**Lab 13: Introduction to Network Traffic Analysis using Wireshark**

**Theory**

**Wireshark** is a popular open-source network protocol analyzer that captures and analyzes network traffic in real time. It allows users to inspect data packets, making it invaluable for troubleshooting, monitoring network performance, and identifying security issues. With its graphical interface, Wireshark supports a wide range of protocols, offering detailed insights into network communications. This tool is essential for network administrators and cybersecurity professionals alike.

**Key Concepts of Wireshark**

**Packet Capture**: Wireshark captures data packets traveling through a network in real time. Each packet contains headers and data that can be analyzed for network performance or security issues.

**Protocols**: Wireshark supports thousands of protocols (e.g., TCP, UDP, HTTP, DNS), allowing you to analyze traffic from different network layers.

**Filters**: Filters in Wireshark help narrow down the data by specific criteria (e.g., IP addresses, port numbers, protocol types). Display filters show only relevant packets, while capture filters limit the data being recorded.

**Frames and Layers**: Wireshark displays packet data in different layers (e.g., Ethernet, IP, TCP) following the OSI model, which helps break down packet content at various levels of networking.

**Hexadecimal and ASCII Views**: The packet data is shown in both hexadecimal and ASCII formats, allowing you to inspect the raw content of a packet and understand its structure.

**Packet Dissection**: Wireshark breaks down packet contents into readable components, showing fields, flags, and values related to the specific protocol in use.

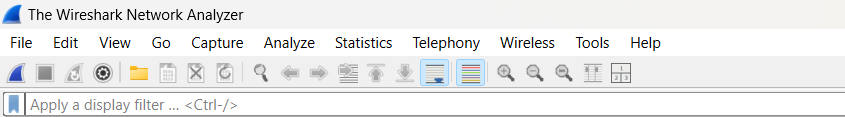
**Statistics and Graphs**: Wireshark provides statistical tools (e.g., flow graphs, IO graphs) to visualize network data. It can capture packets from different interfaces, including Ethernet, Wi-Fi, and loopback, based on the system’s available interfaces.

**Real-Time and Offline Analysis**: Wireshark can analyze both live network traffic and saved capture files (PCAP), making it versatile for both real-time troubleshooting and post-event analysis.

**Security**: Wireshark helps detect network security issues like malware, unauthorized access, and other vulnerabilities by analyzing unusual patterns in packet flows.

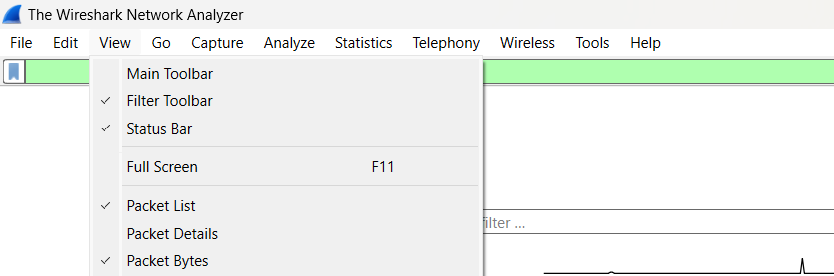
**Interface of Wireshark**

**Main Toolbar**: The toolbar offers quick access to essential functions like opening, saving, or closing captures. It includes buttons for starting, stopping, and restarting packet captures, as well as filtering the packet display. Key features include a field for applying display filters, navigating through packets, and zooming in on packet views.

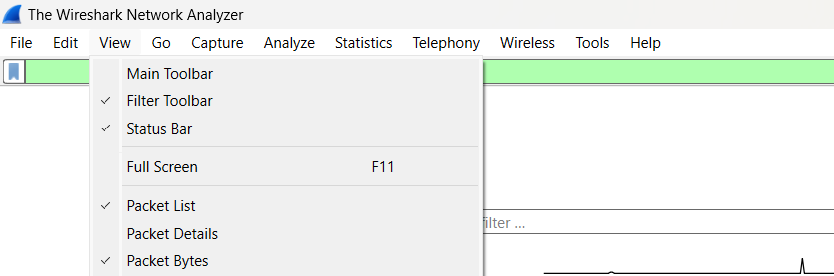


*Fig: Toolbar*

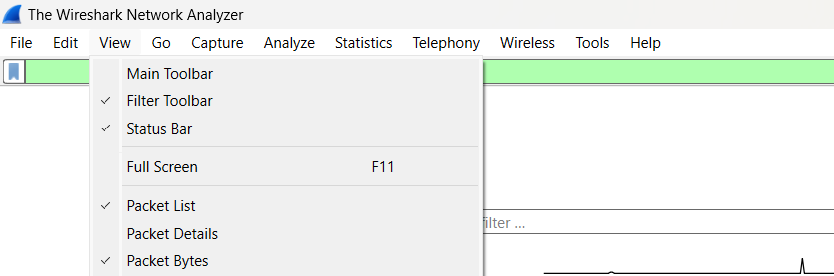
**Packet List Pane**: This pane displays all captured packets in real time. Each row represents a packet, with columns showing key details such as packet number, timestamp, source, destination, protocol, length, and additional info. Clicking on a packet allows you to view its detailed information in the other panes.

