COMPUTER NETWORK

Packet Analysis Using Wireshark

CODEINTERN

Rimsha Bibi

CI/DEC116

28-DEC-2025

Objective

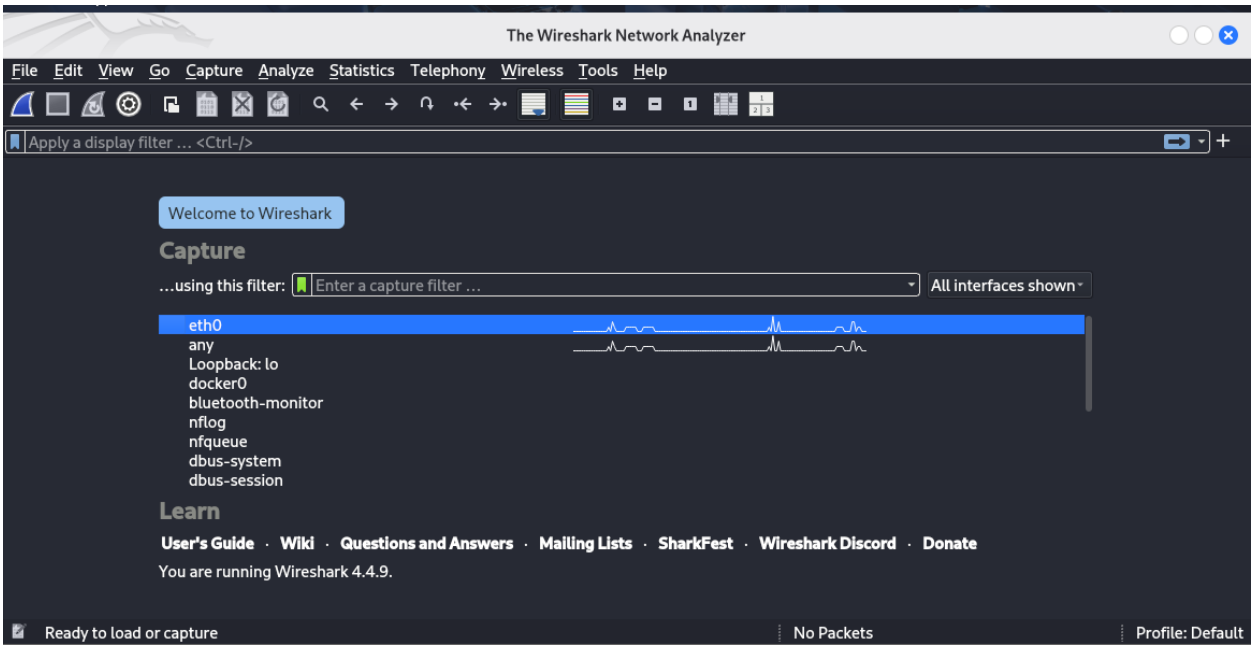
To analyze network traffic using **Wireshark** by capturing and identifying different protocols (HTTPS, DNS, ARP, TCP, UDP), understanding TCP connection establishment, and applying packet filters.

1. Install Wireshark

Steps

1. Download Wireshark from: https://www.wireshark.org
2. Choose your operating system (Windows / Linux / macOS).
3. During installation on Windows, **enable Npcap** (required for packet capture).
4. Launch Wireshark after installation.

📸 **Screenshot to include**:



2. Capture Network Packets

Steps

1. Open Wireshark.
2. Select your active network interface (Wi-Fi or Ethernet).
3. Click **Start Capturing Packets** (blue shark fin).
4. Perform normal network activities:

- Open a website (HTTPS)
- Run ping google.com
- Browse any webpage

5. Stop capture after ~1–2 minutes.

Screenshot to include:

