Task 5: Capture and Analyze Network Traffic Using Wireshark

System: Ubuntu/Linux using Wireshark

Step 1: Install Wireshark

sudo apt update
sudo apt install wireshark -y

Step 2: Start Wireshark & Begin Capture

- Launched Wireshark.
- Selected the active interface (e.g., `eth0`, `wlan0`, or other listed interfaces).
- Clicked **Start Capturing Packets**.

Step 3: Generate Network Traffic

- Opened a web browser and visited a few websites.
- This triggered various types of traffic including DNS lookups and secure connections.

Step 4: Stop Capture

- After ~1 minute of traffic generation, stopped the capture to review collected packets.

Step 5: Filter and Analyze by Protocol

From Image 1 (QUIC, DNS traffic):

- **QUIC (UDP-based)**:
    - Connected to `142.250.183.100`, `142.250.77.35`, and `142.251.220.10`.
    - Payloads show typical QUIC handshake patterns including `Initial`, `Crypto`, `Ping`, `Protected Payload`.
- **DNS**:
    - Resolved names such as:
        - `udit.mu.ac.in`
        - `fonts.googleapis.com`
        - `gstatic.com`
    - All DNS queries directed to `myIP`, the likely gateway or DNS resolver.

From Image 2 (TCP, TLSv1.3):

- **TCP**:
    - Source IP `myIP` communicated with `121.241.25.72` on port `443` (HTTPS).
    - Includes TCP flags like `SYN`, `ACK`, and `RST` as part of the handshake.
- **TLSv1.3**:
    - Application data and Client Hello messages confirm encrypted HTTPS communication.
    - Session shows secure negotiation between local machine and remote host.

Step 6: Protocols Identified

Protocol	Description
QUIC	Secure, fast UDP-based protocol used by Google.
DNS	Resolving domain names to IP addresses.

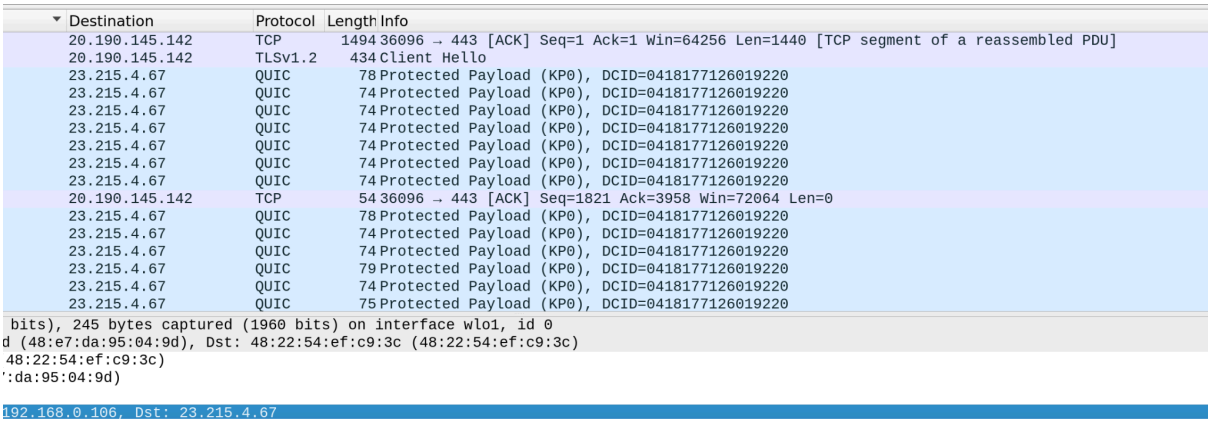
TCP	Underlying transport protocol for secure web traffic.
TLSv1.3	Latest version of Transport Layer Security (HTTPS).

Step 7: Summary of Findings

The Wireshark capture clearly demonstrates real-world internet activity:

- DNS queries resolving web domains.
- QUIC traffic establishing fast encrypted connections.
- TCP and TLSv1.3 ensuring secure HTTPS communication.

All protocols reflect typical behavior when browsing modern websites, including Google services. The presence of QUIC and TLSv1.3 indicates that the system is communicating securely.



Destination	Protocol	Length	Info
20.190.145.142	TCP	1494	36096 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=1440 [TCP segment of a reassembled PDU]
20.190.145.142	TLSv1.2	434	Client Hello
23.215.4.67	QUIC	78	Protected Payload (KP0), DCID=0418177126019220
23.215.4.67	QUIC	74	Protected Payload (KP0), DCID=0418177126019220
23.215.4.67	QUIC	74	Protected Payload (KP0), DCID=0418177126019220
23.215.4.67	QUIC	74	Protected Payload (KP0), DCID=0418177126019220
23.215.4.67	QUIC	74	Protected Payload (KP0), DCID=0418177126019220
23.215.4.67	QUIC	74	Protected Payload (KP0), DCID=0418177126019220
23.215.4.67	QUIC	74	Protected Payload (KP0), DCID=0418177126019220
20.190.145.142	TCP	54	36096 → 443 [ACK] Seq=1821 Ack=3958 Win=72064 Len=0
23.215.4.67	QUIC	78	Protected Payload (KP0), DCID=0418177126019220
23.215.4.67	QUIC	74	Protected Payload (KP0), DCID=0418177126019220
23.215.4.67	QUIC	74	Protected Payload (KP0), DCID=0418177126019220
23.215.4.67	QUIC	79	Protected Payload (KP0), DCID=0418177126019220
23.215.4.67	QUIC	74	Protected Payload (KP0), DCID=0418177126019220
23.215.4.67	QUIC	75	Protected Payload (KP0), DCID=0418177126019220