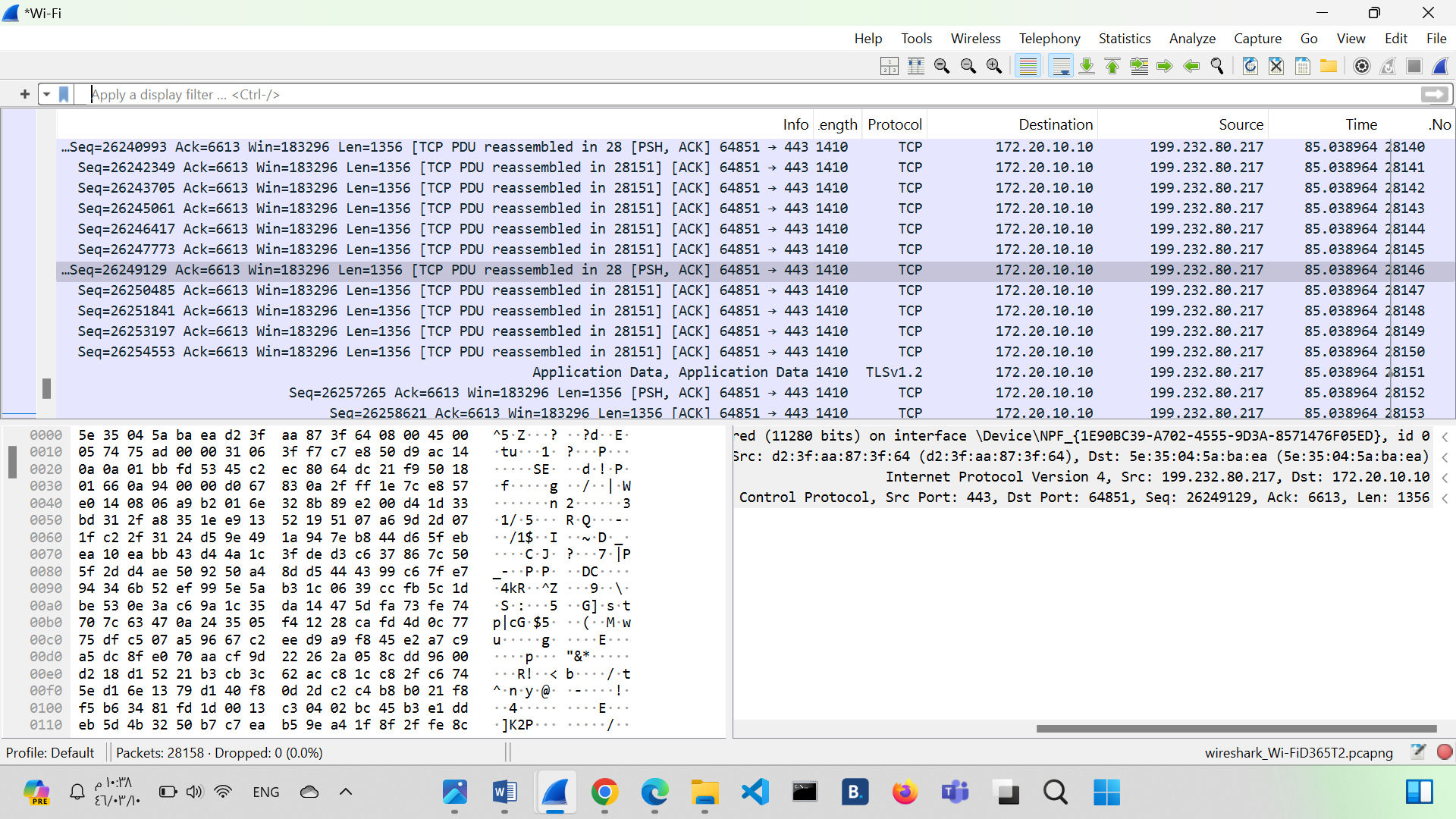
Lab week 2

Alanod

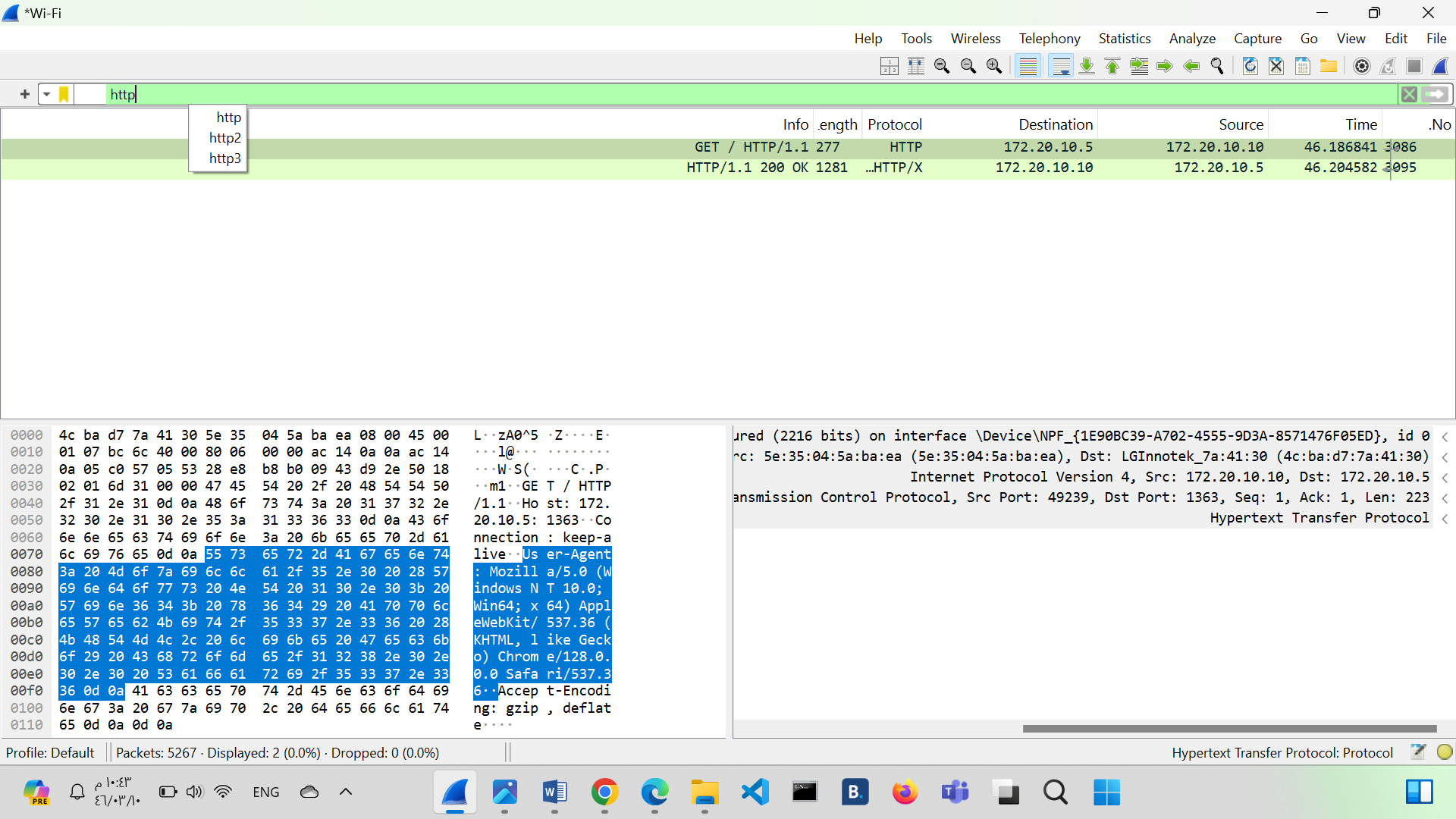
**Part 1: Capturing HTTP Traffic.**

**Task 1: Start Wireshark and capture packets.**



**Task 2: Filter HTTP packets and analyze them.**

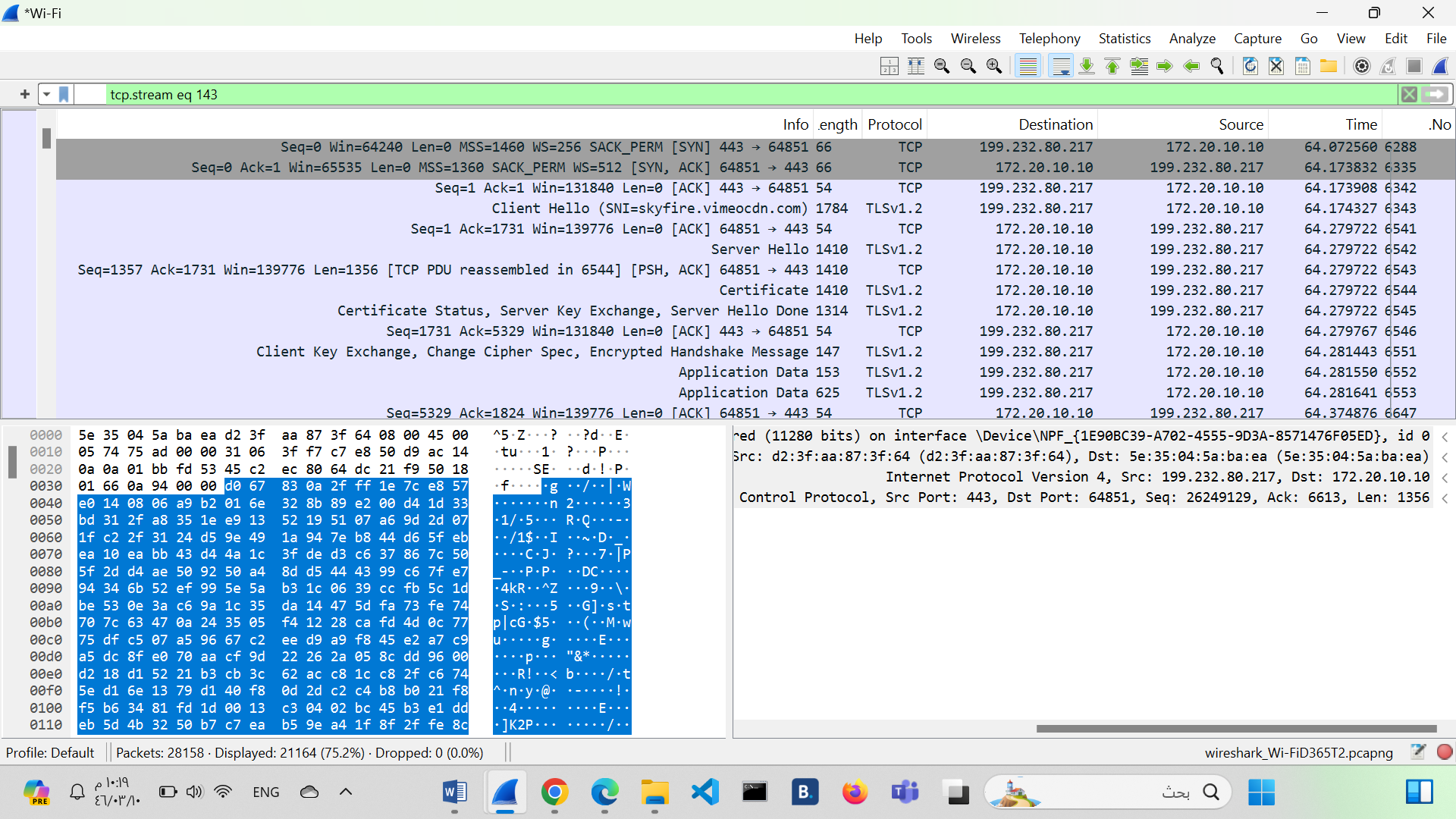
**GET method**

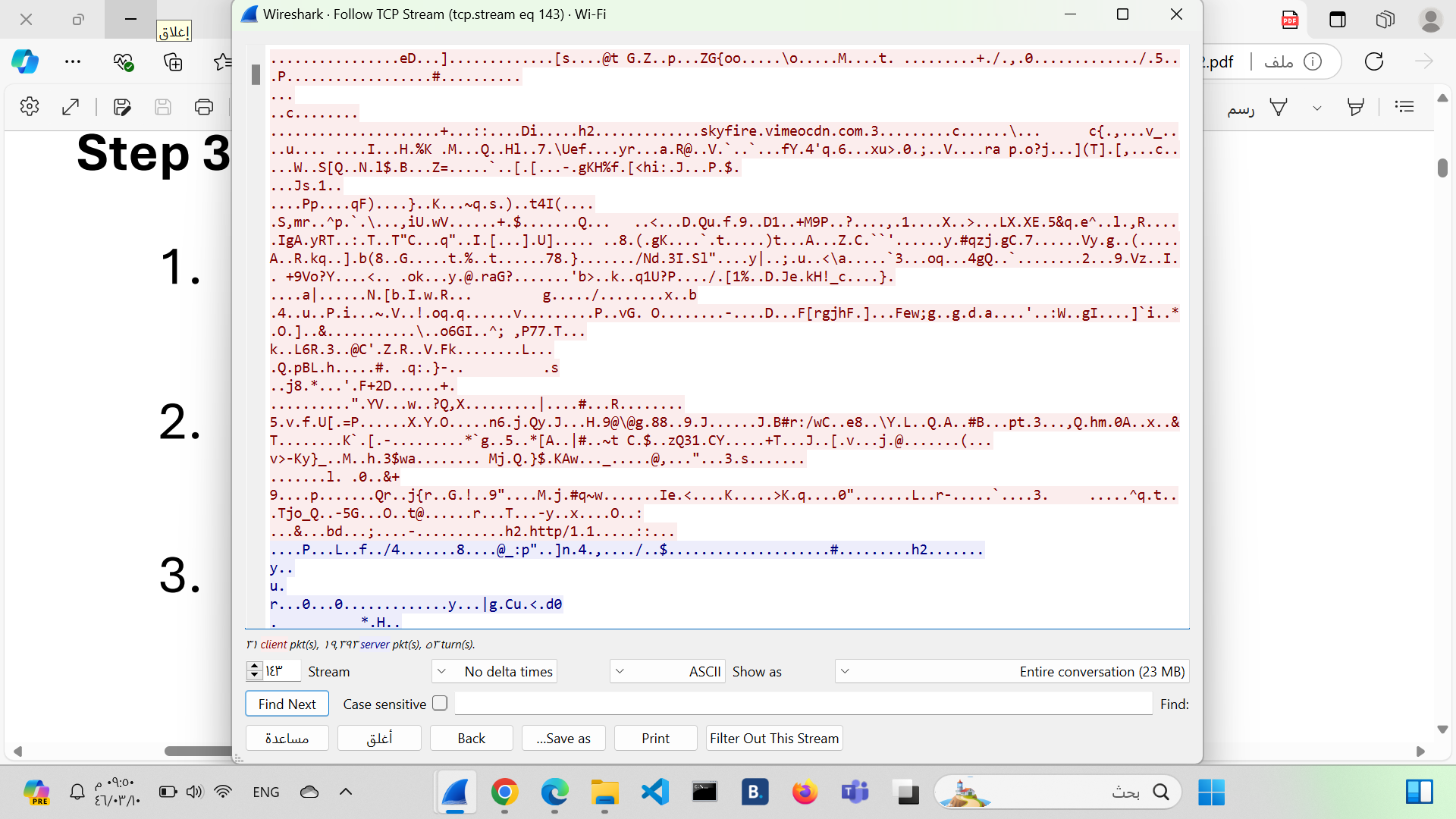


