Lab Activities

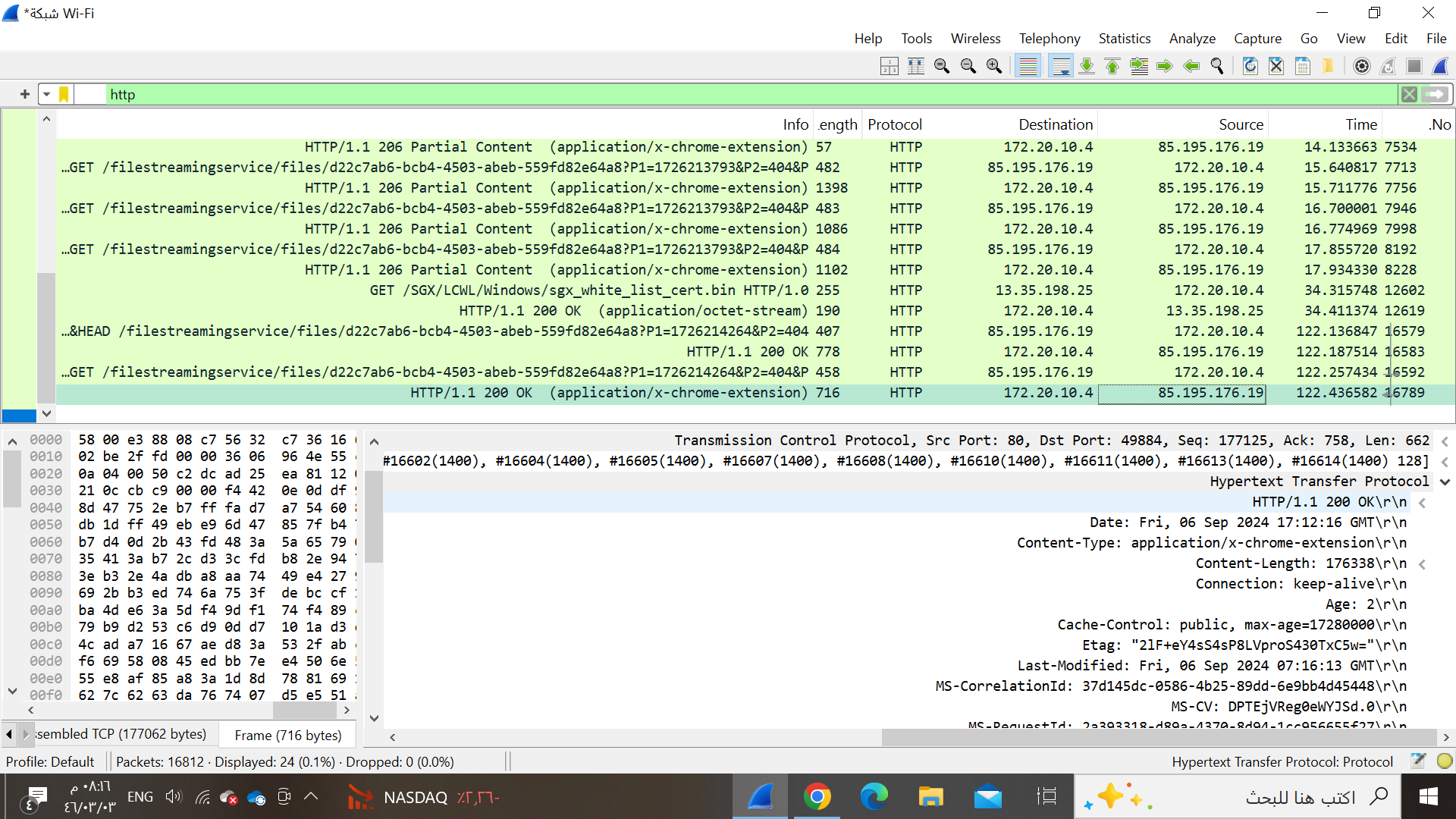
**Part 1: Capturing HTTP Traffic**

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

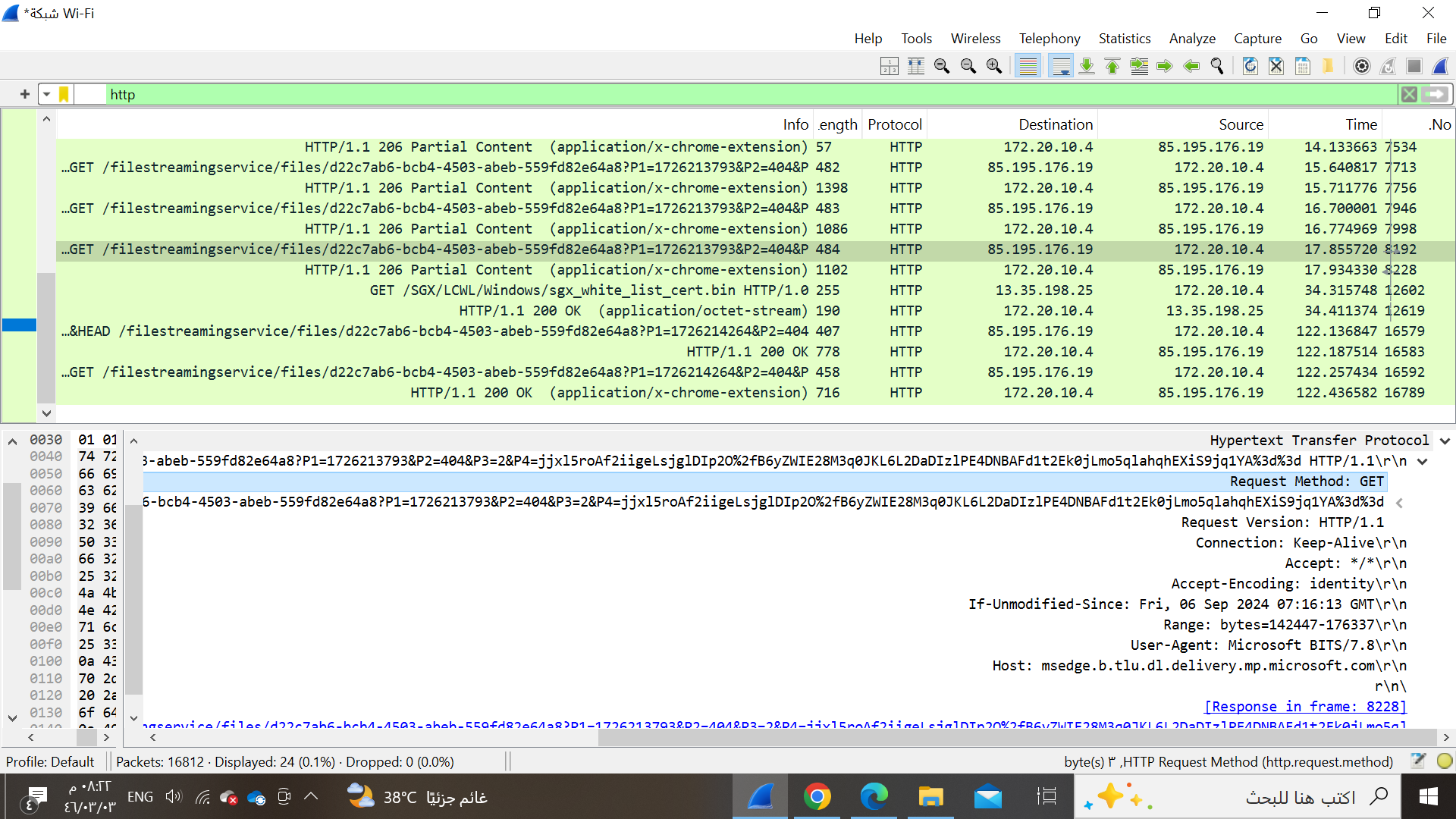
Step 2: Select the network interface connected to the internet (e.g., Ethernet or Wi-Fi).

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

Step 3: Observe the HTTP request and response messages. Note the method (GET, POST), URL, response codes (200 OK, 404 Not Found), etc.

.