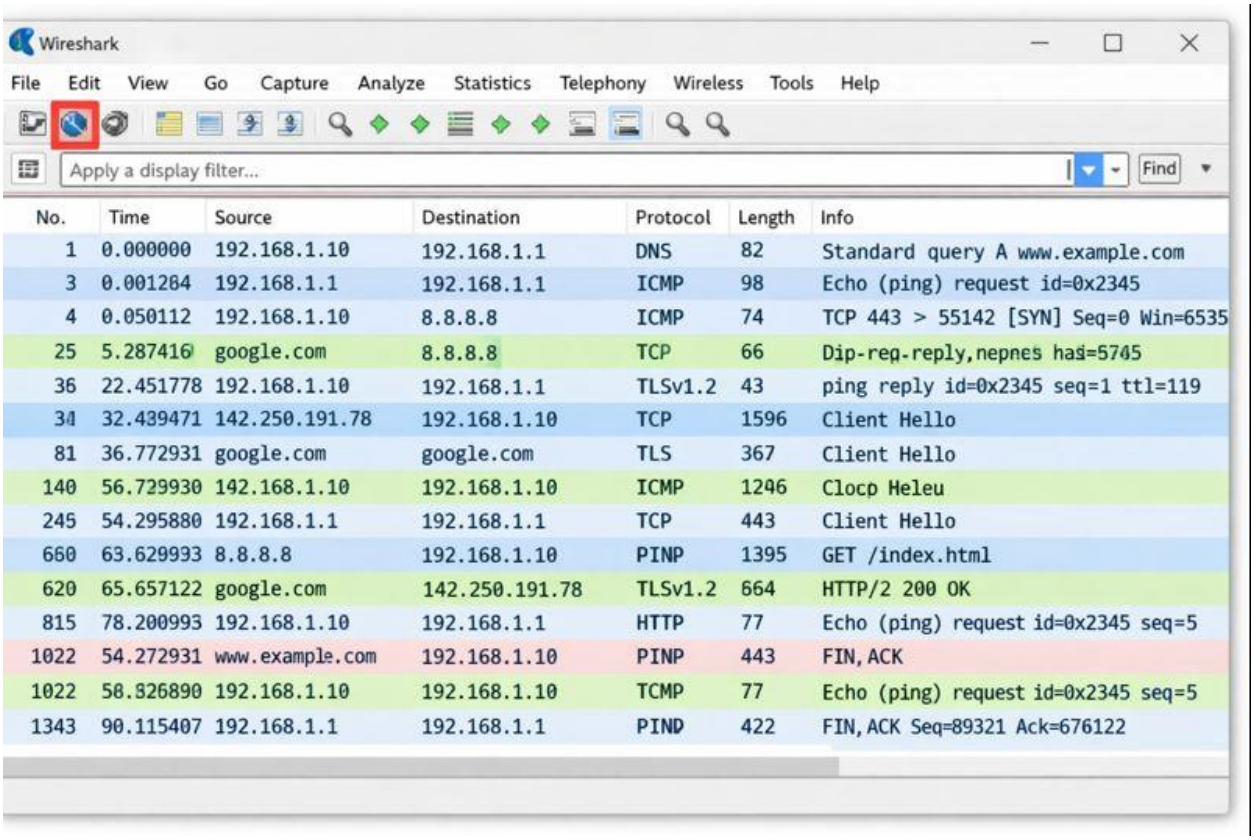


Figure 1Live packet capture showing multiple protocols.

3. Identify Protocols in Captured Traffic

Common Protocols Observed

Protocol	Description
HTTPS	Secure web communication (encrypted HTTP)
DNS	Resolves domain names to IP addresses
ARP	Maps IP addresses to MAC addresses
TCP	Reliable, connection-oriented protocol
UDP	Fast, connectionless protocol

You can see the protocol name in the **Protocol** column.

Screenshot to include:

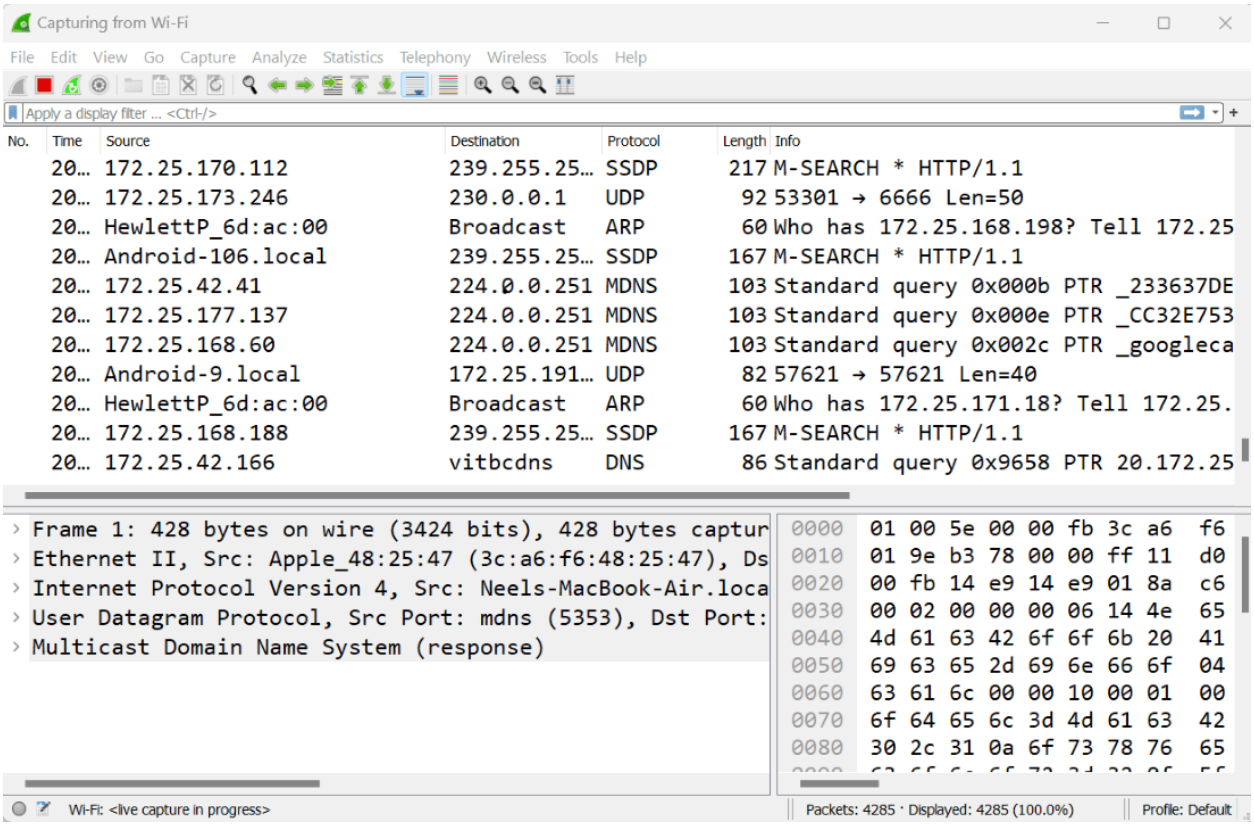
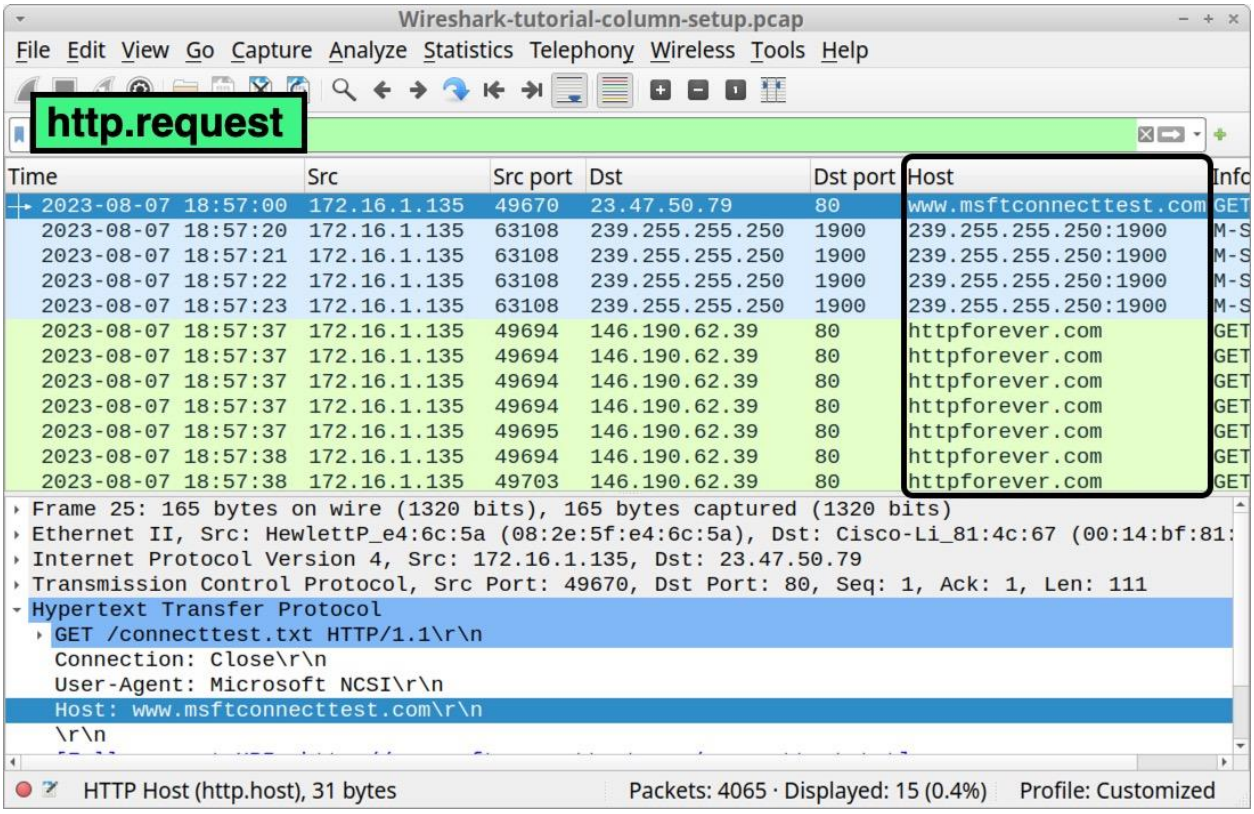
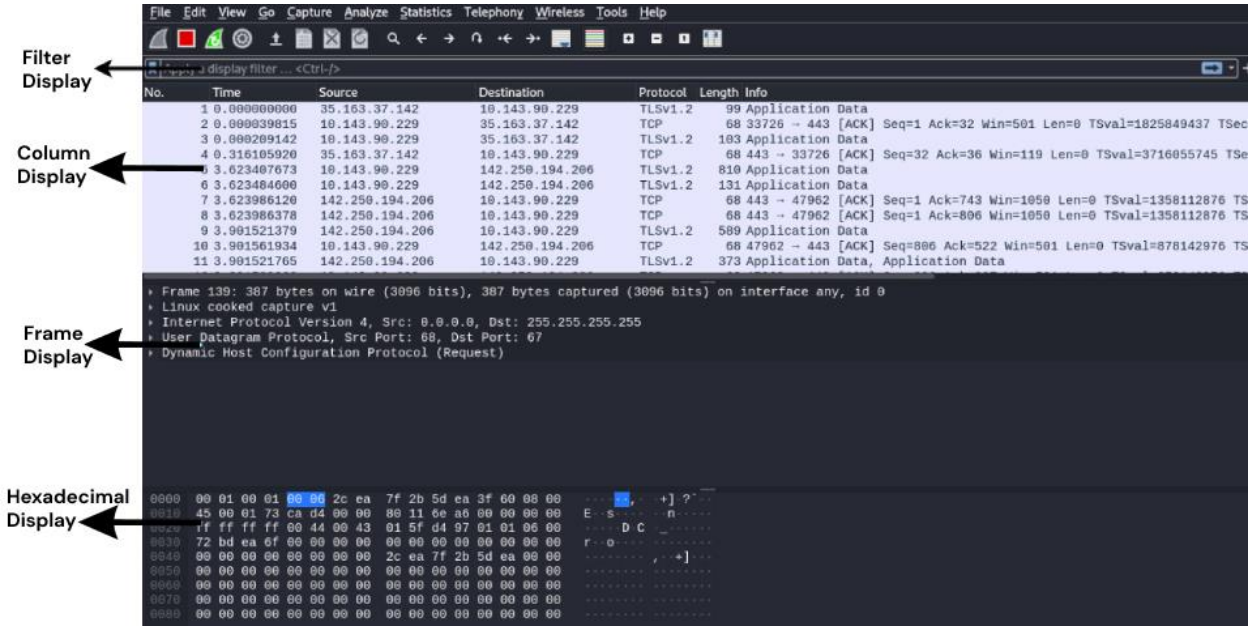


Figure 2 Packet list showing DNS, TCP, UDP, ARP traffic.





4. TCP 3-Way Handshake

The **TCP 3-way handshake** is used to establish a reliable connection between a client and server.

Steps in the Handshake

1. **SYN**
    - Client → Server
    - Requests connection
2. **SYN-ACK**
    - Server → Client
    - Acknowledges request
3. **ACK**
    - Client → Server
    - Connection established

Why it's important

- Ensures both sides are ready
- Synchronizes sequence numbers
- Prevents half-open connections

Screenshot to include:

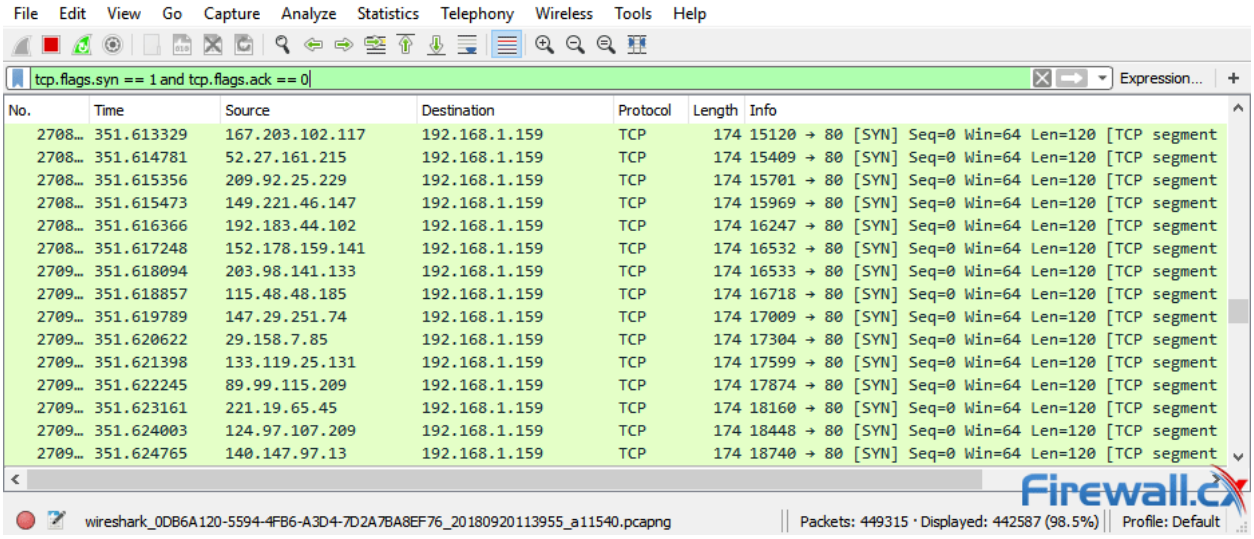


Figure 3. TCP 3-Way Handshake

```
    Checksum: 0x262f [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▼ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
    ▶ TCP Option - No-Operation (NOP)
    ▶ TCP Option - No-Operation (NOP)
    ▶ TCP Option - Timestamps: TSval 824635422, TSecr 3249934137
  ▼ [SEQ/ACK analysis]
      [This is an ACK to the segment in frame: 15]
      [The RTT to ACK the segment was: 0.002592000 seconds]
    ▼ [TCP Analysis Flags]
      ▼ [Expert Info (Warning/Sequence): Previous segment not captured (common at capture start)]
          [Previous segment not captured (common at capture start)]
          [Severity level: Warning]
          [Group: Sequence]
```

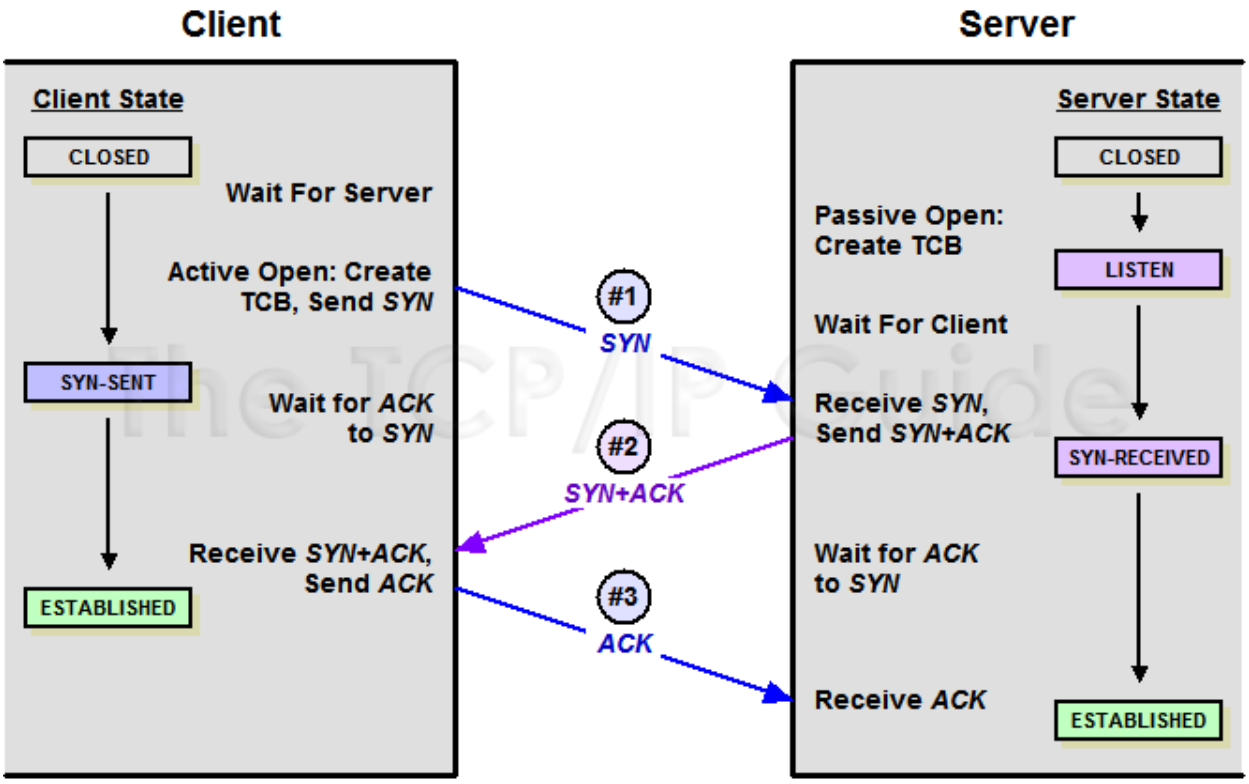


Figure 4 TCP packets showing SYN, SYN-ACK, ACK flags.

