Task 5: Network Traffic Analysis Report

Introduction and Objective

The primary objective of this task was to gain practical, hands-on experience in packet capture and protocol analysis. Using the free, industry-standard tool 'Wireshark', I captured live network traffic from my active interface, analyzed the data, and identified at least three distinct network protocols. This process is foundational to network troubleshooting and security monitoring.

ICMP TRAFFIC CAPTURING (Example)

No.	Time	Source	Destination	Protocol	Length	Info
1398	19.194996	172.16.4.139	72.163.4.185	ICMP	74	Echo (ping) request id=0x0001, seq=1/256, ttl=128 (reply in 1440)
1440	19.526914	72.163.4.185	172.16.4.139	ICMP	74	Echo (ping) reply id=0x0001, seq=1/256, ttl=41 (request in 1398)
1492	20.209980	172.16.4.139	72.163.4.185	ICMP	74	Echo (ping) request id=0x0001, seq=2/512, ttl=128 (reply in 1514)
1514	20.540155	72.163.4.185	172.16.4.139	ICMP	74	Echo (ping) reply id=0x0001, seq=2/512, ttl=41 (request in 1492)
1560	21.236866	172.16.4.139	72.163.4.185	ICMP	74	Echo (ping) request id=0x0001, seq=3/768, ttl=128 (reply in 1606)
1606	21.566356	72.163.4.185	172.16.4.139	ICMP	74	Echo (ping) reply id=0x0001, seq=3/768, ttl=41 (request in 1560)
1678	22.249879	172.16.4.139	72.163.4.185	ICMP	74	Echo (ping) request id=0x0001, seq=4/1024, ttl=128 (reply in 1707)
1707	22.569104	72.163.4.185	172.16.4.139	ICMP	74	Echo (ping) reply id=0x0001, seq=4/1024, ttl=41 (request in 1678)
1752	23.263394	172.16.4.139	72.163.4.185	ICMP	74	Echo (ping) request id=0x0001, seq=5/1280, ttl=128 (reply in 1770)
1770	23.582927	72.163.4.185	172.16.4.139	ICMP	74	Echo (ping) reply id=0x0001, seq=5/1280, ttl=41 (request in 1752)
1822	24.275359	172.16.4.139	72.163.4.185	ICMP	74	Echo (ping) request id=0x0001, seq=6/1536, ttl=128 (reply in 1847)
1847	24.607754	72.163.4.185	172.16.4.139	ICMP	74	Echo (ping) reply id=0x0001, seq=6/1536, ttl=41 (request in 1822)
1893	25.290343	172.16.4.139	72.163.4.185	ICMP	74	Echo (ping) request id=0x0001, seq=7/1792, ttl=128 (reply in 1917)
1917	25.623019	72.163.4.185	172.16.4.139	ICMP	74	Echo (ping) reply id=0x0001, seq=7/1792, ttl=41 (request in 1893)
1967	26.300453	172.16.4.139	72.163.4.185	ICMP	74	Echo (ping) request id=0x0001, seq=8/2048, ttl=128 (reply in 2017)
2017	26.637390	72.163.4.185	172.16.4.139	ICMP	74	Echo (ping) reply id=0x0001, seq=8/2048, ttl=41 (request in 1967)
2061	27.314714	172.16.4.139	72.163.4.185	ICMP	74	Echo (ping) request id=0x0001, seq=9/2304, ttl=128 (reply in 2102)
2102	27.659396	72.163.4.185	172.16.4.139	ICMP	74	Echo (ping) reply id=0x0001, seq=9/2304, ttl=41 (request in 2061)
2157	28.334166	172.16.4.139	72.163.4.185	ICMP	74	Echo (ping) request id=0x0001, seq=10/2560, ttl=128 (reply in 2171)
2171	28.665154	72.163.4.185	172.16.4.139	ICMP	74	Echo (ping) reply id=0x0001, seq=10/2560, ttl=41 (request in 2157)
2199	29.349145	172.16.4.139	72.163.4.185	ICMP	74	Echo (ping) request id=0x0001, seq=11/2816, ttl=128 (reply in 2228)

Packet Capture Details

Tool Used: Wireshark (Version: [Insert Wireshark Version])
Network Interface: Wifi
Traffic Generation Method: Pinging (ICMP Traffic)
Capture Duration: 60 seconds (Stopped manually)
Delieverable: "capture.pcap"

