.237	TCP	66	50621 → 443 [ACK] Seq=1 Ack=547 Win=2048 Len=0 TSval=459684528 TSecr=300609999
37	4.227755	10.0.0.44	52.70.172.237	TCP	66	50621 → 443 [ACK] Seq=1 Ack=579 Win=2047 Len=0 TSval=459684528 TSecr=300609999
38	4.227946	10.0.0.44	52.70.172.237	TLSv1.2	97	Encrypted Alert
39	4.228589	10.0.0.44	52.70.172.237	TCP	66	50621 → 443 [FIN, ACK] Seq=32 Ack=579 Win=2048 Len=0 TSval=459684528 TSecr=300609999
40	4.229512	2601:193:8302:4620::215c:f5ae:8b40:a27a	2001:558:feed::1	DNS	116	Standard query 0xea78 AAAA ss-prod-ue1-notif-53.aws.adobe.com
41	4.229576	2601:193:8302:4620::215c:f5ae:8b40:a27a	2001:558:feed::1	DNS	116	Standard query 0xb4ac A ss-prod-ue1-notif-53.aws.adobe.com
42	4.490971	10.0.0.44	52.70.172.237	TCP	97	[TCP Retransmission] 50621 → 443 [FIN, PSH, ACK] Seq=1 Ack=579 Win=2048 Len=31 TSval=459684790 TSecr=300609999
43	4.490971	Source 20.3a.2a	Spawning tree (for...)	STP	60	Conf. Root = 3004/6434358::ff02::fb Cost = 0 Port = 0x0001
44	4.794130	2601:193:8302:4620::215c:f5ae:8b40:a27a	2001:558:feed::1	TCP	603	[TCP Retransmission] 50620 → 443 [PSH, ACK] Seq=1 Ack=1 Win=131328 Len=517 TSval=459685035 TSecr=2729032167
45	4.833433	10.0.0.44	52.70.172.237	TCP	97	[TCP Retransmission] 50621 → 443 [FIN, PSH, ACK] Seq=1 Ack=579 Win=2048 Len=31 TSval=459685132 TSecr=300609999
46	4.944008	128.119.240.53	10.0.0.44	TCP	66	4282 → 50018 [ACK] Seq=1 Ack=1 Win=209 Len=0 TSval=2851596312 TSecr=459679293
47	4.041608	10.0.0.44	128.119.240.53	TCP	66	4282 → 50018 [ACK] Seq=1 Ack=1 Win=209 Len=0 TSval=2851596312 TSecr=459679293

Frame 20: 91 bytes on wire (728 bits), 91 bytes captured (728 bits) on interface en0, id 0
Ethernet II, Src: Apple_98:d9:27 (78:4f:43:98:d9:27), Dst: Technico_B1:74:5a (44:1c:12:81:74:5a)
Internet Protocol Version 6, Src: 2601:193:8302:4620:215c:f5ae:8b40:a27a, Dst: 2001:558:feed::1
[115] ...
... 0000 0000 ... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)
... 0110 0011 1110 1000 = 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
User Datagram Protocol, Src Port: 64430, Dst Port: 53
Source Port: 64430
Destination Port: 53
Length: 37
Checksum: 0x3953 [unverified]
[Checksum Status: Unverified]
[Stream index: 3]
[Timestamps]
UDP payload (29 bytes)
Domain Name System (query)
Transaction ID: 0x920d
Flags: 0x0100 Standard query
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
Queries
[Response In: 27]

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

- 2001:558:feed::1

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

- The value of the flow label is 0x63ed0.

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

- The payload length is 37.

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

- The upper layer protocol is UDP (17) for the destination 2001:558:feed::1

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

- There is 1 IPv6 address being returned.

```
main Name System (response)
Transaction ID: 0x920d
Flags: 0x8180 Standard query response, No error
Questions: 1
Answer RRs: 1
Authority RRs: 0
Additional RRs: 0
Queries
Answers
  > youtube.com: type AAAA, class IN, addr 2607:f8b0:4006:815::200e
[Request In: 20]
[Time: 0.140916000 seconds]
```

26. 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 is AAAA Address: 2607:f8b0:4006:815::200e

```
~ Answers
~ youtube.com: type AAAA, class IN, addr 2607:f8b0:4006:815::200e
  Name: youtube.com
  Type: AAAA (IPv6 Address) (28)
  Class: IN (0x0001)
  Time to live: 201 (3 minutes, 21 seconds)
  Data length: 16
  AAAA Address: 2607:f8b0:4006:815::200e
```