



COMPUTER NETWORK

Packet Analysis Using Wireshark

CODEINTERN

Rimsha Bibi

CI/DEC116

28-DEC-2025

Objective

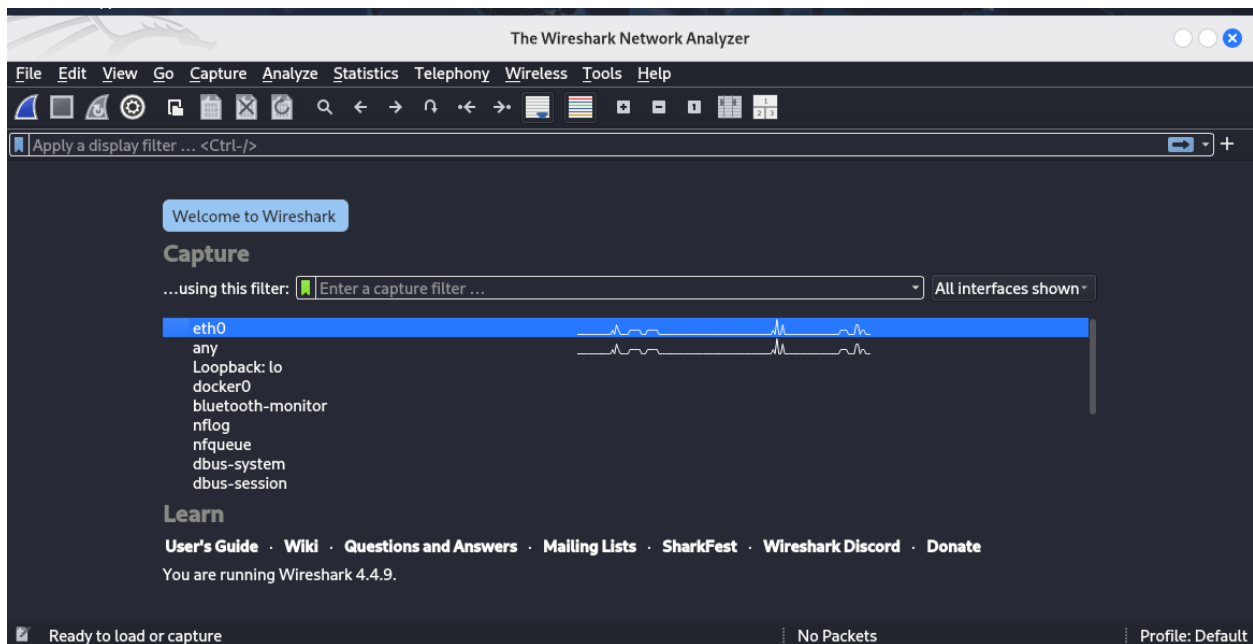
To analyze network traffic using **Wireshark** by capturing and identifying different protocols (HTTPS, DNS, ARP, TCP, UDP), understanding TCP connection establishment, and applying packet filters.

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:

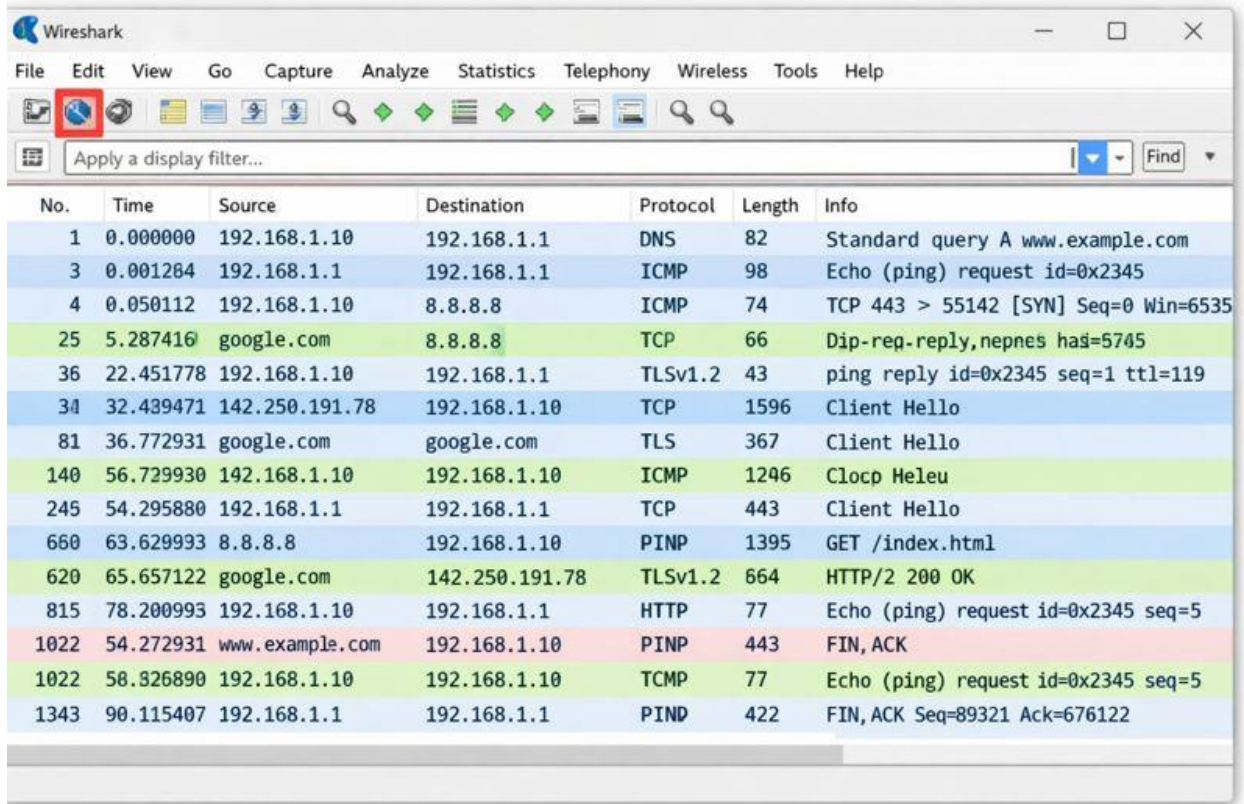


Figure 1Live packet capture showing multiple protocols.

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:

The screenshot shows the Wireshark interface with a packet list on the left and a packet details pane on the right. The packet list contains 12 entries, including SSDP, ARP, and DNS traffic. The selected packet (No. 12) is a DNS response, and the details pane shows the frame structure and raw data.