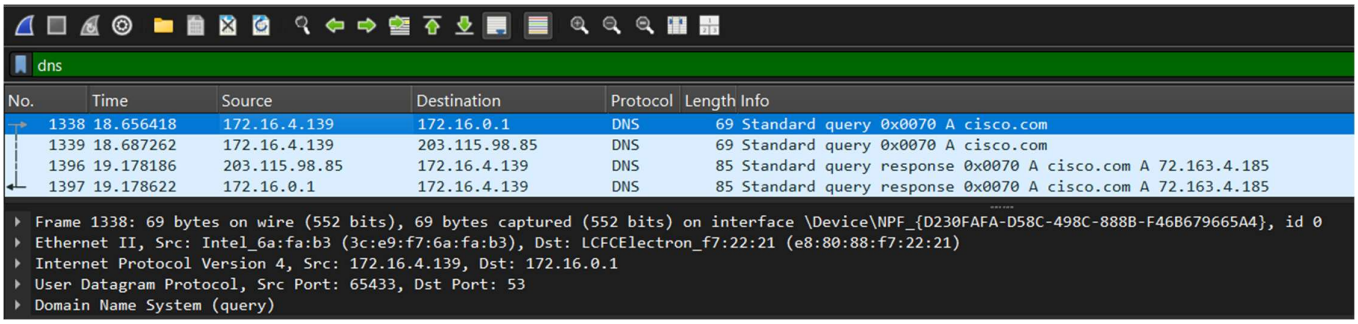
Capture Steps Summary

1. Installed **Wireshark** and supporting libraries.
2. Selected the active network interface and initiated capture.
3. Generated a mix of traffic (ping/browse).
4. Stopped the capture and saved the output as a **.pcap** file.
5. Applied **display filters** to isolate and examine individual protocols.

Protocol Identification and Analysis

The analysis was performed by applying various display filters to the captured traffic. The following three core protocols were successfully identified and analyzed:

Protocol 1: Domain Name System (DNS)



Brief Description: DNS is the protocol responsible for translating human-readable domain names (like www.cisco.com) into numerical IP addresses for routing and connection.

Packet Details (from capture)

Wireshark Filter Used : DNS

Packet No. : 1338

Frame Size: 69 Bytes

IP Version: IPv4

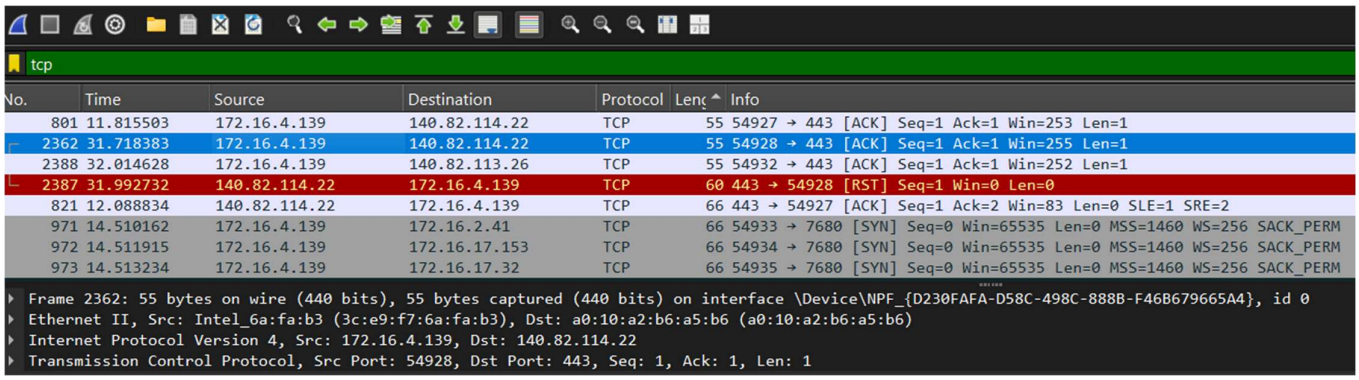
Source/Destination IP: 172.16.4.139 ---> 172.16.0.1

Protocol Layer: Transport Layer (UDP)

