COMPUTER NETWORKS ASSIGNMENT - 1

Prateek Takshak(23110256)

Task1:

I used **make** to build up client.cpp and server.cpp

```
rachit@MEHTA:~/cn$ make
g++ -Wall -std=c++17 -o server.out server.cpp
g++ -Wall -std=c++17 -o client.out client.cpp
```

The server with the help of ./server.out is running, and a port is opened waiting for packets

```
rachit@MEHTA:~/cn$ ./server.out
Server listening on port 8080
```

The Client extracted the DNS queries.
Modified them to custom queries. With modifies headers.

```
rachit@MEHTA:~/cnass/cn_ass_pr$ ./client.out
17103600 | netflix.com. -> 192.168.1.6
17103601 | linkedin.com. -> 192.168.1.7
17103602 | example.com. -> 192.168.1.8
17103603 | google.com. -> 192.168.1.9
17103604 | facebook.com. -> 192.168.1.10
17103605 | amazon.com. -> 192.168.1.6
```

Listens for incoming packets over a UDP socket.
Receives packets containing a custom 8-byte header followed by a DNS payload.
Applies timestamp + ID rules to map each query to an IP address from a predefined pool.
Extracts the domain name and displays it along with the resolved IP address.

```
rachit@MEHTA:~/cnass/cn_ass_pr$ ./server.out
Server listening on port 8080
17103600 | netflix.com. -> 192.168.1.6
17103601 | linkedin.com. -> 192.168.1.7
17103602 | example.com. -> 192.168.1.8
17103603 | google.com. -> 192.168.1.9
17103604 | facebook.com. -> 192.168.1.10
17103605 | amazon.com. -> 192.168.1.6
```

Resultant:

result_table.txt

	CustomHeader	Domain	ResolvedIP
2	17103600	netflix.com.	192.168.1.6
3	17103601	linkedin.com.	192.168.1.7
4	17103602	example.com.	192.168.1.8
5	17103603	google.com.	192.168.1.9
6	17103604	facebook.com.	192.168.1.10
7	17103605	amazon.com.	192.168.1.6

Explanation

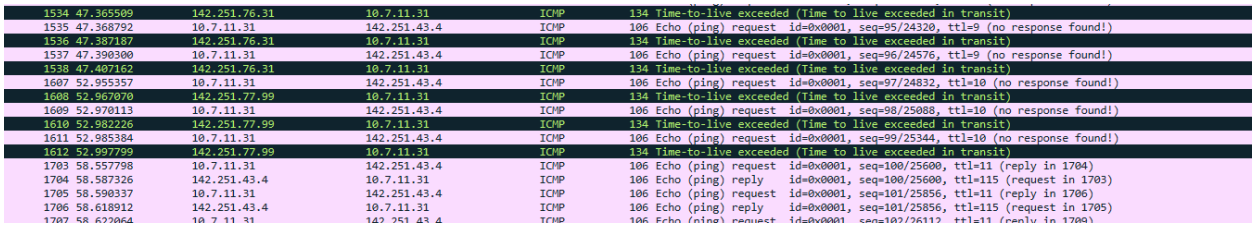
For the DNS query `netflix.com.` with header value **17103600**, the hour field is **17**, which falls in the **Evening slot (16:00–19:59)** where the IP pool start index is **6**. The session ID is **00**, giving `00 % 5 = 0`. Adding this to the pool start index results in **6**, which corresponds to the IP **192.168.1.6** as shown in the table. This confirms that the header parsing and rule-based allocation correctly resolve the query to the expected IP.