**200 OK**

**Part 2: Analyzing TCP/IP Traffic.**

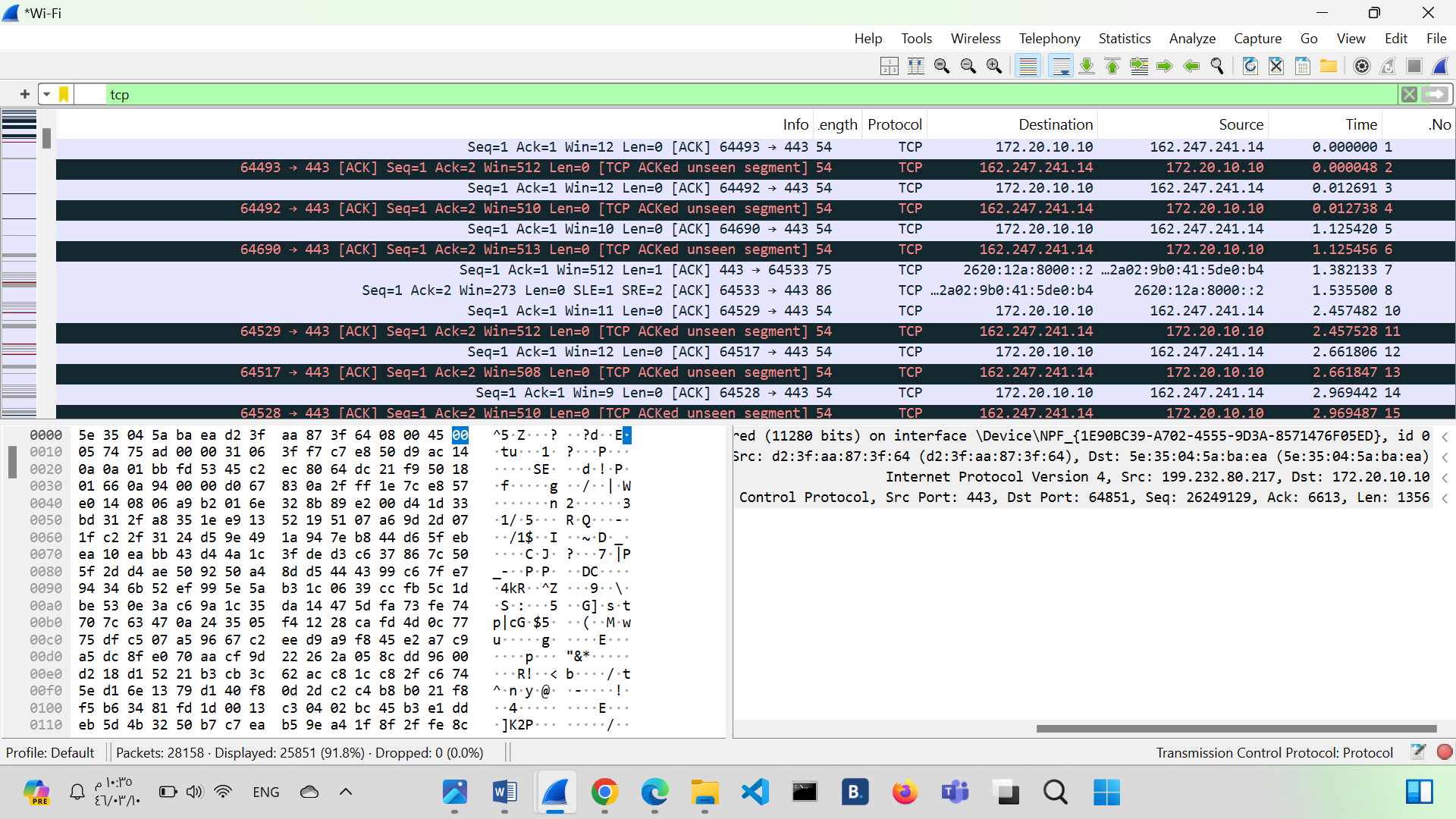
**Task 1: Filter TCP packets**



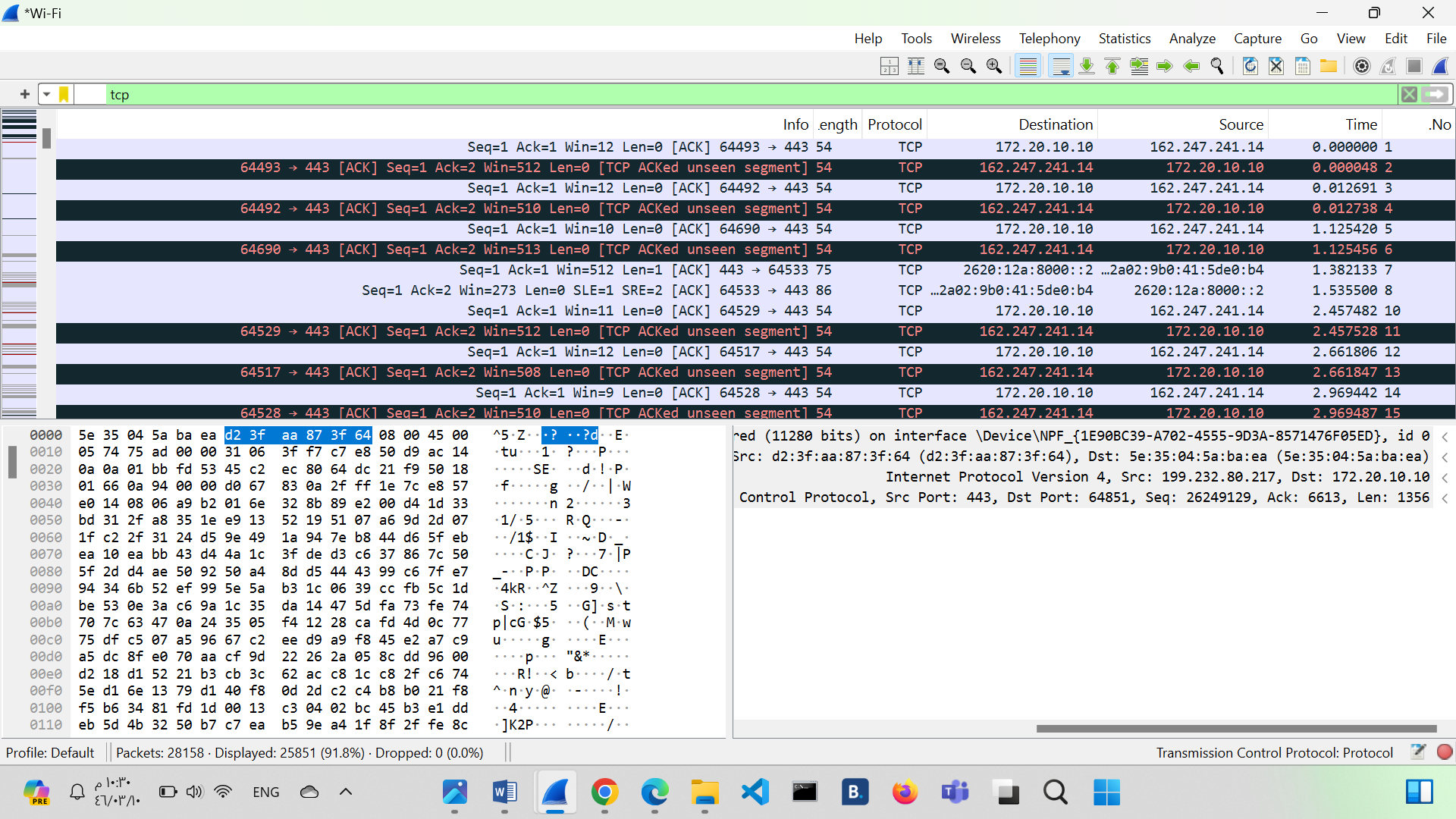


**Task 2: Analyze TCP handshake and investigate Data Transfer and Termination**

**TCP three-way handshake**



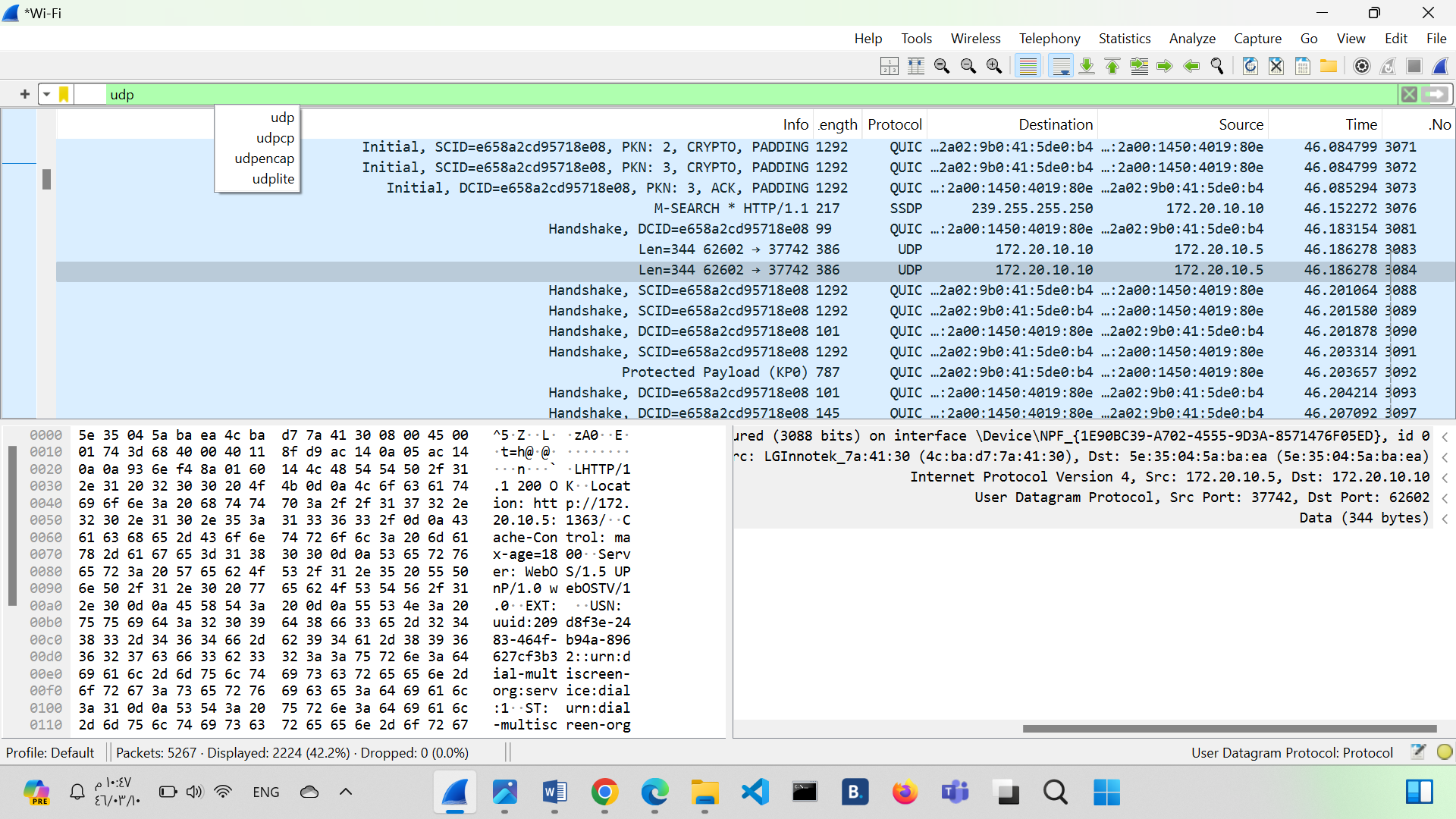
**ACK**

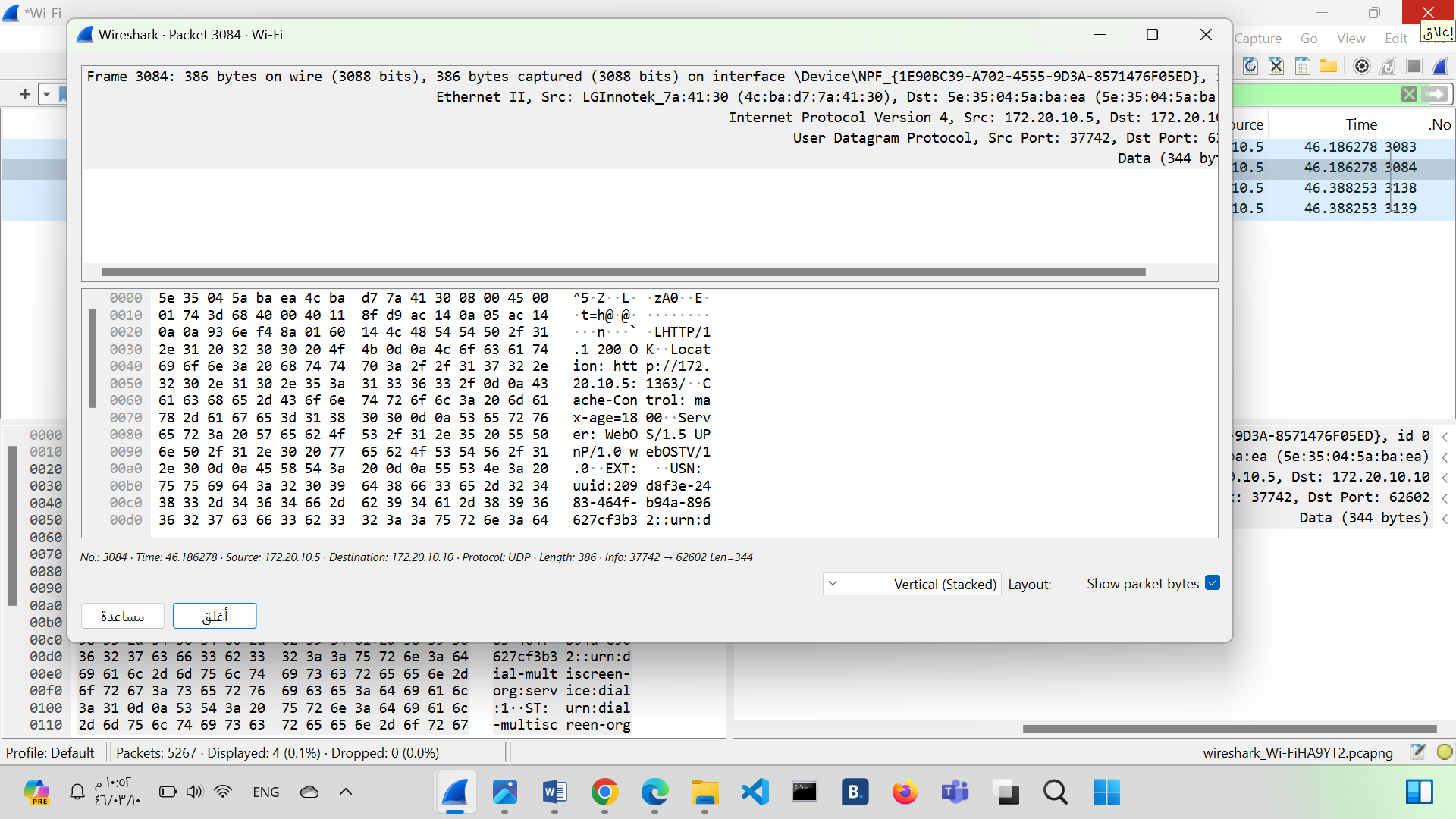