Packet Details Found: DNS Standard query

For other packet details: 'capture.pcap' has given in repository

Protocol 2: Transmission Control Protocol (TCP)



tcp

No.	Time	Source	Destination	Protocol	Leng	Info
801	11.815503	172.16.4.139	140.82.114.22	TCP	55	54927 → 443 [ACK] Seq=1 Ack=1 Win=253 Len=1
2362	31.718383	172.16.4.139	140.82.114.22	TCP	55	54928 → 443 [ACK] Seq=1 Ack=1 Win=255 Len=1
2388	32.014628	172.16.4.139	140.82.113.26	TCP	55	54932 → 443 [ACK] Seq=1 Ack=1 Win=252 Len=1
2387	31.992732	140.82.114.22	172.16.4.139	TCP	60	443 → 54928 [RST] Seq=1 Win=0 Len=0
821	12.088834	140.82.114.22	172.16.4.139	TCP	66	443 → 54927 [ACK] Seq=1 Ack=2 Win=83 Len=0 SLE=1 SRE=2
971	14.510162	172.16.4.139	172.16.2.41	TCP	66	54933 → 7680 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
972	14.511915	172.16.4.139	172.16.17.153	TCP	66	54934 → 7680 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
973	14.513234	172.16.4.139	172.16.17.32	TCP	66	54935 → 7680 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM

```
Frame 2362: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface \Device\NPF_{D230FAFA-D58C-498C-888B-F46B679665A4}, id 0
Ethernet II, Src: Intel_6a:fa:b3 (3c:e9:f7:6a:fa:b3), Dst: a0:10:a2:b6:a5:b6 (a0:10:a2:b6:a5:b6)
Internet Protocol Version 4, Src: 172.16.4.139, Dst: 140.82.114.22
Transmission Control Protocol, Src Port: 54928, Dst Port: 443, Seq: 1, Ack: 1, Len: 1
```

Brief Description: TCP is a reliable, **connection-oriented** protocol that ensures data is delivered correctly and in order. It is fundamental for establishing reliable web connections.

Packet Details (from capture)

Wireshark Filter Used: tcp.flags.syn==1 (to find connection initiation)

Example Packet No.: 2362

Frame Size: 55 Bytes

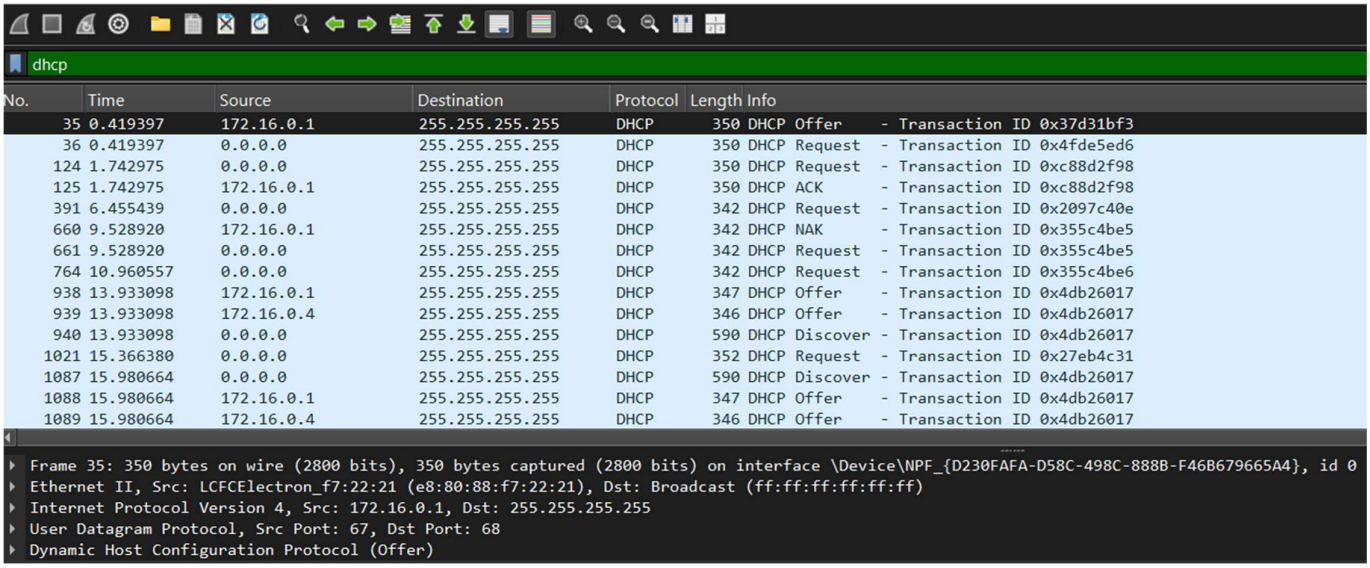
Source/Destination IP: 172.16.4.139 ---> 140.82.114.22

