****

**Experiment No.:10**

| **TITLE: Study of Packet Analyzer tool: Wireshark** |
| --- |

**AIM:** To study and analyse various Protocols using Packet Analyzer tool: Wireshark

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected Outcome of Experiment:**

**CO:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. A. S. Tanenbaum, “Computer Networks”, Pearson Education, Fourth Edition
2. B. A. Forouzan, “Data Communications and Networking”, TMH, Fourth Edition

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Pre Lab/ Prior Concepts:**

IPv4 Addressing, Subnetting, Link State Protocol, Router configuration Commands

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_New Concepts to be learned: Packet Analyzer tool: Wireshark**.

**THEORY:**

Wireshark is a widely-used open-source packet analyzer that is employed for network troubleshooting, analysis, software and protocol development, and security auditing. Originally developed by Gerald Combs in 1998 under the name "Ethereal," it was renamed Wireshark in 2006 due to trademark issues. It supports a wide variety of protocols and provides comprehensive packet-level analysis to understand how data is moving across networks.

### Key Features of Wireshark:

1. **Packet Capture**: Wireshark is primarily used to capture network packets, which are small chunks of data transmitted across the network. These packets can be captured from various interfaces such as Ethernet, Wi-Fi, or even USB connections. The captured packets can be analyzed in real-time or saved for later analysis.
2. **Protocol Analysis**: Wireshark understands a wide range of network protocols, allowing it to dissect the captured packets and display information about the specific protocol headers. It supports both popular and niche protocols such as TCP/IP, UDP, HTTP, FTP, DNS, and more.
3. **Detailed Filtering and Search**: Wireshark offers powerful filtering capabilities to help users focus on specific traffic of interest. Filters can be applied to narrow down captured traffic by IP address, protocol, port number, or even more advanced filters based on packet contents.
4. **Visualization**: Wireshark includes visual tools to graphically represent traffic flows, packet lengths, round-trip times, and other network behaviors. This helps users quickly identify network issues such as latency, packet loss, or abnormal traffic patterns.
5. **Exporting and Reporting**: Captured data can be saved in a variety of file formats, such as pcap (Packet Capture), for later analysis or reporting. Wireshark also allows you to export specific data to CSV, XML, or plain text formats for further processing or documentation.
6. **Decryption**: Wireshark can decrypt certain types of encrypted traffic (such as SSL/TLS) if the necessary keys are available. This is particularly useful for analyzing secure communication and ensuring that data is properly encrypted.
7. **Cross-Platform Support**: Wireshark works on a wide variety of operating systems, including Windows, macOS, and Linux. The interface remains consistent across platforms, providing users with the same tools and features, regardless of their operating system.

### Components and Usage

Wireshark provides several essential components for effective packet analysis:

#### 1. Capture Interface:

* Before you can analyze packets, you need to choose an interface (e.g., Ethernet, Wi-Fi, or other network interfaces) through which Wireshark will listen and capture traffic. You can either capture all traffic or specify a filter to capture only certain types of packets (such as TCP or HTTP packets).

#### 2. Packet Capture:

* When capturing data, Wireshark collects raw packet data from the selected network interface. Each packet contains headers and payload data that represent the communication between devices on the network.

#### 3. Packet List Pane:

* This is the main window in Wireshark, where each row represents an individual packet. Information shown in this pane includes the timestamp, source and destination IP addresses, protocol, length, and info (which provides a quick summary of the packet's contents). You can click on individual packets for more detailed analysis.

#### 4. Packet Details Pane:

* Once a packet is selected in the Packet List Pane, the Packet Details Pane displays the decoded packet information in a hierarchical structure. Each layer of the protocol stack is displayed in a nested format, from the physical layer (Ethernet) to the application layer (HTTP, DNS, etc.).
* For example, a TCP packet might display:
  + Ethernet Header (MAC addresses)
  + IP Header (source and destination IPs)
  + TCP Header (ports, sequence numbers, flags)
  + Application Data (e.g., HTTP request/response)

#### 5. Packet Bytes Pane:

* This pane shows the raw hexadecimal values of the packet data along with the corresponding ASCII values. This is helpful for inspecting the actual data contained in the packet, especially when troubleshooting low-level protocol issues.

