



COMPUTER NETWORK

Packet Analysis Using Wireshark

CODEINTERN

Rimsha Bibi

CI/DEC116

28-DEC-2025

Objective

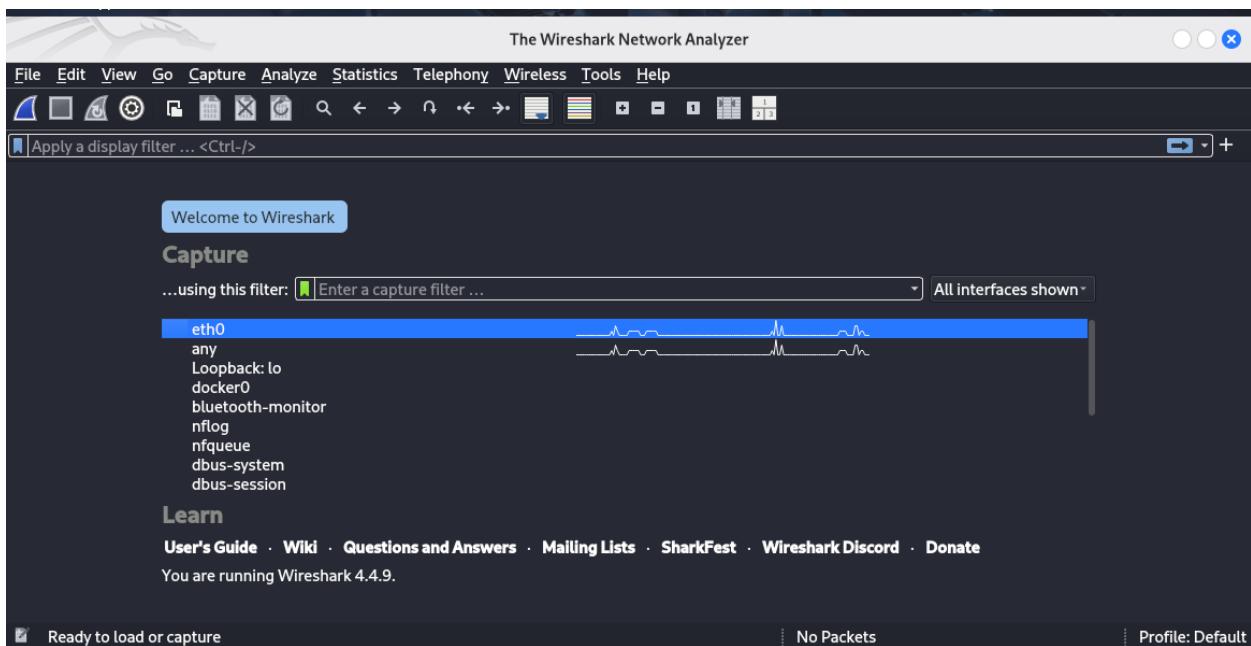
To analyze network traffic using **Wireshark** by capturing and identifying different protocols (HTTPS, DNS, ARP, TCP, UDP), understanding TCP connection establishment, and applying packet filters.

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:

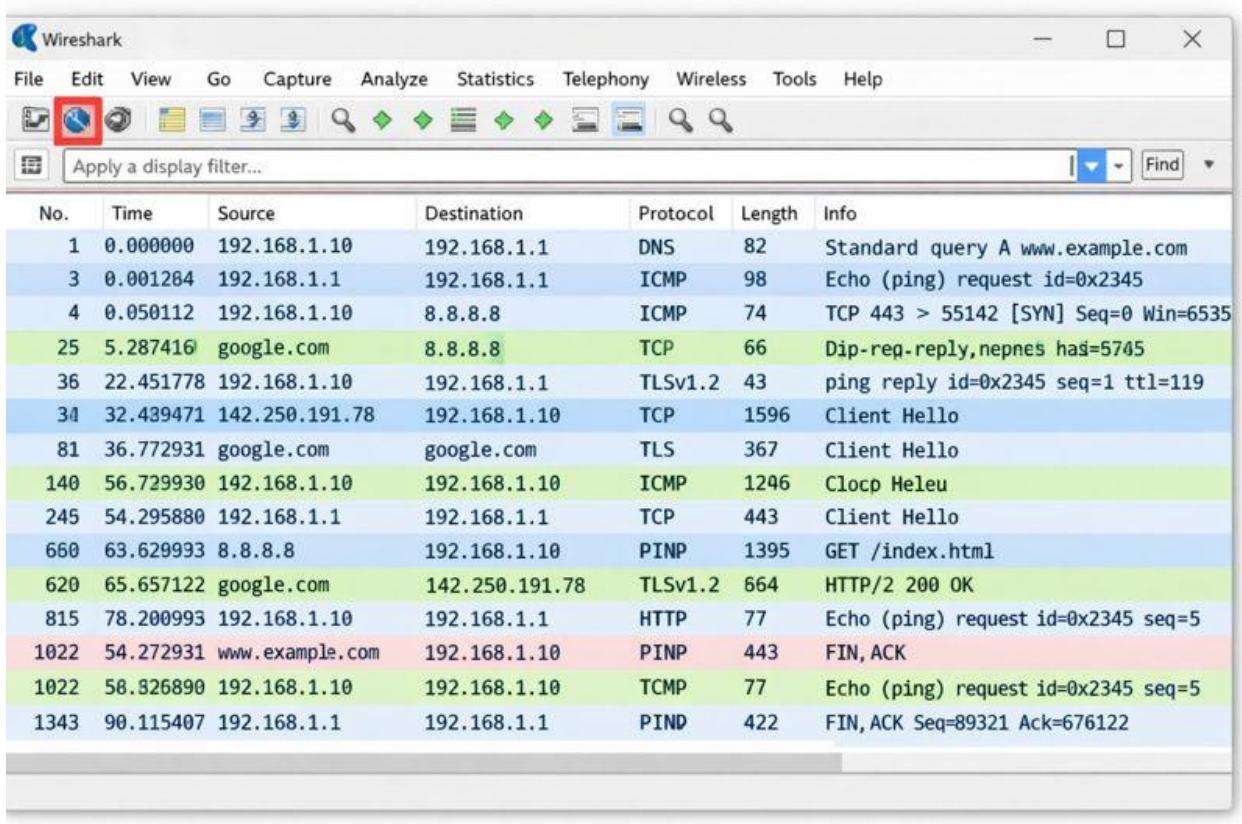


Figure 1Live packet capture showing multiple protocols.

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:

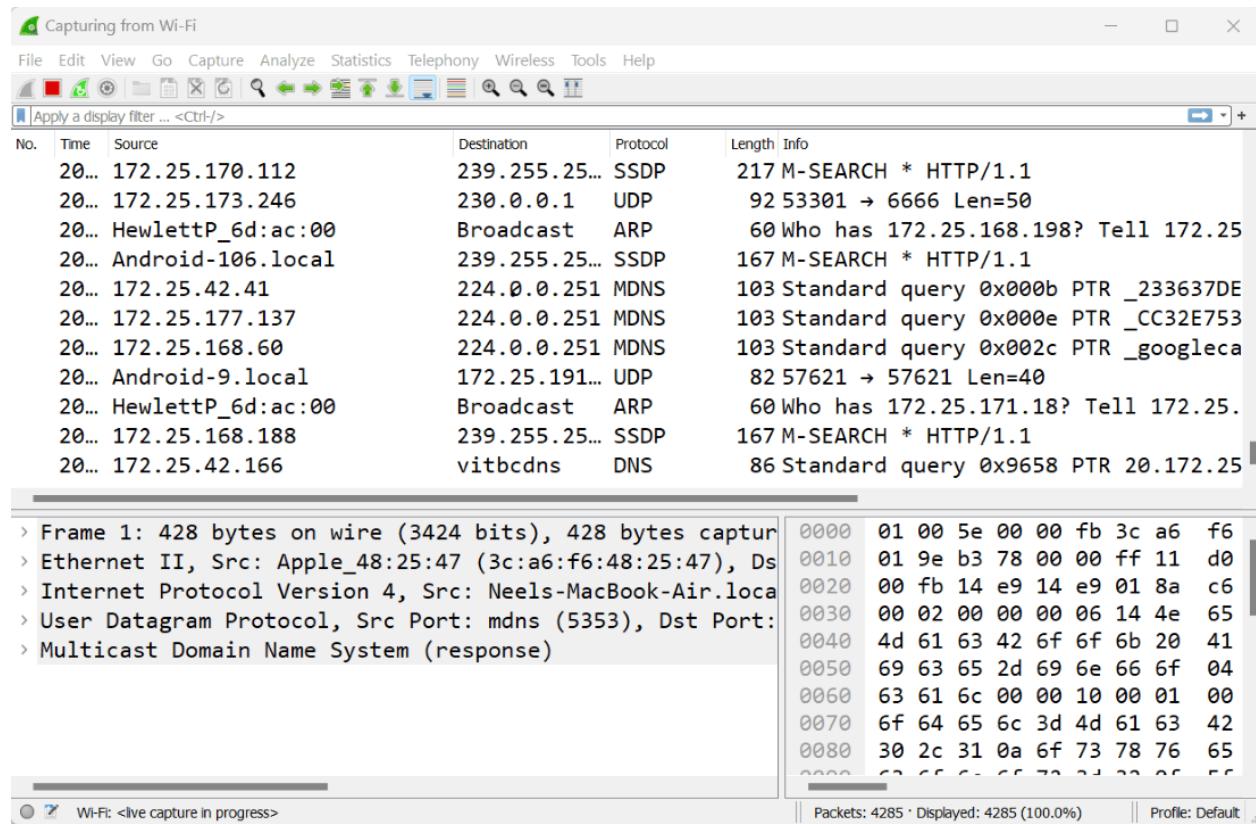


Figure 2 Packet list showing DNS, TCP, UDP, ARP traffic.

Wireshark-tutorial-column-setup.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http.request