**200 OK**



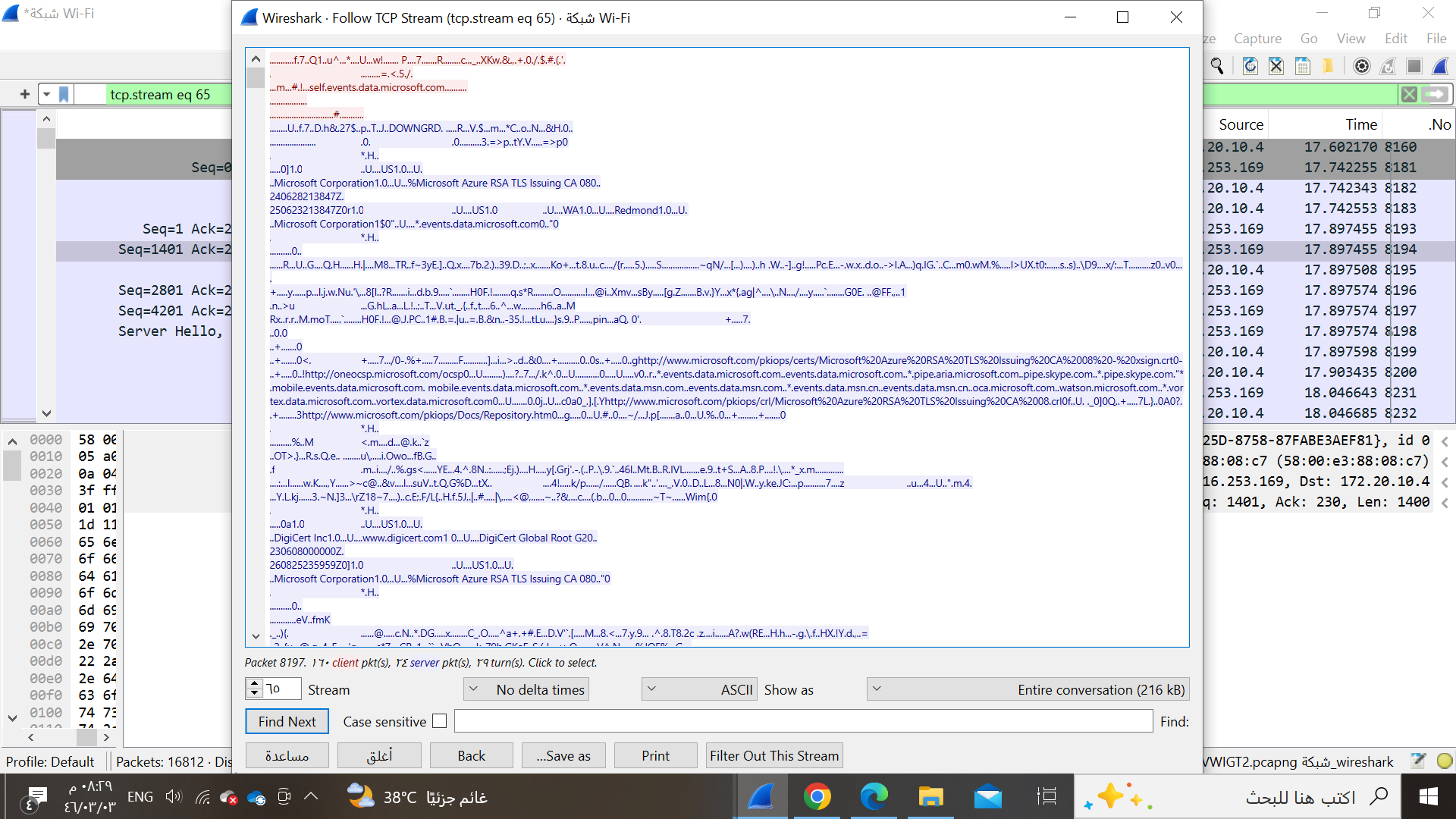
Request method :GET

URL: <http://msedge.b.tlu.dl.delivery.mp.microsoft.com/filestreamingservice/files/d22c7ab6-bcb4-4503-abeb-559fd82e64a8?P1=1726213793&P2=404&P3=2&P4=jjxl5roAf2iigeLsjglDIp2O%2fB6yZWIE28M3q0JKL6L2DaDIzlPE4DNBAFd1t2Ek0jLmo5q>]