Task2:

1. Protocols used by default for tracert and traceroute:

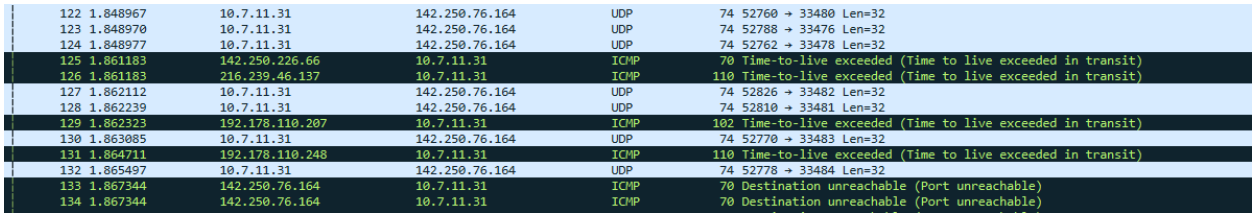
- **Windows tracert** works with **ICMP Echo Request** packets.
    - **Outgoing:** ICMP Echo Request is sent.
    - **Incoming (from routers):** ICMP Time Exceeded message.
    - **Incoming (from final destination):** ICMP Echo Reply.
- **Linux traceroute** works with **UDP packets** sent to very high port numbers (33434 and above).
    - **Outgoing:** UDP probe packet is sent.
    - **Incoming (from routers):** ICMP Time Exceeded message.
    - **Incoming (from final destination):** ICMP Destination Unreachable – Port Unreachable.

Using tracert in Windows



1534	47.365509	142.251.76.31	10.7.11.31	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
1535	47.368792	10.7.11.31	142.251.43.4	ICMP	106 Echo (ping) request id=0x0001, seq=95/24320, ttl=9 (no response found!)
1536	47.387187	142.251.76.31	10.7.11.31	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
1537	47.390300	10.7.11.31	142.251.43.4	ICMP	106 Echo (ping) request id=0x0001, seq=96/24576, ttl=9 (no response found!)
1538	47.407162	142.251.76.31	10.7.11.31	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
1607	52.955357	10.7.11.31	142.251.43.4	ICMP	106 Echo (ping) request id=0x0001, seq=97/24832, ttl=10 (no response found!)
1608	52.967070	142.251.77.99	10.7.11.31	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
1609	52.970113	10.7.11.31	142.251.43.4	ICMP	106 Echo (ping) request id=0x0001, seq=98/25088, ttl=10 (no response found!)
1610	52.982226	142.251.77.99	10.7.11.31	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
1611	52.985384	10.7.11.31	142.251.43.4	ICMP	106 Echo (ping) request id=0x0001, seq=99/25344, ttl=10 (no response found!)
1612	52.997799	142.251.77.99	10.7.11.31	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
1703	58.557798	10.7.11.31	142.251.43.4	ICMP	106 Echo (ping) request id=0x0001, seq=100/25600, ttl=11 (reply in 1704)
1704	58.587326	142.251.43.4	10.7.11.31	ICMP	106 Echo (ping) reply id=0x0001, seq=100/25600, ttl=115 (request in 1703)
1705	58.590337	10.7.11.31	142.251.43.4	ICMP	106 Echo (ping) request id=0x0001, seq=101/25856, ttl=11 (reply in 1706)
1706	58.618912	142.251.43.4	10.7.11.31	ICMP	106 Echo (ping) reply id=0x0001, seq=101/25856, ttl=115 (request in 1705)