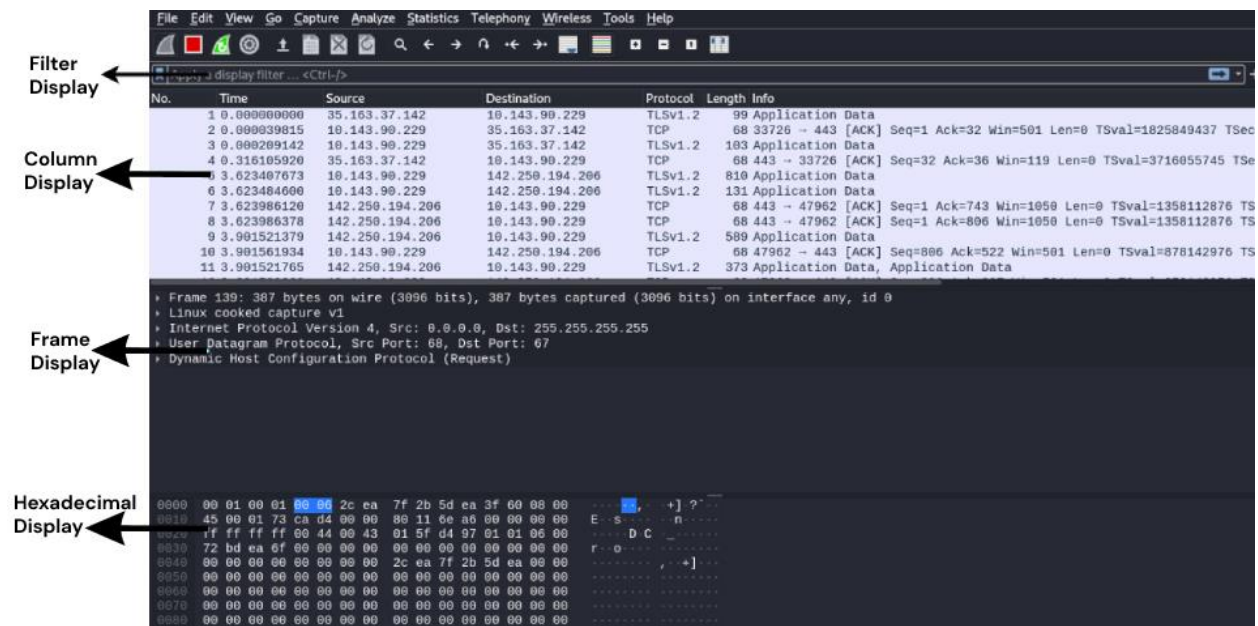
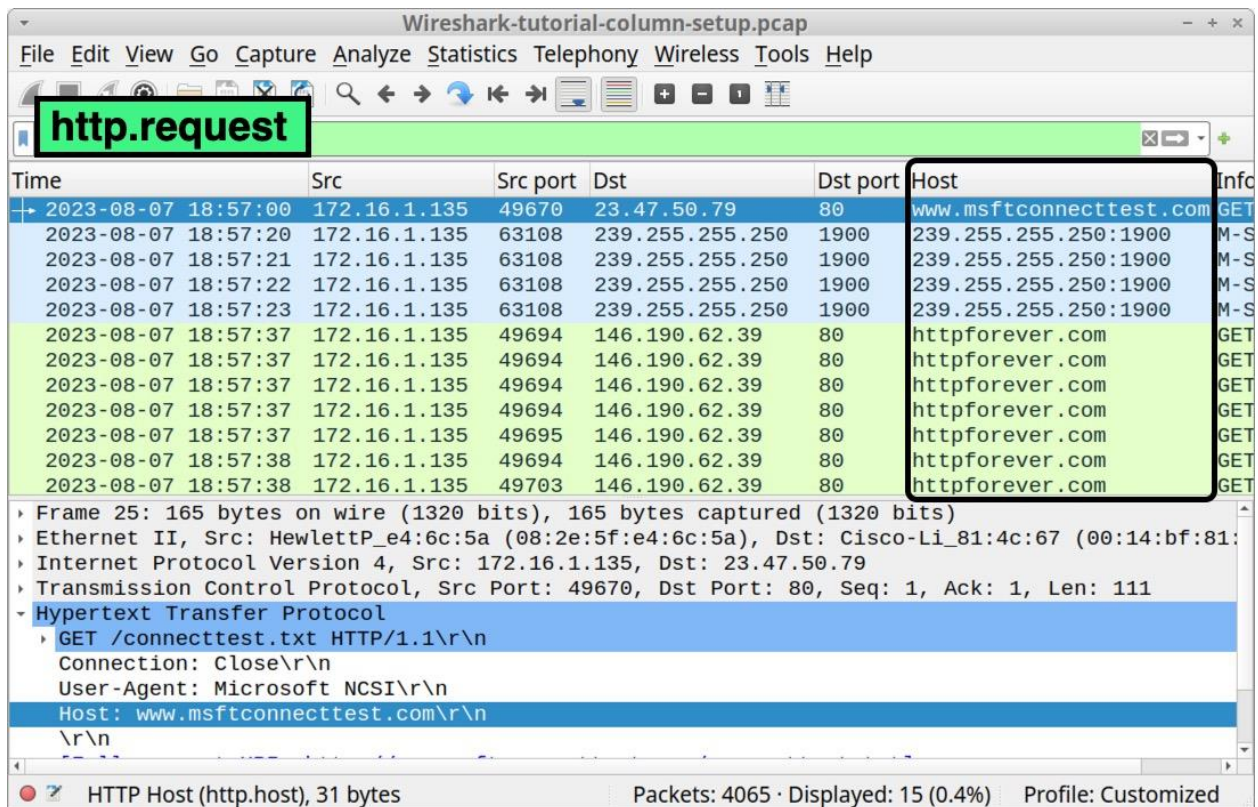
No.	Time	Source	Destination	Protocol	Length	Info
20...	172.25.170.112	239.255.25...	SSDP	217	M-SEARCH * HTTP/1.1	
20...	172.25.173.246	230.0.0.1	UDP	92	53301 → 6666 Len=50	
20...	HewlettP_6d:ac:00	Broadcast	ARP	60	Who has 172.25.168.198? Tell 172.25.	
20...	Android-106.local	239.255.25...	SSDP	167	M-SEARCH * HTTP/1.1	
20...	172.25.42.41	224.0.0.251	MDNS	103	Standard query 0x000b PTR _233637DE	
20...	172.25.177.137	224.0.0.251	MDNS	103	Standard query 0x000e PTR _CC32E753	
20...	172.25.168.60	224.0.0.251	MDNS	103	Standard query 0x002c PTR _googleca	
20...	Android-9.local	172.25.191...	UDP	82	57621 → 57621 Len=40	
20...	HewlettP_6d:ac:00	Broadcast	ARP	60	Who has 172.25.171.18? Tell 172.25.	
20...	172.25.168.188	239.255.25...	SSDP	167	M-SEARCH * HTTP/1.1	
20...	172.25.42.166	vitbcdns	DNS	86	Standard query 0x9658 PTR 20.172.25	

