Computer Network

Assignment 1

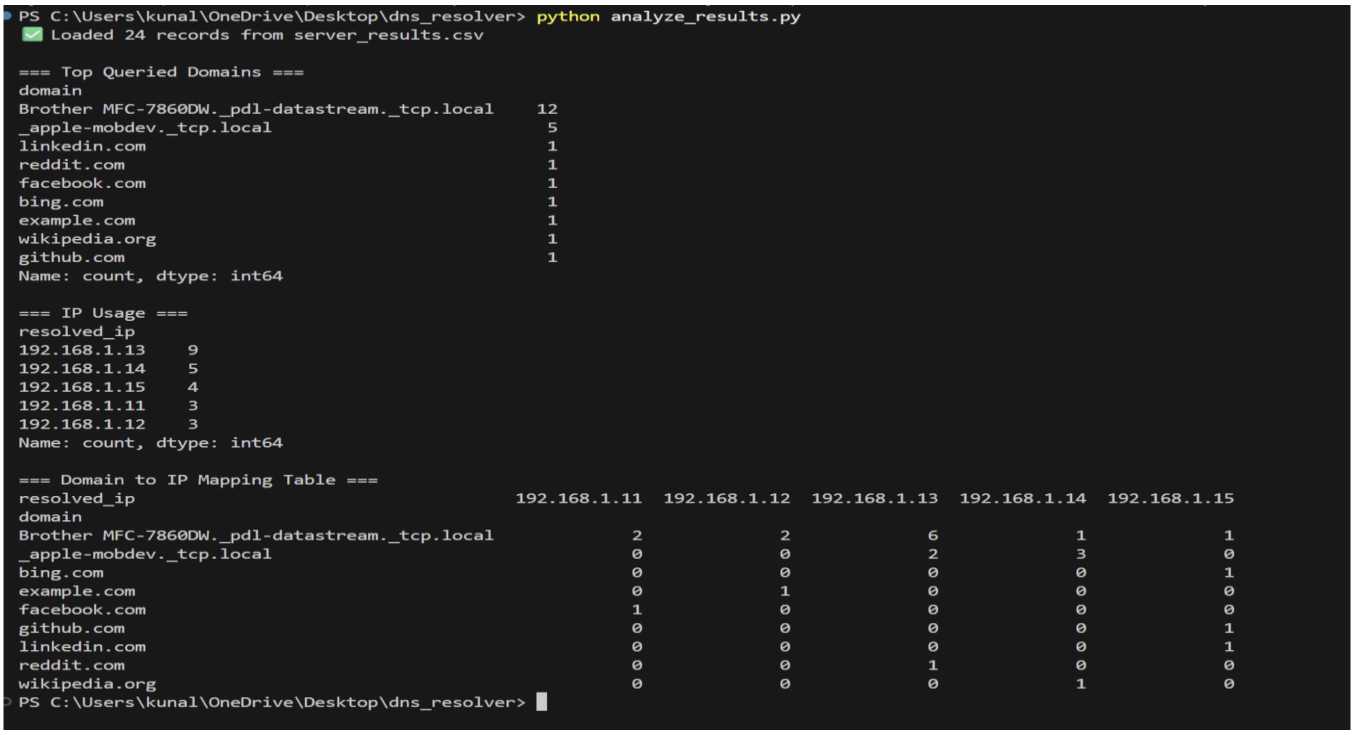
Kunal Singh Rathore_23110181

Gurudayal Meena_23110125

Task 1 : Table

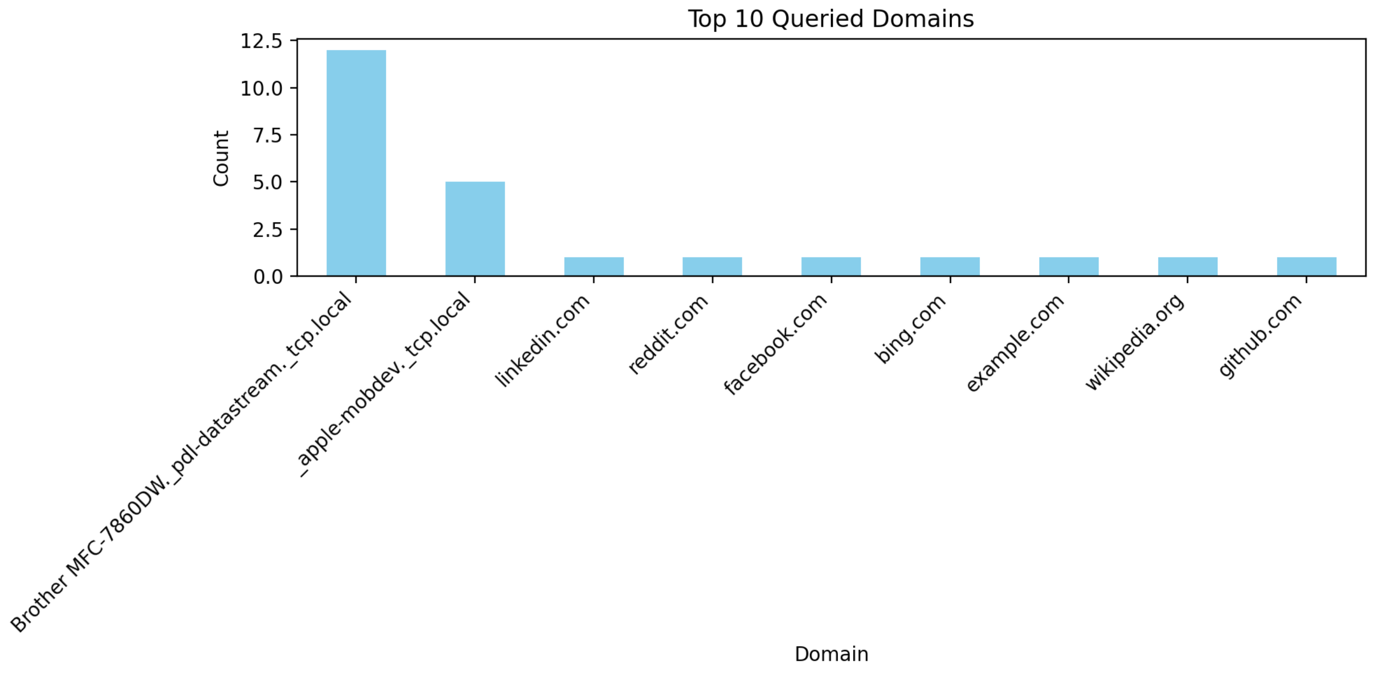
Custom header value (HHMMSSID)	Domain name	Resolved IP address
22524537	_apple-mobdev._tcp.local	192.168.1.13
22524538	_apple-mobdev._tcp.local	192.168.1.14
22524579	linkedin.com	192.168.1.15
22524502	reddit.com	192.168.1.13
22524530	facebook.com	192.168.1.11
22524577	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.13
22524532	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.13
22524619	bing.com	192.168.1.15
22524673	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.14
22524671	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.12
22524761	example.com	192.168.1.12
22524798	_apple-mobdev._tcp.local	192.168.1.14
22524742	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.13
22524745	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.11
22524783	wikipedia.org	192.168.1.14
22524751	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.12
22524724	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.15
22524842	_apple-mobdev._tcp.local	192.168.1.13
22524843	_apple-mobdev._tcp.local	192.168.1.14
22524870	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.11
22524862	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.13
22524849	github.com	192.168.1.15
22524827	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.13
22524897	Brother MFC-7860DW._pdl-datastream._tcp.local	192.168.1.13

The DNS analysis of 24 records shows that most queries were for **local services** (printer and Apple discovery), with 192.168.1.13 being the most resolved IP. Local domains mapped to multiple internal IPs, while external sites (GitHub, Bing, LinkedIn) appeared only once, indicating mainly **local network activity**.

