

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?

Time	No.	Source	Destination	Protocol	Length	Info
5.635364690	40	10.240.118.97	142.250.196.174	ICMP	98	Echo (ping) request id=0x0002, seq=1/256, ttl=64 (reply in 45)
5.650506842	45	142.250.196.174	10.240.118.97	ICMP	98	Echo (ping) reply id=0x0002, seq=1/256, ttl=58 (request in 45)
6.637225159	59	10.240.118.97	142.250.196.174	ICMP	98	Echo (ping) request id=0x0002, seq=2/512, ttl=64 (reply in 60)
6.652856657	60	142.250.196.174	10.240.118.97	ICMP	98	Echo (ping) reply id=0x0002, seq=2/512, ttl=58 (request in 60)
7.638910711	70	10.240.118.97	142.250.196.174	ICMP	98	Echo (ping) request id=0x0002, seq=3/768, ttl=64 (reply in 71)
7.662849400	71	142.250.196.174	10.240.118.97	ICMP	98	Echo (ping) reply id=0x0002, seq=3/768, ttl=58 (request in 71)
8.640993296	87	10.240.118.97	142.250.196.174	ICMP	98	Echo (ping) request id=0x0002, seq=4/1024, ttl=64 (reply in 88)
8.656239122	88	142.250.196.174	10.240.118.97	ICMP	98	Echo (ping) reply id=0x0002, seq=4/1024, ttl=58 (request in 88)
9.642333364	94	10.240.118.97	142.250.196.174	ICMP	98	Echo (ping) request id=0x0002, seq=5/1280, ttl=64 (reply in 95)
9.662376772	95	142.250.196.174	10.240.118.97	ICMP	98	Echo (ping) reply id=0x0002, seq=5/1280, ttl=58 (request in 95)

- **Source IP address:** 10.240.118.97
- **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 image shows a Wireshark packet capture of a DNS query and response. The packet list on the left shows four packets:

- 5.634354357: Standard query 0x8a77 A google.com (DNS, 70 bytes)
- 5.634572365: Standard query 0xcd7f AAAA google.com (DNS, 70 bytes)
- 5.634696180: Standard query response 0x8a77 A google.com A 142.250.196.174 (DNS, 86 bytes)
- 5.634800423: Standard query response 0xcd7f AAAA google.com AAAA 2404:6e00:0000:0000:0000:0000:0000:0000 (DNS, 98 bytes)

The packet details pane for the selected packet (5.634696180) shows the following structure:

- Frame 38: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface eno0
- Ethernet II, Src: Cisco_13:e0:82 (bc:d2:95:13:e0:82), Dst: HP_0a:99:9a (e0:73:e7:0a:00:10)
- Internet Protocol Version 4, Src: 10.250.200.3, Dst: 10.240.118.97
- User Datagram Protocol, Src Port: 53, Dst Port: 52268
- Domain Name System (response)
 - Transaction ID: 0x8a77
 - Flags: 0x8180 Standard query response, No error
 - Questions: 1
 - Answer RRs: 1
 - Authority RRs: 0
 - Additional RRs: 0
 - Queries
 - Answers
 - google.com: type A, class IN, addr 142.250.196.174

The packet bytes pane shows the raw data of the response, with the IP address 142.250.196.174 highlighted in blue.

- 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)

dns

- 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:

dns						
Time	No.	Source	Destination	Protocol	Length	Info
5.634354357	36	10.240.118.97	10.250.200.3	DNS	70	Standard query 0x8a77 A google.com
5.634572365	37	10.240.118.97	10.250.200.3	DNS	70	Standard query 0xcd7f AAAA google.com
5.634696180	38	10.250.200.3	10.240.118.97	DNS	86	Standard query response 0x8a77 A google.com A 142.250.196.1
5.634800423	39	10.250.200.3	10.240.118.97	DNS	98	Standard query response 0xcd7f AAAA google.com AAAA 2404:68
5.651217578	46	10.240.118.97	10.250.200.3	DNS	88	Standard query 0x996d PTR 174.196.250.142.in-addr.arpa
5.651454102	47	10.250.200.3	10.240.118.97	DNS	127	Standard query response 0x996d PTR 174.196.250.142.in-addr.

Ethernet II, Src: Cisco 13:e0:82 (bc:d2:95:13:e0:82), Dst: HP_0a:99:9a (e0:73:e7:0a:99:9a)	0000	e0 73 e7 0a 99 9a bc d2 95 13 e0 82 08 00 45 00	...
Internet Protocol Version 4, Src: 10.250.200.3, Dst: 10.240.118.97	0010	00 48 7c a8 40 00 3f 11 6a ae 0a fa c8 03 0a f0	...
Version: 4	0020	76 61 00 35 cc 2c 00 34 ac 39 8a 77 81 80 00 01	...

The version of the IP address used for the above request=IPv4

dns

Time	No.	Source	Destination	Protocol	Length	Info
5.634354357	36	10.240.118.97	10.250.200.3	DNS	70	Standard query 0x8a77 A google.com
5.634572365	37	10.240.118.97	10.250.200.3	DNS	70	Standard query 0xcd7f AAAA google.com
5.634696180	38	10.250.200.3	10.240.118.97	DNS	86	Standard query response 0x8a77 A google.com A 142.250.196.174
5.634800423	39	10.250.200.3	10.240.118.97	DNS	98	Standard query response 0xcd7f AAAA google.com AAAA 2404:6800:4007:813::200e
5.651217578	46	10.240.118.97	10.250.200.3	DNS	88	Standard query 0x996d PTR 174.196.250.142.in-addr.arpa
5.651454102	47	10.250.200.3	10.240.118.97	DNS	127	Standard query response 0x996d PTR 174.196.250.142.in-addr.arpa

Ethernet II, Src: Cisco_13:e0:82 (bc:d2:95:13:e0:82), Dst: HP_0a:99:9a (e0:73:e7:0a:99:9a)

Internet Protocol Version 4, Src: 10.250.200.3, Dst: 10.240.118.97

Version: 4

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?