Source/Destination Port: 54928 ---> 443

Protocol Layer: Transport (Layer 4)

Packet Details Found: **Flags:** Acknowledgement (ACK)

Protocol 3: Dynamic Host Configuration Protocol (DHCP)



dhcp

No.	Time	Source	Destination	Protocol	Length	Info
35	0.419397	172.16.0.1	255.255.255.255	DHCP	350	DHCP Offer - Transaction ID 0x37d31bf3
36	0.419397	0.0.0.0	255.255.255.255	DHCP	350	DHCP Request - Transaction ID 0x4fde5ed6
124	1.742975	0.0.0.0	255.255.255.255	DHCP	350	DHCP Request - Transaction ID 0xc88d2f98
125	1.742975	172.16.0.1	255.255.255.255	DHCP	350	DHCP ACK - Transaction ID 0xc88d2f98
391	6.455439	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x2097c40e
660	9.528920	172.16.0.1	255.255.255.255	DHCP	342	DHCP NAK - Transaction ID 0x355c4be5
661	9.528920	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x355c4be5
764	10.960557	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x355c4be6
938	13.933098	172.16.0.1	255.255.255.255	DHCP	347	DHCP Offer - Transaction ID 0x4db26017
939	13.933098	172.16.0.4	255.255.255.255	DHCP	346	DHCP Offer - Transaction ID 0x4db26017
940	13.933098	0.0.0.0	255.255.255.255	DHCP	590	DHCP Discover - Transaction ID 0x4db26017
1021	15.366380	0.0.0.0	255.255.255.255	DHCP	352	DHCP Request - Transaction ID 0x27eb4c31
1087	15.980664	0.0.0.0	255.255.255.255	DHCP	590	DHCP Discover - Transaction ID 0x4db26017
1088	15.980664	172.16.0.1	255.255.255.255	DHCP	347	DHCP Offer - Transaction ID 0x4db26017
1089	15.980664	172.16.0.4	255.255.255.255	DHCP	346	DHCP Offer - Transaction ID 0x4db26017

```
Frame 35: 350 bytes on wire (2800 bits), 350 bytes captured (2800 bits) on interface \Device\NPF_{D230FAFA-D58C-498C-888B-F46B679665A4}, id 0
Ethernet II, Src: LCFCElectron_f7:22:21 (e8:80:88:f7:22:21), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 172.16.0.1, Dst: 255.255.255.255
User Datagram Protocol, Src Port: 67, Dst Port: 68
Dynamic Host Configuration Protocol (Offer)
```

Brief Description: DHCP is an application layer protocol used for automatically assigning IP addresses and other network configuration parameters (like the subnet mask and default gateway) to devices on a network. It often uses the "DORA" process (Discover, Offer, Request, Acknowledge).

Packet Details (from capture)