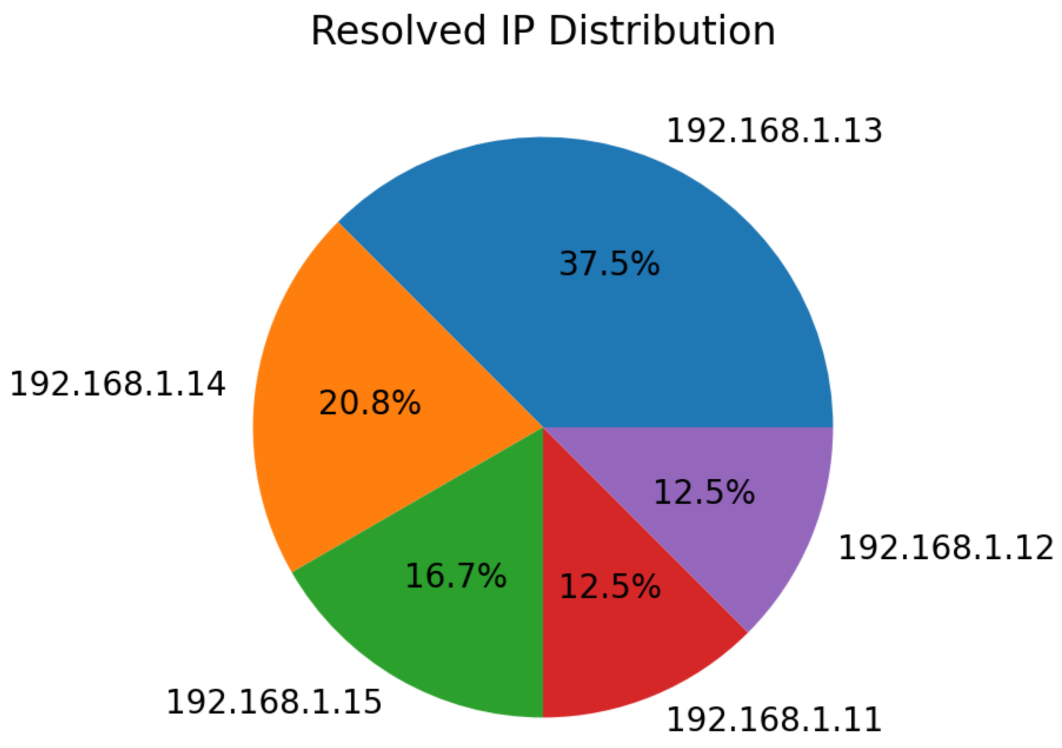


```
PS C:\Users\kunal\OneDrive\Desktop\dns_resolver> python analyze_results.py
✅ Loaded 24 records from server_results.csv

=== Top Queried Domains ===
domain
Brother MFC-7860DW._pdl-datastream._tcp.local    12
_apple-mobdev._tcp.local                          5
linkedin.com                                      1
reddit.com                                        1
facebook.com                                      1
bing.com                                          1
example.com                                       1
wikipedia.org                                     1
github.com                                        1
Name: count, dtype: int64

=== IP Usage ===
resolved_ip
192.168.1.13    9
192.168.1.14    5
192.168.1.15    4
192.168.1.11    3
192.168.1.12    3
Name: count, dtype: int64

=== Domain to IP Mapping Table ===
resolved_ip                                    192.168.1.11  192.168.1.12  192.168.1.13  192.168.1.14  192.168.1.15
domain
Brother MFC-7860DW._pdl-datastream._tcp.local             2             2             6             1             1
_apple-mobdev._tcp.local                                  0             0             2             3             0
bing.com                                                  0             0             0             0             1
example.com                                               0             1             0             0             0
facebook.com                                              1             0             0             0             0
github.com                                                0             0             0             0             1
linkedin.com                                              0             0             0             0             1
reddit.com                                                0             0             1             0             0
wikipedia.org                                             0             0             0             1             0
PS C:\Users\kunal\OneDrive\Desktop\dns_resolver> █
```

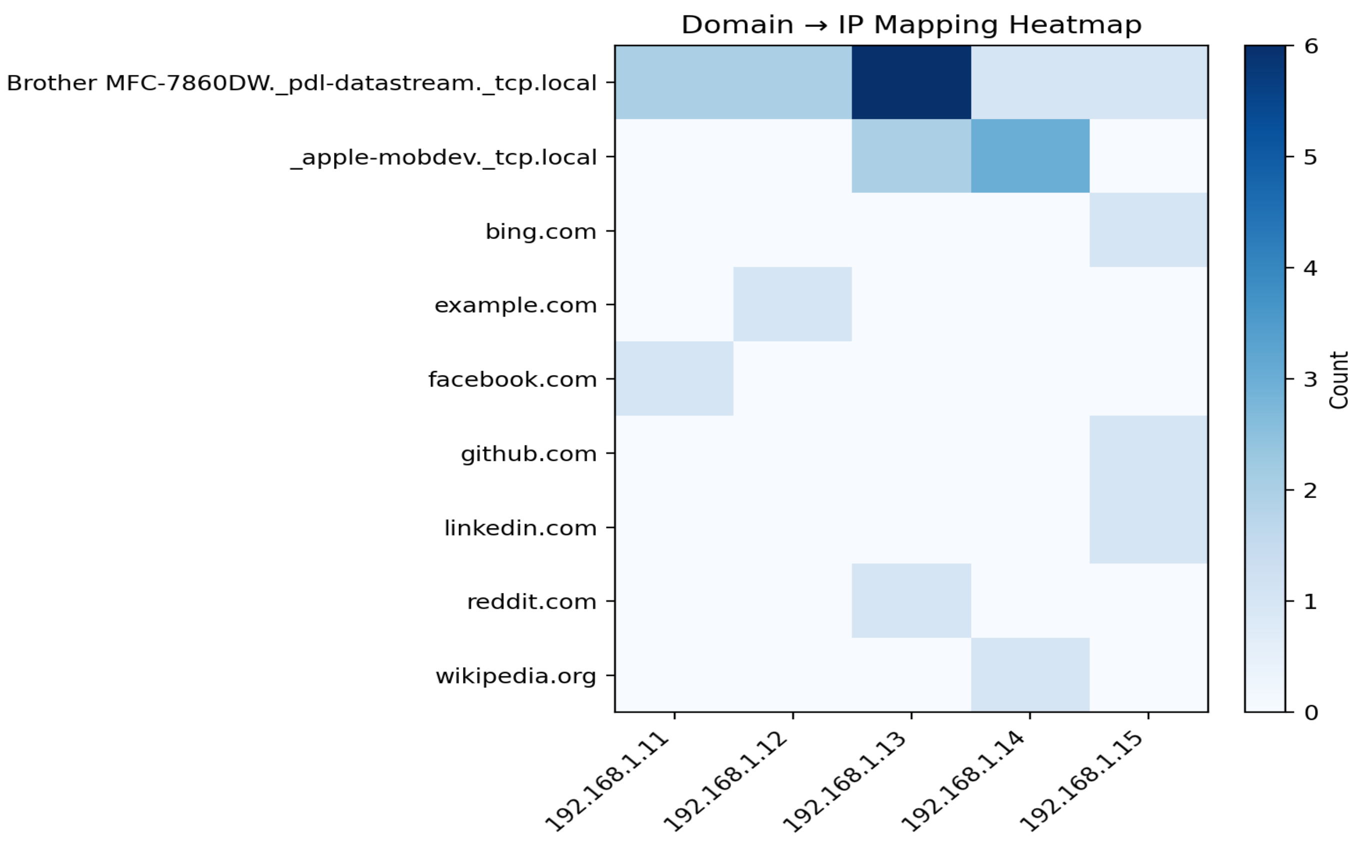
The bar chart shows the top 10 most frequently queried domains, highlighting which websites are accessed the most.