Time	No.	Source	Destination	Protocol	Length	Info
5.634354357	36	10.240.118.97	10.250.200.3	DNS	70	Standard query 0x8a77 A google.com
5.634572365	37	10.240.118.97	10.250.200.3	DNS	70	Standard query 0xcd7f AAAA google.com
5.634696180	38	10.250.200.3	10.240.118.97	DNS	86	Standard query response 0x8a77 A google.com A 142.250.196.174
5.634800423	39	10.250.200.3	10.240.118.97	DNS	98	Standard query response 0xcd7f AAAA google.com AAAA 2404:6800:4000:0000:0000:0000:0000:0000
5.651217578	46	10.240.118.97	10.250.200.3	DNS	88	Standard query 0x996d PTR 174.196.250.142.in-addr.arpa
5.651454102	47	10.250.200.3	10.240.118.97	DNS	127	Standard query response 0x996d PTR 174.196.250.142.in-addr.arpa
13.541846170	138	10.250.200.3	10.240.119.85	DNS	425	Standard query response 0x3c49 AAAA connectivity-check.ubuntu.com

Internet Protocol Version 4, Src: 10.240.118.97, Dst: 10.250.200.3	0000 44 b6 be 0a 8f 70 e0 73 e7 0a 99 9a 08 00 45 00
0100 = Version: 4	0010 00 38 1d e4 00 00 40 11 08 83 0a f0 76 61 0a fa
.... 0101 = Header Length: 20 bytes (5)	0020 c8 03 cc 2c 00 35 00 24 54 84 8a 77 01 00 00 01
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)	0030 00 00 00 00 00 00 06 67 6f 6f 67 6c 65 03 63 6f
Total Length: 56	0040 6d 00 00 01 00 01
Identification: 0x1de4 (7652)	
0000 = Flags: 0x0	
...0 0000 0000 0000 = Fragment Offset: 0	
Time to Live: 64	

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?

Time	No.	Source	Destination	Protocol	Length	Info
5.634354357	36	10.240.118.97	10.250.200.3	DNS	70	Standard query 0x8a77 A google.com
5.634572365	37	10.240.118.97	10.250.200.3	DNS	70	Standard query 0xcd7f AAAA google.com
5.634696180	38	10.250.200.3	10.240.118.97	DNS	86	Standard query response 0x8a77 A google.com A 142.250.196.174
5.634800423	39	10.250.200.3	10.240.118.97	DNS	98	Standard query response 0xcd7f AAAA google.com AAAA 2404:6800:4000:0000:0000:0000:0000:0000
5.651217578	46	10.240.118.97	10.250.200.3	DNS	88	Standard query 0x996d PTR 174.196.250.142.in-addr.arpa
5.651454102	47	10.250.200.3	10.240.118.97	DNS	127	Standard query response 0x996d PTR 174.196.250.142.in-addr.arpa
13.541846170	138	10.250.200.3	10.240.119.85	DNS	425	Standard query response 0x3c49 AAAA connectivity-check.ubuntu.com

Internet Protocol Version 4, Src: 10.240.118.97, Dst: 10.250.200.3	0000 44 b6 be 0a 8f 70 e0 73 e7 0a 99 9a 08 00 45 00
0100 = Version: 4	0010 00 38 1d e4 00 00 40 11 08 83 0a f0 76 61 0a fa
.... 0101 = Header Length: 20 bytes (5)	0020 c8 03 cc 2c 00 35 00 24 54 84 8a 77 01 00 00 01
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)	0030 00 00 00 00 00 00 06 67 6f 6f 67 6c 65 03 63 6f
Total Length: 56	0040 6d 00 00 01 00 01
Identification: 0x1de4 (7652)	
0000 = Flags: 0x0	
...0 0000 0000 0000 = Fragment Offset: 0	
Time to Live: 64	
Protocol: UDP (17)	

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.

Time	No.	Source	Destination	Protocol	Length	Info
5.634354357	36	10.240.118.97	10.250.200.3	DNS	70	Standard query 0x8a77 A google.com
5.634572365	37	10.240.118.97	10.250.200.3	DNS	70	Standard query 0xcd7f AAAA google.com
5.634696180	38	10.250.200.3	10.240.118.97	DNS	86	Standard query response 0x8a77 A google.com A 142.250.196.174
5.634800423	39	10.250.200.3	10.240.118.97	DNS	98	Standard query response 0xcd7f AAAA google.com AAAA 2404:6800:4000:0000:0000:0000:0000:0000
5.651217578	46	10.240.118.97	10.250.200.3	DNS	88	Standard query 0x996d PTR 174.196.250.142.in-addr.arpa
5.651454102	47	10.250.200.3	10.240.118.97	DNS	127	Standard query response 0x996d PTR 174.196.250.142.in-addr.arpa
13.541846170	138	10.250.200.3	10.240.119.85	DNS	425	Standard query response 0x3c49 AAAA connectivity-check.ubuntu.com

Internet Protocol Version 4, Src: 10.240.118.97, Dst: 10.250.200.3	0000 44 b6 be 0a 8f 70 e0 73 e7 0a 99 9a 08 00 45 00
0100 = Version: 4	0010 00 38 1d e4 00 00 40 11 08 83 0a f0 76 61 0a fa
.... 0101 = Header Length: 20 bytes (5)	0020 c8 03 cc 2c 00 35 00 24 54 84 8a 77 01 00 00 01
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)	0030 00 00 00 00 00 00 06 67 6f 6f 67 6c 65 03 63 6f
Total Length: 56	0040 6d 00 00 01 00 01