| Time | Src | Src port | Dst | Dst port | Host | Info |
|---|--------------|----------|-----------------|----------|-------------------------|------|
| 2023-08-07 18:57:00 | 172.16.1.135 | 49670 | 23.47.50.79 | 80 | www.msftconnecttest.com | GET |
| 2023-08-07 18:57:20 | 172.16.1.135 | 63108 | 239.255.255.250 | 1900 | 239.255.255.250:1900 | M-S |
| 2023-08-07 18:57:21 | 172.16.1.135 | 63108 | 239.255.255.250 | 1900 | 239.255.255.250:1900 | M-S |
| 2023-08-07 18:57:22 | 172.16.1.135 | 63108 | 239.255.255.250 | 1900 | 239.255.255.250:1900 | M-S |
| 2023-08-07 18:57:23 | 172.16.1.135 | 63108 | 239.255.255.250 | 1900 | 239.255.255.250:1900 | M-S |
| 2023-08-07 18:57:37 | 172.16.1.135 | 49694 | 146.190.62.39 | 80 | httpforever.com | GET |
| 2023-08-07 18:57:37 | 172.16.1.135 | 49694 | 146.190.62.39 | 80 | httpforever.com | GET |
| 2023-08-07 18:57:37 | 172.16.1.135 | 49694 | 146.190.62.39 | 80 | httpforever.com | GET |
| 2023-08-07 18:57:37 | 172.16.1.135 | 49694 | 146.190.62.39 | 80 | httpforever.com | GET |
| 2023-08-07 18:57:37 | 172.16.1.135 | 49694 | 146.190.62.39 | 80 | httpforever.com | GET |
| 2023-08-07 18:57:38 | 172.16.1.135 | 49694 | 146.190.62.39 | 80 | httpforever.com | GET |
| 2023-08-07 18:57:38 | 172.16.1.135 | 49703 | 146.190.62.39 | 80 | httpforever.com | GET |
| Frame 25: 165 bytes on wire (1320 bits), 165 bytes captured (1320 bits) | | | | | | |
| › Ethernet II, Src: HewlettP_e4:6c:5a (08:2e:f5:e4:6c:5a), Dst: Cisco-Li_81:4c:67 (00:14:bf:81:00:00) | | | | | | |
| › Internet Protocol Version 4, Src: 172.16.1.135, Dst: 23.47.50.79 | | | | | | |
| › Transmission Control Protocol, Src Port: 49670, Dst Port: 80, Seq: 1, Ack: 1, Len: 111 | | | | | | |
| ‐ Hypertext Transfer Protocol | | | | | | |
| › GET /connecttest.txt HTTP/1.1\r\n | | | | | | |
| Connection: Close\r\n | | | | | | |
| User-Agent: Microsoft NCSI\r\n | | | | | | |
| Host: www.msftconnecttest.com\r\n | | | | | | |
| \r\n | | | | | | |
| HTTP Host (http.host), 31 bytes | | | | | | |
| Packets: 4065 · Displayed: 15 (0.4%) · Profile: Customized | | | | | | |

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Filter Display

No. Time Source Destination Protocol Length Info

| | | | | | | |
|---|---------------|-----------------|-----------------|---------|---|--|
| 1 | 0.000000000 | 35.163.37.142 | 10.143.90.229 | TLSV1.2 | 99 Application Data | |
| 2 | 0.000039815 | 10.143.90.229 | 35.163.37.142 | TCP | 68 33726 → 443 [ACK] Seq=1 Ack=32 Win=501 Len=0 TSval=1825849437 TSec | |
| 3 | 0.000209142 | 10.143.90.229 | 35.163.37.142 | TLSV1.2 | 183 Application Data | |
| 4 | 0.315105920 | 35.163.37.142 | 10.143.90.229 | TCP | 68 443 → 33726 [ACK] Seq=32 Ack=36 Win=119 Len=0 TSval=3716055745 TSe | |
| 5 | 3.623407673 | 10.143.90.229 | 142.250.194.266 | TLSV1.2 | 810 Application Data | |
| 6 | 3.623484600 | 10.143.90.229 | 142.250.194.266 | TLSV1.2 | 131 Application Data | |
| 7 | 3.623986120 | 142.250.194.266 | 10.143.90.229 | TCP | 68 443 → 47962 [ACK] Seq=1 Ack=743 Win=0 Len=0 TSval=1358112876 TS | |
| 8 | 3.623986378 | 142.250.194.266 | 10.143.90.229 | TCP | 68 443 → 47962 [ACK] Seq=1 Ack=806 Win=0 Len=0 TSval=1358112876 TS | |
| 9 | 3.90151521379 | 142.250.194.266 | 10.143.90.229 | TLSV1.2 | 589 Application Data | |
| 10 | 3.90151561934 | 10.143.90.229 | 142.250.194.266 | TCP | 68 47962 → 443 [ACK] Seq=806 Ack=522 Win=501 Len=0 TSval=878142976 TS | |
| 11 | 3.901521765 | 142.250.194.266 | 10.143.90.229 | TLSV1.2 | 373 Application Data, Application Data | |
| Frame 139: 387 bytes on wire (3096 bits), 387 bytes captured (3096 bits) on interface any, id 0 | | | | | | |
| Linux cooked capture v1 | | | | | | |
| Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255 | | | | | | |
| User Datagram Protocol, Src Port: 68, Dst Port: 67 | | | | | | |
| Dynamic Host Configuration Protocol (Request) | | | | | | |
| Hexadecimal Display | | | | | | |

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:

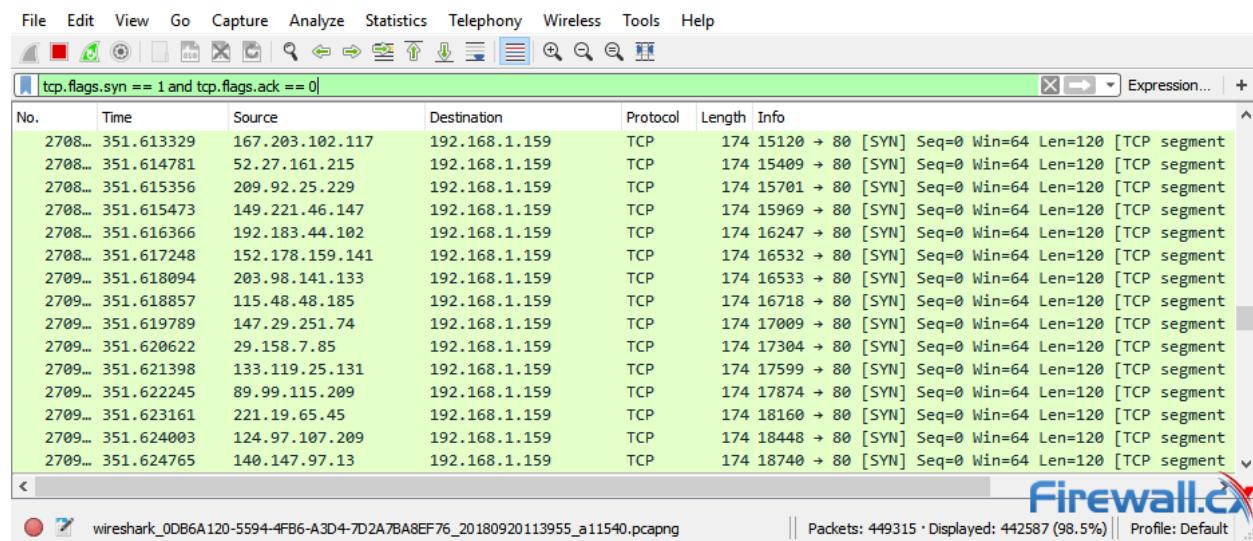


Figure 3. TCP 3-Way Handshake

```

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]

```

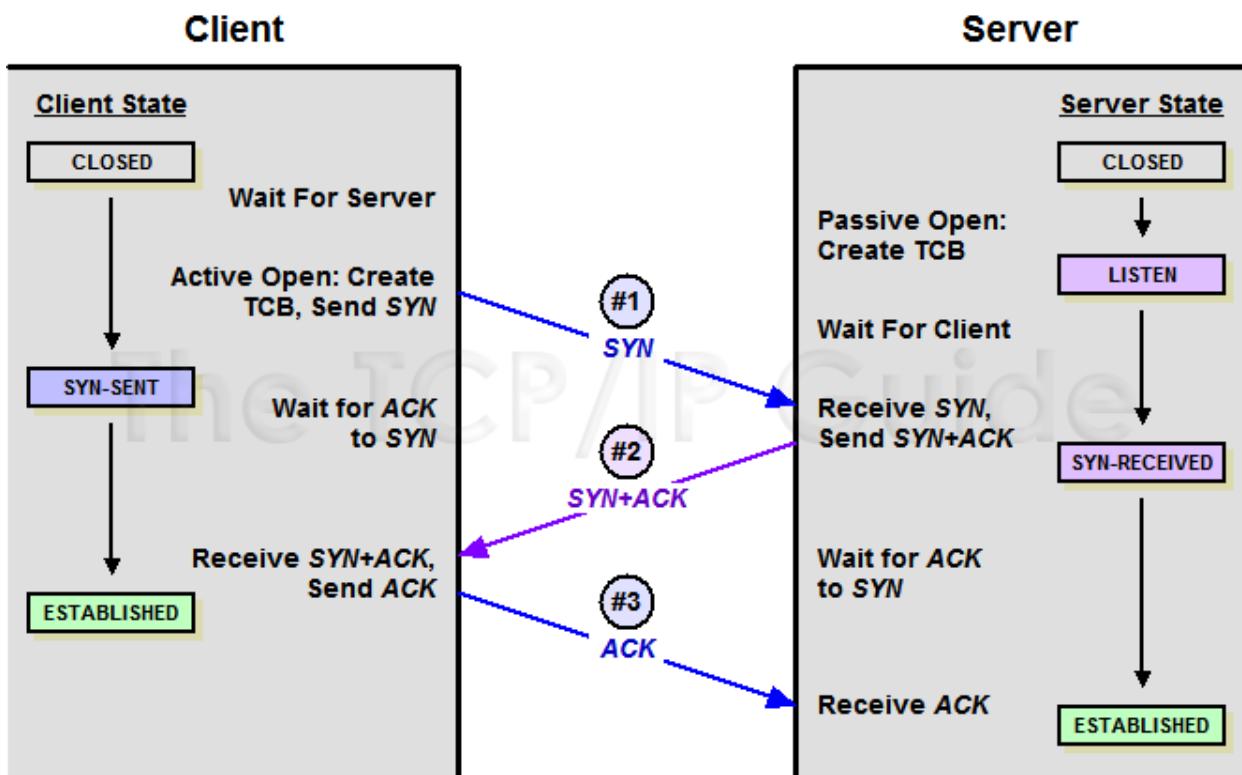


Figure 4 TCP packets showing SYN, SYN-ACK, ACK flags.