The pie chart displays the distribution of resolved IP addresses, indicating which servers or devices handle the majority of requests. The domain-to-IP mapping table and heatmap visualize how each domain resolves to different IPs, with the heatmap using color intensity to show frequency.



Together, these graphs provide insights into network traffic patterns, popular domains, and load distribution across IPs. They help identify high-traffic resources and potential anomalies in DNS resolution.



Task 2

1.What protocol does Windows tracert use by default, and what protocol does Linux traceroute use by default?

Answer: Windows tracert uses **ICMP Echo Request** packets by default. These packets behave like ping requests, and each router along the path responds with an ICMP Time Exceeded message until the destination replies with an Echo Reply. In contrast, Linux traceroute uses **UDP packets** with incrementing port numbers, and the destination replies with an ICMP Port Unreachable message, allowing the path to be mapped.

2. Some hops in your traceroute output may show * * *. Provide at least two reasons why a router might not reply.

Answer: A hop may not respond if a router or firewall is **configured to block ICMP Time Exceeded messages** for security reasons. Another possibility is **packet loss or congestion**, where the probe never reaches the router or the reply never returns. In some cases, the router may prioritize forwarding traffic over responding to diagnostic packets, resulting in no visible reply.

Two reason for showing *** in hop 9:

1.) The router at hop 9 is likely configured to not send ICMP "Time-to-live exceeded" messages. This is a common security practice to hide network topology. The router simply drops the packet without notifying the sender.
2.) The ICMP reply from the router at hop 9 was lost on its way back to your computer due to network congestion or other issues.

3.In Linux traceroute, which field in the probe packets changes between successive probes sent to the destination?

Answer: In Linux traceroute, the **UDP destination port number** changes with each successive probe packet. This ensures that each probe can be uniquely identified and matched with the corresponding ICMP response, avoiding confusion when multiple packets are in transit.

4. At the final hop, how is the response different compared to the intermediate hop?

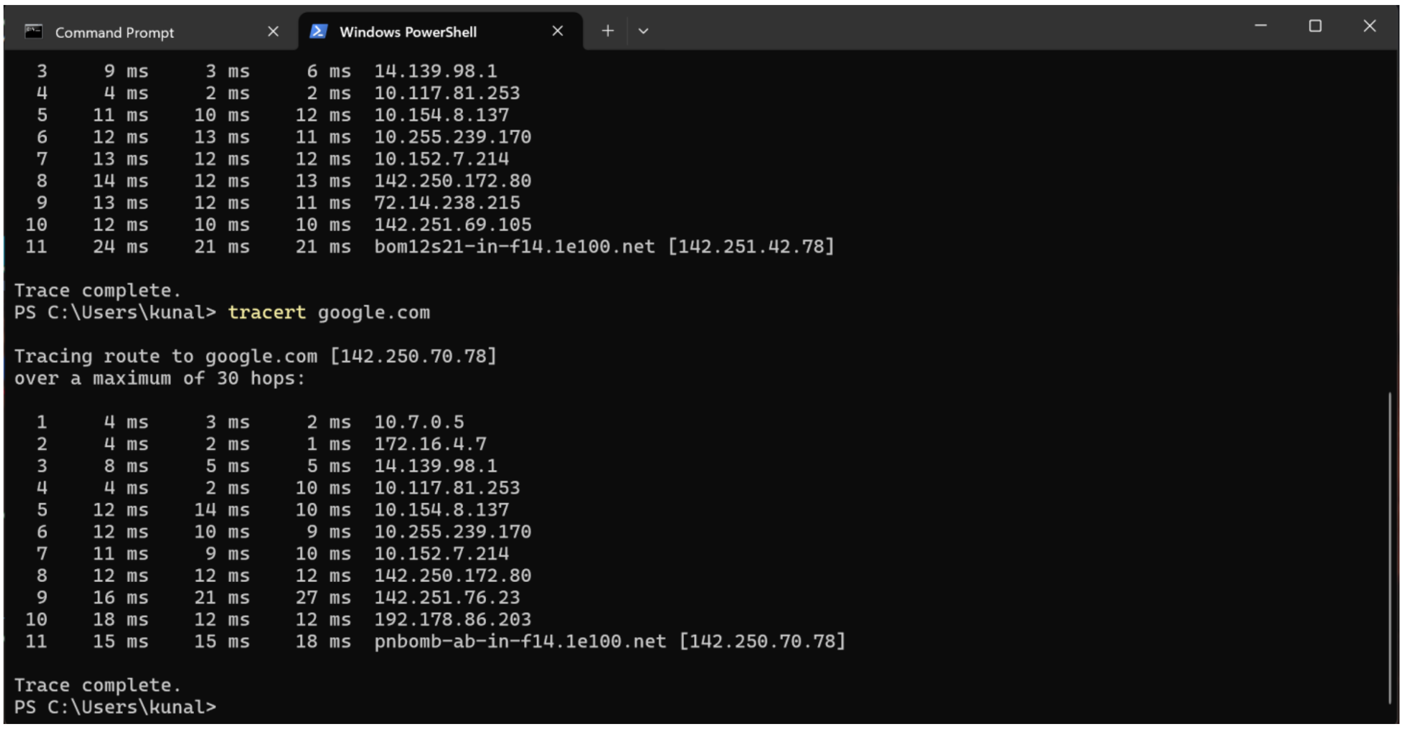
Answer: At intermediate hops, routers return an **ICMP Time Exceeded** message when the TTL reaches zero. At the final hop, however, the destination host responds differently: in Linux (UDP-based), it sends an **ICMP Port Unreachable** message because the UDP port is closed, while in Windows (ICMP-based), it replies with an **ICMP Echo Reply** since the probe is an echo request.

5.Suppose a firewall blocks UDP traffic but allows ICMP — how would this affect the results of Linux traceroute vs. Windows tracert?

Answer: If UDP is blocked, **Linux traceroute would fail** since its default probes are UDP packets, and no valid responses would be received. On the other hand, **Windows tracert would still succeed** because it sends ICMP Echo Requests, which are allowed by the firewall. This demonstrates how protocol choice directly affects traceroutes ability to work in different network environments.