Payload Size = Total Length - IP Header Size

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

Time	No.	Source	Destination	Protocol	Length	Info
5.634354357	36	10.240.118.97	10.250.200.3	DNS	70	Standard query 0x8a77 A google.com
5.634572365	37	10.240.118.97	10.250.200.3	DNS	70	Standard query 0xcd7f AAAA google.com
5.634696180	38	10.250.200.3	10.240.118.97	DNS	86	Standard query response 0x8a77 A google.com A 142.250.196.174
5.634800423	39	10.250.200.3	10.240.118.97	DNS	98	Standard query response 0xcd7f AAAA google.com AAAA 2404:6800:4...
5.651217578	46	10.240.118.97	10.250.200.3	DNS	88	Standard query 0x996d PTR 174.196.250.142.in-addr.arpa
5.651454102	47	10.250.200.3	10.240.118.97	DNS	127	Standard query response 0x996d PTR 174.196.250.142.in-addr.arpa...
13.541846170	138	10.250.200.3	10.240.119.85	DNS	425	Standard query response 0x3c49 AAAA connectivity-check.ubuntu.c...

Frame 36: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface eno...	0000	44 b6 be 0a 8f 70 e0 73 e7 0a 99 9a 08 00 45 00	D... p s
Ethernet II, Src: HP_0a:99:9a (e0:73:e7:0a:99:9a), Dst: Cisco_0a:8f:70 (44:b6:be:0a:...	0010	00 38 1d e4 00 00 40 11 08 83 0a f0 76 61 0a fa	8...@...
Internet Protocol Version 4, Src: 10.240.118.97, Dst: 10.250.200.3	0020	c8 03 cc 2c 00 35 00 24 54 84 8a 77 01 00 00 015... T
User Datagram Protocol, Src Port: 52268, Dst Port: 53	0030	00 00 00 00 00 00 06 67 6f 6f 67 6c 65 03 63 6fg o
Source Port: 52268	0040	6d 00 00 01 00 01	m.....
Destination Port: 53			
Length: 36			

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.

Time	No.	Source	Destination	Protocol	Length	Info
5.634354357	36	10.240.118.97	10.250.200.3	DNS	70	Standard query 0x8a77 A google.com
5.634572365	37	10.240.118.97	10.250.200.3	DNS	70	Standard query 0xcd7f AAAA google.com
5.634696180	38	10.250.200.3	10.240.118.97	DNS	86	Standard query response 0x8a77 A google.com A 142.250.196.174
5.634800423	39	10.250.200.3	10.240.118.97	DNS	98	Standard query response 0xcd7f AAAA google.com AAAA 2404:6800:4...
5.651217578	46	10.240.118.97	10.250.200.3	DNS	88	Standard query 0x996d PTR 174.196.250.142.in-addr.arpa
5.651454102	47	10.250.200.3	10.240.118.97	DNS	127	Standard query response 0x996d PTR 174.196.250.142.in-addr.arpa...
13.541846170	138	10.250.200.3	10.240.119.85	DNS	425	Standard query response 0x3c49 AAAA connectivity-check.ubuntu.c...

Internet Protocol Version 4, Src: 10.240.118.97, Dst: 10.250.200.3	0000	44 b6 be 0a 8f 70 e0 73 e7 0a 99 9a 08 00 45 00	D... p s
0100 = Version: 4	0010	00 38 1d e4 00 00 40 11 08 83 0a f0 76 61 0a fa	8...@...
.... 0101 = Header Length: 20 bytes (5)	0020	c8 03 cc 2c 00 35 00 24 54 84 8a 77 01 00 00 015... T
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)	0030	00 00 00 00 00 00 06 67 6f 6f 67 6c 65 03 63 6fg o
Total Length: 56	0040	6d 00 00 01 00 01	m.....
Identification: 0x1de4 (7652)			
0000 = Flags: 0x0			
0... = Reserved bit: Not set			
.0.. = Don't fragment: Not set			
..0. = More fragments: Not set			
...0 0000 0000 = Fragment Offset: 0			

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.

Time	No.	Source	Destination	Protocol	Length	Info
5.635364690	40	10.240.118.97	142.250.196.174	ICMP	98	Echo (ping) request id=0x0002, seq=1/256, ttl=64 (reply in 45)
5.650506842	45	142.250.196.174	10.240.118.97	ICMP	98	Echo (ping) reply id=0x0002, seq=1/256, ttl=58 (request in 4...)
6.637225159	59	10.240.118.97	142.250.196.174	ICMP	98	Echo (ping) request id=0x0002, seq=2/512, ttl=64 (reply in 60)
6.652856657	60	142.250.196.174	10.240.118.97	ICMP	98	Echo (ping) reply id=0x0002, seq=2/512, ttl=58 (request in 5...)
7.638910711	70	10.240.118.97	142.250.196.174	ICMP	98	Echo (ping) request id=0x0002, seq=3/768, ttl=64 (reply in 71)
7.662849400	71	142.250.196.174	10.240.118.97	ICMP	98	Echo (ping) reply id=0x0002, seq=3/768, ttl=58 (request in 7...)
8.640993296	87	10.240.118.97	142.250.196.174	ICMP	98	Echo (ping) request id=0x0002, seq=4/1024, ttl=64 (reply in 88)
8.656239122	88	142.250.196.174	10.240.118.97	ICMP	98	Echo (ping) reply id=0x0002, seq=4/1024, ttl=58 (request in ...)
9.642333364	94	10.240.118.97	142.250.196.174	ICMP	98	Echo (ping) request id=0x0002, seq=5/1280, ttl=64 (reply in 95)
9.662376772	95	142.250.196.174	10.240.118.97	ICMP	98	Echo (ping) reply id=0x0002, seq=5/1280, ttl=58 (request in ...)