Frame 12: 428 bytes on wire (3424 bits), 428 bytes captured (3424 bits) on interface 0
Ethernet II, Src: Apple_48:25:47 (3c:a6:f6:48:25:47), Dst: Neels-MacBook-Air_08:00:27:8c:6f:11 (08:00:27:8c:6f:11)
Internet Protocol Version 4, Src: Neels-MacBook-Air.local (172.25.42.166), Dst: vitbcdns (172.25.191.1)
User Datagram Protocol, Src Port: mdns (5353), Dst Port: mdns (5353)
Multicast Domain Name System (response)

0000 01 00 5e 00 00 fb 3c a6 f6
0010 01 9e b3 78 00 00 ff 11 d0
0020 00 fb 14 e9 14 e9 01 8a c6
0030 00 02 00 00 00 06 14 4e 65
0040 4d 61 63 42 6f 6f 6b 20 41
0050 69 63 65 2d 69 6e 66 6f 04
0060 63 61 6c 00 00 10 00 01 00
0070 6f 64 65 6c 3d 4d 61 63 42
0080 30 2c 31 0a 6f 73 78 76 65
0090 63 6f 6f 6f 73 73 73 73 73

Wi-Fi: <live capture in progress> | Packets: 4285 · Displayed: 4285 (100.0%) | Profile: Default

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:

No.	Time	Source	Destination	Protocol	Length	Info
2708...	351.613329	167.203.102.117	192.168.1.159	TCP	174	15120 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2708...	351.614781	52.27.161.215	192.168.1.159	TCP	174	15409 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2708...	351.615356	209.92.25.229	192.168.1.159	TCP	174	15701 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2708...	351.615473	149.221.46.147	192.168.1.159	TCP	174	15969 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2708...	351.616366	192.183.44.102	192.168.1.159	TCP	174	16247 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2708...	351.617248	152.178.159.141	192.168.1.159	TCP	174	16532 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709...	351.618094	203.98.141.133	192.168.1.159	TCP	174	16533 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709...	351.618857	115.48.48.185	192.168.1.159	TCP	174	16718 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709...	351.619789	147.29.251.74	192.168.1.159	TCP	174	17009 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709...	351.620622	29.158.7.85	192.168.1.159	TCP	174	17304 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709...	351.621398	133.119.25.131	192.168.1.159	TCP	174	17599 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709...	351.622245	89.99.115.209	192.168.1.159	TCP	174	17874 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709...	351.623161	221.19.65.45	192.168.1.159	TCP	174	18160 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709...	351.624003	124.97.107.209	192.168.1.159	TCP	174	18448 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709...	351.624765	140.147.97.13	192.168.1.159	TCP	174	18740 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment

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]

```

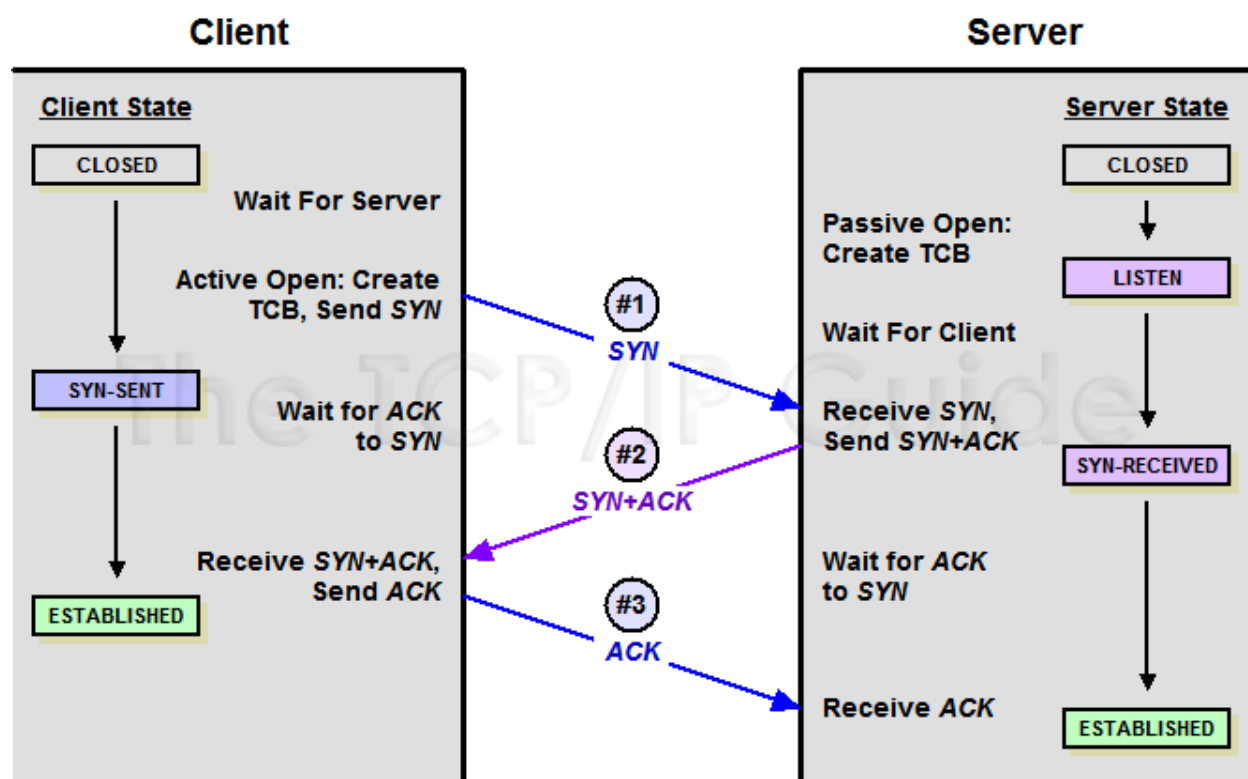


Figure 4 TCP packets showing SYN, SYN-ACK, ACK flags.

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:

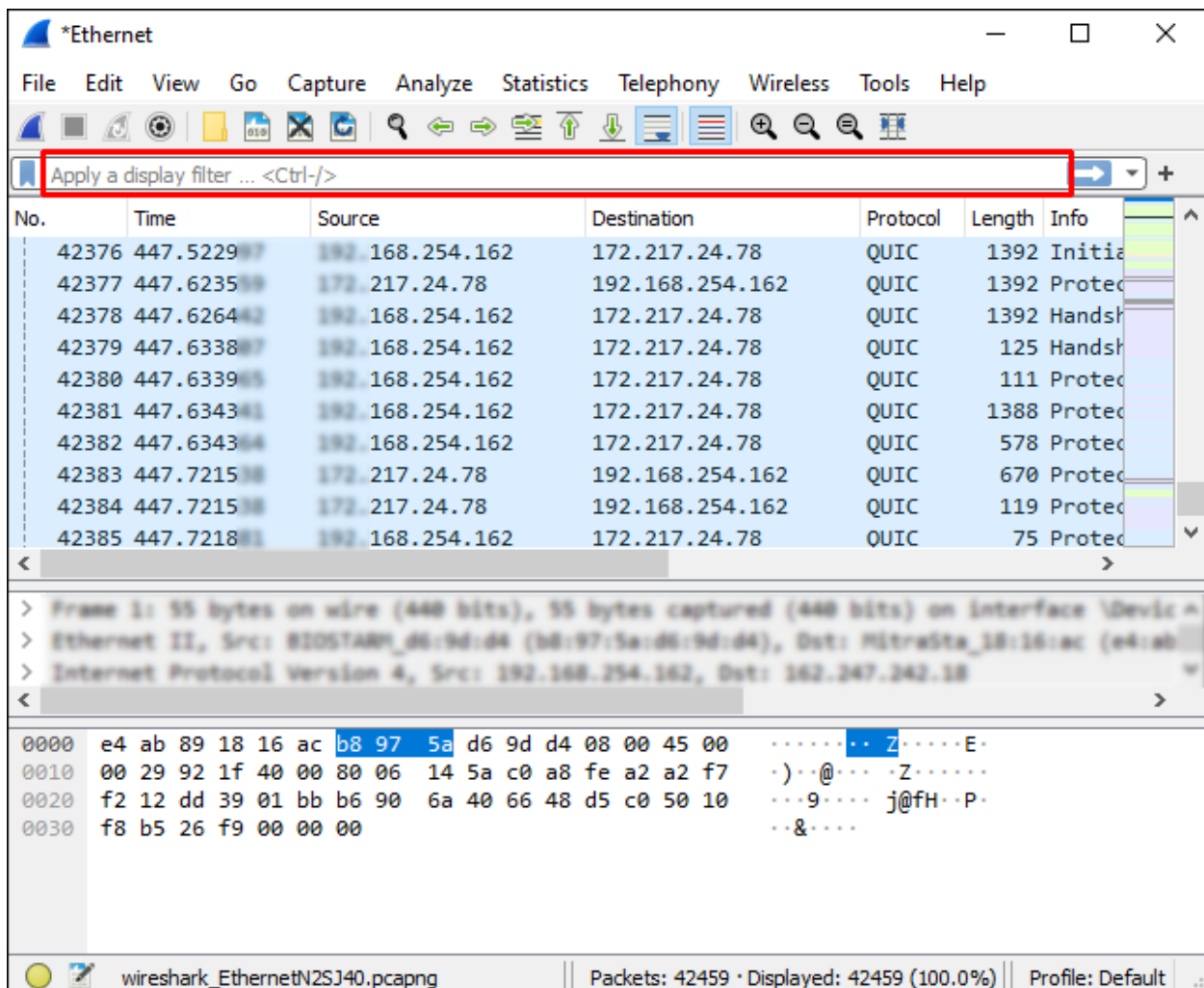


Figure 5 Filter bar showing a protocol filter applied.

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 10.143.90.14? 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)

Wireshark-tutorial-filter-expressions-5-of-5.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