Using traceroute in ubuntu

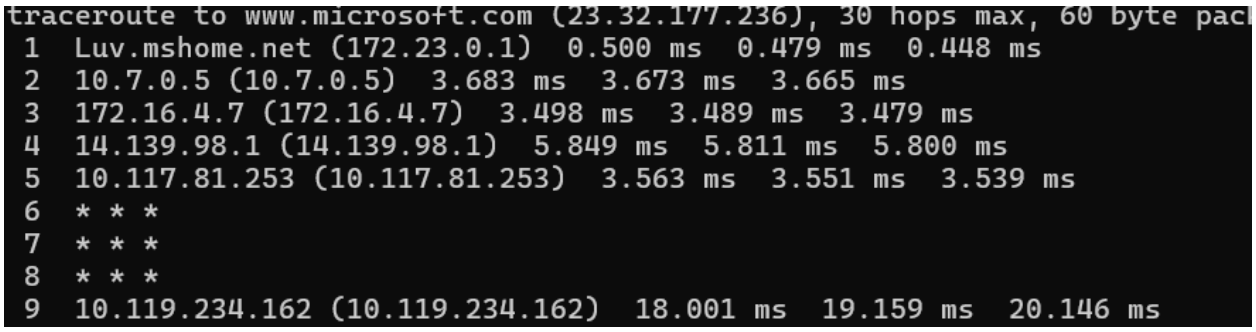


122	1.848967	10.7.11.31	142.250.76.164	UDP	74 52760 → 33480 Len=32
123	1.848970	10.7.11.31	142.250.76.164	UDP	74 52788 → 33476 Len=32
124	1.848977	10.7.11.31	142.250.76.164	UDP	74 52762 → 33478 Len=32
125	1.861183	142.250.226.66	10.7.11.31	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
126	1.861183	216.239.46.137	10.7.11.31	ICMP	110 Time-to-live exceeded (Time to live exceeded in transit)
127	1.862112	10.7.11.31	142.250.76.164	UDP	74 52826 → 33482 Len=32
128	1.862239	10.7.11.31	142.250.76.164	UDP	74 52810 → 33481 Len=32
129	1.862323	192.178.110.207	10.7.11.31	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
130	1.863085	10.7.11.31	142.250.76.164	UDP	74 52770 → 33483 Len=32
131	1.864711	192.178.110.248	10.7.11.31	ICMP	110 Time-to-live exceeded (Time to live exceeded in transit)
132	1.865497	10.7.11.31	142.250.76.164	UDP	74 52778 → 33484 Len=32
133	1.867344	142.250.76.164	10.7.11.31	ICMP	70 Destination unreachable (Port unreachable)
134	1.867344	142.250.76.164	10.7.11.31	ICMP	70 Destination unreachable (Port unreachable)

2. Why some hops show ***

When you see *** in the traceroute output, it means the router didn't reply. This can happen because:

- A router or firewall is set up to **block ICMP replies**.
- A router is **too busy forwarding traffic** and chooses not to respond to traceroute probes.



```
traceroute to www.microsoft.com (23.32.177.236), 30 hops max, 60 byte pacl
 1  Luv.mshome.net (172.23.0.1)  0.500 ms  0.479 ms  0.448 ms
 2  10.7.0.5 (10.7.0.5)  3.683 ms  3.673 ms  3.665 ms
 3  172.16.4.7 (172.16.4.7)  3.498 ms  3.489 ms  3.479 ms
 4  14.139.98.1 (14.139.98.1)  5.849 ms  5.811 ms  5.800 ms
 5  10.117.81.253 (10.117.81.253)  3.563 ms  3.551 ms  3.539 ms
 6  * * *
 7  * * *
 8  * * *
 9  10.119.234.162 (10.119.234.162)  18.001 ms  19.159 ms  20.146 ms
```

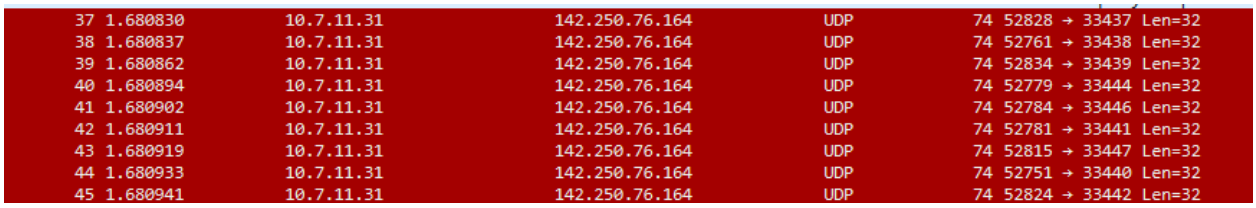
3. Changing field in Linux traceroute probes

In Linux traceroute, the **UDP destination port number** changes for each probe.

For example:

- 1st probe → port 33437
- 2nd probe → port 33438
- 3rd probe → port 33439

This way, traceroute can keep track of which reply belongs to which probe.