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

Task 1: Filter TCP packets

Step 3: Right-click on the packet and select "Follow" -> "TCP Stream"



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

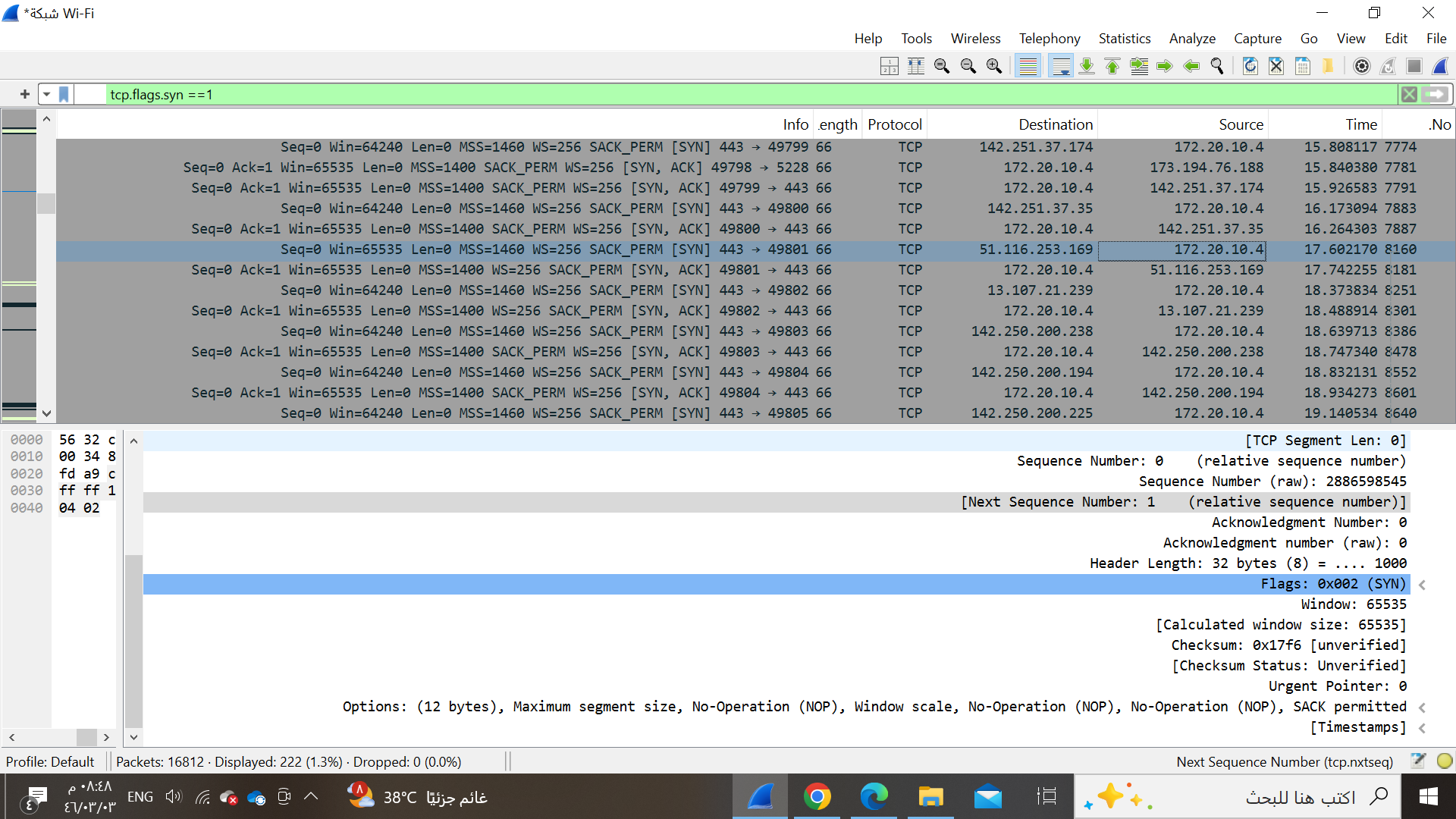
Step 1: Find and select packets related to the TCP three-way handshake:

SYN: Initiates a connection.

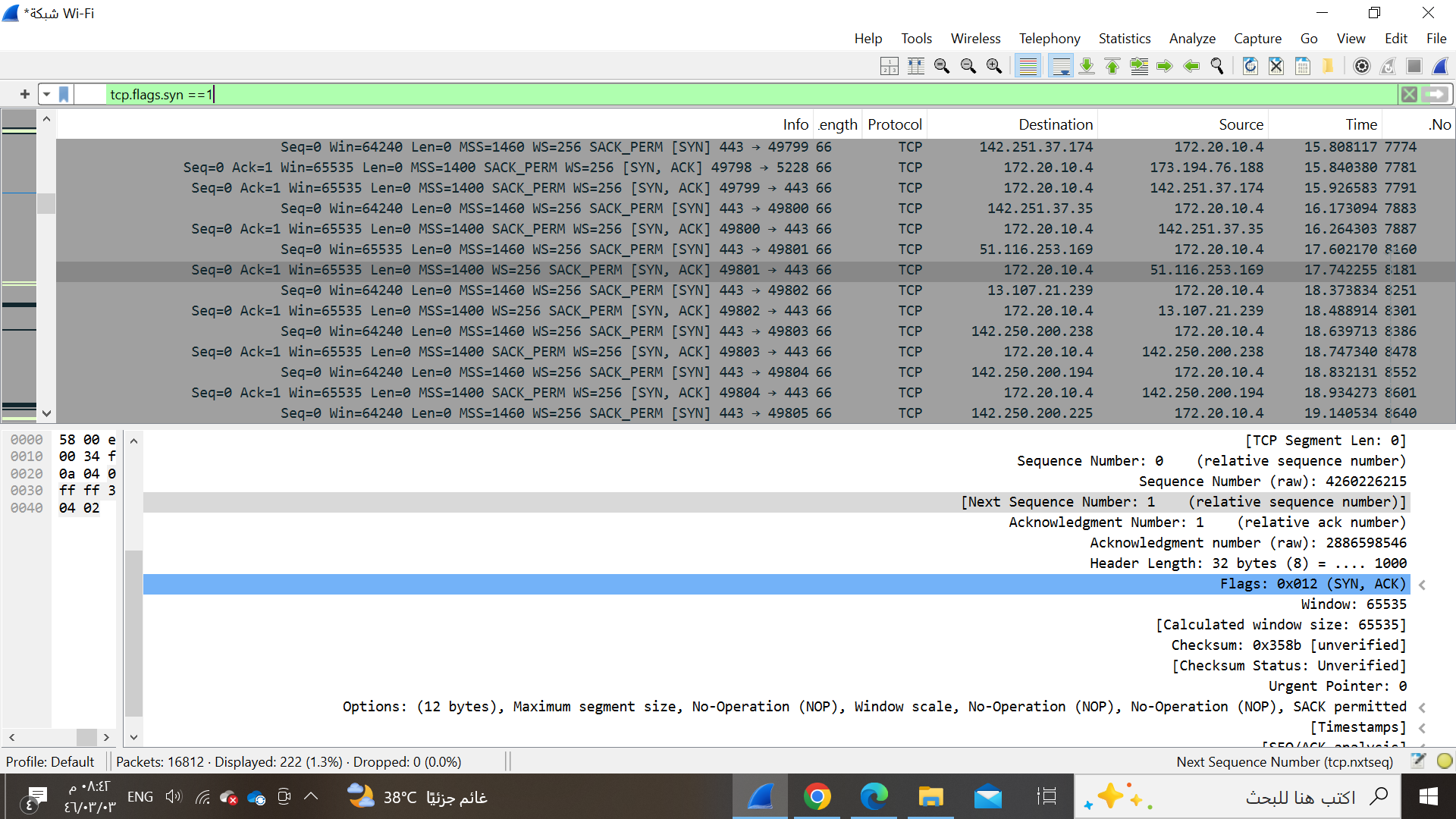
SYN-ACK: Acknowledges and responds to the SYN.

ACK: Acknowledges the SYN-ACK and establishes the connection.

