Task 2: Network Path Analysis using Traceroute and Tracert

Your Name Your Partner's Name

September 15, 2025

Objective

The objective of this task is to analyze the path taken by packets to a remote host using `tracert` (Windows) and `traceroute` (Linux/macOS). We will examine the intermediate hops, final hop responses, and compare protocol behavior based on captured network traffic.

Methodology

To understand the underlying mechanisms of the traceroute utilities, experiments were conducted on both Windows and macOS. The network traffic for each was captured to analyze the protocols used.

Windows (`tracert` – ICMP based)

The `tracert www.google.com` command was executed on Windows. The output in Figure 1 shows the successful trace to the destination. Wireshark captures (Figures 2 and 3) confirm that `tracert` uses the ICMP protocol.

```
C:\Users\sawan>tracert www.google.com

Tracing route to www.google.com [142.250.192.68]
over a maximum of 30 hops:

  1     2 ms     3 ms     4 ms  10.7.0.5
  2    13 ms     2 ms     2 ms  172.16.4.7
  3    81 ms    61 ms    29 ms  14.139.98.1
  4     2 ms     1 ms    19 ms  10.117.81.253
  5    10 ms    10 ms    10 ms  10.154.8.137
  6    21 ms    25 ms    15 ms  10.255.239.170
  7    26 ms     9 ms    10 ms  10.152.7.214
  8    70 ms    39 ms    12 ms  72.14.204.62
  9    27 ms    13 ms    11 ms  142.251.49.177
 10    29 ms    40 ms    17 ms  108.170.226.131
 11    31 ms    12 ms    12 ms  bom12s16-in-f4.1e100.net [142.250.192.68]

Trace complete.
```

Figure 1: Output of the `tracert www.google.com` command on Windows.

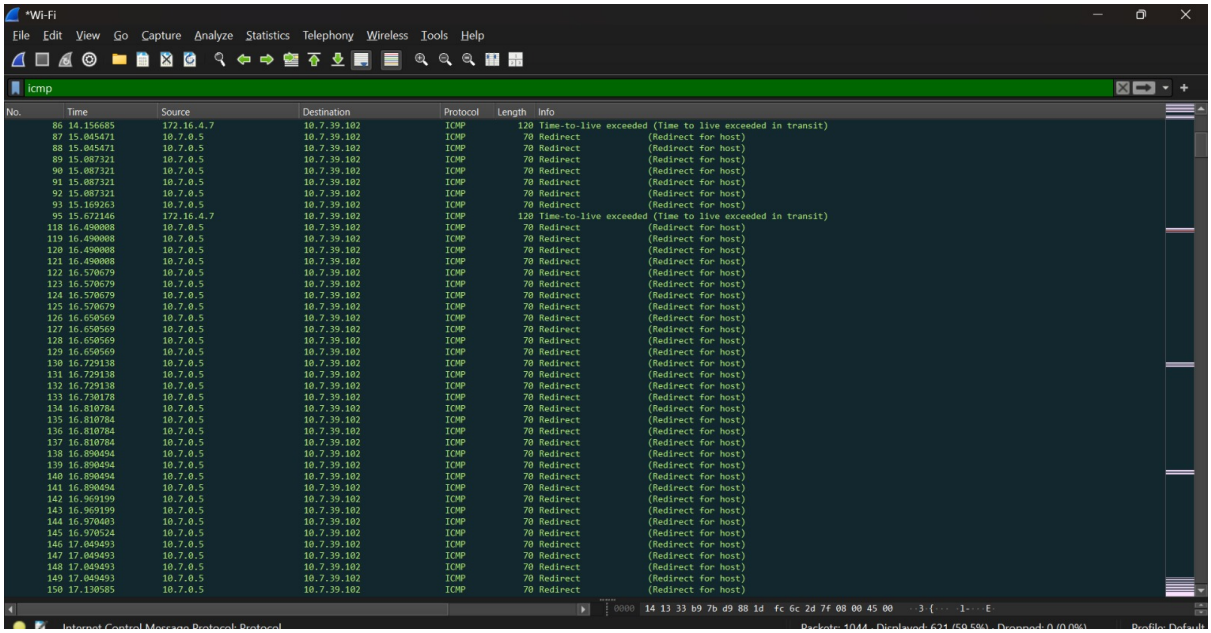

Figure 2: Wireshark capture showing the various ICMP packets generated by `tracert`.