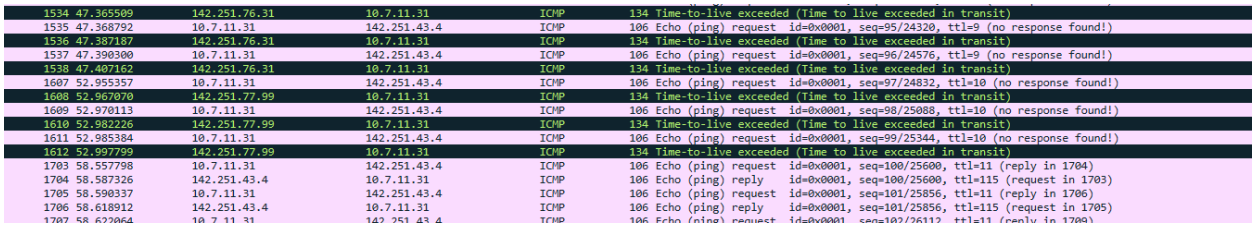


---

4. Final hop vs. intermediate hop response

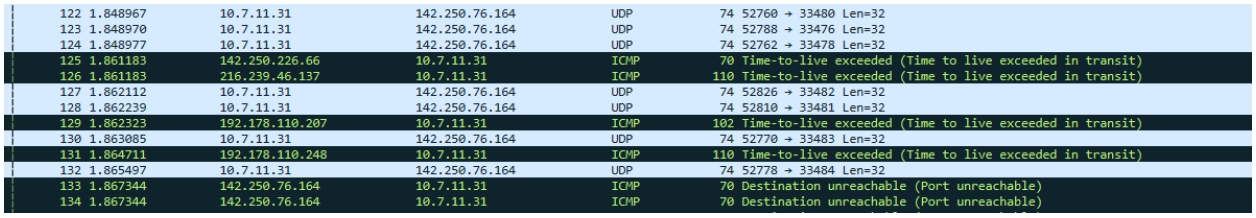
- **Intermediate hop:**
  - **Outgoing:** Probe packet with TTL = N.
  - **Incoming:** ICMP Time Exceeded (router drops packet when TTL = 0).
- **Final hop (destination):**
  - **Windows tracert:** Gets an ICMP Echo Reply → shows the host is reached.
  - **Linux traceroute:** Gets an ICMP Destination Unreachable – Port Unreachable → means the packet reached the host, but no service is listening on that high port.

For windows



1534	47.365509	142.251.76.31	10.7.11.31	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
1535	47.368792	10.7.11.31	142.251.43.4	ICMP	106 Echo (ping) request id=0x0001, seq=95/24320, ttl=9 (no response found!)
1536	47.387187	142.251.76.31	10.7.11.31	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
1537	47.390300	10.7.11.31	142.251.43.4	ICMP	106 Echo (ping) request id=0x0001, seq=96/24576, ttl=9 (no response found!)
1538	47.407162	142.251.76.31	10.7.11.31	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
1607	52.955357	10.7.11.31	142.251.43.4	ICMP	106 Echo (ping) request id=0x0001, seq=97/24832, ttl=10 (no response found!)
1608	52.967070	142.251.77.99	10.7.11.31	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
1609	52.970113	10.7.11.31	142.251.43.4	ICMP	106 Echo (ping) request id=0x0001, seq=98/25088, ttl=10 (no response found!)
1610	52.982226	142.251.77.99	10.7.11.31	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
1611	52.985384	10.7.11.31	142.251.43.4	ICMP	106 Echo (ping) request id=0x0001, seq=99/25344, ttl=10 (no response found!)
1612	52.997799	142.251.77.99	10.7.11.31	ICMP	134 Time-to-live exceeded (Time to live exceeded in transit)
1703	58.557798	10.7.11.31	142.251.43.4	ICMP	106 Echo (ping) request id=0x0001, seq=100/25600, ttl=11 (reply in 1704)
1704	58.587326	142.251.43.4	10.7.11.31	ICMP	106 Echo (ping) reply id=0x0001, seq=100/25600, ttl=115 (request in 1703)
1705	58.590337	10.7.11.31	142.251.43.4	ICMP	106 Echo (ping) request id=0x0001, seq=101/25856, ttl=11 (reply in 1706)
1706	58.618912	142.251.43.4	10.7.11.31	ICMP	106 Echo (ping) reply id=0x0001, seq=101/25856, ttl=115 (request in 1705)
1707	58.622064	10.7.11.31	142.251.43.4	ICMP	106 Echo (ping) request id=0x0001, seq=102/26112, ttl=11 (reply in 1709)