5. Filtering Packets Using Wireshark Filters

Display Filters (Used After Capture)

Protocol	Filter
DNS	dns
ARP	arp

TCP	tcp
UDP	udp
HTTPS	tcp.port == 443
TCP SYN packets	tcp.flags.syn == 1

Example

- To see only DNS packets:

dns

Screenshot to include:

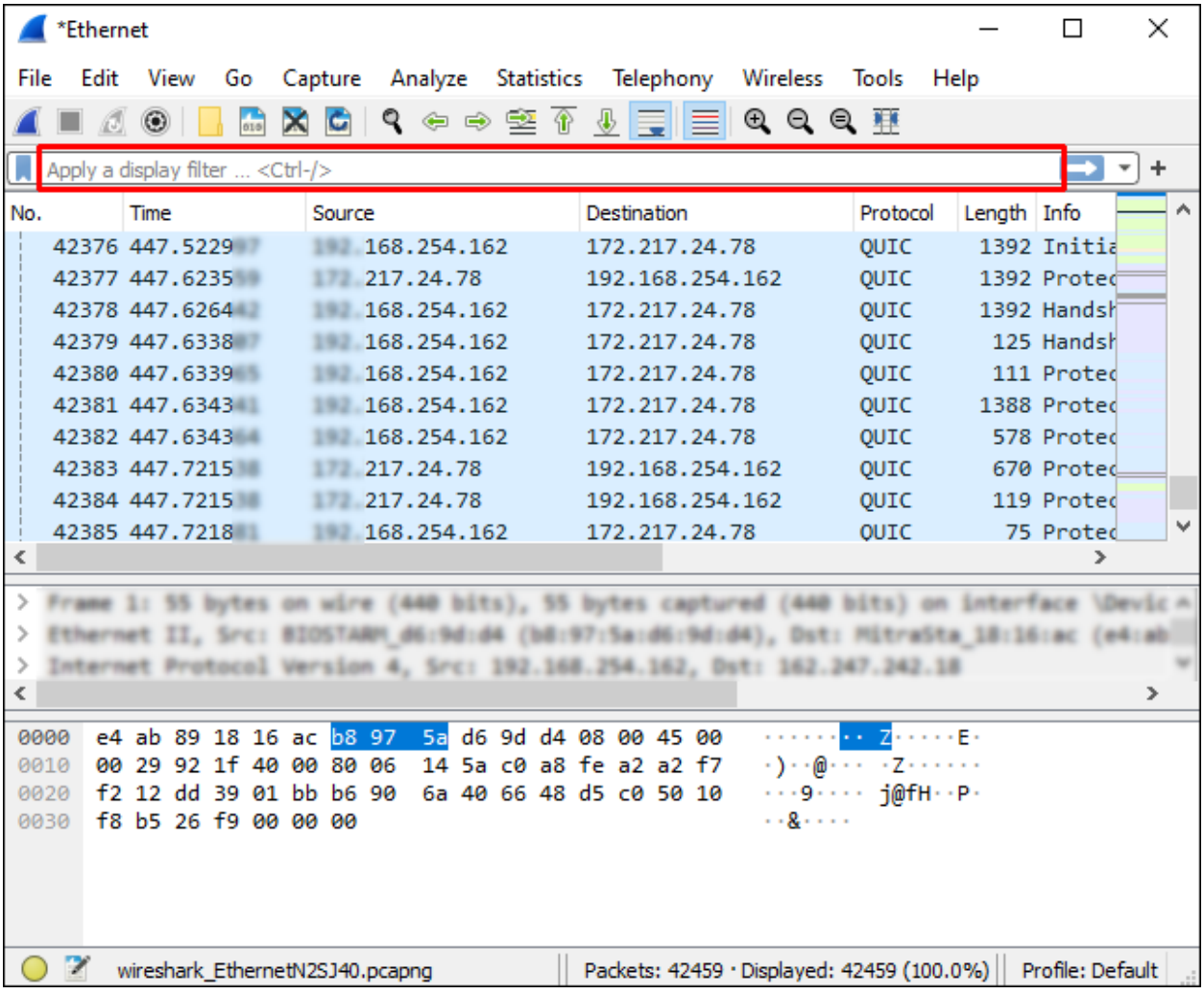
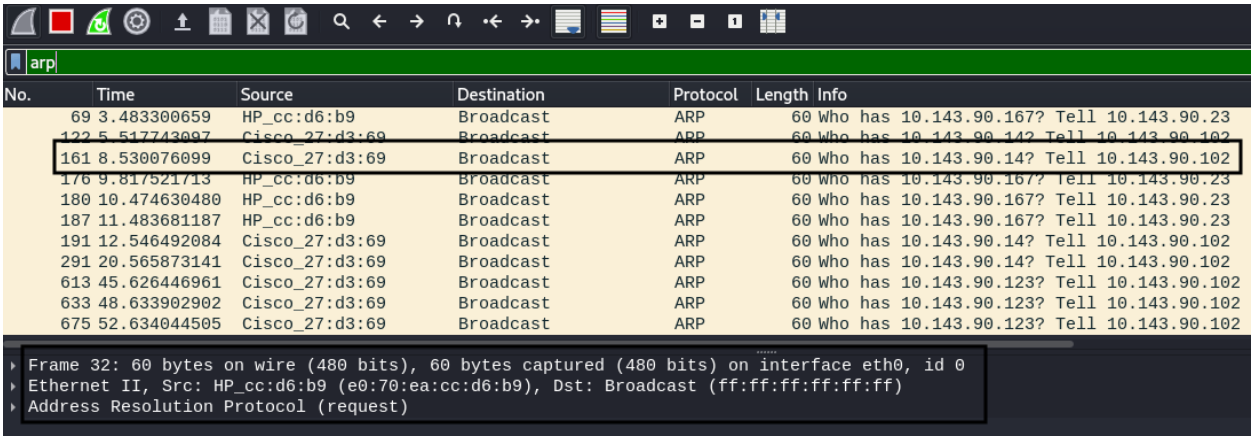
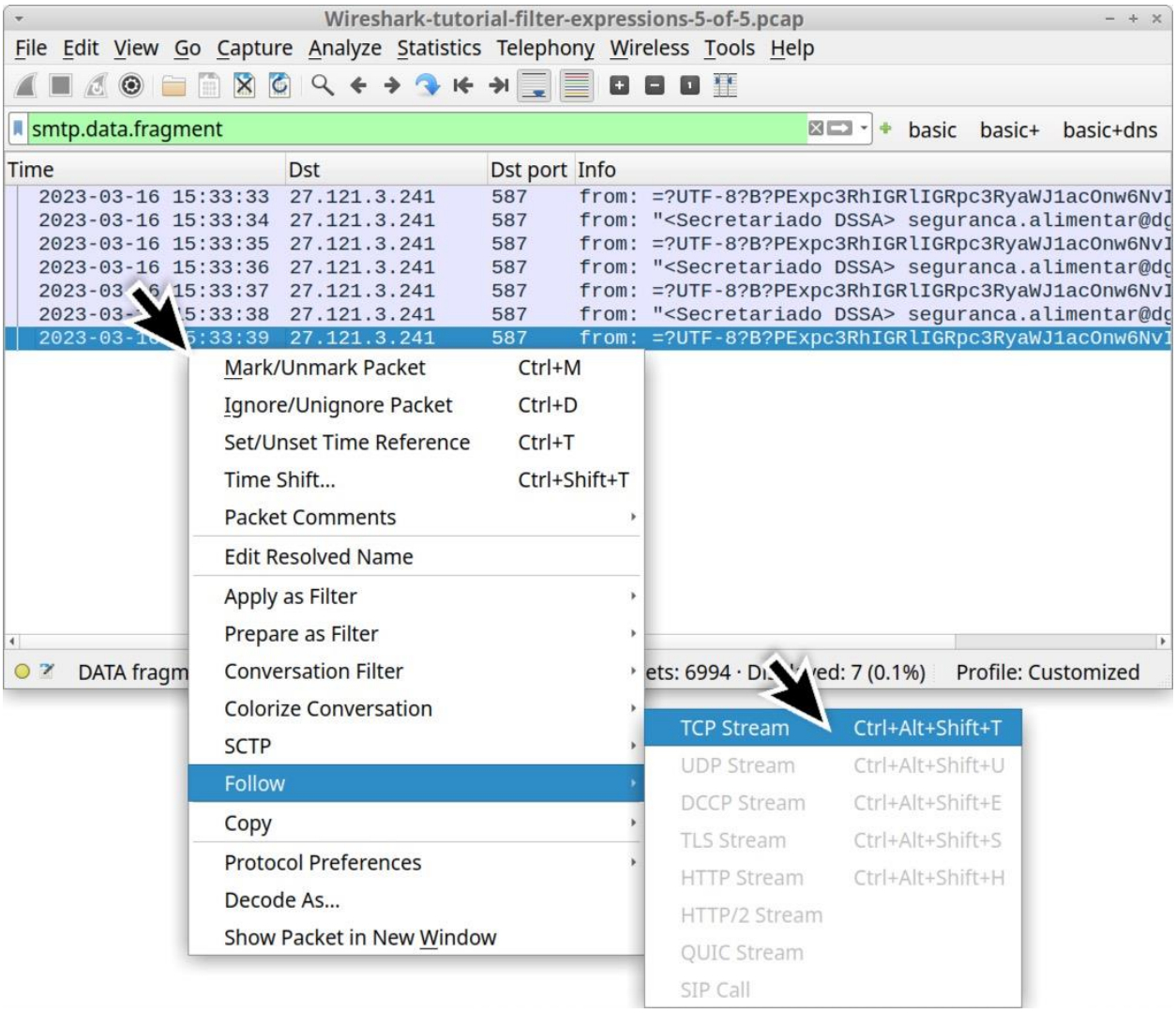