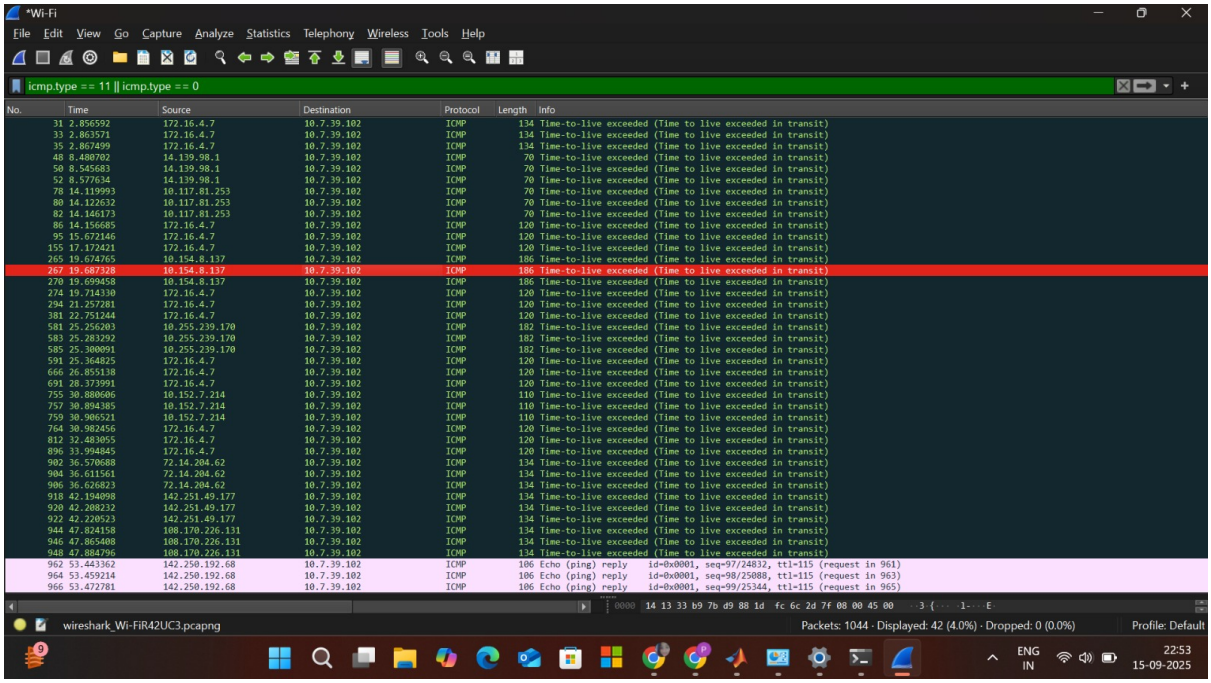

Figure 3: Filtered Wireshark capture highlighting the final ICMP Echo Reply (bottom packet) from the destination, which is different from the intermediate "Time-to-live exceeded" responses.

macOS (`traceroute` – UDP based)

The `traceroute www.google.com` command was executed on macOS, and the traffic was captured using `tcpdump`. The command output is shown below, followed by the corresponding packet capture.

Traceroute Command Output:

```
traceroute to www.google.com (216.58.203.4), 64 hops max, 40 byte packets
 1  10.7.0.5 (10.7.0.5)  4.529 ms  3.718 ms  3.772 ms
 2  172.16.4.7 (172.16.4.7)  3.841 ms  3.521 ms  3.228 ms
 3  14.139.98.1 (14.139.98.1)  5.281 ms  5.321 ms  5.182 ms
 4  10.117.81.253 (10.117.81.253)  16.463 ms  3.935 ms  3.693 ms
 5  10.154.8.137 (10.154.8.137)  12.214 ms  11.907 ms  11.876 ms
 6  10.255.239.170 (10.255.239.170)  13.410 ms  11.888 ms  11.877 ms
 7  10.152.7.214 (10.152.7.214)  12.171 ms  15.959 ms  13.713 ms
 8  72.14.204.62 (72.14.204.62)  12.172 ms * *
 9  * * *
10  142.250.228.48 (142.250.228.48)  26.352 ms
    216.239.58.18 (216.239.58.18)  13.912 ms
    192.178.86.238 (192.178.86.238)  29.365 ms
11  142.250.226.134 (142.250.226.134)  13.885 ms  13.686 ms  14.401 ms
12  bom12s04-in-f4.1e100.net (216.58.203.4)  12.268 ms  12.472 ms  12.551 ms
```

Tcpdump Packet Capture Analysis: The `tcpdump` output below provides clear evidence of the UDP-based approach on macOS.