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

Time	No.	Source	Destination	Protocol	Length	Info
5.635364690	40	10.240.118.97	142.250.196.174	ICMP	98	Echo (ping) request id=0x0002, seq=1/256, ttl=64 (reply in
5.650506842	45	142.250.196.174	10.240.118.97	ICMP	98	Echo (ping) reply id=0x0002, seq=1/256, ttl=58 (request i
6.637225159	59	10.240.118.97	142.250.196.174	ICMP	98	Echo (ping) request id=0x0002, seq=2/512, ttl=64 (reply in
6.652856657	60	142.250.196.174	10.240.118.97	ICMP	98	Echo (ping) reply id=0x0002, seq=2/512, ttl=58 (request i
7.638910711	70	10.240.118.97	142.250.196.174	ICMP	98	Echo (ping) request id=0x0002, seq=3/768, ttl=64 (reply in
7.662849400	71	142.250.196.174	10.240.118.97	ICMP	98	Echo (ping) reply id=0x0002, seq=3/768, ttl=58 (request i
8.640993296	87	10.240.118.97	142.250.196.174	ICMP	98	Echo (ping) request id=0x0002, seq=4/1024, ttl=64 (reply in
8.656239122	88	142.250.196.174	10.240.118.97	ICMP	98	Echo (ping) reply id=0x0002, seq=4/1024, ttl=58 (request
9.642333364	94	10.240.118.97	142.250.196.174	ICMP	98	Echo (ping) request id=0x0002, seq=5/1280, ttl=64 (reply in
9.662376772	95	142.250.196.174	10.240.118.97	ICMP	98	Echo (ping) reply id=0x0002, seq=5/1280, ttl=58 (request

Frame 40: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eno000
Ethernet II, Src: HP_0a:99:9a (e0:73:e7:0a:99:9a), Dst: Cisco_0a:8f:70 (44:b6:be:0a:8f:70)
Internet Protocol Version 4, Src: 10.240.118.97, Dst: 142.250.196.174
Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0

ICMP Echo Request (Type 8, Code 0)

Time	No.	Source	Destination	Protocol	Length	Info
5.635364690	40	10.240.118.97	142.250.196.174	ICMP	98	Echo (ping) request id=0x0002, seq=1/256, ttl=64 (reply in
5.650506842	45	142.250.196.174	10.240.118.97	ICMP	98	Echo (ping) reply id=0x0002, seq=1/256, ttl=58 (request
6.637225159	59	10.240.118.97	142.250.196.174	ICMP	98	Echo (ping) request id=0x0002, seq=2/512, ttl=64 (reply in
6.652856657	60	142.250.196.174	10.240.118.97	ICMP	98	Echo (ping) reply id=0x0002, seq=2/512, ttl=58 (request
7.638910711	70	10.240.118.97	142.250.196.174	ICMP	98	Echo (ping) request id=0x0002, seq=3/768, ttl=64 (reply in
7.662849400	71	142.250.196.174	10.240.118.97	ICMP	98	Echo (ping) reply id=0x0002, seq=3/768, ttl=58 (request
8.640993296	87	10.240.118.97	142.250.196.174	ICMP	98	Echo (ping) request id=0x0002, seq=4/1024, ttl=64 (reply in
8.656239122	88	142.250.196.174	10.240.118.97	ICMP	98	Echo (ping) reply id=0x0002, seq=4/1024, ttl=58 (request
9.642333364	94	10.240.118.97	142.250.196.174	ICMP	98	Echo (ping) request id=0x0002, seq=5/1280, ttl=64 (reply in
9.662376772	95	142.250.196.174	10.240.118.97	ICMP	98	Echo (ping) reply id=0x0002, seq=5/1280, ttl=58 (request

Frame 45: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eno000
Ethernet II, Src: Cisco_13:e0:82 (bc:d2:95:13:e0:82), Dst: HP_0a:99:9a (e0:73:e7:0a:99:9a)
Internet Protocol Version 4, Src: 142.250.196.174, Dst: 10.240.118.97
Internet Control Message Protocol
Type: 0 (Echo (ping) reply)
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**

Fields that remain constant

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)**.

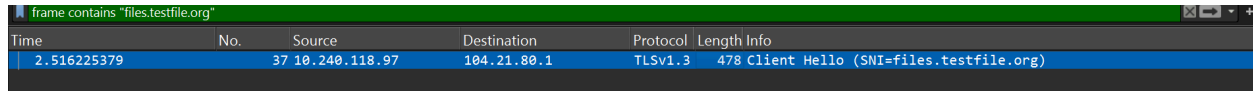
Part 2: Fragmentation

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?

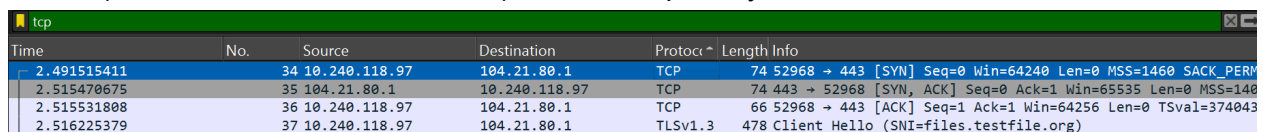


Time	No.	Source	Destination	Protocol	Length	Info
2.516225379	37	10.240.118.97	104.21.80.1	TLSv1.3	478	Client Hello (SNI=files.testfile.org)

- 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?

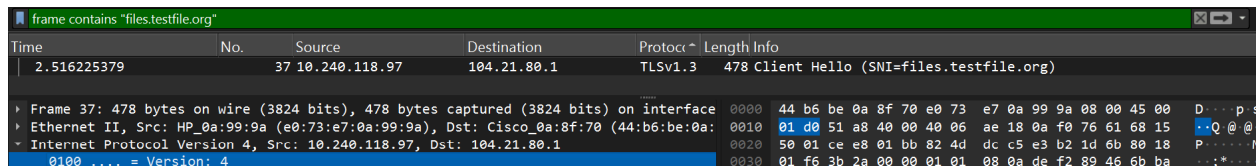
Since **wget** is used to download files over HTTP or HTTPS, it relies on the TCP (Transmission Control Protocol) at the transport layer.