Figure 5Filter bar showing a protocol filter applied.

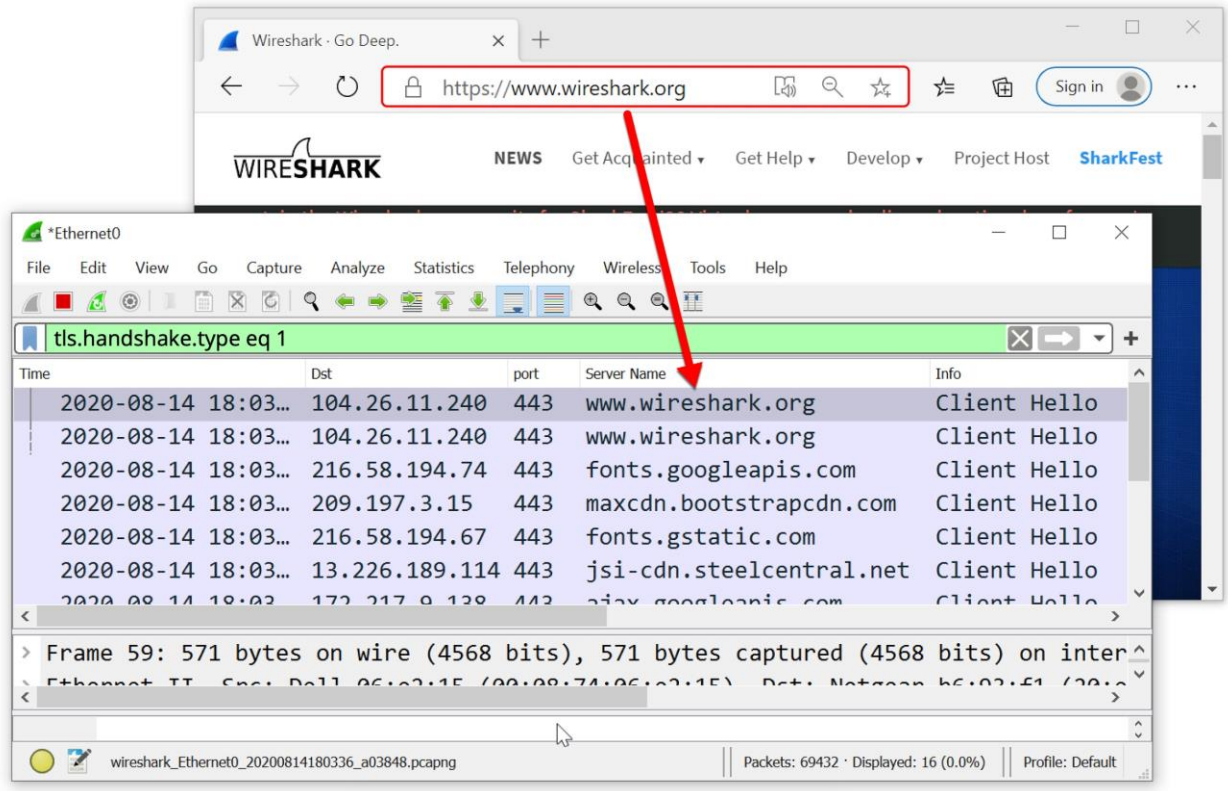


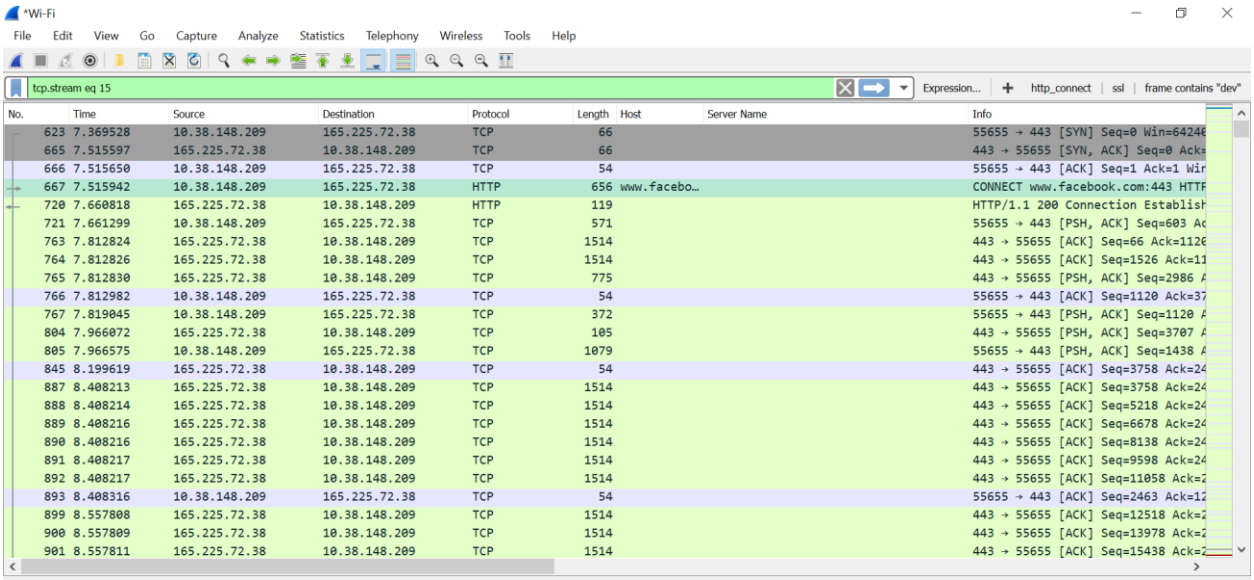


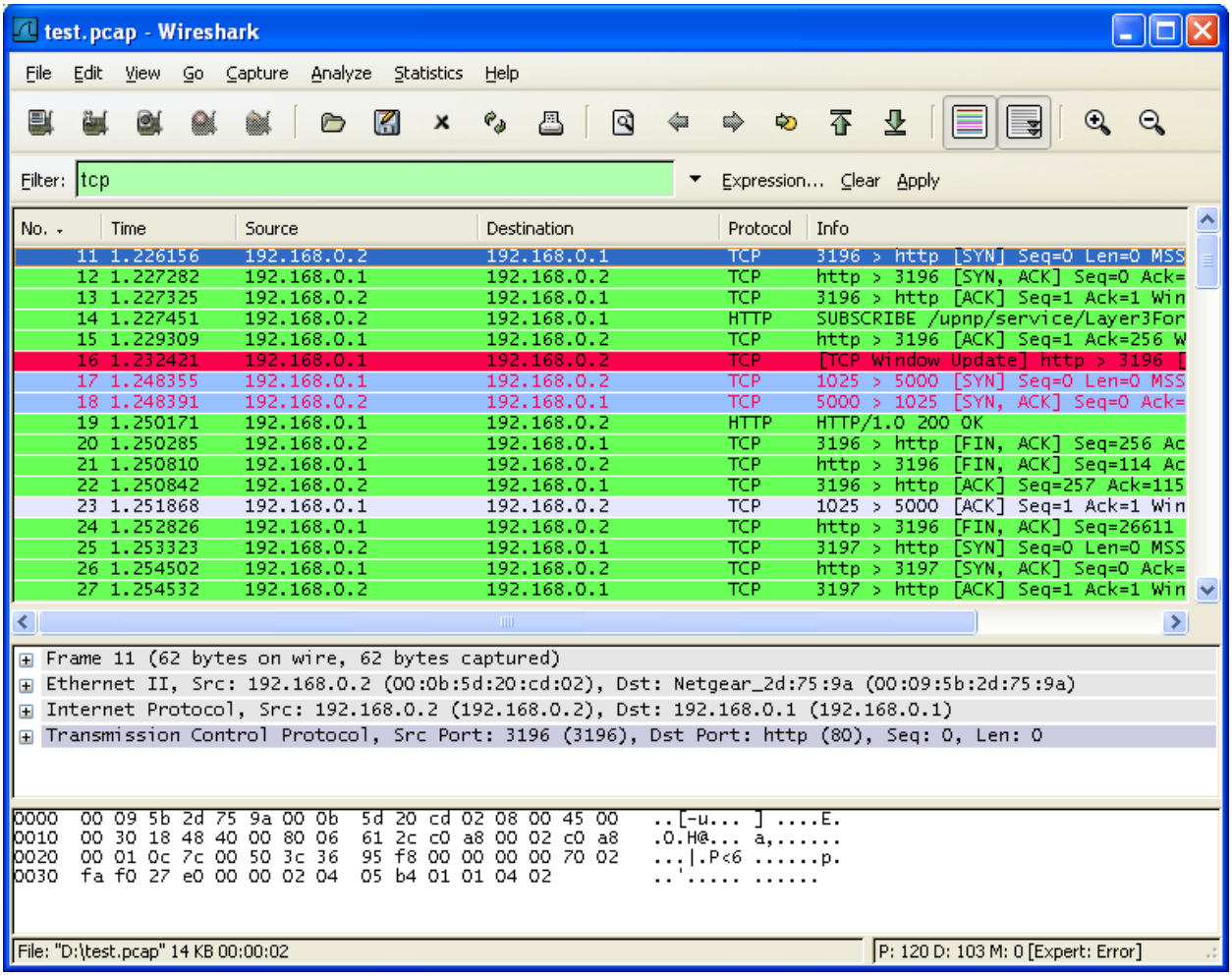
6. HTTPS Traffic Analysis

- HTTPS uses **TCP port 443**
- Payload is encrypted
- Only metadata (IP, port, handshake) is visible