smtp.data.fragment

Time	Dst	Dst port	Info
2023-03-16 15:33:33	27.121.3.241	587	from: =?UTF-8?B?PExp3RhIGRlIGRpc3RyYWJ1ac0nw6NvI
2023-03-16 15:33:34	27.121.3.241	587	from: "<Secretariado DSSA> seguranca.alimentar@dc
2023-03-16 15:33:35	27.121.3.241	587	from: =?UTF-8?B?PExp3RhIGRlIGRpc3RyYWJ1ac0nw6NvI
2023-03-16 15:33:36	27.121.3.241	587	from: "<Secretariado DSSA> seguranca.alimentar@dc
2023-03-16 15:33:37	27.121.3.241	587	from: =?UTF-8?B?PExp3RhIGRlIGRpc3RyYWJ1ac0nw6NvI
2023-03-16 15:33:38	27.121.3.241	587	from: "<Secretariado DSSA> seguranca.alimentar@dc
2023-03-16 15:33:39	27.121.3.241	587	from: =?UTF-8?B?PExp3RhIGRlIGRpc3RyYWJ1ac0nw6NvI

Mark/Unmark Packet Ctrl+M
 Ignore/Unignore Packet Ctrl+D
 Set/Unset Time Reference Ctrl+T
 Time Shift... Ctrl+Shift+T
 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

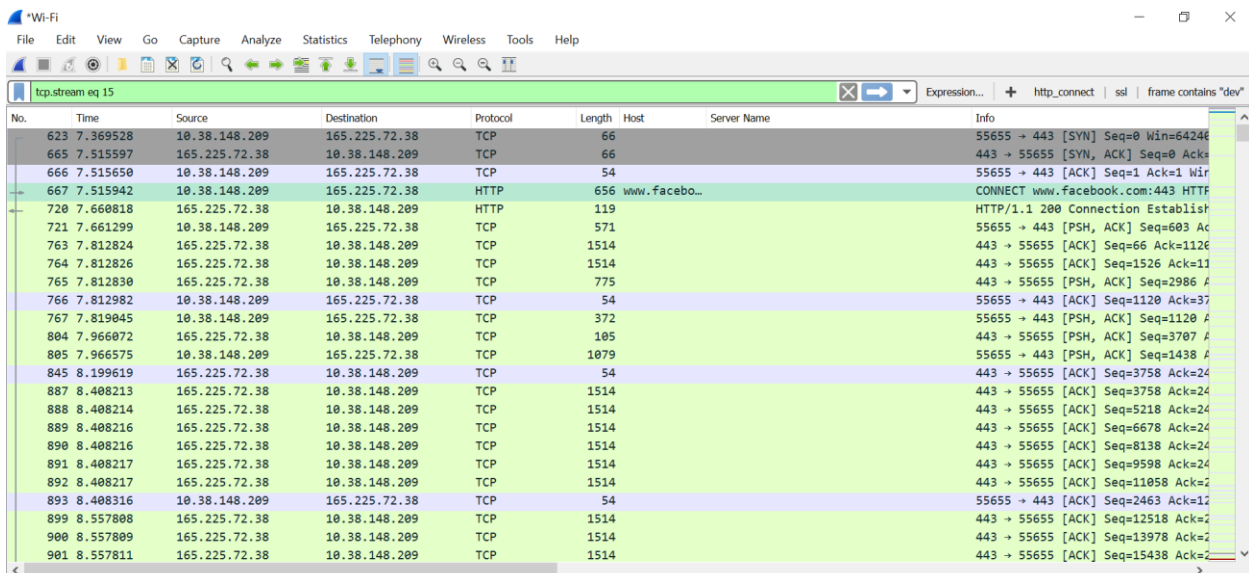
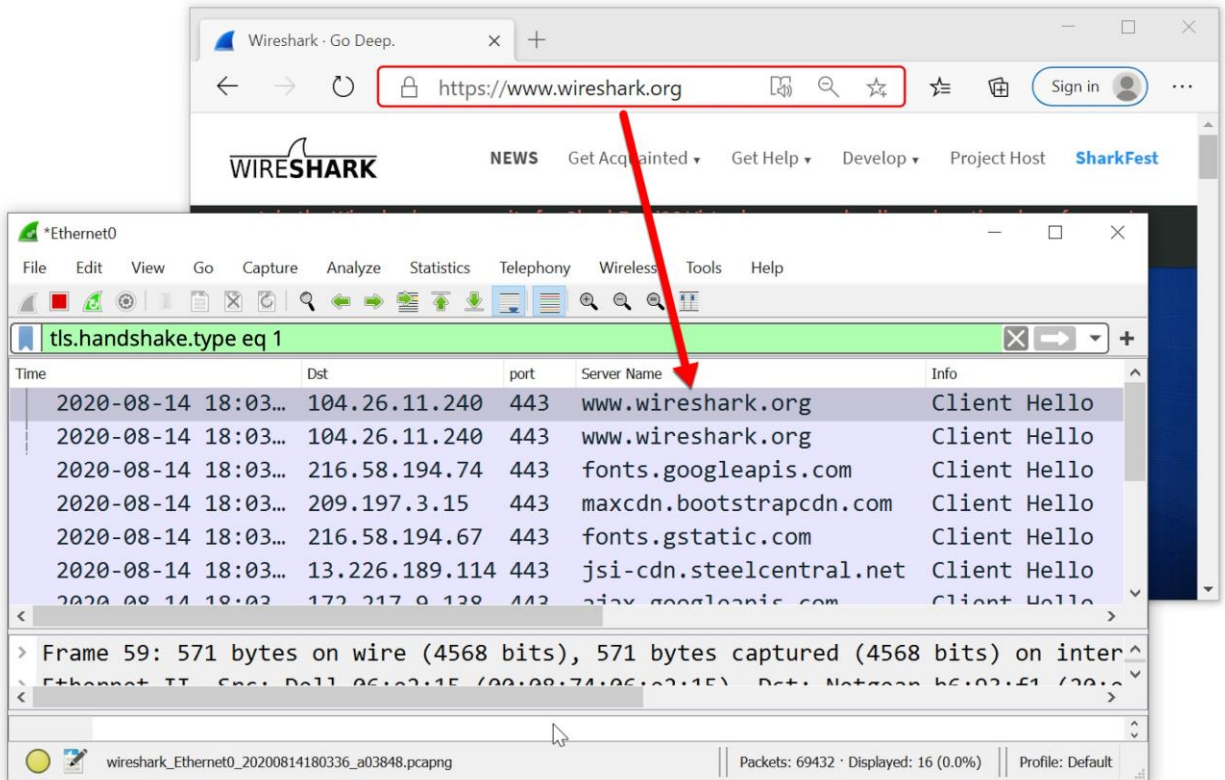
TCP Stream Ctrl+Alt+Shift+T
 UDP Stream Ctrl+Alt+Shift+U
 DCCP Stream Ctrl+Alt+Shift+E
 TLS Stream Ctrl+Alt+Shift+S
 HTTP Stream Ctrl+Alt+Shift+H
 HTTP/2 Stream
 QUIC Stream
 SIP Call