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:

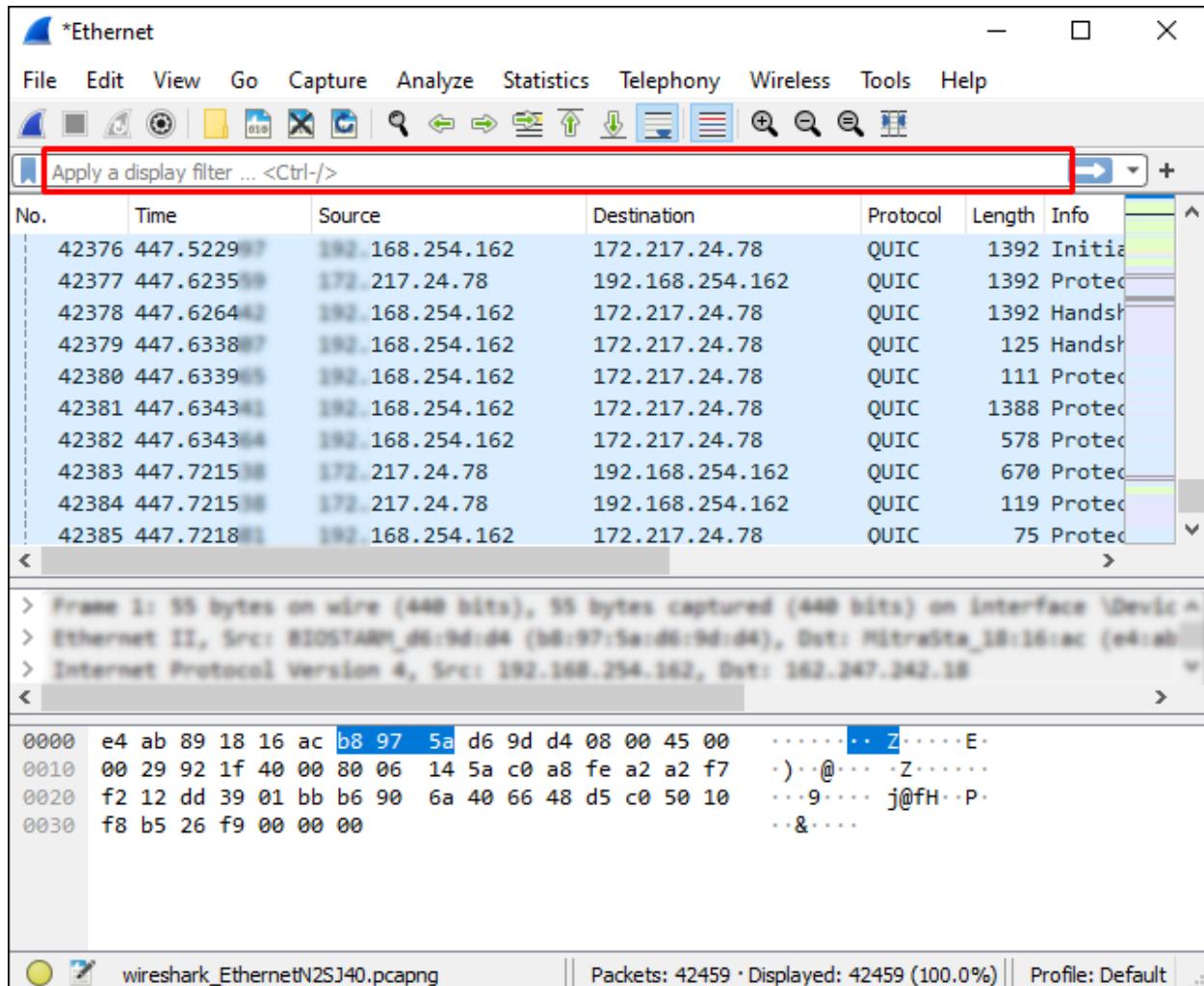


Figure 5 Filter bar showing a protocol filter applied.

Wireshark interface showing an ARP request frame (Frame 32) highlighted in yellow. The frame details are as follows:

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|--------------|----------------|-------------|----------|--------|---|
| 69 | 3.483300659 | HP_cc:d6:b9 | Broadcast | ARP | 60 | Who has 10.143.90.167? Tell 10.143.90.23 |
| 122 | 5.517743097 | Cisco_27:d3:69 | Broadcast | ARP | 60 | Who has 10.143.90.142? Tell 10.143.90.102 |
| 161 | 8.530076099 | Cisco_27:d3:69 | Broadcast | ARP | 60 | Who has 10.143.90.14? Tell 10.143.90.102 |
| 176 | 9.817521713 | HP_cc:d6:b9 | Broadcast | ARP | 60 | Who has 10.143.90.167? Tell 10.143.90.23 |
| 180 | 10.474630480 | HP_cc:d6:b9 | Broadcast | ARP | 60 | Who has 10.143.90.167? Tell 10.143.90.23 |
| 187 | 11.483681187 | HP_cc:d6:b9 | Broadcast | ARP | 60 | Who has 10.143.90.167? Tell 10.143.90.23 |
| 191 | 12.546492084 | Cisco_27:d3:69 | Broadcast | ARP | 60 | Who has 10.143.90.14? Tell 10.143.90.102 |
| 291 | 20.565873141 | Cisco_27:d3:69 | Broadcast | ARP | 60 | Who has 10.143.90.14? Tell 10.143.90.102 |
| 613 | 45.626446961 | Cisco_27:d3:69 | Broadcast | ARP | 60 | Who has 10.143.90.123? Tell 10.143.90.102 |
| 633 | 48.633902902 | Cisco_27:d3:69 | Broadcast | ARP | 60 | Who has 10.143.90.123? Tell 10.143.90.102 |
| 675 | 52.634044505 | Cisco_27:d3:69 | Broadcast | ARP | 60 | Who has 10.143.90.123? Tell 10.143.90.102 |

Frame details:

- Frame 32: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0, id 0
- Ethernet II, Src: HP_cc:d6:b9 (e0:70:ea:cc:d6:b9), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
- Address Resolution Protocol (request)