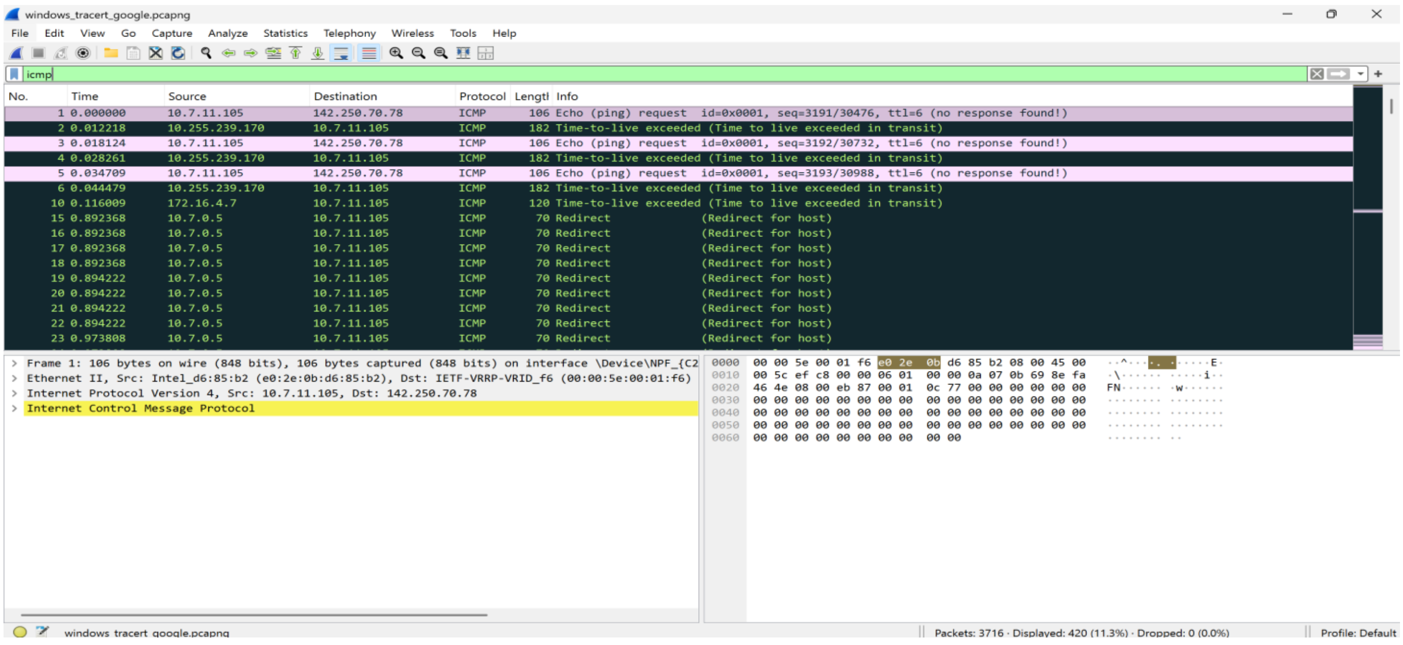
Screenshots of the Task 2 done in Windows:

This screenshot shows the command-line output of running tracert google.com in Windows PowerShell. It lists the sequence of routers (hops) that packets travel through to reach the destination, along with the round-trip time for three probes sent to each hop



```
  3     9 ms     3 ms     6 ms  14.139.98.1
  4     4 ms     2 ms     2 ms  10.117.81.253
  5    11 ms    10 ms    12 ms  10.154.8.137
  6    12 ms    13 ms    11 ms  10.255.239.170
  7    13 ms    12 ms    12 ms  10.152.7.214
  8    14 ms    12 ms    13 ms  142.250.172.80
  9    13 ms    12 ms    11 ms  72.14.238.215
 10    12 ms    10 ms    10 ms  142.251.69.105
 11    24 ms    21 ms    21 ms  bom12s21-in-f14.1e100.net [142.251.42.78]

Trace complete.
PS C:\Users\kunal> tracert google.com

Tracing route to google.com [142.250.70.78]
over a maximum of 30 hops:

  1     4 ms     3 ms     2 ms  10.7.0.5
  2     4 ms     2 ms     1 ms  172.16.4.7
  3     8 ms     5 ms     5 ms  14.139.98.1
  4     4 ms     2 ms    10 ms  10.117.81.253
  5    12 ms    14 ms    10 ms  10.154.8.137
  6    12 ms    10 ms     9 ms  10.255.239.170
  7    11 ms     9 ms    10 ms  10.152.7.214
  8    12 ms    12 ms    12 ms  142.250.172.80
  9    16 ms    21 ms    27 ms  142.251.76.23
 10    18 ms    12 ms    12 ms  192.178.86.203
 11    15 ms    15 ms    18 ms  pnbomb-ab-in-f14.1e100.net [142.250.70.78]

Trace complete.
PS C:\Users\kunal>
```

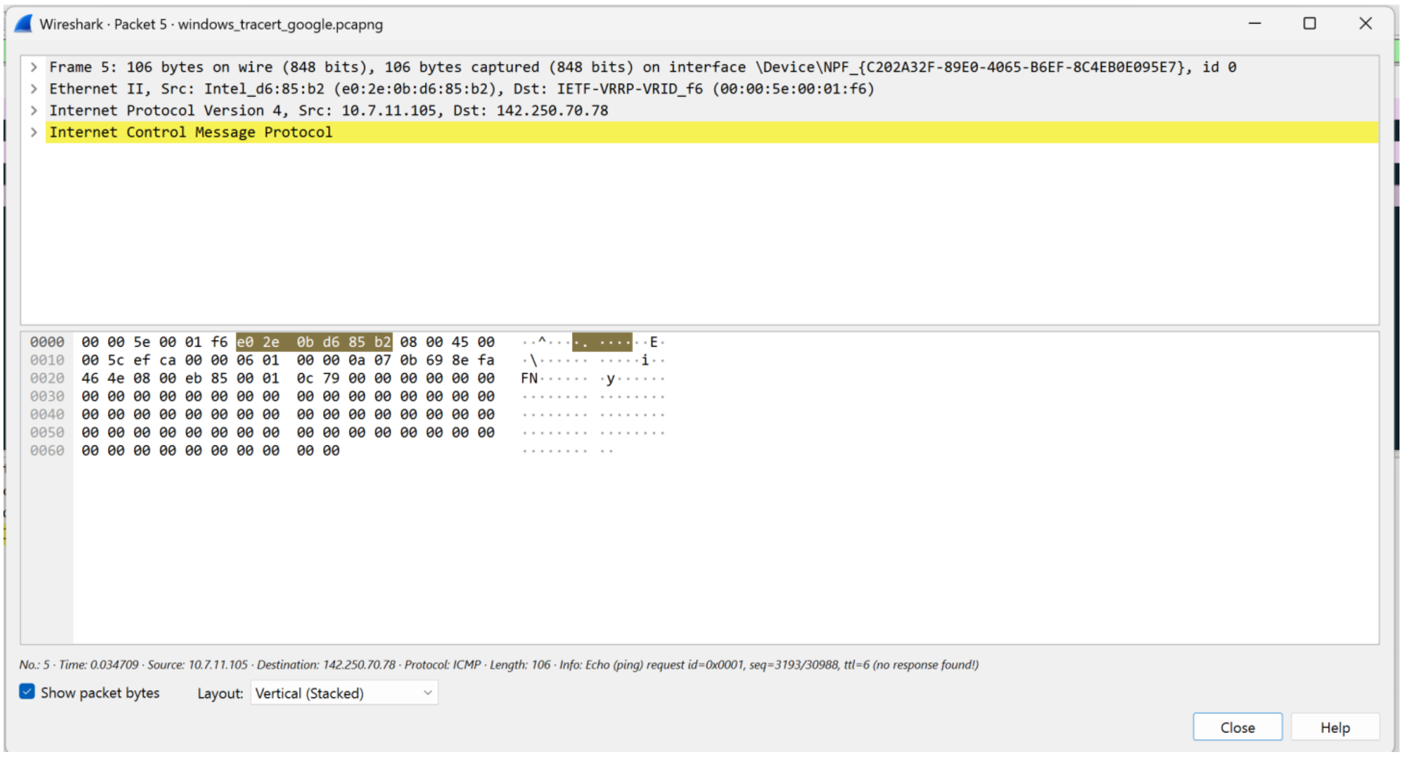
This is a Wireshark packet capture log taken during the tracert execution. It shows the specific packets being sent and received. We can clearly see the protocol is ICMP, and the key message types are "Echo (ping) request" and "Time-to-live exceeded".



