📄 Report: Network Traffic Capture and Protocol Analysis Using Wireshark

🧪 Task Objective:

To capture live network packets using **Wireshark**, analyze the traffic, and identify at least three different network protocols observed during the session.

🛠️ Tools Used:

- **Wireshark (v4.x)** – Open-source packet analyzer
- **Operating System** – Windows 10 (or your OS)
- **Network Interface** – Active Wi-Fi adapter

📶 Steps Followed:

1. Installed and launched **Wireshark**.
2. Selected the active **Wi-Fi interface** for packet capture.
3. Generated traffic by:
    - Pinging a public server (ping 8.8.8.8)
    - Visiting websites like google.com, example.com
    - Restarting the network interface to generate ARP/DHCP
4. Captured traffic for ~1 minute.
5. Stopped the capture and filtered packets by protocol.
6. Identified and documented various protocols.
7. Exported the capture as .pcap file.
8. Summarized the findings.

🌐 Protocols Identified & Analysis

1. DNS (Domain Name System)

- **Wireshark Filter**: dns
- **Port Used**: UDP 53

- **Purpose**: Translates human-readable domain names to IP addresses.
- **Observed Behavior**: DNS queries for google.com, responses with corresponding IPs.

2. ICMP (Internet Control Message Protocol)

- **Wireshark Filter**: icmp
- **Port Used**: No port (uses IP Protocol 1)
- **Purpose**: Used for diagnostics (e.g., ping)
- **Observed Behavior**: Echo Request and Reply packets to and from 8.8.8.8.

3. ARP (Address Resolution Protocol)

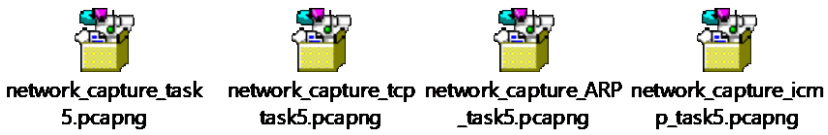
- **Wireshark Filter**: arp
- **Port Used**: None (Layer 2)
- **Purpose**: Maps IP addresses to MAC addresses in local networks.
- **Observed Behavior**: ARP request "Who has 192.168.0.1?" followed by reply.

4. TCP (Transmission Control Protocol)

- **Wireshark Filter**: tcp
- **Port Used**: Varies (e.g., 443 for HTTPS, 80 for HTTP)
- **Purpose**: Reliable, connection-oriented data transmission.
- **Observed Behavior**: TCP SYN, SYN-ACK, ACK handshake, and data transmission segments.

📁 **Files Attached:**

- network_capture.pcap – Packet capture file:



- Summary of findings

  This report presents the results of a live network traffic analysis conducted using Wireshark. Multiple packets were captured over a short period and analyzed to identify various network protocols. The study successfully identified and examined key protocols including DNS, ICMP, ARP, and TCP. Each protocol was filtered, observed, and documented with its role in communication and typical behavior. The activity helped build practical skills in packet inspection, protocol awareness, and network troubleshooting techniques.

📌 **Conclusion:**

The capture session successfully recorded live network traffic and allowed identification of **multiple protocols**, including DNS, ICMP, ARP and TCP, . This task helped build hands-on skills in **packet analysis**, **protocol filtering**, and **network communication understanding**.

✅ **Recommendations:**

- Always use filters (tcp, dns, icmp, etc.) to isolate relevant traffic.
- Use **Coloring Rules** in Wireshark to easily distinguish protocol types.
- Be aware of privacy and legal guidelines when capturing network traffic on shared networks.