Wireshark interface showing a context menu for a selected SMTP packet (Time: 2023-03-16 15:33:39). The menu options include:

- Mark/Unmark Packet
- Ignore/Unignore Packet
- Set/Unset Time Reference
- Time Shift...
- Packet Comments
- Edit Resolved Name
- Apply as Filter
- Prepare as Filter
- Conversation Filter
- Colorize Conversation
- SCTP
- Follow**
- Copy
- Protocol Preferences
- Decode As...
- Show Packet in New Window

The "Follow" option is highlighted. A secondary context menu for "TCP Stream" is displayed, listing:

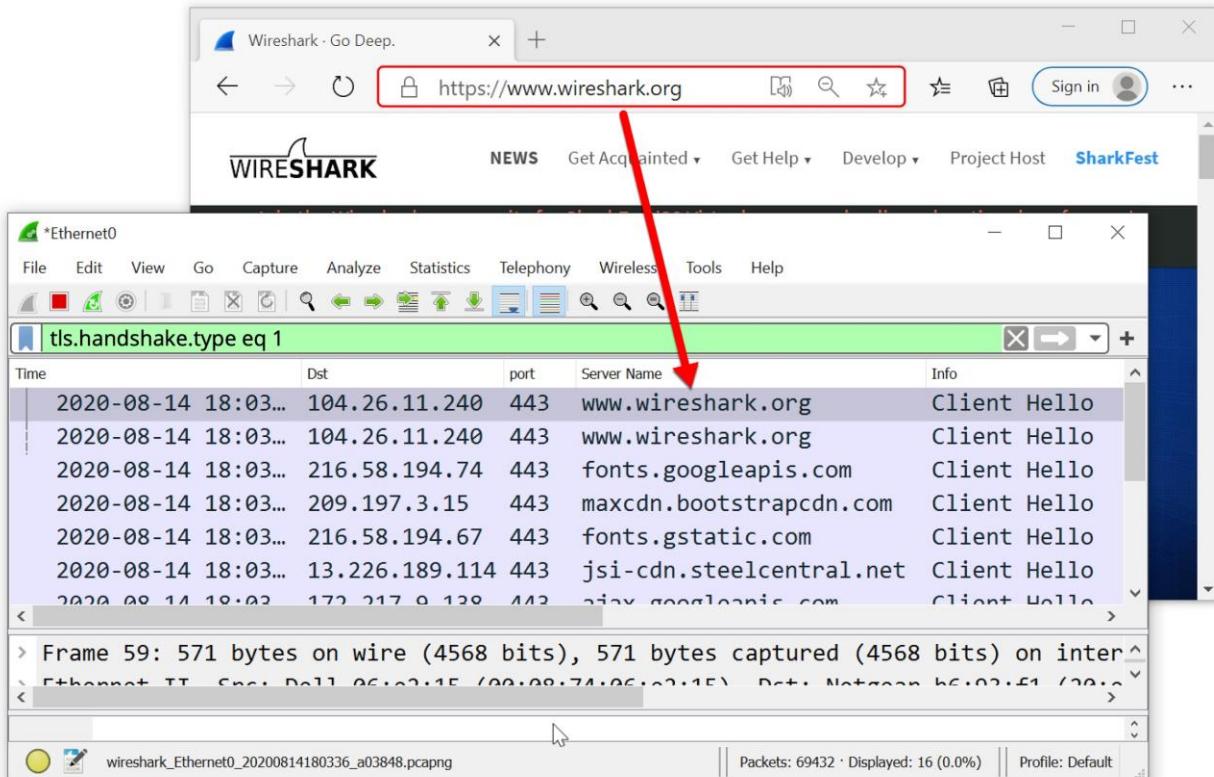
- TCP Stream
- UDP Stream
- DCCP Stream
- TLS Stream
- HTTP Stream
- HTTP/2 Stream
- QUIC Stream
- SIP Call

A tooltip for the TCP Stream option shows: "Selected: 6994 · Discarded: 7 (0.1%) · Profile: Customized".