Time	No.	Source	Destination	Protocol	Length	Info
2.491515411	34	10.240.118.97	104.21.80.1	TCP	74	52968 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
2.515470675	35	104.21.80.1	10.240.118.97	TCP	74	443 → 52968 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
2.515531808	36	10.240.118.97	104.21.80.1	TCP	66	52968 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=374043
2.516225379	37	10.240.118.97	104.21.80.1	TLSv1.3	478	Client Hello (SNI=files.testfile.org)

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

3. What is the IP version?

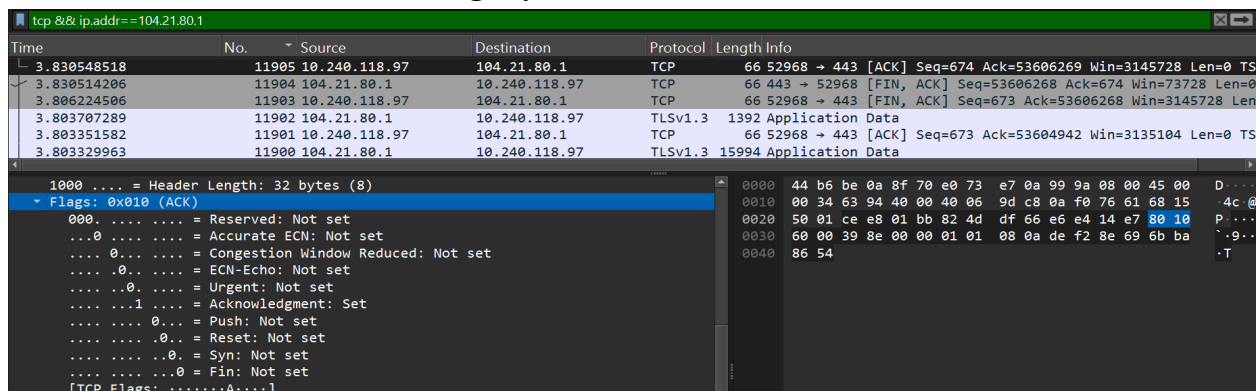


Time	No.	Source	Destination	Protocol	Length	Info
2.516225379	37	10.240.118.97	104.21.80.1	TLSv1.3	478	Client Hello (SNI=files.testfile.org)

Frame 37: 478 bytes on wire (3824 bits), 478 bytes captured (3824 bits) on interface eth0
Ethernet II, Src: HP_0a:99:9a (e0:73:e7:0a:99:9a), Dst: Cisco_0a:8f:70 (44:b6:be:0a:8f:70)
Internet Protocol Version 4, Src: 10.240.118.97, Dst: 104.21.80.1
0100 = Version: 4

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

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



Time	No.	Source	Destination	Protocol	Length	Info
3.830548518	11905	10.240.118.97	104.21.80.1	TCP	66	52968 → 443 [ACK] Seq=674 Ack=53606269 Win=3145728 Len=0
3.830514206	11904	104.21.80.1	10.240.118.97	TCP	66	443 → 52968 [FIN, ACK] Seq=53606268 Ack=674 Win=73728 Len=0
3.806224506	11903	10.240.118.97	104.21.80.1	TCP	66	52968 → 443 [FIN, ACK] Seq=673 Ack=53606268 Win=3145728 Len=0
3.803707289	11902	104.21.80.1	10.240.118.97	TLSv1.3	1392	Application Data
3.803351582	11901	10.240.118.97	104.21.80.1	TCP	66	52968 → 443 [ACK] Seq=673 Ack=53604942 Win=3135104 Len=0
3.803329963	11900	104.21.80.1	10.240.118.97	TLSv1.3	15994	Application Data

1000 = Header Length: 32 bytes (8)
Flags: 0x010 (ACK)
000. = Reserved: Not set
...0. = Accurate ECN: Not set
...0. = Congestion Window Reduced: Not set
...0. = ECN-Echo: Not set
...0. = Urgent: Not set
...1. = Acknowledgment: Set
...0. = Push: Not set
...0. = Reset: Not set
...0. = Syn: Not set
...0. = Fin: Not set
[TCP Flags:A....]

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?

Time	No.	Source	Destination	Protocol	Length	Info
3.814364	19	2601:193:8302:4620::...	2001:558:feed::1	DNS	91	Standard query 0x4667 A youtube.com
3.814489	20	2601:193:8302:4620::...	2001:558:feed::1	DNS	91	Standard query 0x920d AAAA youtube.com
3.819370	21	2601:193:8302:4620::...	2001:558:feed::1	DNS	95	Standard query 0x7884 A www.youtube.com
3.819905	22	2601:193:8302:4620::...	2001:558:feed::1	DNS	95	Standard query 0x04fe AAAA www.youtube.com
3.946846	23	2001:558:feed::1	2601:193:8302:4620::...	DNS	107	Standard query response 0x4667 A youtube.com A 172.217.10...
3.953852	24	2001:558:feed::1	2601:193:8302:4620::...	DNS	241	Standard query response 0x04fe AAAA www.youtube.com CNAME y...
3.954763	25	2601:193:8302:4620::...	2001:558:feed::1	DNS	103	Standard query 0x7884 A youtube-ui.l.google.com
3.955402	26	2001:558:feed::1	2601:193:8302:4620::...	DNS	337	Standard query response 0x7884 A www.youtube.com CNAME yo...
3.955405	27	2001:558:feed::1	2601:193:8302:4620::...	DNS	119	Standard query response 0x920d AAAA youtube.com AAAA 2607:f...
4.099922	30	2001:558:feed::1	2601:193:8302:4620::...	DNS	311	Standard query response 0x7884 A youtube-ui.l.google.com A