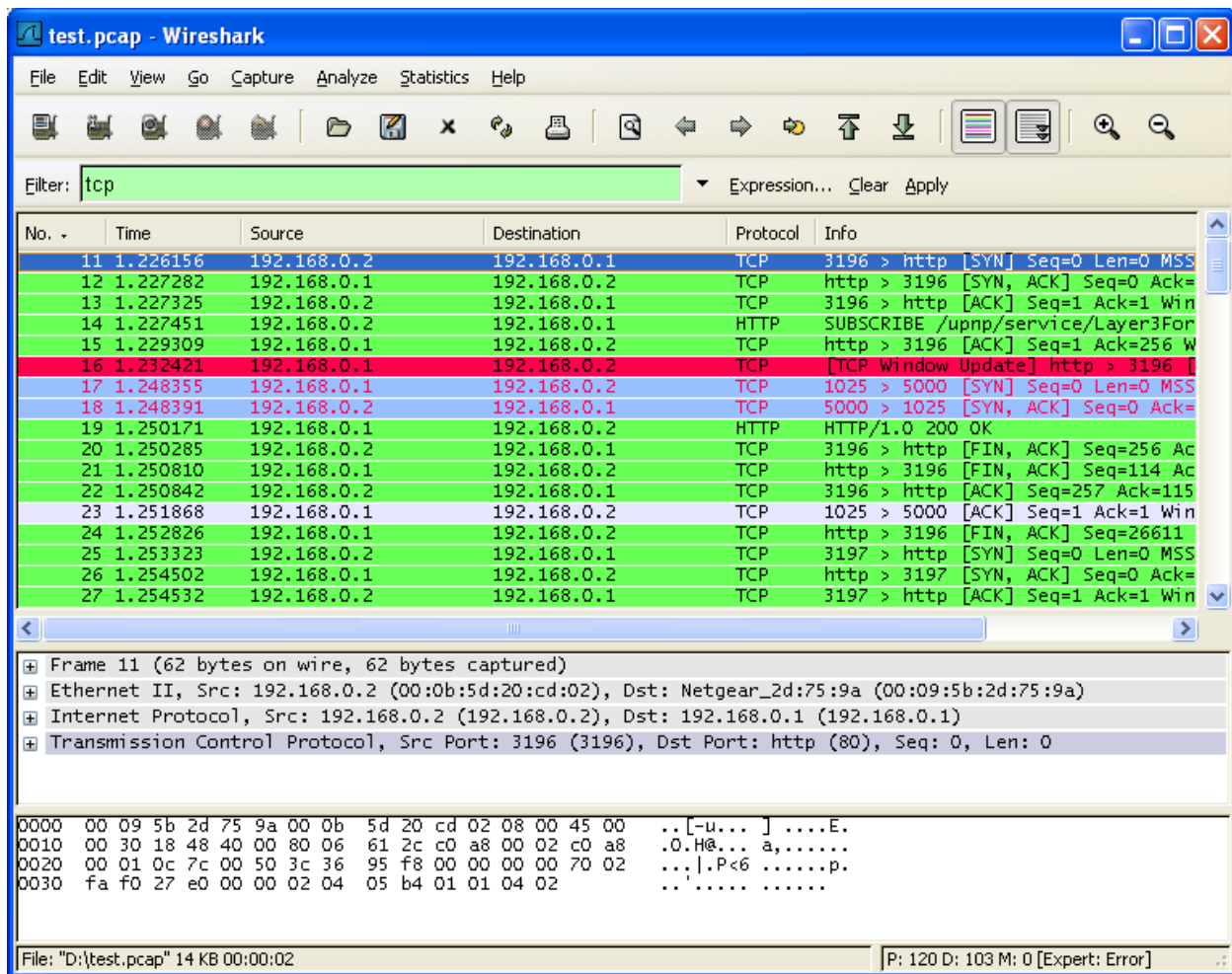
Packets: 6994 · Displayed: 7 (0.1%) Profile: Customized

6. HTTPS Traffic Analysis

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

Screenshot to include:





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: 10.143.90.254, 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 history.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    5.517743097    Cisco_27:d3:69   Broadcast        ARP          60 Who has 10.143.90.14? 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**.