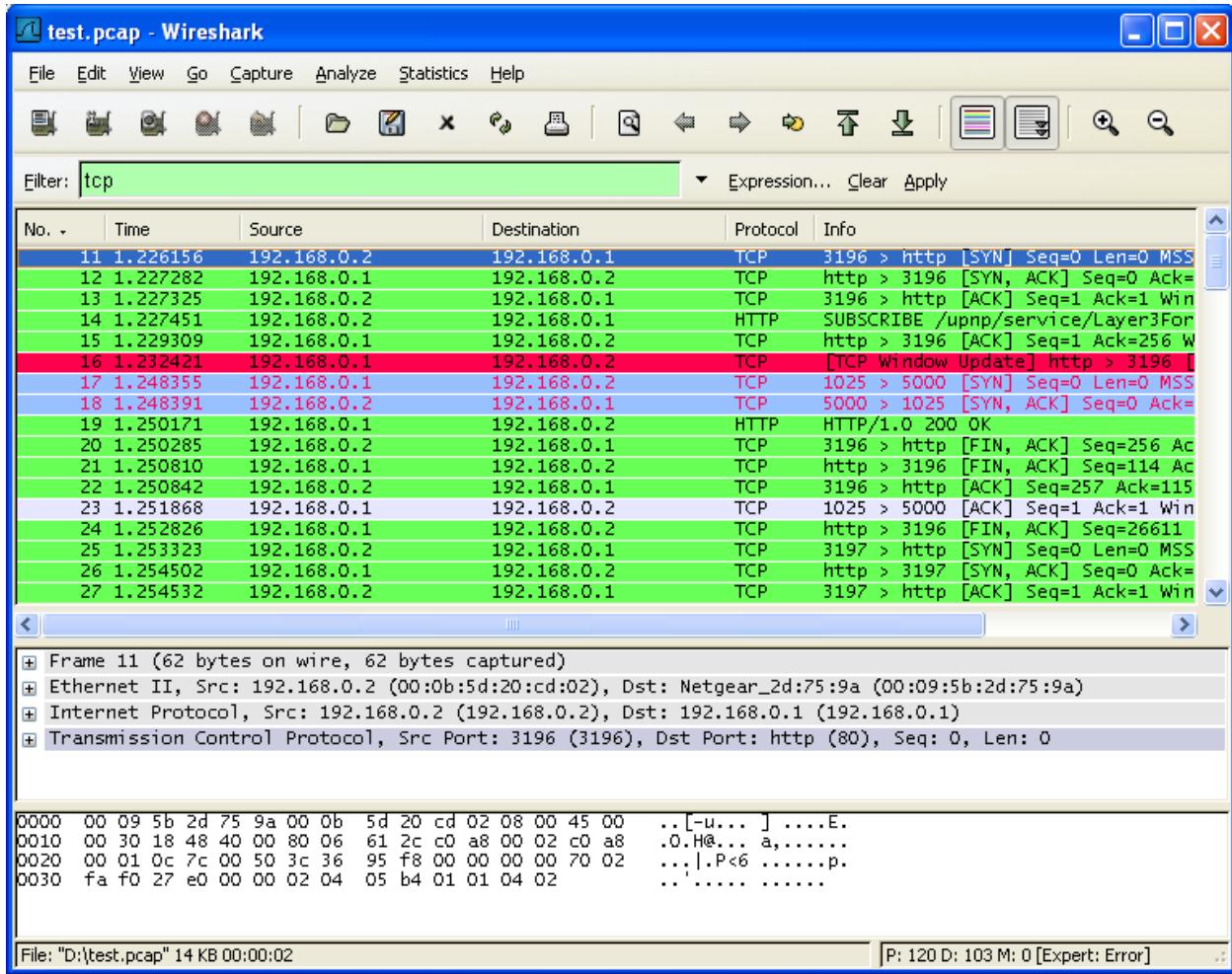
6. HTTPS Traffic Analysis

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

Screenshot to include:



| No. | Time | Source | Destination | Protocol | Length | Host | Server Name | Info |
|-----|----------|---------------|---------------|----------|--------|------------------|-------------|--|
| 623 | 7.369528 | 10.38.148.209 | 165.225.72.38 | TCP | 66 | | | 55655 → 443 [SYN] Seq=0 Win=64246 |
| 665 | 7.515597 | 165.225.72.38 | 10.38.148.209 | TCP | 66 | | | 443 → 55655 [SYN, ACK] Seq=0 Ack=1 Win=64246 |
| 666 | 7.515650 | 10.38.148.209 | 165.225.72.38 | TCP | 54 | | | 55655 → 443 [ACK] Seq=1 Ack=1 Win=64246 |
| 667 | 7.515942 | 10.38.148.209 | 165.225.72.38 | HTTP | 656 | www.facebook.com | | CONNECT www.facebook.com:443 HTTP/1.1 200 Connection Established |
| 720 | 7.666818 | 165.225.72.38 | 10.38.148.209 | HTTP | 119 | | | HTTP/1.1 200 Connection Established |
| 721 | 7.661299 | 10.38.148.209 | 165.225.72.38 | TCP | 571 | | | 55655 → 443 [PSH, ACK] Seq=603 Ack=662 |
| 763 | 7.812824 | 165.225.72.38 | 10.38.148.209 | TCP | 1514 | | | 443 → 55655 [ACK] Seq=66 Ack=1126 |
| 764 | 7.812826 | 165.225.72.38 | 10.38.148.209 | TCP | 1514 | | | 443 → 55655 [ACK] Seq=1526 Ack=1126 |
| 765 | 7.812830 | 165.225.72.38 | 10.38.148.209 | TCP | 775 | | | 443 → 55655 [PSH, ACK] Seq=2986 Ack=1126 |
| 766 | 7.812982 | 10.38.148.209 | 165.225.72.38 | TCP | 54 | | | 55655 → 443 [ACK] Seq=1120 Ack=378 |
| 767 | 7.819045 | 10.38.148.209 | 165.225.72.38 | TCP | 372 | | | 55655 → 443 [PSH, ACK] Seq=1120 Ack=378 |
| 804 | 7.966072 | 165.225.72.38 | 10.38.148.209 | TCP | 105 | | | 443 → 55655 [PSH, ACK] Seq=3707 Ack=443 |
| 805 | 7.966575 | 10.38.148.209 | 165.225.72.38 | TCP | 1079 | | | 55655 → 443 [PSH, ACK] Seq=1438 Ack=443 |
| 845 | 8.199619 | 165.225.72.38 | 10.38.148.209 | TCP | 54 | | | 443 → 55655 [ACK] Seq=3758 Ack=2443 |
| 887 | 8.408213 | 165.225.72.38 | 10.38.148.209 | TCP | 1514 | | | 443 → 55655 [ACK] Seq=3758 Ack=2443 |
| 888 | 8.408214 | 165.225.72.38 | 10.38.148.209 | TCP | 1514 | | | 443 → 55655 [ACK] Seq=5218 Ack=2443 |
| 889 | 8.408216 | 165.225.72.38 | 10.38.148.209 | TCP | 1514 | | | 443 → 55655 [ACK] Seq=6678 Ack=2443 |
| 890 | 8.408216 | 165.225.72.38 | 10.38.148.209 | TCP | 1514 | | | 443 → 55655 [ACK] Seq=8138 Ack=2443 |
| 891 | 8.408217 | 165.225.72.38 | 10.38.148.209 | TCP | 1514 | | | 443 → 55655 [ACK] Seq=9598 Ack=2443 |
| 892 | 8.408217 | 165.225.72.38 | 10.38.148.209 | TCP | 1514 | | | 443 → 55655 [ACK] Seq=11058 Ack=2443 |
| 893 | 8.408316 | 10.38.148.209 | 165.225.72.38 | TCP | 54 | | | 55655 → 443 [ACK] Seq=2463 Ack=12443 |
| 899 | 8.557888 | 165.225.72.38 | 10.38.148.209 | TCP | 1514 | | | 443 → 55655 [ACK] Seq=12518 Ack=12443 |
| 900 | 8.557889 | 165.225.72.38 | 10.38.148.209 | TCP | 1514 | | | 443 → 55655 [ACK] Seq=13978 Ack=2443 |
| 901 | 8.557811 | 165.225.72.38 | 10.38.148.209 | TCP | 1514 | | | 443 → 55655 [ACK] Seq=15438 Ack=2443 |



