EXPERIMENT 13 - TO CAPTURE, SAVE, AND ANALYZE NETWORK TRAFFIC ON TCP / UDP / IP / HTTP / ARP /DHCP /ICMP /DNS USING WIRESHARK TOOL

Aim:

To study and analyze different network protocol packets such as TCP, UDP, IP, HTTP, ARP, DHCP, ICMP, and DNS by capturing live network traffic using **Wireshark** tool.

Introduction:

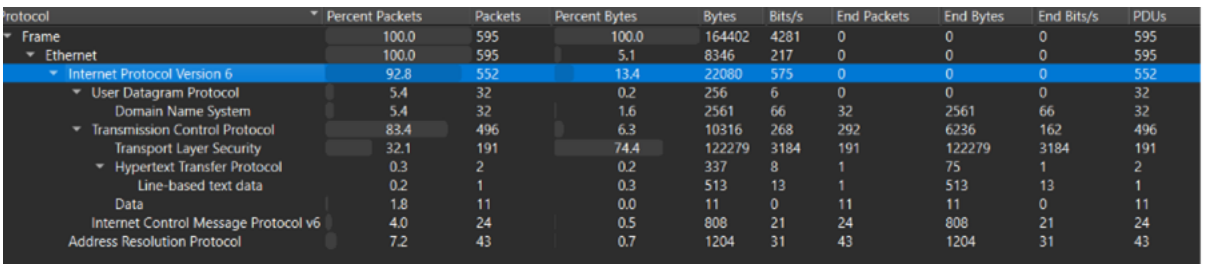
Wireshark is a powerful network protocol analyzer used to monitor and capture live network packets. It allows users to inspect protocols at each layer of the OSI model and understand how data is transmitted over the network.

By analyzing captured packets, we can identify communication patterns, troubleshoot connectivity issues, and study protocol behavior such as TCP handshakes, DNS lookups, HTTP requests, and ICMP pings.

Algorithm:

1. Open Wireshark application on the system.
2. Select the active network interface (Wi-Fi or Ethernet) to capture packets.
3. Click on Start Capturing Packets.
4. Open Command Prompt in Windows and execute the commands to generate various types of traffic
5. After running the commands, return to Wireshark and click Stop Capture.
6. Save the captured packets as a .pcap file.
7. Observe packet details such as Source & Destination IP, MAC addresses, ports, and payload.

Output:



Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	595	100.0	164402	4281	0	0	0	595
Ethernet	100.0	595	5.1	8346	217	0	0	0	595
Internet Protocol Version 6	92.8	552	13.4	22080	575	0	0	0	552
User Datagram Protocol	5.4	32	0.2	256	6	0	0	0	32
Domain Name System	5.4	32	1.6	2561	66	32	2561	66	32
Transmission Control Protocol	83.4	496	6.3	10316	268	292	6236	162	496
Transport Layer Security	32.1	191	74.4	122279	3184	191	122279	3184	191
Hypertext Transfer Protocol	0.3	2	0.2	337	8	1	75	1	2
Line-based text data	0.2	1	0.3	513	13	1	513	13	1
Data	1.8	11	0.0	11	0	11	11	0	11
Internet Control Message Protocol v6	4.0	24	0.5	808	21	24	808	21	24
Address Resolution Protocol	7.2	43	0.7	1204	31	43	1204	31	43

Result:

The experiment to capture and analyse network traffic using **Wireshark** was successfully performed.
Packets for TCP, UDP, IP (IPv6), HTTP, ARP, ICMP, and DNS were captured and analysed.
Hence, the objective of the experiment is achieved successfully.