*Fig: Packet list pane*

**Packet Details Pane**: When you select a packet from the Packet List, this pane provides an expandable tree view of the packet’s structure. It breaks down the packet into its various protocol layers, such as Ethernet, IP, TCP/UDP, etc., offering detailed protocol information for analysis.

*Fig: Packet Details pane*

**Packet Bytes Pane**: This pane displays the raw data of the selected packet in both hexadecimal and ASCII formats. It allows users to view the actual bytes transmitted, which can be useful for low-level protocol analysis or detecting anomalies at the byte level.

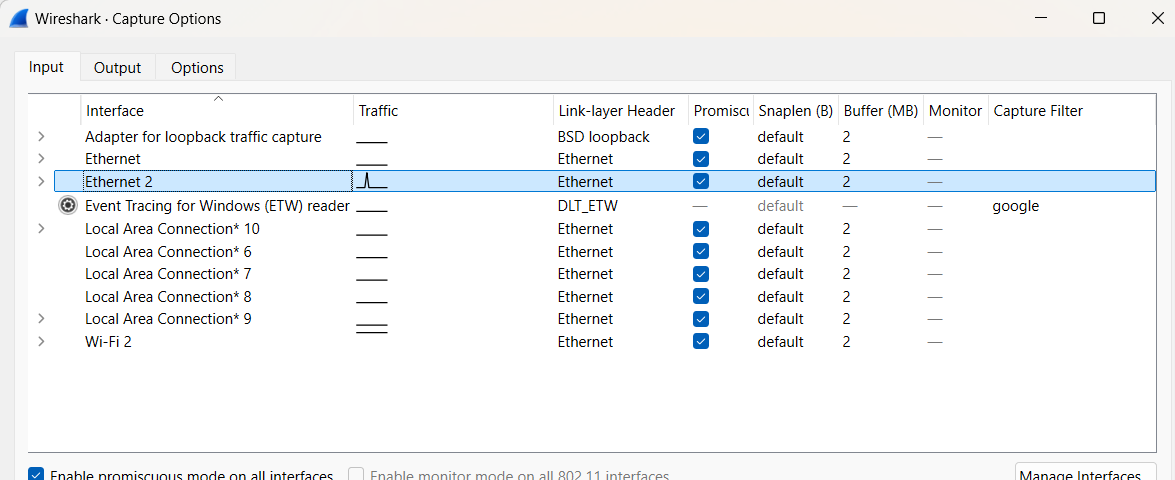
*Fig: Packet Bytes pane*

**Basic Network Capture and Analysis**

**Selecting a Network Interface**

**Steps:**

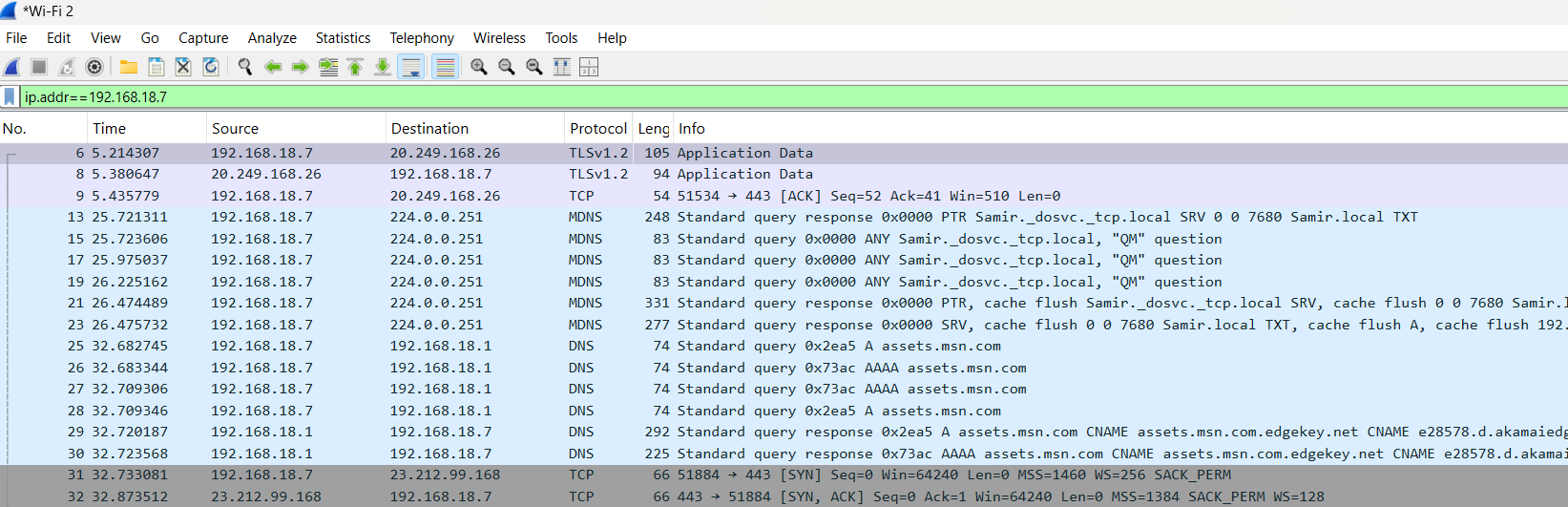
* Open Wireshark.
* You’ll see a list of available network interfaces (Wi-Fi, Ethernet, etc.).
* Look for the interface capturing active traffic (typically with a moving graph or traffic count).
* Select the desired interface by clicking on it.