7. DNS Traffic Analysis

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

Screenshot to include:

```

> Ethernet II, Src: Cisco_a5:8d:69 (70:01:b5:a5:8d:69), Dst: Dell_91:ff:73 (70:b5:e8:91:ff:73)
> Internet Protocol Version 4, Src: 10.143.70.254, Dst: 10.143.90.167
> User Datagram Protocol, Src Port: 53, Dst Port: 50302
> Domain Name System (response)
  Transaction ID: 0x7b13
  Flags: 0x8180 Standard query response, No error
  Questions: 1
  Answer RRs: 7
  Authority RRs: 0
  Additional RRs: 0
  > Queries
  > Answers
    [Unsolicited: True]

0020  5a a7 00 35 c4 7e 00 a4 61 02 7b 13 81 80 00 01 Z...5~... a[.....
0030  00 07 00 00 00 00 07 68 69 73 74 6f 72 79 06 67 ....h istory.g
0040  6f 6f 67 6c 65 03 63 6f 6d 00 00 01 00 01 c0 0c oogle.co m.....
0050  00 05 00 01 00 00 00 08 00 0c 07 68 69 73 74 6f ..... .histo
0060  72 79 01 6c c0 14 c0 30 00 01 00 01 00 00 01 1f ry.l...0 .....
0070  00 04 4a 7d 44 8a c0 30 00 01 00 01 00 00 01 1f ..J}D..0 .....
0080  00 04 4a 7d 44 64 c0 30 00 01 00 01 00 00 01 1f ..J}Dd..0 .....
0090  00 04 4a 7d 44 65 c0 30 00 01 00 01 00 00 01 1f ..J}De..0 .....
00a0  00 04 4a 7d 44 71 c0 30 00 01 00 01 00 00 01 1f ..J}Dq..0 .....

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:

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|--------------|----------------|-------------|----------|--------|---|
| 69 | 3.483300659 | HP_cc:d6:b9 | Broadcast | ARP | 60 | Who has 10.143.90.167? Tell 10.143.90.23 |
| 122 | 5.51743097 | Cisco_27:d3:69 | Broadcast | ARP | 60 | Who has 10.143.90.142? Tell 10.143.90.102 |
| 161 | 8.530076099 | Cisco_27:d3:69 | Broadcast | ARP | 60 | Who has 10.143.90.14? Tell 10.143.90.102 |
| 176 | 9.817521713 | HP_cc:d6:b9 | Broadcast | ARP | 60 | Who has 10.143.90.167? Tell 10.143.90.23 |
| 180 | 10.474630480 | HP_cc:d6:b9 | Broadcast | ARP | 60 | Who has 10.143.90.167? Tell 10.143.90.23 |
| 187 | 11.483681187 | HP_cc:d6:b9 | Broadcast | ARP | 60 | Who has 10.143.90.167? Tell 10.143.90.23 |
| 191 | 12.546492084 | Cisco_27:d3:69 | Broadcast | ARP | 60 | Who has 10.143.90.14? Tell 10.143.90.102 |
| 291 | 20.565873141 | Cisco_27:d3:69 | Broadcast | ARP | 60 | Who has 10.143.90.14? Tell 10.143.90.102 |
| 613 | 45.626446961 | Cisco_27:d3:69 | Broadcast | ARP | 60 | Who has 10.143.90.123? Tell 10.143.90.102 |
| 633 | 48.633902902 | Cisco_27:d3:69 | Broadcast | ARP | 60 | Who has 10.143.90.123? Tell 10.143.90.102 |
| 675 | 52.634044505 | Cisco_27:d3:69 | Broadcast | ARP | 60 | Who has 10.143.90.123? Tell 10.143.90.102 |

> Frame 32: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0, id 0
 > Ethernet II, Src: HP_cc:d6:b9 (e0:70:ea:cc:d6:b9), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 > Address Resolution Protocol (request)

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.**