Frame 22: 95 bytes on wire (760 bits), 95 bytes captured (760 bits) on interface en0	0000	44 1c 12 81 74 5a 78 4f	43 98 d9 27 86 dd 60 08	D...
Ethernet II, Src: Apple_98:d9:27 (78:4f:43:98:d9:27), Dst: VantivaUSA_81:74:5a (44:1	0010	f0 f4 00 29 11 ff 26 01	01 93 83 02 46 20 21 5c	...
Internet Protocol Version 6, Src: 2601:193:8302:4620:215c:f5ae:8b40:a27a, Dst: 2001:	0020	f5 ae 8b 40 a2 7a 20 01	05 58 fe ed 00 00 00 00	...
0110 = Version: 6	0030	00 00 00 00 00 01 df 56	00 35 00 29 67 c4 04 fe	...
.... 0000 0000 = Traffic Class: 0x00 (DSCP: CS0, ECN: Not	0040	01 00 00 01 00 00 00 00	00 00 03 77 77 07 79	...
.... 1000 1111 0000 1111 0100 = Flow Label: 0x8f0f4	0050	6f 75 74 75 62 65 03 63	6f 6d 00 00 1c 00 01	out
Payload Length: 41				
Next Header: UDP (17)				
Hop Limit: 255				
Source Address: 2601:193:8302:4620:215c:f5ae:8b40:a27a				
Destination Address: 2001:558:feed::1				

- **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?

Time	No.	Source	Destination	Protocol	Length	Info
3.814364	19	2601:193:8302:4620::...	2001:558:feed::1	DNS	91	Standard query 0x4667 A youtube.com
3.814489	20	2601:193:8302:4620::...	2001:558:feed::1	DNS	91	Standard query 0x920d AAAA youtube.com
3.819370	21	2601:193:8302:4620::...	2001:558:feed::1	DNS	95	Standard query 0x7884 A www.youtube.com
3.819905	22	2601:193:8302:4620::...	2001:558:feed::1	DNS	95	Standard query 0x04fe AAAA www.youtube.com
3.946846	23	2001:558:feed::1	2601:193:8302:4620::...	DNS	107	Standard query response 0x4667 A youtube.com A 172.217.10.1
3.953852	24	2001:558:feed::1	2601:193:8302:4620::...	DNS	241	Standard query response 0x04fe AAAA www.youtube.com CNAME y
3.954763	25	2601:193:8302:4620::...	2001:558:feed::1	DNS	103	Standard query 0x7884 A youtube-ui.l.google.com
3.955402	26	2001:558:feed::1	2601:193:8302:4620::...	DNS	337	Standard query response 0x7884 A www.youtube.com CNAME yout
3.955405	27	2001:558:feed::1	2601:193:8302:4620::...	DNS	119	Standard query response 0x920d AAAA youtube.com AAAA 2607:f
4.099922	30	2001:558:feed::1	2601:193:8302:4620::...	DNS	311	Standard query response 0x7884 A youtube-ui.l.google.com A

Frame 22: 95 bytes on wire (760 bits), 95 bytes captured (760 bits) on interface en0	0000	44 1c 12 81 74 5a 78 4f	43 98 d9 27 86 dd 60 08	D...
Ethernet II, Src: Apple_98:d9:27 (78:4f:43:98:d9:27), Dst: VantivaUSA_81:74:5a (44:1	0010	f0 f4 00 29 11 ff 26 01	01 93 83 02 46 20 21 5c	...
Internet Protocol Version 6, Src: 2601:193:8302:4620:215c:f5ae:8b40:a27a, Dst: 2001:	0020	f5 ae 8b 40 a2 7a 20 01	05 58 fe ed 00 00 00 00	...
0110 = Version: 6	0030	00 00 00 00 00 01 df 56	00 35 00 29 67 c4 04 fe	...
.... 0000 0000 = Traffic Class: 0x00 (DSCP: CS0, ECN: Not	0040	01 00 00 01 00 00 00 00	00 00 03 77 77 07 79	...
.... 1000 1111 0000 1111 0100 = Flow Label: 0x8f0f4	0050	6f 75 74 75 62 65 03 63	6f 6d 00 00 1c 00 01	out
Payload Length: 41				

- 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., 0x000000 or 0x8f0f4).
 - If no special QoS or flow classification is applied, the value is often 0x000000.
3. How much payload data are carried for these IPv6 datagrams? What does this signify?

Time	No.	Source	Destination	Protocol	Length	Info
3.814364	19	2601:193:8302:4620::...	2001:558:feed::1	DNS	91	Standard query 0x4667 A youtube.com
3.814489	20	2601:193:8302:4620::...	2001:558:feed::1	DNS	91	Standard query 0x920d AAAA youtube.com
3.819370	21	2601:193:8302:4620::...	2001:558:feed::1	DNS	95	Standard query 0x7884 A www.youtube.com
3.819905	22	2601:193:8302:4620::...	2001:558:feed::1	DNS	95	Standard query 0x04fe AAAA www.youtube.com
3.946846	23	2001:558:feed::1	2601:193:8302:4620::...	DNS	107	Standard query response 0x4667 A youtube.com A 172.217.10.1
3.953852	24	2001:558:feed::1	2601:193:8302:4620::...	DNS	241	Standard query response 0x04fe AAAA www.youtube.com CNAME y
3.954763	25	2601:193:8302:4620::...	2001:558:feed::1	DNS	103	Standard query 0x7884 A youtube-ui.l.google.com
3.955402	26	2001:558:feed::1	2601:193:8302:4620::...	DNS	337	Standard query response 0x7884 A www.youtube.com CNAME yout
3.955405	27	2001:558:feed::1	2601:193:8302:4620::...	DNS	119	Standard query response 0x920d AAAA youtube.com AAAA 2607:f
4.099922	30	2001:558:feed::1	2601:193:8302:4620::...	DNS	311	Standard query response 0x7884 A youtube-ui.l.google.com A