#### 6. Display Filters:

* Display filters are used to focus on specific traffic patterns or packets of interest. Examples include:
  + ip.addr == 192.168.1.1: Show all packets involving the IP address 192.168.1.1.
  + tcp.port == 80: Show only HTTP traffic on port 80.
  + http.request: Show only HTTP requests.
  + Filters can be combined with logical operators like and, or, and not.
* Wireshark provides an intuitive filtering language that allows for complex queries like filtering based on packet content, protocol flags, or even packet length.

#### 7. Statistics and Graphs:

* Wireshark includes powerful statistical tools that can visualize traffic patterns, analyze conversations between endpoints, and track protocol distribution. Key statistics features include:
  + **Protocol Hierarchy**: Breaks down packet traffic by protocol, showing what percentage of traffic is being used by various protocols.
  + **Conversations**: Displays communication sessions between pairs of devices, which can be useful for tracking down issues such as dropped connections or miscommunications.
  + **IO Graphs**: Visualizes packet counts, bytes, or other traffic metrics over time, helping to identify trends or anomalies in network behavior.

#### 8. Expert Information:

* Wireshark features an expert system that flags unusual or abnormal events in the captured traffic. These are indicated by color-coded warnings and summaries that help users quickly identify potential network issues (such as retransmissions, protocol errors, or suspicious packets).

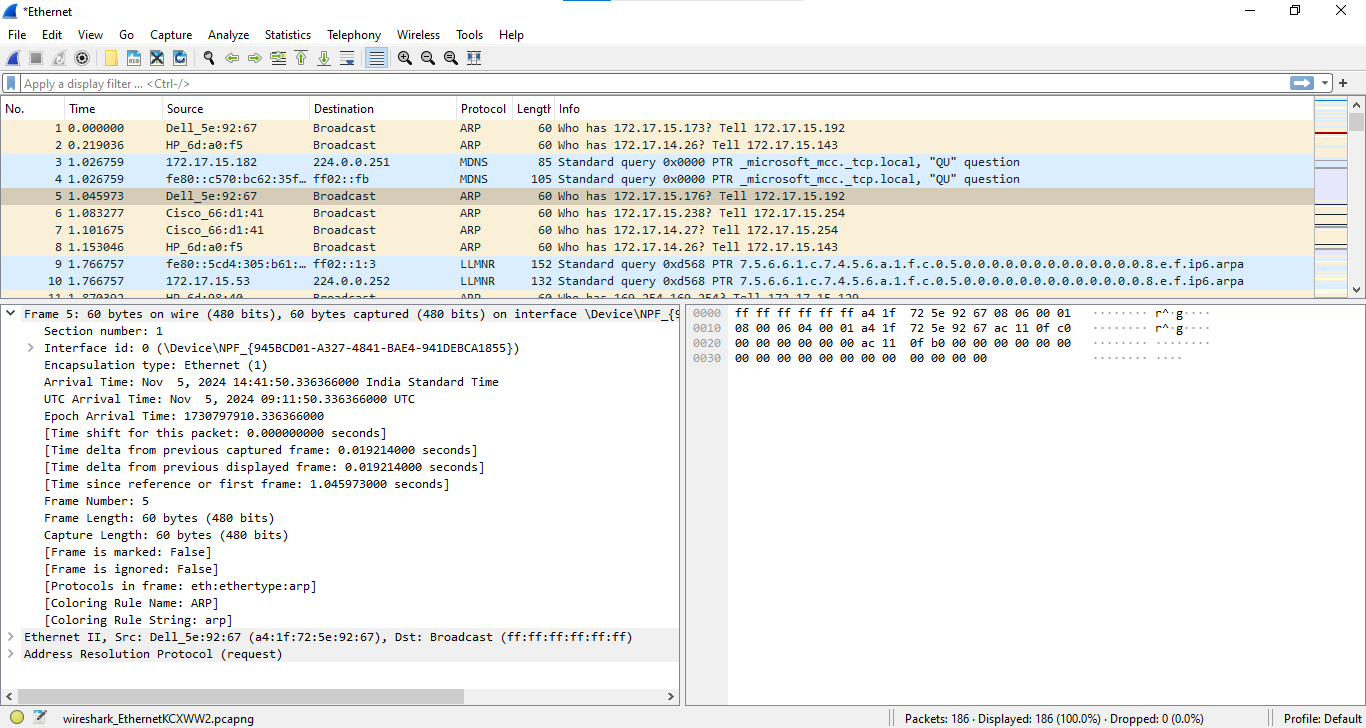
#### 9. Coloring Rules:

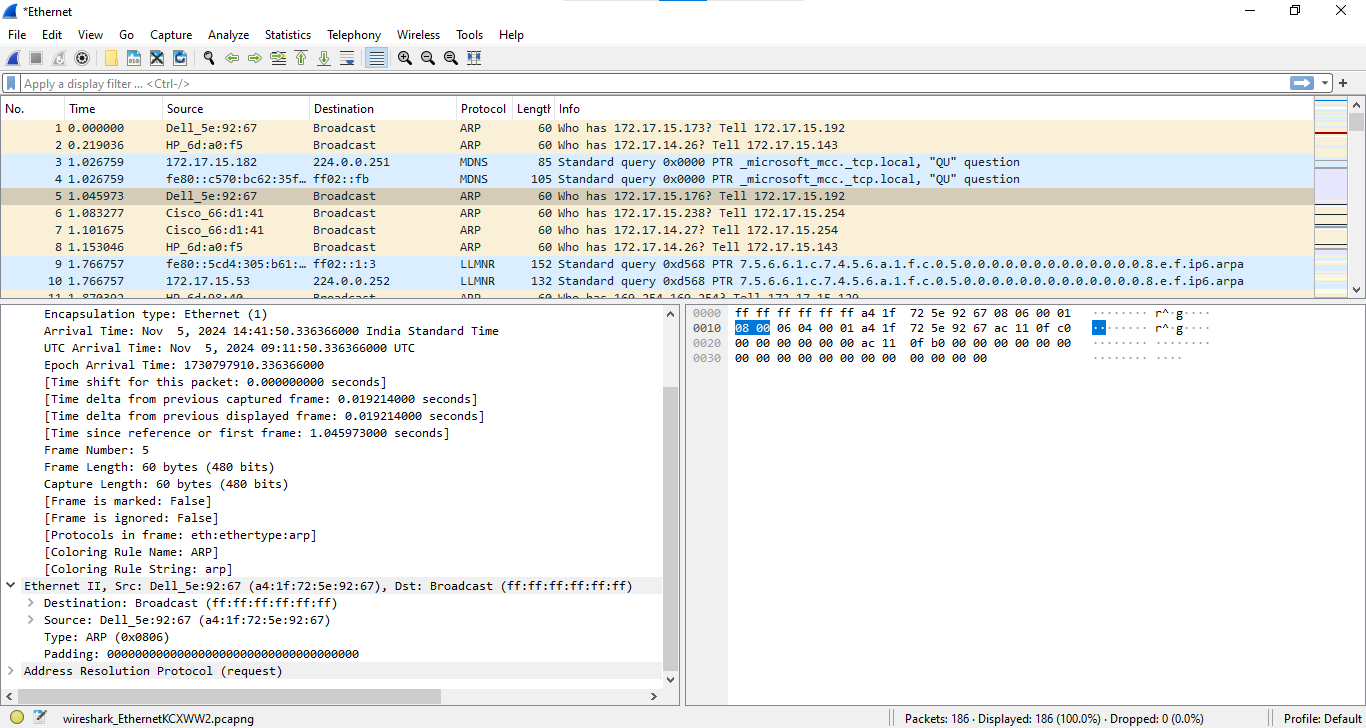
* Users can customize the display using coloring rules, which color packets based on defined criteria. For instance, TCP packets that are retransmissions might be displayed in red, while packets containing errors might be shown in yellow. This visual cueing can significantly speed up the troubleshooting process.