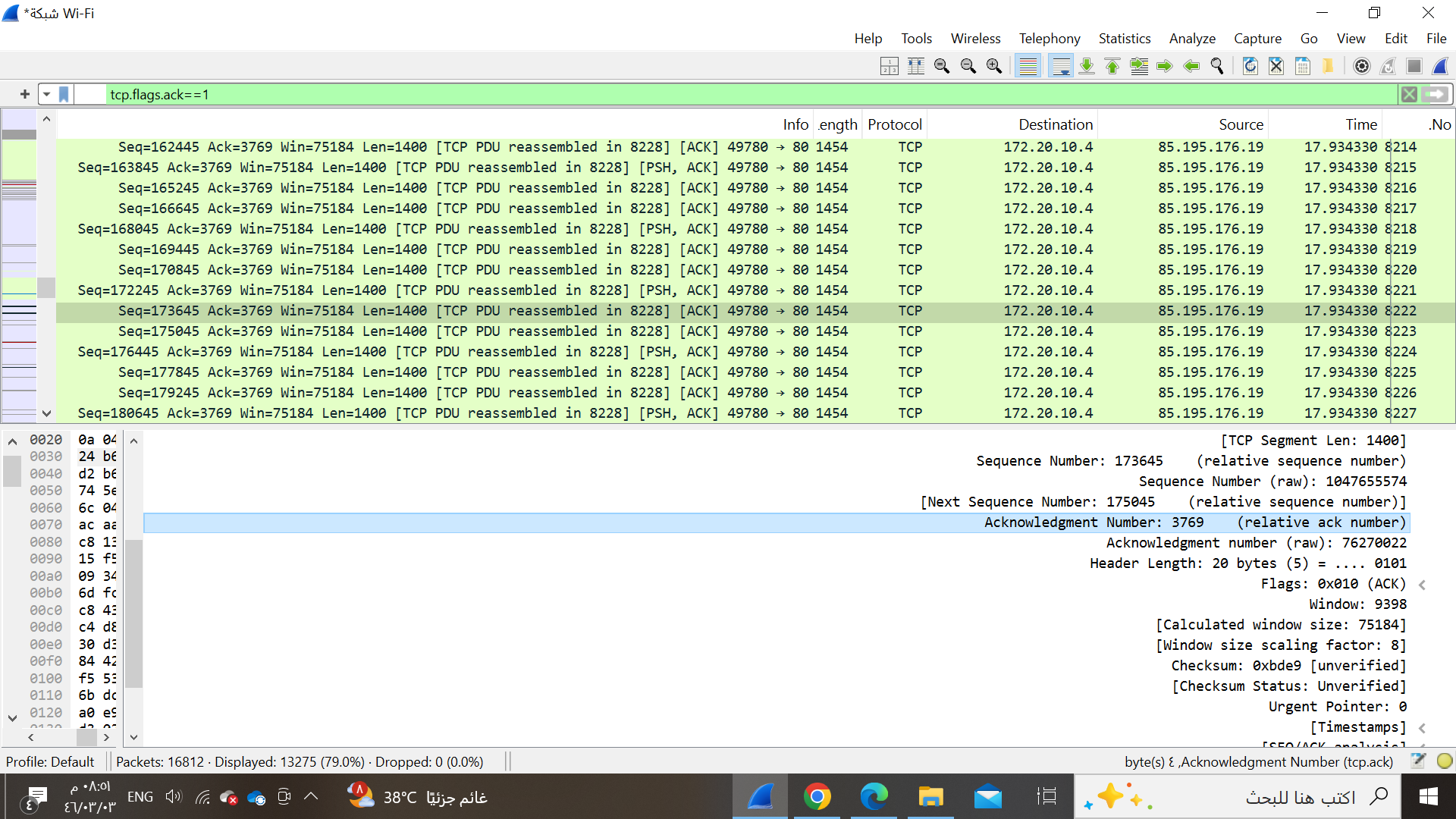
SYN



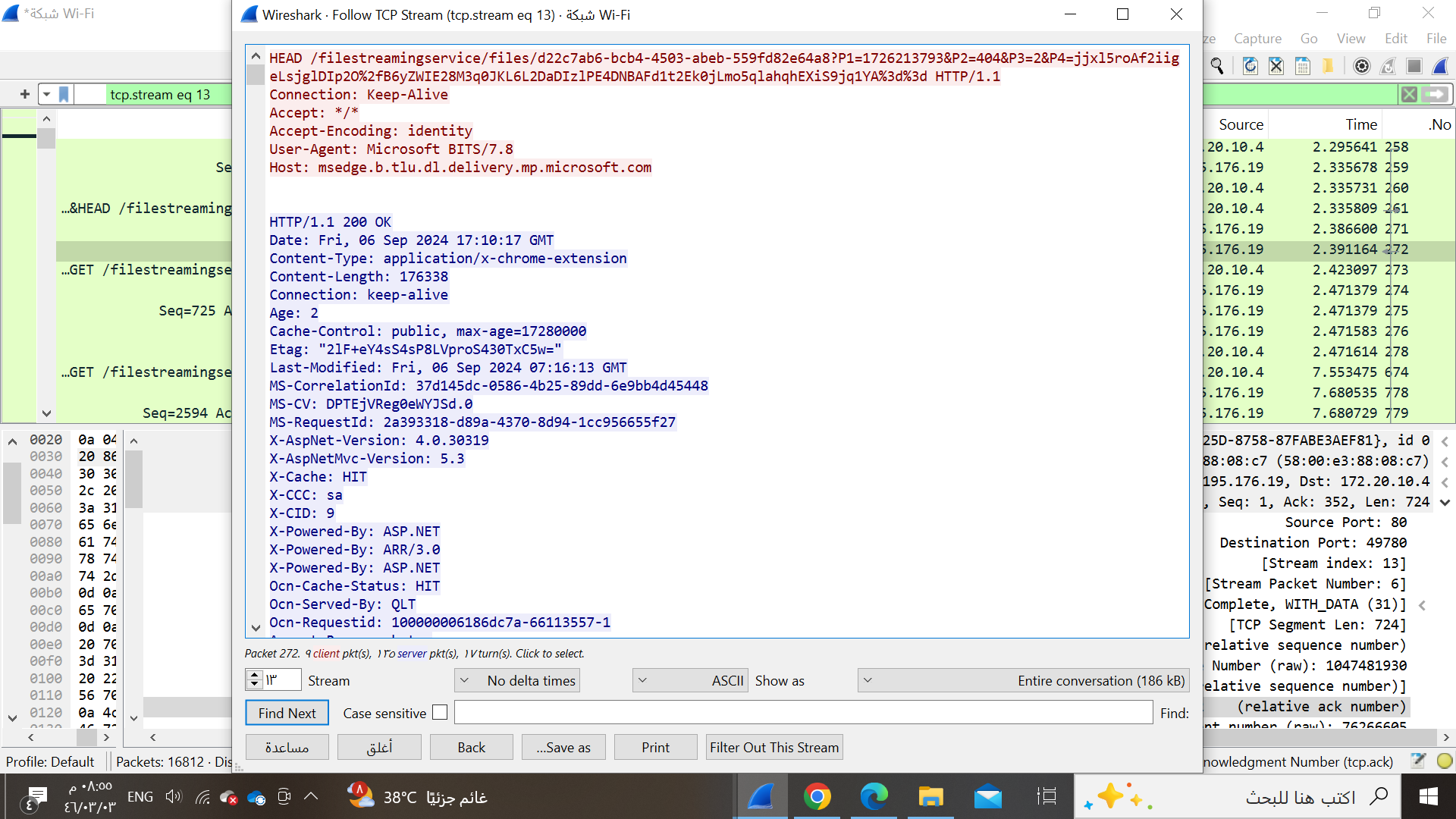
SYN-ACK:



ACK: Acknowledges numbers