Screenshot to include:







7. DNS Traffic Analysis

- DNS queries typically use **UDP port 53**
- Shows domain name resolution

Screenshot to include:

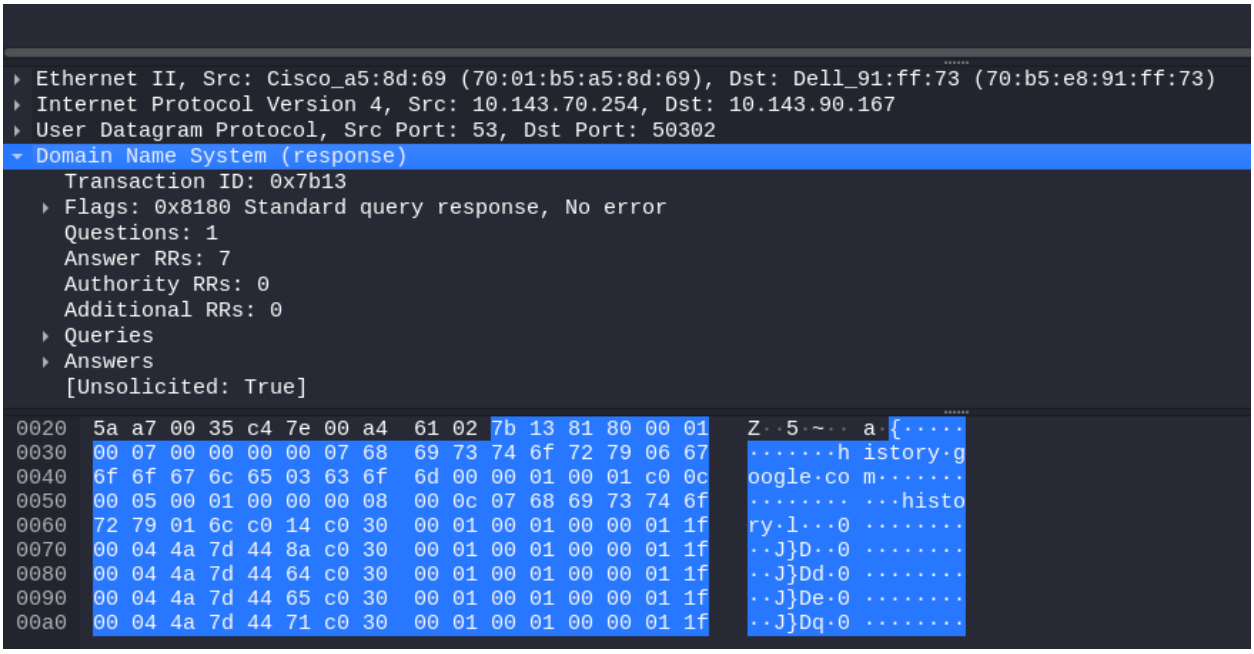


Figure 6 DNS query and response packets

.

8. ARP Traffic Analysis

- ARP works within the local network
- No ports used
- Resolves IP → MAC address

Screenshot to include:

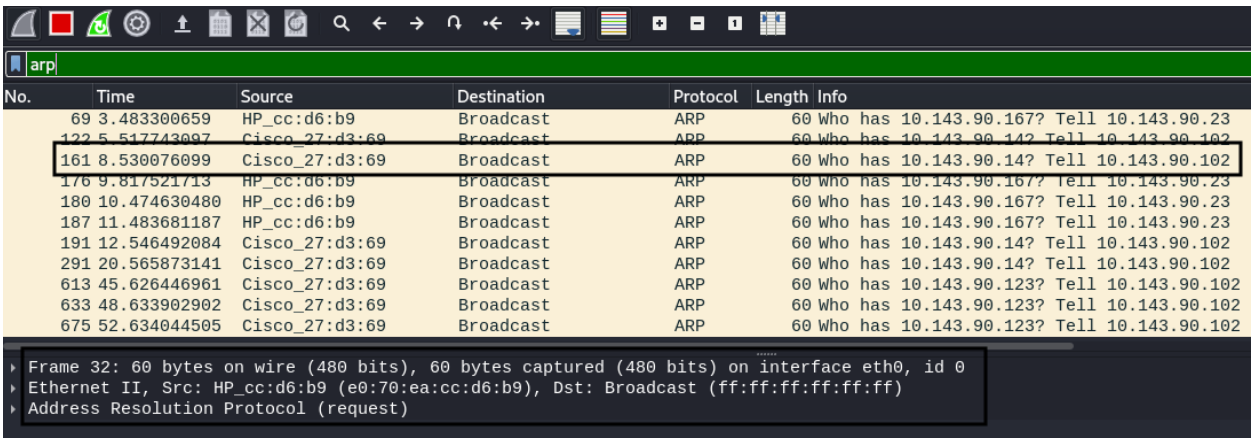


Figure 7 ARP request and reply packets.

9. TCP vs UDP (Observation)

Feature	TCP	UDP
Connection	Yes	No
Reliability	High	Low
Speed	Slower	Faster
Example Use	HTTPS, FTP	DNS, VoIP

Conclusion

In this task, Wireshark was used to:

- Capture real network traffic
- Identify common network protocols
- Analyze TCP connection establishment
- Apply display filters for packet analysis
- Understand protocol behavior in real-time

Wireshark is a powerful tool for **network troubleshooting, security analysis, and protocol learning**.