**

*Fig: Selecting network interface*

**Starting Packet Capture**

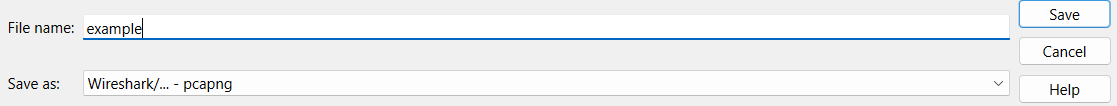
**Steps:**

1. **Start Capture**: After selecting the network interface, click the blue shark fin icon in the toolbar to start the packet capture.
2. **Navigate to a Website**: Open your browser and navigate to any website (e.g., example.com).
3. **Capture Traffic**: Wireshark will start capturing all network traffic on the selected interface.
4. **Apply Display Filter**: To filter the capture for packets related to example.com, you can use a display filter.
5. **Enter Filter Criteria**: In the filter bar, type ip.addr == <ip address of src>.
6. ****Apply Filter**: Press Enter to apply the filter.

*Fig: Packet Capture*

**Stopping and Saving Captures**

**Steps:**

1. Once you’ve captured the desired traffic, click the red square icon in the toolbar to stop the capture.
2. Go to **File > Save As**.
3. Select a file name and location.
4. Choose a format (default is .pcapng), then click **Save**.

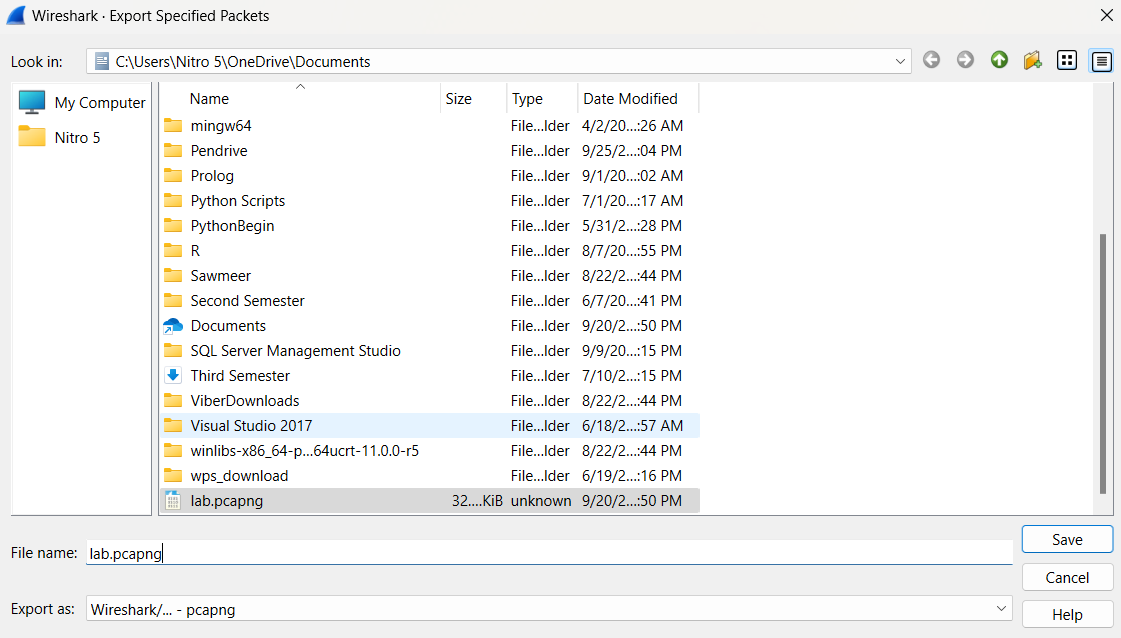
*Fig: Saving a file*

**Exporting the Captured Data**

**Steps:**

1.To export specific packets or data:.Go to File > Export Specified Packets.

2.Choose the range or filter for the packets you want to export.



**Conclusion**

In this lab, we learned the fundamentals of network traffic analysis using Wireshark. By capturing and examining live network traffic, we gained practical experience in identifying network protocols, analyzing packet details, and filtering traffic for specific hosts or services. This hands-on approach helped us understand how data flows across networks and equipped us with the skills to troubleshoot and secure network communication effectively.