the data packets exchanged between the client and server



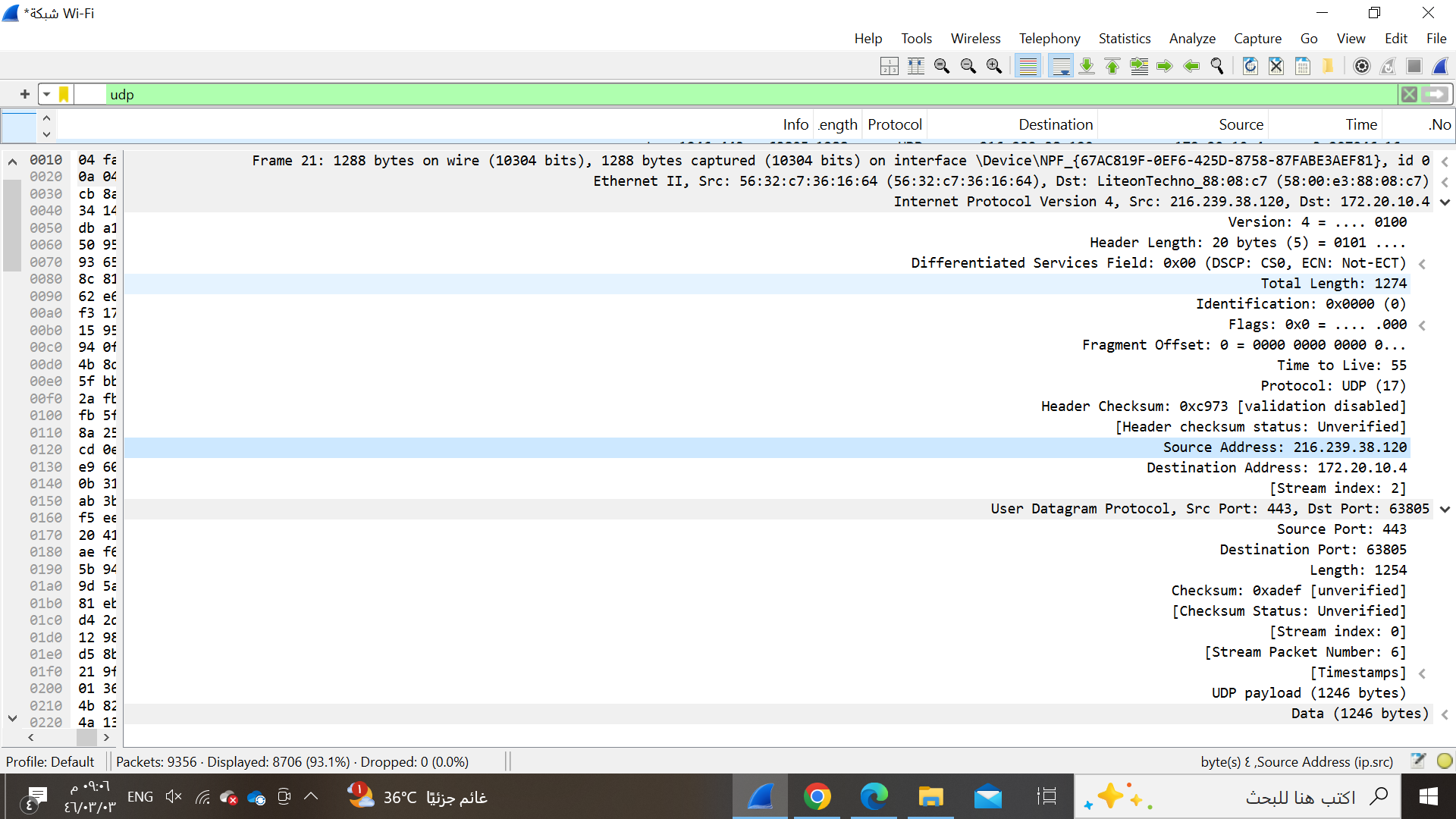
**Part 3: Capturing and Analyzing UDP Traffic**