For ubuntu



122	1.848967	10.7.11.31	142.250.76.164	UDP	74 52760 → 33480 Len=32
123	1.848970	10.7.11.31	142.250.76.164	UDP	74 52788 → 33476 Len=32
124	1.848977	10.7.11.31	142.250.76.164	UDP	74 52762 → 33478 Len=32
125	1.861183	142.250.226.66	10.7.11.31	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
126	1.861183	216.239.46.137	10.7.11.31	ICMP	110 Time-to-live exceeded (Time to live exceeded in transit)
127	1.862112	10.7.11.31	142.250.76.164	UDP	74 52826 → 33482 Len=32
128	1.862239	10.7.11.31	142.250.76.164	UDP	74 52810 → 33481 Len=32
129	1.862323	192.178.110.207	10.7.11.31	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
130	1.863085	10.7.11.31	142.250.76.164	UDP	74 52770 → 33483 Len=32
131	1.864711	192.178.110.248	10.7.11.31	ICMP	110 Time-to-live exceeded (Time to live exceeded in transit)
132	1.865497	10.7.11.31	142.250.76.164	UDP	74 52778 → 33484 Len=32
133	1.867344	142.250.76.164	10.7.11.31	ICMP	70 Destination unreachable (Port unreachable)
134	1.867344	142.250.76.164	10.7.11.31	ICMP	70 Destination unreachable (Port unreachable)

5. Effect of firewall blocking UDP but allowing ICMP

- **Linux traceroute:** Will fail. Since UDP is blocked, the probes never reach the routers/destination == output shows only ***.
- **Windows tracert:** Works fine. ICMP is allowed, so you still get normal hop-by-hop replies.

In this we using same server for both the task.

In case of ubuntu:

```
luvag@Luv:~$ traceroute www.microsoft.com
traceroute to www.microsoft.com (23.32.177.236), 30 hops max, 60 byte packets
 1  Luv.mshome.net (172.23.0.1)  0.500 ms  0.479 ms  0.448 ms
 2  10.7.0.5 (10.7.0.5)  3.683 ms  3.673 ms  3.665 ms
 3  172.16.4.7 (172.16.4.7)  3.498 ms  3.489 ms  3.479 ms
 4  14.139.98.1 (14.139.98.1)  5.849 ms  5.811 ms  5.800 ms
 5  10.117.81.253 (10.117.81.253)  3.563 ms  3.551 ms  3.539 ms
 6  * * *
 7  * * *
 8  * * *
 9  10.119.234.162 (10.119.234.162)  18.001 ms  19.159 ms  20.146 ms
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

In case of windows:

```
C:\Users\luvag>tracert www.microsoft.com

Tracing route to e13678.dscb.akamaiedge.net [23.32.177.236]
over a maximum of 30 hops:

  1     2 ms     2 ms     2 ms  10.7.0.5
  2     2 ms     2 ms     3 ms  172.16.4.7
  3     7 ms     4 ms     4 ms  14.139.98.1
  4     4 ms     2 ms     2 ms  10.117.81.253
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7     *        *        *     Request timed out.
  8    22 ms    18 ms    19 ms  10.119.234.162
  9    15 ms    16 ms    15 ms  a23-32-177-236.deploy.static.akamaitechnologies.com [23.32.177.236]

Trace complete.
```