**Part 3: Capturing and Analyzing UDP Traffic**

**Task 1: Generate UDP traffic and capture packets**

**Task 2: Filter and analysis UDP Packets**





**Part 4: Comparing TCP and UDP by filling in the following tables. Save your work (e.g., in an MS Word document), and upload it to your online git repo.**

**Task 1: Fill in the following table and provide reasons**

|  |  |  |
| --- | --- | --- |
| **Reason** | **TCP or UDP** |  |
| **TCP** establishes a connection using a three-way handshake before data transfer, ensuring reliable communication.  **UDP** does not establish a connection, making it faster but less reliable. | **UDP** -> Connectionless  **TCP** -> Connection-oriented | **Reliability and Connection Establishment** |
| **TCP** ensures data is received in the correct order and checks for errors, retransmitting lost packets.  **UDP** sends packets independently, without checking for errors or order, suitable for real-time applications. | **UDP** -> No guarantee of data integrity/order  **TCP** -> Ensures data integrity and ordering | **Data Integrity and Ordering** |

**Task 2: Identify the use Cases and Performance of TCP and UDP.**

|  |  |  |
| --- | --- | --- |
| **UDP** | **TCP** |  |
| Live Streaming - Online Gaming - Voice over IP - Broadcast and Multicast. | Web Browsing – Email - File Transfers - Remote Access. | **Use cases** |
| **Speed:** Faster than TCP because it has lower overhead and does not establish a connection.  **Connectionless:** Sends data without establishing a connection, which reduces latency.  **Unreliable:** Does not guarantee data delivery or order, making it less reliable than TCP.  **Efficiency:** Suitable for applications where speed is more critical than reliability. | **Reliability:** TCP ensures data is delivered accurately and in the correct order. It uses error-checking, acknowledgments, and retransmissions.  **Connection-Oriented:** Establishes a connection before data transfer, using a three-way handshake.  Flow Control: Manages data flow to prevent congestion and ensure efficient transmission.  **Overhead:** Higher due to error-checking and connection management, which can result in slower performance compared to UDP. | **Performance** |