windows_tracert_google.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
icmp

No.  Time      Source          Destination     Protocol Lengtl Info
   1 0.000000  10.7.11.105     142.250.70.78   ICMP     106 Echo (ping) request  id=0x0001, seq=3191/30476, ttl=6 (no response found!)
   2 0.012218  10.255.239.170  10.7.11.105     ICMP     182 Time-to-live exceeded (Time to live exceeded in transit)
   3 0.018124  10.7.11.105     142.250.70.78   ICMP     106 Echo (ping) request  id=0x0001, seq=3192/30732, ttl=6 (no response found!)
   4 0.028261  10.255.239.170  10.7.11.105     ICMP     182 Time-to-live exceeded (Time to live exceeded in transit)
   5 0.034709  10.7.11.105     142.250.70.78   ICMP     106 Echo (ping) request  id=0x0001, seq=3193/30988, ttl=6 (no response found!)
   6 0.044479  10.255.239.170  10.7.11.105     ICMP     182 Time-to-live exceeded (Time to live exceeded in transit)
  10 0.116009  172.16.4.7      10.7.11.105     ICMP     120 Time-to-live exceeded (Time to live exceeded in transit)
  15 0.892368  10.7.0.5        10.7.11.105     ICMP      70 Redirect             (Redirect for host)
  16 0.892368  10.7.0.5        10.7.11.105     ICMP      70 Redirect             (Redirect for host)
  17 0.892368  10.7.0.5        10.7.11.105     ICMP      70 Redirect             (Redirect for host)
  18 0.892368  10.7.0.5        10.7.11.105     ICMP      70 Redirect             (Redirect for host)
  19 0.894222  10.7.0.5        10.7.11.105     ICMP      70 Redirect             (Redirect for host)
  20 0.894222  10.7.0.5        10.7.11.105     ICMP      70 Redirect             (Redirect for host)
  21 0.894222  10.7.0.5        10.7.11.105     ICMP      70 Redirect             (Redirect for host)
  22 0.894222  10.7.0.5        10.7.11.105     ICMP      70 Redirect             (Redirect for host)
  23 0.973808  10.7.0.5        10.7.11.105     ICMP      70 Redirect             (Redirect for host)

> Frame 1: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface \Device\NPF_{C2
> Ethernet II, Src: Intel_d6:85:b2 (e0:2e:0b:d6:85:b2), Dst: IETF-VRRP-VRID_f6 (00:00:5e:00:01:f6)
> Internet Protocol Version 4, Src: 10.7.11.105, Dst: 142.250.70.78
> Internet Control Message Protocol

0000  00 00 5e 00 01 f6 e0 2e  0b d6 85 b2 08 00 45 00   ··^·····  ······E·
0010  00 5c ef c8 00 00 06 01  00 00 0a 07 0b 69 8e fa   ·\······  ·····i··
0020  46 4e 08 00 eb 87 00 01  0c 77 00 00 00 00 00 00   FN······  ·w······
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
0060  00 00 00 00 00 00 00 00  00 00                     ········  ··

windows_tracert_google.pcapng    Packets: 3716 · Displayed: 420 (11.3%) · Dropped: 0 (0.0%)    Profile: Default
```

This is a detailed view of a single packet (Packet 5) from the Wireshark capture. It confirms that the packet is an ICMP "Echo (ping) request" being sent from your machine (10.7.11.105) towards the destination (142.250.70.78).