Frame 22: 95 bytes on wire (760 bits), 95 bytes captured (760 bits) on interface en0	0000	44 1c 12 81 74 5a 78 4f	43 98 d9 27 86 dd 60 08	D...
Ethernet II, Src: Apple_98:d9:27 (78:4f:43:98:d9:27), Dst: VantivaUSA_81:74:5a (44:1	0010	f0 f4 00 29 11 ff 26 01	01 93 83 02 46 20 21 5c	...
Internet Protocol Version 6, Src: 2601:193:8302:4620:215c:f5ae:8b40:a27a, Dst: 2001:	0020	f5 ae 8b 40 a2 7a 20 01	05 58 fe ed 00 00 00 00	...
0110 = Version: 6	0030	00 00 00 00 00 01 df 56	00 35 00 29 67 c4 04 fe	...
.... 0000 0000 = Traffic Class: 0x00 (DSCP: CS0, ECN: Not	0040	01 00 00 01 00 00 00 00	00 00 03 77 77 07 79	...
.... 1000 1111 0000 1111 0100 = Flow Label: 0x8f0f4	0050	6f 75 74 75 62 65 03 63	6f 6d 00 00 1c 00 01	out
Payload Length: 41				

- 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.

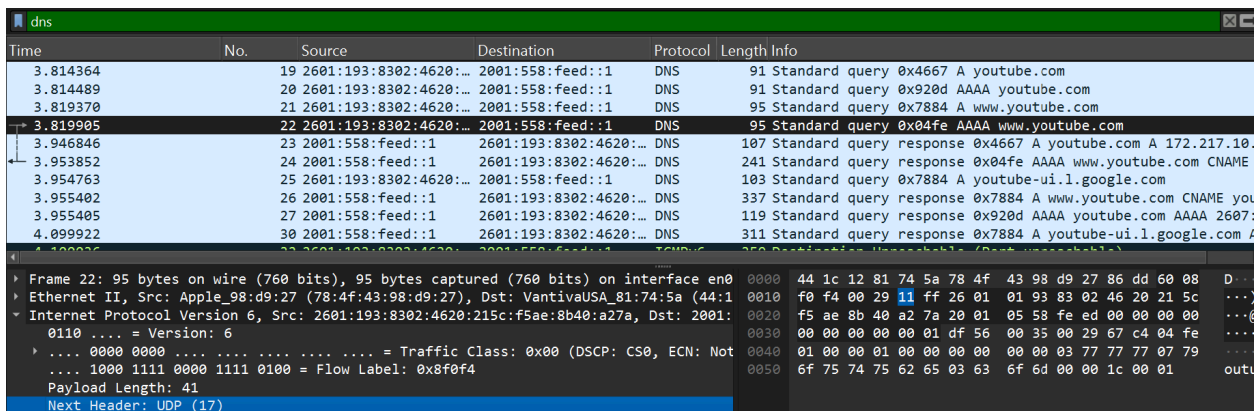
→ **Payload Length = Total Length-Ethernet Header length – IPv6 Header Size**

- 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?



Time	No.	Source	Destination	Protocol	Length	Info
3.814364	19	2601:193:8302:4620::...	2001:558:feed::1	DNS	91	Standard query 0x4667 A youtube.com
3.814489	20	2601:193:8302:4620::...	2001:558:feed::1	DNS	91	Standard query 0x920d AAAA youtube.com
3.819370	21	2601:193:8302:4620::...	2001:558:feed::1	DNS	95	Standard query 0x7884 A www.youtube.com
3.819905	22	2601:193:8302:4620::...	2001:558:feed::1	DNS	95	Standard query 0x04fe AAAA www.youtube.com
3.946846	23	2001:558:feed::1	2601:193:8302:4620::...	DNS	107	Standard query response 0x4667 A youtube.com A 172.217.10...
3.953852	24	2001:558:feed::1	2601:193:8302:4620::...	DNS	241	Standard query response 0x04fe AAAA www.youtube.com CNAME
3.954763	25	2601:193:8302:4620::...	2001:558:feed::1	DNS	103	Standard query 0x7884 A youtube-ui.l.google.com
3.955402	26	2001:558:feed::1	2601:193:8302:4620::...	DNS	337	Standard query response 0x7884 A www.youtube.com CNAME you
3.955405	27	2001:558:feed::1	2601:193:8302:4620::...	DNS	119	Standard query response 0x920d AAAA youtube.com AAAA 2607:
4.099922	30	2001:558:feed::1	2601:193:8302:4620::...	DNS	311	Standard query response 0x7884 A youtube-ui.l.google.com /

Frame	Length	Wire	Captured	Interface	Protocol	Details
22	95	95	95	en0	IPv6	Frame 22: 95 bytes on wire (760 bits), 95 bytes captured (760 bits) on interface en0 Ethernet II, Src: Apple_98:d9:27 (78:4f:43:98:d9:27), Dst: VantivaUSA_81:74:5a (44:1... Internet Protocol Version 6, Src: 2601:193:8302:4620::215c:f5ae:8b40:a27a, Dst: 2001:... 0110 = Version: 6 0000 0000 = Traffic Class: 0x00 (DSCP: CS0, ECN: Not... 1000 1111 0000 1111 0100 = Flow Label: 0x8f0f4 Payload Length: 41 Next Header: UDP (17)

- 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?