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

Step 3: Start capturing packets in Wireshark while the UDP application is running.

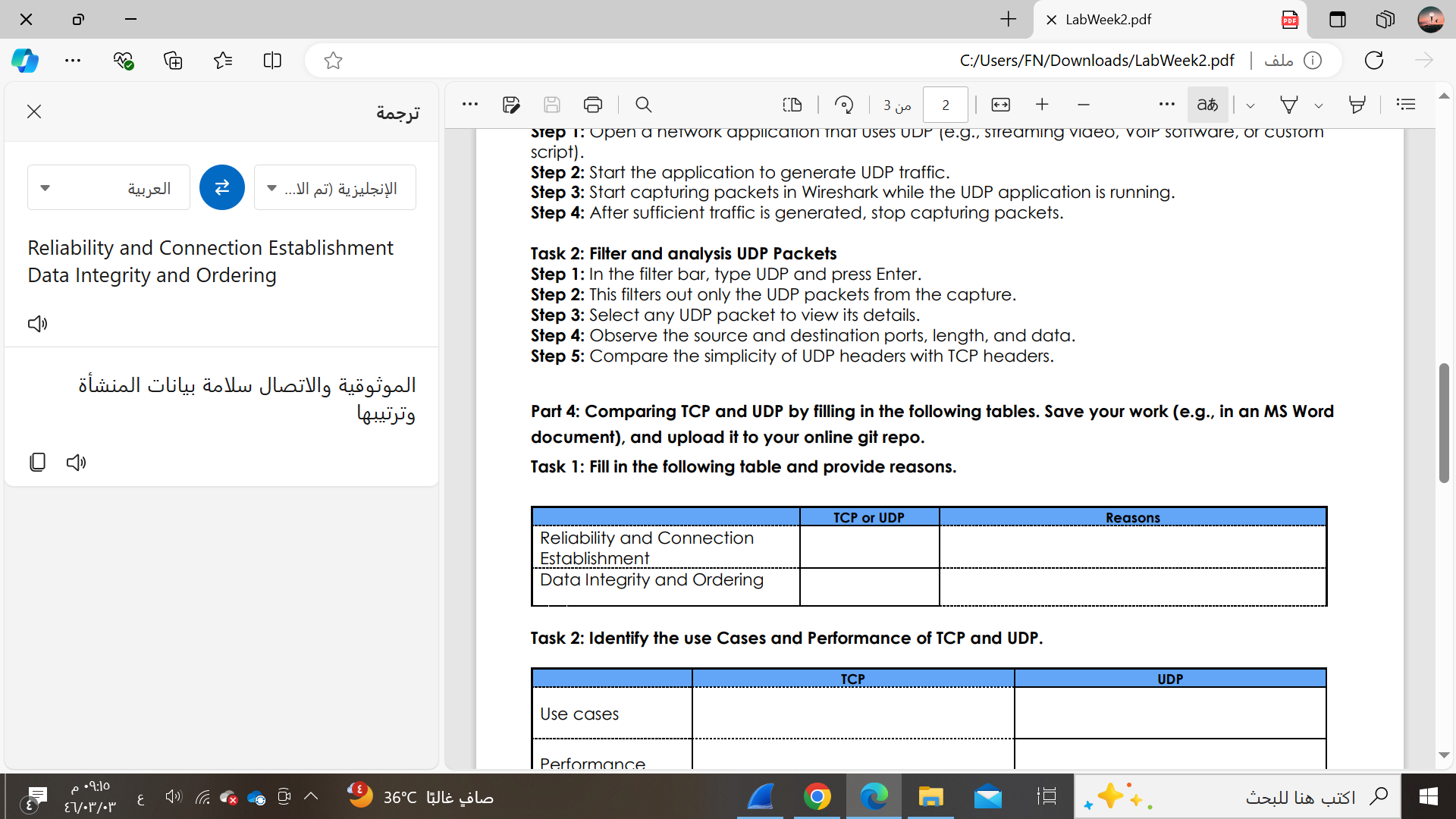
**Task 2: Filter and analysis UDP Packets**

Step 4: Observe the source and destination ports, length, and data.