```
> Frame 5: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface \Device\NPF_{C202A32F-89E0-4065-B6EF-8C4EB0E095E7}, id 0
> Ethernet II, Src: Intel_d6:85:b2 (e0:2e:0b:d6:85:b2), Dst: IETF-VRRP-VRID_f6 (00:00:5e:00:01:f6)
> Internet Protocol Version 4, Src: 10.7.11.105, Dst: 142.250.70.78
> Internet Control Message Protocol
```

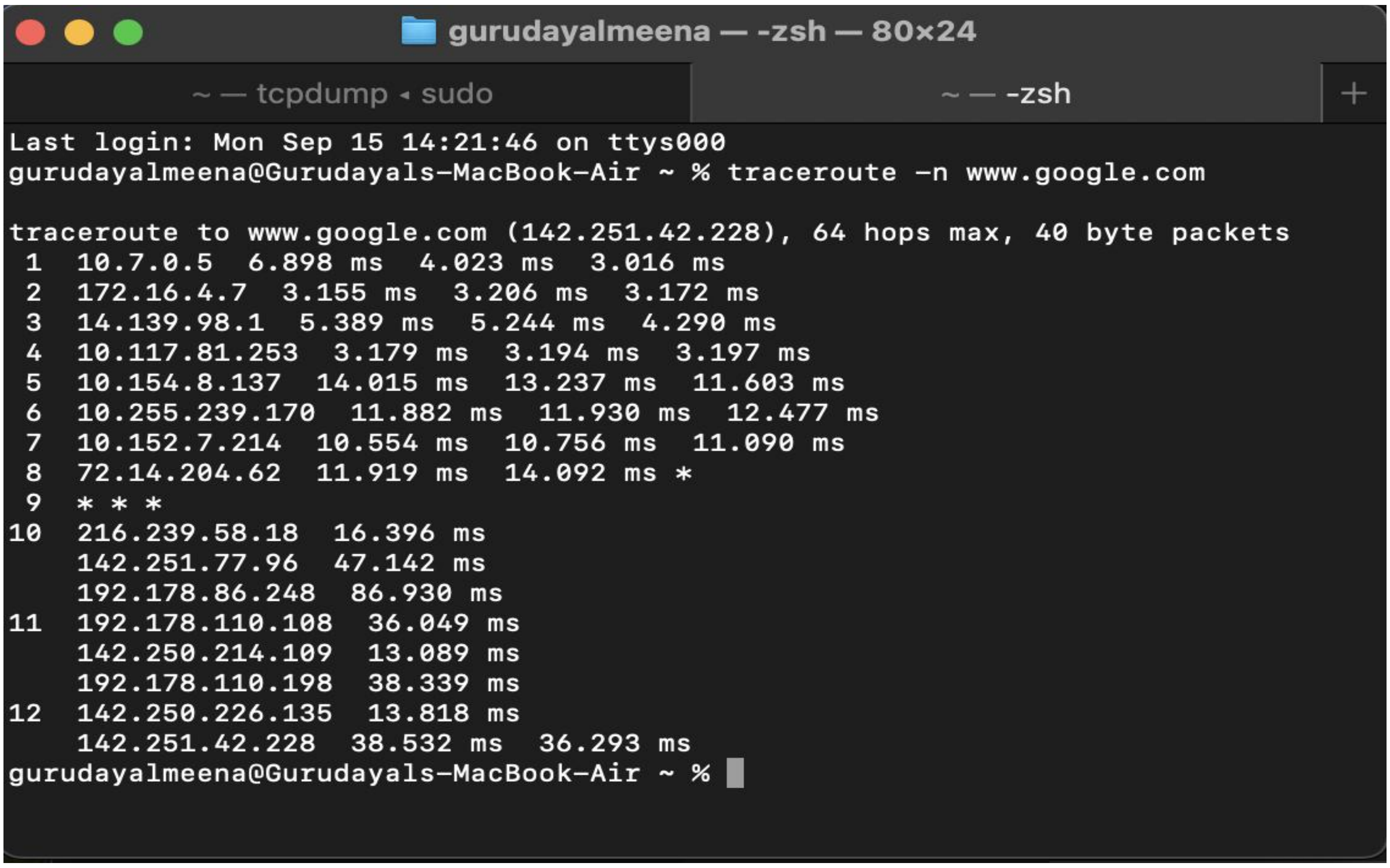
```
0000  00 00 5e 00 01 f6 e0 2e  0b d6 85 b2 08 00 45 00   ··^···.. ······E·
0010  00 5c ef ca 00 00 06 01  00 00 0a 07 0b 69 8e fa   ·\······ ·····i··
0020  46 4e 08 00 eb 85 00 01  0c 79 00 00 00 00 00 00   FN······ ·y······
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0060  00 00 00 00 00 00 00 00  00 00                     ········ ··
```

No.: 5 · Time: 0.034709 · Source: 10.7.11.105 · Destination: 142.250.70.78 · Protocol: ICMP · Length: 106 · Info: Echo (ping) request id=0x0001, seq=3193/30988, ttl=6 (no response found!)

Show packet bytes ✓   Layout: Vertical (Stacked)

Screenshots of the Task 2 done in Mac.

This shows the output of the default traceroute -n www.google.com command. The -n flag prevents DNS lookups, showing only IP addresses, which is great for analysis. Note the *** at hop 9.



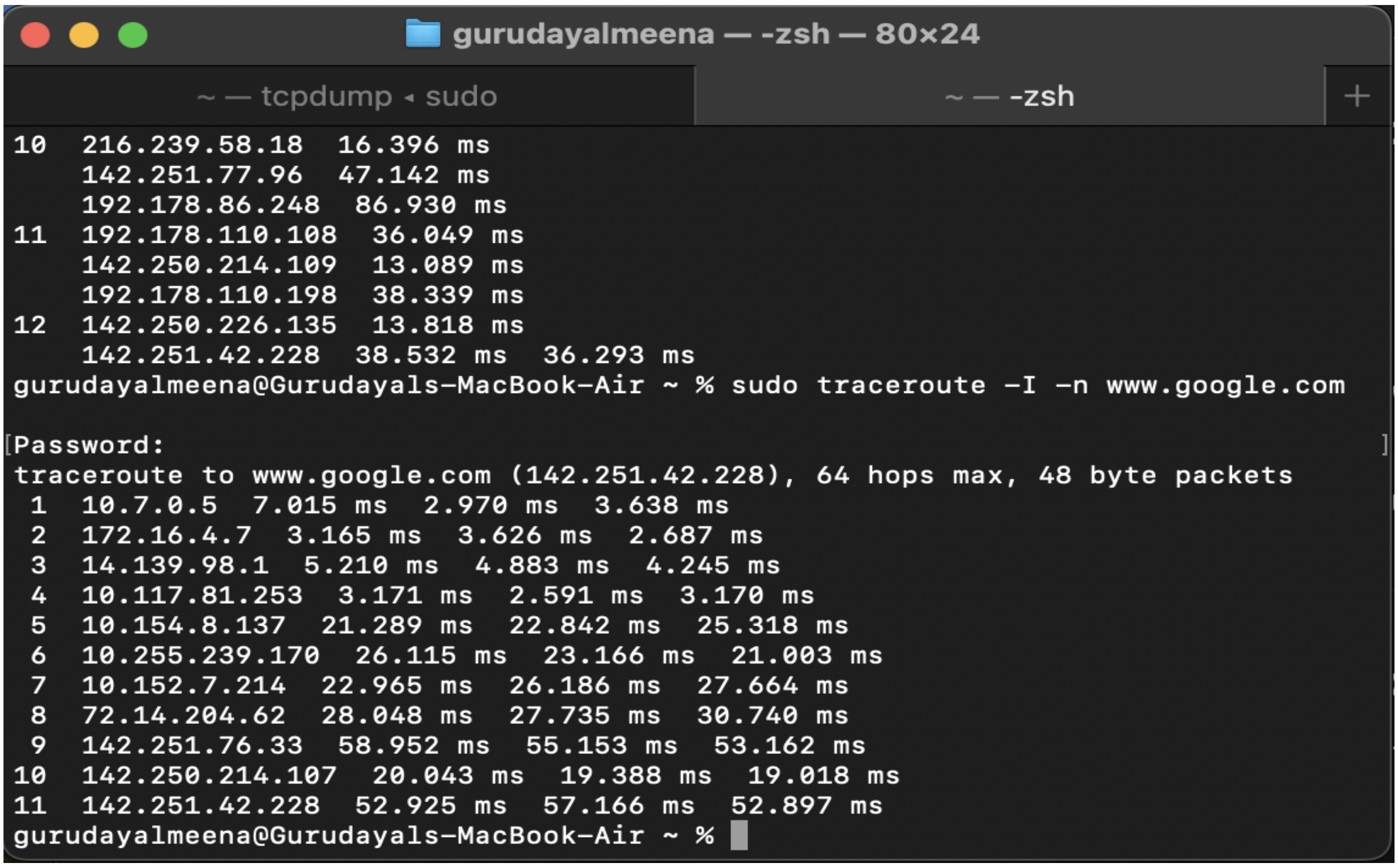
```
Last login: Mon Sep 15 14:21:46 on ttys000
gurudayalmeena@Gurudayals-MacBook-Air ~ % traceroute -n www.google.com