Time	No.	Source	Destination	Protocol	Length	Info
3.819905	22	2601:193:8302:4620::...	2601:558:feed::1	DNS	95	Standard query 0x04fe AAAA www.youtube.com
3.946846	23	2601:558:feed::1	2601:193:8302:4620::...	DNS	107	Standard query response 0x4667 A youtube.com A 172.217.10.1
3.953852	24	2601:558:feed::1	2601:193:8302:4620::...	DNS	241	Standard query response 0x04fe AAAA www.youtube.com CNAME y
3.954763	25	2601:193:8302:4620::...	2601:558:feed::1	DNS	103	Standard query 0x7884 A youtube-ui.l.google.com
3.955402	26	2601:558:feed::1	2601:193:8302:4620::...	DNS	337	Standard query response 0x7884 A www.youtube.com CNAME yout
3.955405	27	2601:558:feed::1	2601:193:8302:4620::...	DNS	119	Standard query response 0x920d AAAA youtube.com AAAA 2607:f
4.099922	30	2601:558:feed::1	2601:193:8302:4620::...	DNS	311	Standard query response 0x7884 A youtube-ui.l.google.com A

Frame 27: 119 bytes on wire (952 bits), 119 bytes captured (952 bits) on interface e	0000	78 4f 43 98 d9 27 44 1c 12 81 74 5a 86 dd 60 00	xOC ..
Ethernet II, Src: VantivaUSA_81:74:5a (44:1c:12:81:74:5a), Dst: Apple_98:d9:27 (78:4	0010	00 00 00 41 11 3a 20 01 05 58 fe ed 00 00 00 00	... A
Internet Protocol Version 6, Src: 2601:558:feed::1, Dst: 2601:193:8302:4620:215c:f5a	0020	00 00 00 00 00 01 26 01 01 93 83 02 46 20 21 5c
User Datagram Protocol, Src Port: 53, Dst Port: 64430	0030	f5 ae 8b 40 a2 7a 00 35 fb ae 00 41 d4 51 92 0d	...@
Domain Name System (response)	0040	81 80 00 01 00 01 00 00 00 00 07 79 6f 75 74 75
Transaction ID: 0x920d	0050	62 65 03 63 6f 6d 00 00 1c 00 01 c0 0c 00 1c 00	be:co
Flags: 0x8180 Standard query response, No error	0060	01 00 00 00 c9 00 10 26 07 f8 b0 40 06 08 15 00
Questions: 1	0070	00 00 00 00 00 20 0e
Answer RRs: 1			
Authority RRs: 0			
Additional RRs: 0			
Queries			
Answers			
youtube.com: type AAAA, class IN, addr 2607:f8b0:4006:815::200e			

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

Time	No.	Source	Destination	Protocol	Length	Info
3.814364	19	2601:193:8302:4620::...	2601:558:feed::1	DNS	91	Standard query 0x4667 A youtube.com
3.814489	20	2601:193:8302:4620::...	2601:558:feed::1	DNS	91	Standard query 0x920d AAAA youtube.com
3.819370	21	2601:193:8302:4620::...	2601:558:feed::1	DNS	95	Standard query 0x7884 A www.youtube.com
3.819905	22	2601:193:8302:4620::...	2601:558:feed::1	DNS	95	Standard query 0x04fe AAAA www.youtube.com
3.946846	23	2601:558:feed::1	2601:193:8302:4620::...	DNS	107	Standard query response 0x4667 A youtube.com A 172.217.10.1
3.953852	24	2601:558:feed::1	2601:193:8302:4620::...	DNS	241	Standard query response 0x04fe AAAA www.youtube.com CNAME y
3.954763	25	2601:193:8302:4620::...	2601:558:feed::1	DNS	103	Standard query 0x7884 A youtube-ui.l.google.com
3.955402	26	2601:558:feed::1	2601:193:8302:4620::...	DNS	337	Standard query response 0x7884 A www.youtube.com CNAME yout
3.955405	27	2601:558:feed::1	2601:193:8302:4620::...	DNS	119	Standard query response 0x920d AAAA youtube.com AAAA 2607:f
4.099922	30	2601:558:feed::1	2601:193:8302:4620::...	DNS	311	Standard query response 0x7884 A youtube-ui.l.google.com A

User Datagram Protocol, Src Port: 53, Dst Port: 57174	0000	78 4f 43 98 d9 27 44 1c 12 81 74 5a 86 dd 60 00	xOC ..
Domain Name System (response)	0010	00 00 00 bb 11 3a 20 01 05 58 fe ed 00 00 00 00
Transaction ID: 0x04fe	0020	00 00 00 00 00 01 26 01 01 93 83 02 46 20 21 5c
Flags: 0x8180 Standard query response, No error	0030	f5 ae 8b 40 a2 7a 00 35 df 56 00 bb 8c 03 04 fe	...@
Questions: 1	0040	81 80 00 01 00 05 00 00 00 00 03 77 77 77 07 79
Answer RRs: 5	0050	6f 75 74 75 62 65 03 63 6f 6d 00 00 1c 00 01 c0	outub
Authority RRs: 0	0060	0c 00 05 00 01 00 00 e0 b2 00 16 0a 79 6f 75 74
Additional RRs: 0	0070	75 62 65 2d 75 69 01 6c 06 67 6f 6f 67 6c 65 c0	ube-u
Queries	0080	18 c0 2d 00 1c 00 01 00 00 00 c1 00 10 26 07 f8
Answers	0090	b0 40 06 08 06 00 00 00 00 00 20 0e c0 2d 00	...@
www.youtube.com: type CNAME, class IN, cname youtube-ui.l.google.com	00a0	1c 00 01 00 00 00 c1 00 10 26 07 f8 b0 40 06 08
youtube-ui.l.google.com: type AAAA, class IN, addr 2607:f8b0:4006:806::200e	00b0	1a 00 00 00 00 00 20 0e c0 2d 00 1c 00 01 00
youtube-ui.l.google.com: type AAAA, class IN, addr 2607:f8b0:4006:81a::200e	00c0	00 00 c1 00 10 26 07 f8 b0 40 06 08 1b 00 00 00
youtube-ui.l.google.com: type AAAA, class IN, addr 2607:f8b0:4006:81b::200e	00d0	00 00 00 20 0e c0 2d 00 1c 00 01 00 00 00 c1 00
youtube-ui.l.google.com: type AAAA, class IN, addr 2607:f8b0:4006:807::200e	00e0	10 26 07 f8 b0 40 06 08 07 00 00 00 00 00 20	...&

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.