Part 4: Comparing TCP and UDP

Task 1: Fill in the following table and provide reasons



**Reliable**:

TCP is designed to ensure that data is transmitted accurately and in the correct order. It achieves this through error-checking, acknowledgments, retransmissions of lost packets, and ordering of out-of-sequence data. This makes TCP highly reliable

**Connection-Oriented**:

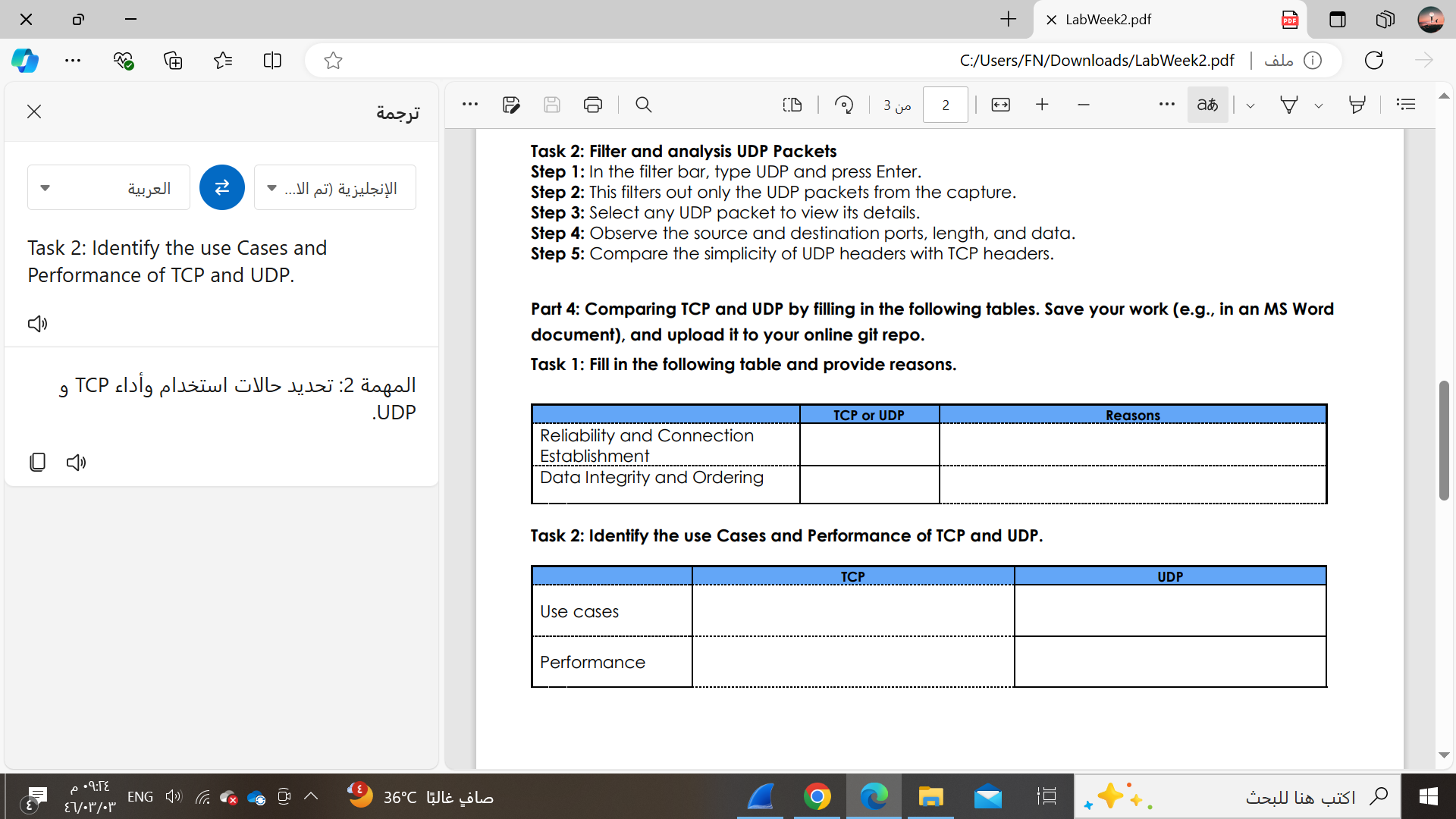
TCP requires a connection to be established before data transfer begins. This is done through the three-way handshake process, which ensures that both the sender and receiver are ready and synchronized for data transfer

**Data Integrity**:

TCP uses checksums to detect errors in the data and retransmits any corrupted or missing packets to ensure that the data received is complete and accurate.

**Guarantees Ordering**:

TCP assigns sequence numbers to packets, ensuring that they are reassembled in the correct order by the receiver. If packets arrive out of order, TCP will reorder them before passing them to the application.



Use Cases

TCP: Web Browsing

UDP: Streaming Media (Video, Audio)

Performance:

**TCP** is best suited for use cases where reliability, data integrity, and ordered delivery are critical, even at the cost of speed and higher overhead.

**UDP** is preferred in scenarios where speed and low latency are more important than reliability, making it ideal for real-time applications where occasional data loss is acceptable