traceroute to www.google.com (142.251.42.228), 64 hops max, 40 byte packets
 1  10.7.0.5  6.898 ms  4.023 ms  3.016 ms
 2  172.16.4.7  3.155 ms  3.206 ms  3.172 ms
 3  14.139.98.1  5.389 ms  5.244 ms  4.290 ms
 4  10.117.81.253  3.179 ms  3.194 ms  3.197 ms
 5  10.154.8.137  14.015 ms  13.237 ms  11.603 ms
 6  10.255.239.170  11.882 ms  11.930 ms  12.477 ms
 7  10.152.7.214  10.554 ms  10.756 ms  11.090 ms
 8  72.14.204.62  11.919 ms  14.092 ms *
 9  * * *
10  216.239.58.18  16.396 ms
    142.251.77.96  47.142 ms
    192.178.86.248  86.930 ms
11  192.178.110.108  36.049 ms
    142.250.214.109  13.089 ms
    192.178.110.198  38.339 ms
12  142.250.226.135  13.818 ms
    142.251.42.228  38.532 ms  36.293 ms
gurudayalmeena@Gurudayals-MacBook-Air ~ %
```

In above image shows the three probe packets sent with a TTL of 9, no response was received within the timeout period have two reason:

3.) The router at hop 9 is likely configured to not send ICMP "Time-to-live exceeded" messages. This is a common security practice to hide network topology. The router simply drops the packet without notifying the sender.

4.) The ICMP reply from the router at hop 9 was lost on its way back to your computer due to network congestion or other issues.

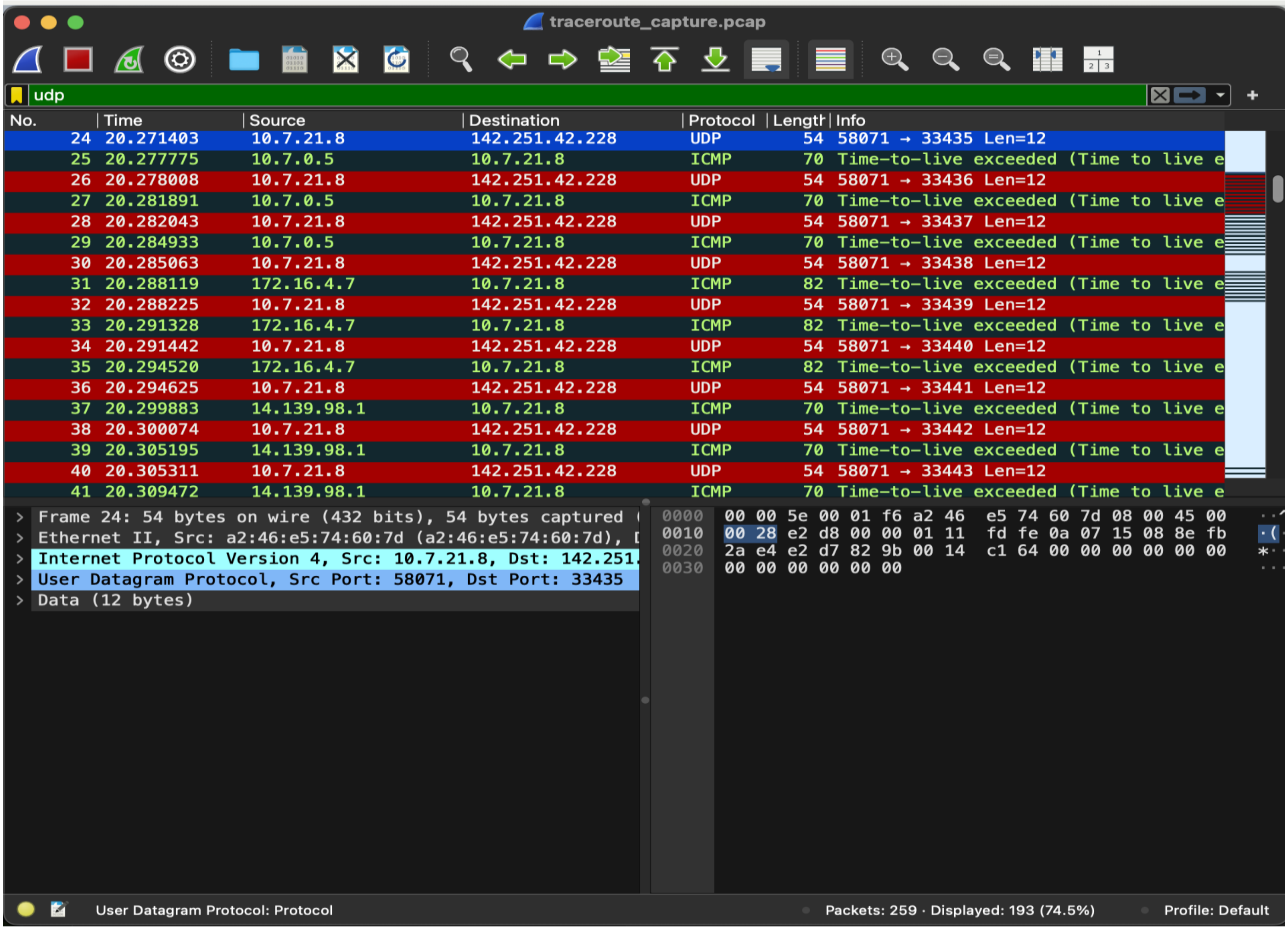
This shows the output of the default traceroute -I-n www.google.com command which give 11-hop network trace executed in **ICMP mode.**



```
10  216.239.58.18  16.396 ms
    142.251.77.96  47.142 ms
    192.178.86.248  86.930 ms
11  192.178.110.108  36.049 ms
    142.250.214.109  13.089 ms
    192.178.110.198  38.339 ms
12  142.250.226.135  13.818 ms
    142.251.42.228  38.532 ms  36.293 ms
gurudayalmeena@Gurudayals-MacBook-Air ~ % sudo traceroute -I -n www.google.com