Wireshark Filter Used: dhcp

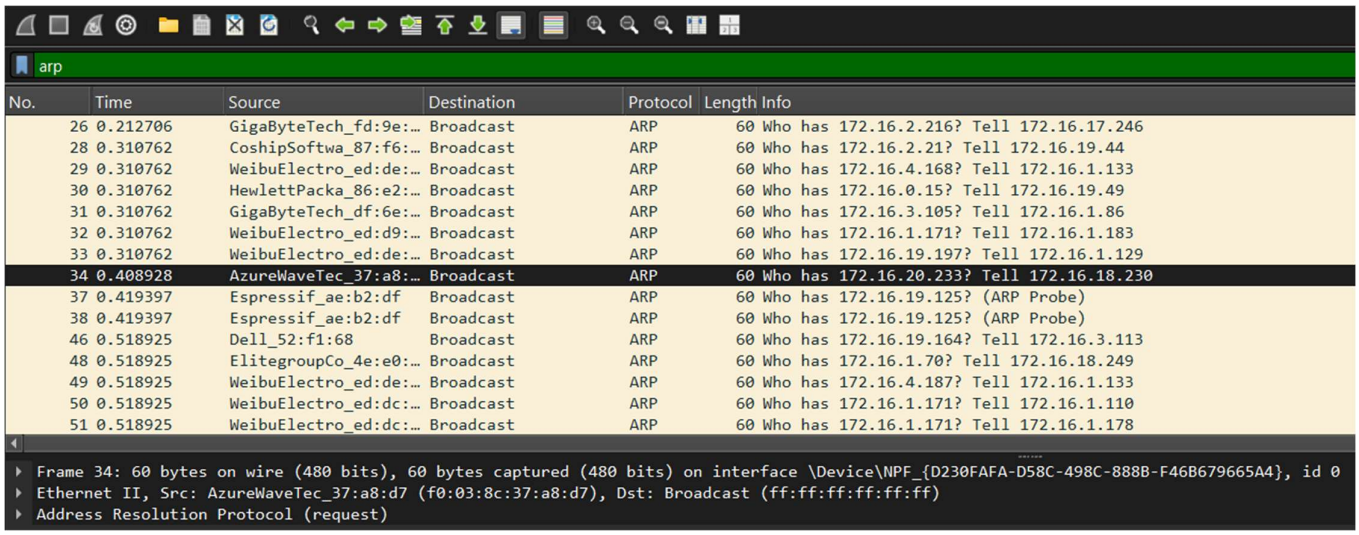
Example Packet No. : 940

Source/Destination IP: 0.0.0.0→255.255.255.255 (Discover Broadcast)

Protocol Layer: Application (Layer 7) / Transport (UDP)

Packet Details Found: (Message Type: DHCP Discover (Client seeking an IP address))

Protocol 4: Address Resolution Protocol (ARP)



arp

No.	Time	Source	Destination	Protocol	Length	Info
26	0.212706	GigaByteTech_fd:9e:…	Broadcast	ARP	60	Who has 172.16.2.216? Tell 172.16.17.246
28	0.310762	CoshipSoftwa_87:f6:…	Broadcast	ARP	60	Who has 172.16.2.21? Tell 172.16.19.44
29	0.310762	WeibuElectro_ed:de:…	Broadcast	ARP	60	Who has 172.16.4.168? Tell 172.16.1.133
30	0.310762	HewlettPacka_86:e2:…	Broadcast	ARP	60	Who has 172.16.0.15? Tell 172.16.19.49
31	0.310762	GigaByteTech_df:6e:…	Broadcast	ARP	60	Who has 172.16.3.105? Tell 172.16.1.86
32	0.310762	WeibuElectro_ed:d9:…	Broadcast	ARP	60	Who has 172.16.1.171? Tell 172.16.1.183
33	0.310762	WeibuElectro_ed:de:…	Broadcast	ARP	60	Who has 172.16.19.197? Tell 172.16.1.129
34	0.408928	AzureWaveTec_37:a8:…	Broadcast	ARP	60	Who has 172.16.20.233? Tell 172.16.18.230
37	0.419397	Espressif_ae:b2:df	Broadcast	ARP	60	Who has 172.16.19.125? (ARP Probe)
38	0.419397	Espressif_ae:b2:df	Broadcast	ARP	60	Who has 172.16.19.125? (ARP Probe)
46	0.518925	Dell_52:f1:68	Broadcast	ARP	60	Who has 172.16.19.164? Tell 172.16.3.113
48	0.518925	ElitegroupCo_4e:e0:…	Broadcast	ARP	60	Who has 172.16.1.70? Tell 172.16.18.249
49	0.518925	WeibuElectro_ed:de:…	Broadcast	ARP	60	Who has 172.16.4.187? Tell 172.16.1.133
50	0.518925	WeibuElectro_ed:dc:…	Broadcast	ARP	60	Who has 172.16.1.171? Tell 172.16.1.110
51	0.518925	WeibuElectro_ed:dc:…	Broadcast	ARP	60	Who has 172.16.1.171? Tell 172.16.1.178

```
▸ Frame 34: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{D230FAFA-D58C-498C-888B-F46B679665A4}, id 0
▸ Ethernet II, Src: AzureWaveTec_37:a8:d7 (f0:03:8c:37:a8:d7), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▸ Address Resolution Protocol (request)
```

Brief Description: ARP is a protocol used to map an Internet Layer (Layer 3) IP address to a Data Link Layer (Layer 2) physical **MAC address**. This is necessary for local network communication.

Packet Details (from capture)

Wireshark Filter Used: arp

Example Packet No: 34

Frame Size: 60 Bytes

Source/Destination MAC: *AzureWave* → ff:ff:ff:ff:ff:ff (Broadcast)]

Protocol Layer Data Link (Layer 2)

Packet Details Found: Who has 172.16.20.233? Tell 172.16.18.230

Note: All the Screenshots were taken from capture.pcap file which is already given in repository.