**🧾 Network Traffic Analysis Report using Wireshark**

**🛰️ 1. Capture Setup**

\* Tool Installed: Wireshark

\* Capture Interface: Wi-Fi (wlan0)

\* Duration of Capture: 2 minute

**📦 2. Packet Filtering and Protocol Analysis**

**➤ Applied Filters in Wireshark:**

\* `tcp` – to view TCP traffic (HTTP/S)

\* `udp` – for UDP-based protocols (e.g., DNS)

\* `dns` – to isolate domain name resolution packets

**🧩 3. Identified Protocols**

| Protocol | Description | Port | Function |

| -------- | --------------------------------- | ------- | ------------------------- |

| \*\*TCP\*\* | Transmission Control Protocol | 80, 443 | Web traffic (HTTP/HTTPS) |

| \*\*UDP\*\* | User Datagram Protocol | 53 | Lightweight data transfer |

| \*\*DNS\*\* | Domain Name System | 53 | Domain name resolution |

| \*\*ICMP\*\* | Internet Control Message Protocol | N/A | Ping requests/replies |

| \*\*TLS\*\* | Transport Layer Security | 443 | Secure web traffic |

**🔎 4. Packet Details**



**📁 5. Exported Capture**

\* File Name: `network\_traffic.pcap`

\* File includes traffic from TCP, DNS, UDP protocols with timestamps and packet lengths.

**🧠 6. Summary & Observations**

\* The network generated standard traffic during basic browsing and pinging.

\* DNS queries were sent before each website visit.

\* TLS handshake and HTTP GET requests were observed.

\* ICMP packets show successful ping responses from 8.8.8.8.

\* No anomalies or suspicious traffic was observed in the short window of capture.

**📌 Conclusion**

This analysis demonstrates how common protocols like TCP, UDP, DNS, and ICMP appear in a network capture. Wireshark provides deep visibility into packet-level communication, making it an essential tool for network diagnostics and forensic analysis.