[Password:
traceroute to www.google.com (142.251.42.228), 64 hops max, 48 byte packets
 1  10.7.0.5  7.015 ms  2.970 ms  3.638 ms
 2  172.16.4.7  3.165 ms  3.626 ms  2.687 ms
 3  14.139.98.1  5.210 ms  4.883 ms  4.245 ms
 4  10.117.81.253  3.171 ms  2.591 ms  3.170 ms
 5  10.154.8.137  21.289 ms  22.842 ms  25.318 ms
 6  10.255.239.170  26.115 ms  23.166 ms  21.003 ms
 7  10.152.7.214  22.965 ms  26.186 ms  27.664 ms
 8  72.14.204.62  28.048 ms  27.735 ms  30.740 ms
 9  142.251.76.33  58.952 ms  55.153 ms  53.162 ms
10  142.250.214.107  20.043 ms  19.388 ms  19.018 ms
11  142.251.42.228  52.925 ms  57.166 ms  52.897 ms
gurudayalmeena@Gurudayals-MacBook-Air ~ % 
```

In this screenshot the network traffic generated by a traceroute to the destination 142.251.42.228 (a Google server) from your machine (10.7.21.8). It reveals the two-protocol "conversation" that makes it work:

1. sends **UDP packets** as probes.
2. The intermediate routers on the internet reply with **ICMP packets**.



traceroute_capture.pcap

udp

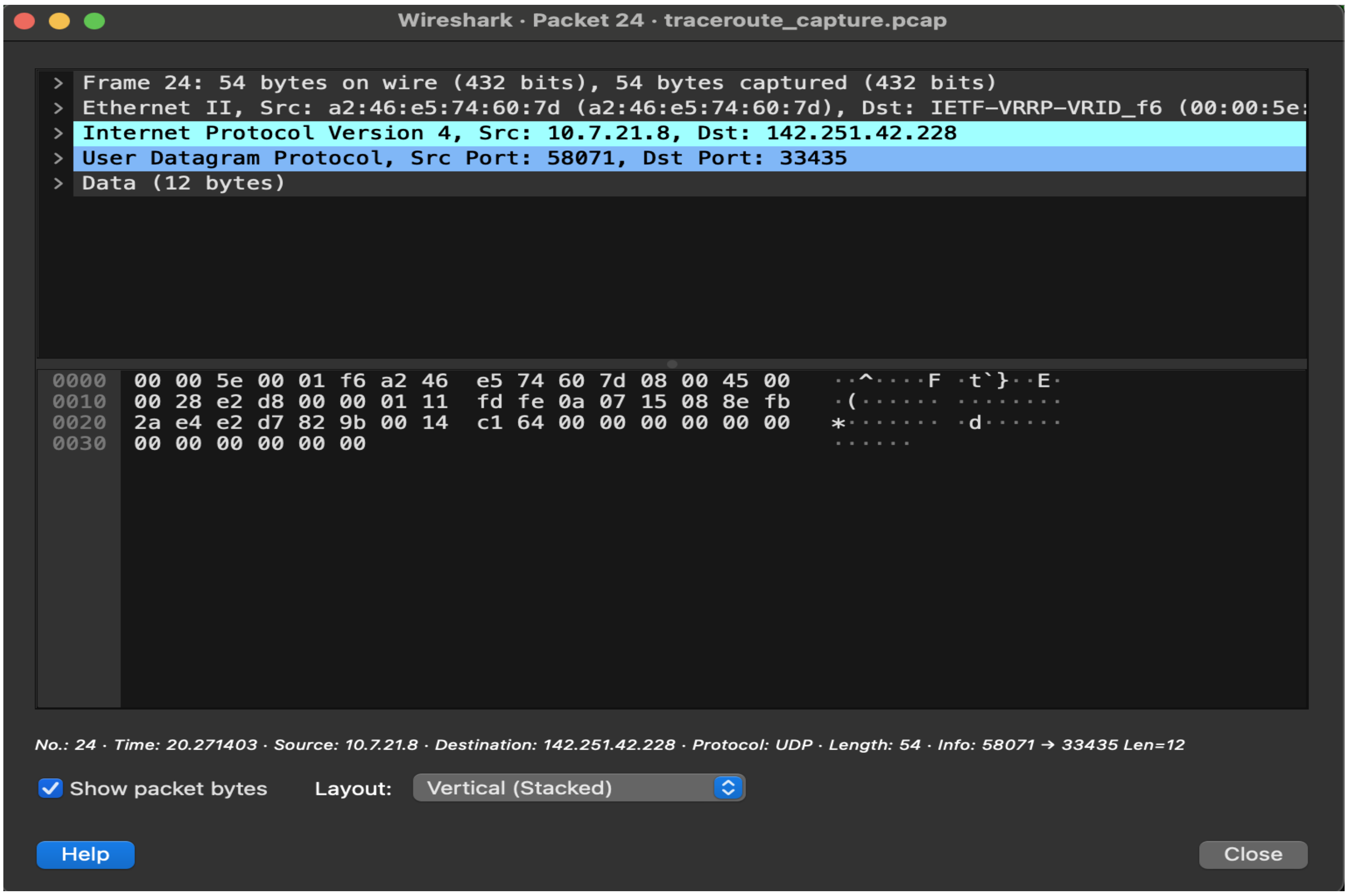
```
No.  Time       Source        Destination     Protocol Length Info
 24  20.271403  10.7.21.8     142.251.42.228  UDP      54  58071 → 33435 Len=12
 25  20.277775  10.7.0.5      10.7.21.8       ICMP     70  Time-to-live exceeded (Time to live e
 26  20.278008  10.7.21.8     142.251.42.228  UDP      54  58071 → 33436 Len=12
 27  20.281891  10.7.0.5      10.7.21.8       ICMP     70  Time-to-live exceeded (Time to live e
 28  20.282043  10.7.21.8     142.251.42.228  UDP      54  58071 → 33437 Len=12
 29  20.284933  10.7.0.5      10.7.21.8       ICMP     70  Time-to-live exceeded (Time to live e
 30  20.285063  10.7.21.8     142.251.42.228  UDP      54  58071 → 33438 Len=12
 31  20.288119  172.16.4.7    10.7.21.8       ICMP     82  Time-to-live exceeded (Time to live e
 32  20.288225  10.7.21.8     142.251.42.228  UDP      54  58071 → 33439 Len=12
 33  20.291328  172.16.4.7    10.7.21.8       ICMP     82  Time-to-live exceeded (Time to live e
 34  20.291442  10.7.21.8     142.251.42.228  UDP      54  58071 → 33440 Len=12
 35  20.294520  172.16.4.7    10.7.21.8       ICMP     82  Time-to-live exceeded (Time to live e
 36  20.294625  10.7.21.8     142.251.42.228  UDP      54  58071 → 33441 Len=12
 37  20.299883  14.139.98.1   10.7.21.8       ICMP     70  Time-to-live exceeded (Time to live e
 38  20.300074  10.7.21.8     142.251.42.228  UDP      54  58071 → 33442 Len=12
 39  20.305195  14.139.98.1   10.7.21.8       ICMP     70  Time-to-live exceeded (Time to live e
 40  20.305311  10.7.21.8     142.251.42.228  UDP      54  58071 → 33443 Len=12
 41  20.309472  14.139.98.1   10.7.21.8       ICMP     70  Time-to-live exceeded (Time to live e
```

```
> Frame 24: 54 bytes on wire (432 bits), 54 bytes captured (
> Ethernet II, Src: a2:46:e5:74:60:7d (a2:46:e5:74:60:7d), D
> Internet Protocol Version 4, Src: 10.7.21.8, Dst: 142.251.
> User Datagram Protocol, Src Port: 58071, Dst Port: 33435
> Data (12 bytes)
```

```
0000  00 00 5e 00 01 f6 a2 46  e5 74 60 7d 08 00 45 00
0010  00 28 e2 d8 00 00 01 11  fd fe 0a 07 15 08 8e fb
0020  2a e4 e2 d7 82 9b 00 14  c1 64 00 00 00 00 00 00
0030  00 00 00 00 00 00
```

User Datagram Protocol: Protocol   Packets: 259 · Displayed: 193 (74.5%)   Profile: Default

it's a UDP packet being sent to a high-numbered, non-standard port (Dst Port: 33435). This is the classic signature of a default traceroute on macOS.



Wireshark · Packet 24 · traceroute_capture.pcap

```
> Frame 24: 54 bytes on wire (432 bits), 54 bytes captured (432 bits)
> Ethernet II, Src: a2:46:e5:74:60:7d (a2:46:e5:74:60:7d), Dst: IETF-VRRP-VRID_f6 (00:00:5e:
> Internet Protocol Version 4, Src: 10.7.21.8, Dst: 142.251.42.228
> User Datagram Protocol, Src Port: 58071, Dst Port: 33435
> Data (12 bytes)
```

```
0000  00 00 5e 00 01 f6 a2 46  e5 74 60 7d 08 00 45 00   ··^····F ·t`}··E·
0010  00 28 e2 d8 00 00 01 11  fd fe 0a 07 15 08 8e fb   ·(······ ········
0020  2a e4 e2 d7 82 9b 00 14  c1 64 00 00 00 00 00 00   *······· ·d······
0030  00 00 00 00 00 00                                  ······
```

No.: 24 · Time: 20.271403 · Source: 10.7.21.8 · Destination: 142.251.42.228 · Protocol: UDP · Length: 54 · Info: 58071 → 33435 Len=12

[✓] Show packet bytes   Layout: Vertical (Stacked)