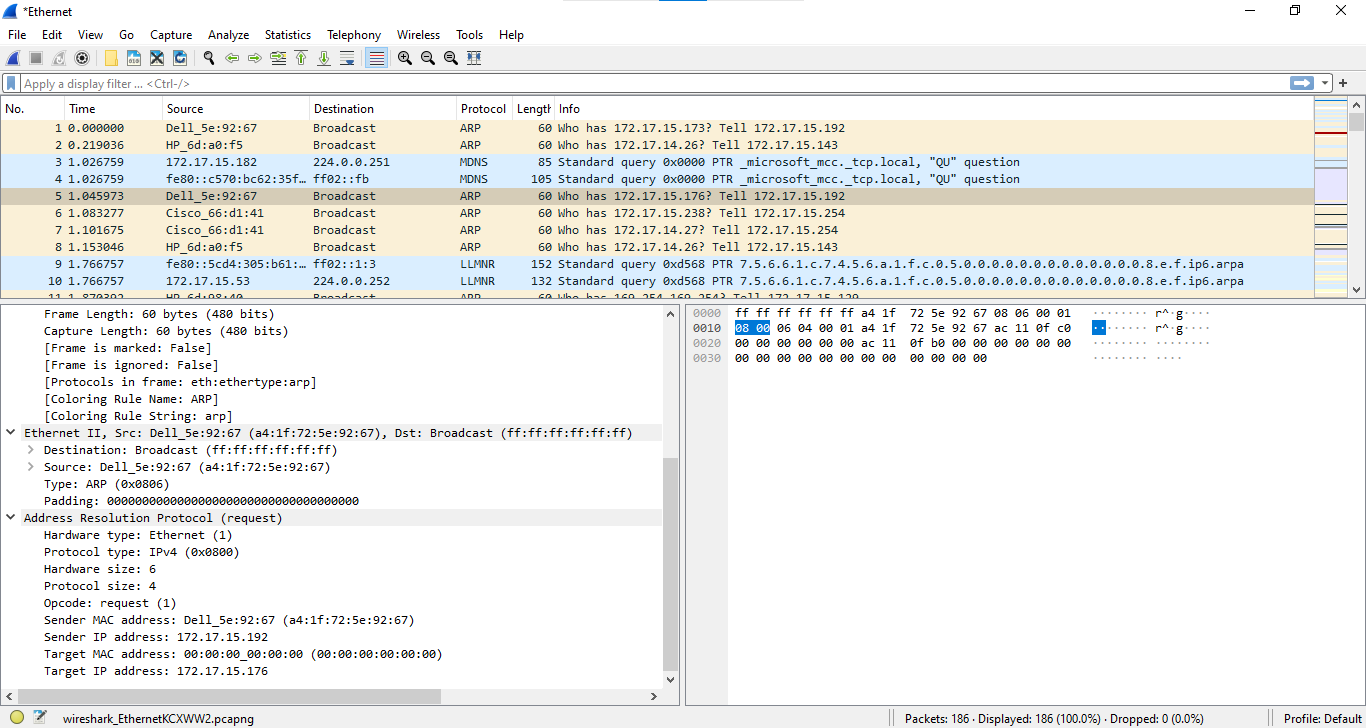
### Use Cases for Wireshark

1. **Network Troubleshooting**:
   * **Latency & Packet Loss**: If a network application is slow, Wireshark can help identify packet retransmissions, lost packets, or network congestion that could be causing delays.
   * **Network Errors**: Wireshark can help detect problems like misconfigured DNS settings, firewall issues, or dropped TCP connections.
   * **Bandwidth Usage**: It can reveal which applications are using the most bandwidth or whether there is any unusual traffic (such as rogue devices or unauthorized data exfiltration).
2. **Security Auditing**:
   * **Malicious Traffic Detection**: Wireshark is used by security professionals to identify malicious or suspicious traffic, such as man-in-the-middle attacks, ARP poisoning, or unauthorized access attempts.
   * **Protocol Anomalies**: It helps spot unusual patterns that might indicate a vulnerability, such as non-standard ports being used or unexpected payload data in packets.
   * **Decryption of Encrypted Traffic**: With the right keys, Wireshark can decrypt secure traffic and help inspect it for signs of a security breach.
3. **Protocol Development & Debugging**:
   * Developers use Wireshark to analyze the behavior of network protocols, test new protocol features, and troubleshoot issues in network communication between their software components.
4. **Forensics & Incident Response**:
   * In the event of a network breach or suspected intrusion, Wireshark can be used to capture evidence of unauthorized activity, track the attacker's movements on the network, and identify what data was exfiltrated or corrupted.

**IMPLEMENTATION:**

****

****

****

**CONCLUSION:** Successfully studied Wireshark, equipped the ability to capture, analyze, and troubleshoot network traffic at a granular level, enhancing network security, performance, and protocol development.

**Date: \_\_05/11/2024 Signature of faculty in-charge**