## Experiment 7: Analyze the Network Traffic using Wireshark Tool / Packet Tracer Tool

### Aim

To capture and analyze network traffic using Wireshark (or Packet Tracer Simulation Mode) and study different network protocols and packet structures.

### Background Theory

Network traffic analysis is the process of capturing and inspecting packets flowing through a network. It helps network administrators understand performance, troubleshoot issues, and identify security threats.

### Wireshark

1. A widely used open-source packet analyzer.
2. Allows packet capturing and in-depth analysis.
3. Supports hundreds of protocols (TCP, UDP, HTTP, DNS, ICMP, etc.).
4. Provides filtering, color coding, and statistics for traffic analysis.

### Packet Tracer(as an alternative):

1. Has a Simulation Mode to capture and inspect packets.
2. Provides protocol-specific information but with limited depth compared to Wireshark.

### Common Uses of Network Traffic Analysis:

1. Troubleshooting connectivity issues.
2. Identifying latency and congestion.
3. Security monitoring (detecting malicious activity).
4. Performance optimization.

### Objectives

1. Capture packets using Wireshark / Packet Tracer.
2. Identify protocols such as TCP, UDP, ICMP, ARP, HTTP, and DNS.
3. Analyze fields in packet headers (Source IP, Destination IP, Port Numbers, Flags, etc.).
4. Understand packet flow in client-server communication.
5. Gain hands-on experience in traffic filtering and analysis.

### Software/Tools Required

1. Wireshark (latest version)

2. Cisco Packet Tracer (if Wireshark is not available)
3. Windows/Linux system with internet connectivity

## Algorithm

1. Start Wireshark and select the active network interface.
2. Begin packet capture.
3. Generate traffic (ping, web browsing, DNS lookup, file transfer).
4. Stop the capture after sufficient data is collected.
5. Apply filters (e.g., `icmp`, `tcp`, `udp`, `http`).
6. Inspect packet headers and analyze communication.
7. Save the capture file for reporting.

## Step-by-Step Procedure

Step 1: Setup Wireshark

Install Wireshark on your computer.
Launch the application.

Step 2: Select Interface

1. From the interface list, select the active adapter (e.g., Wi-Fi, Ethernet).
2. Click **Start Capture**.

Step 3: Generate Traffic

1. Open Command Prompt/Terminal.
2. Run: `ping www.google.com` (for ICMP traffic).
3. Open a browser and visit a website (for HTTP/HTTPS traffic).
4. Use `nslookup www.google.com` (for DNS traffic).
5. Transfer a file or use FTP if available.

Step 4: Capture Packets

1. Observe live packets scrolling in Wireshark.
2. Stop capture after 1–2 minutes.

Step 5: Apply Filters

- In Wireshark filter bar:
    1. `icmp` → to view ping packets.
    2. `http` → to view HTTP requests.
    3. `tcp` → to analyze TCP handshake.

4. `udp` → to capture DNS queries.

Step 6: Analyze Packets

1. Expand packet details (Ethernet, IP, TCP/UDP layers).
2. Note down:
   i. Source/Destination IP
   ii. Port numbers
   iii. Flags (SYN, ACK in TCP)
   iv. Packet size and time delay

Step 7: Save Results

i. Save the capture file (`.pcapng`) for documentation.

**Packet Tracer Alternative**:

1. Use Simulation Mode.
2. Generate pings between PCs.
3. Observe packet movement (Ethernet → IP → ICMP).

**Expected Output**

1. Packet captures showing ICMP requests/replies.
2. TCP 3-way handshake (SYN, SYN-ACK, ACK).
3. DNS query/response packets.
4. HTTP GET request and response.
5. Graph/statistics (packet length distribution, protocol hierarchy).

**Result**

The network traffic was successfully captured and analyzed using Wireshark / Packet Tracer. Different protocols (ICMP, TCP, UDP, DNS, HTTP) were identified, and packet-level details were studied.

**Pre-Viva Questions**

1. What is Wireshark used for?

2. Differentiate between TCP and UDP.

3. What does the ICMP protocol do?

4. How does DNS work in a network?

5. What is the difference between real-time and offline traffic analysis?

**Post-Viva Questions**

1. How can Wireshark help in detecting security threats?

2. Why is filtering important in packet analysis?

3. What are some limitations of using Packet Tracer for traffic analysis?

4. Can Wireshark capture encrypted traffic (HTTPS)? If yes, what are the challenges?