- **Intermediate Hop Probes:** For each intermediate hop (e.g., lines starting with `23:16:36`), we can see an `ICMP time exceeded in-transit` message. This is the router's response when it receives a UDP packet with a TTL of 1. Inside this ICMP message, the original UDP probe packet is quoted.
- **Final Hop Probe:** The final packet in the trace (timestamp `23:17:02.194677`) shows the destination server `216.58.203.4` responding with an `ICMP 216.58.203.4 udp port 33468 unreachable` message. This confirms that the traceroute has reached its destination.

```
23:16:36.560938 IP ... 10.7.0.5 > 10.7.44.10: ICMP time exceeded in-transit...
    10.7.44.10.56667 > 216.58.203.4.33435: UDP, length 12
...
23:17:02.194677 IP ... 216.58.203.4 > 10.7.44.10: ICMP 216.58.203.4 udp port 33468 unreachable...
    10.7.44.10.56667 > 216.58.203.4.33468: UDP, length 12
```

Results Summary

OS	Command	Intermediate Hops	Final Hop
Windows	`tracert www.google.com`	ICMP Time Exceeded	ICMP Echo Reply
Linux/macOS	`traceroute www.google.com`	ICMP Time Exceeded	ICMP Port Unreachable

Table 1: Summary of Tracert and Traceroute Behavior

Questions and Answers

Q1: What protocol does Windows tracert use by default, and what protocol does Linux/macOS traceroute use by default?
As shown in the Wireshark and `tcpdump` captures, Windows `tracert` uses **ICMP** Echo Request packets. In contrast, the default behavior for Linux/macOS `traceroute` is to use **UDP** probes sent to high-numbered ports.

Q2: Some hops in your traceroute output may show `* * *`. Provide at least two reasons why a router might not reply.
A router might not reply for several reasons, leading to a timeout (`* * *`):

- **Firewall Configuration:** The router or a firewall in front of it may be configured to block incoming probes or outgoing ICMP "Time Exceeded" messages for security reasons.
- **Low Priority of ICMP:** Routers prioritize their primary task of forwarding traffic. Generating ICMP error messages is a low-priority task that may be ignored if the router is experiencing high traffic loads.

Q3: In Linux/macOS traceroute, which field in the probe packets changes between successive probes sent to the destination?
In Linux/macOS `traceroute`, the **Time-to-Live (TTL)** field in the IP header of the outgoing UDP probe packets is incremented by one for each successive hop. It starts with TTL=1 for the first hop, TTL=2 for the second, and so on. The destination UDP port also increments for each probe.

Q4: At the final hop, how is the response different compared to the intermediate hop?
The responses are fundamentally different and signal the end of the trace, as evidenced by the packet captures:

- **Intermediate Hops:** All intermediate routers respond with an **ICMP "Time-to-live exceeded"** message when the packet's TTL value drops to zero.
- **Final Hop:** The final destination responds differently. On Windows, it sends an **ICMP "Echo reply"**. As seen in the `tcpdump` output, on macOS it sends an **ICMP "Destination unreachable (Port unreachable)"** message because it receives a UDP packet on a port it isn't listening on.

Q5: Suppose a firewall blocks UDP traffic but allows ICMP — how would this affect the results of Linux/macOS traceroute vs. Windows tracert?
This scenario would have drastically different outcomes for the two utilities:

- **Linux/macOS `traceroute`:** It would **fail**. The outgoing UDP probe packets would be blocked by the firewall. The command would time out at every hop, resulting in an output of only `* * *`.
- **Windows `tracert`:** It would **succeed**. Since it uses ICMP packets for its probes and the firewall allows ICMP, the trace would complete normally.