GHAIDA ALAMMAR

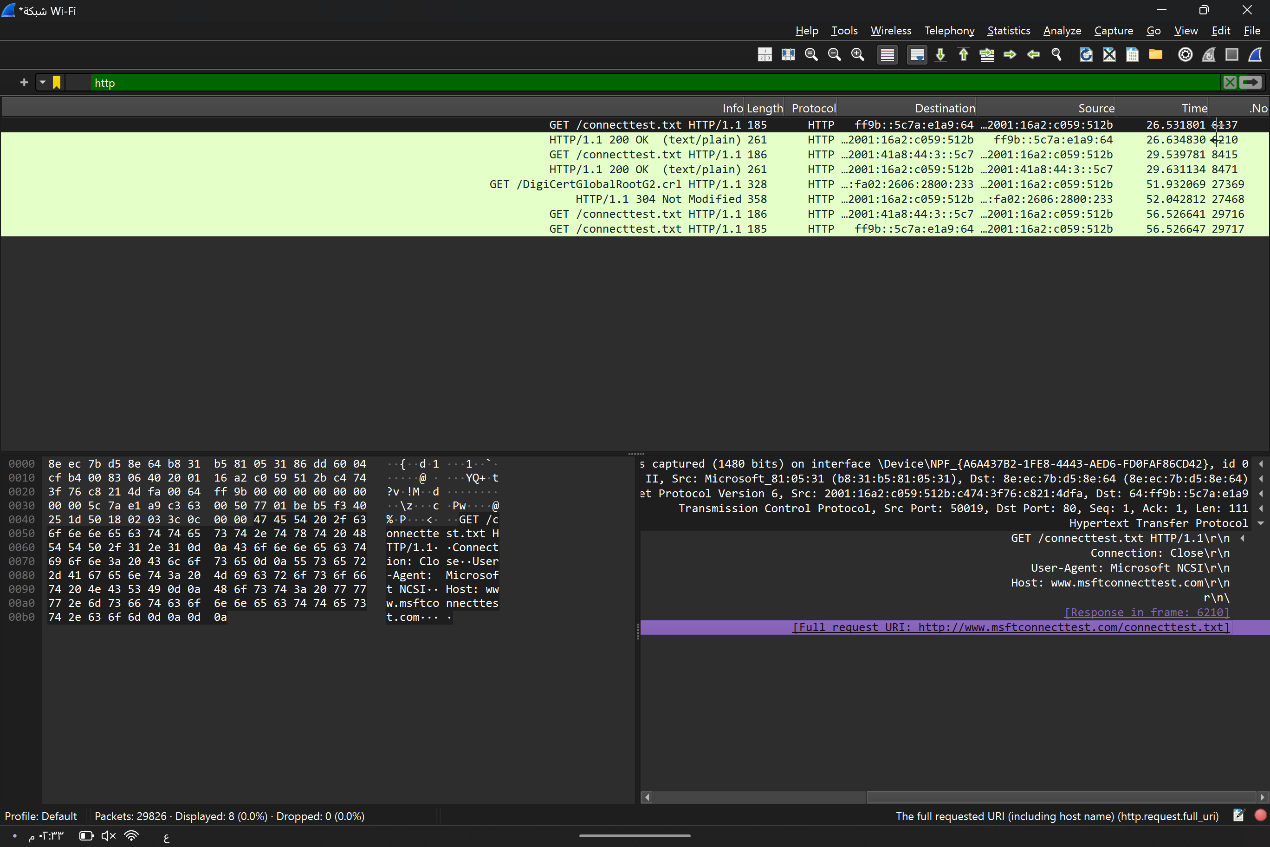
**Part1:**

Task 2:

First packet contains: GET method

Second packet : RESPONSE 200 OK

[Full request URI: <http://www.msftconnecttest.com/connecttest.txt>]



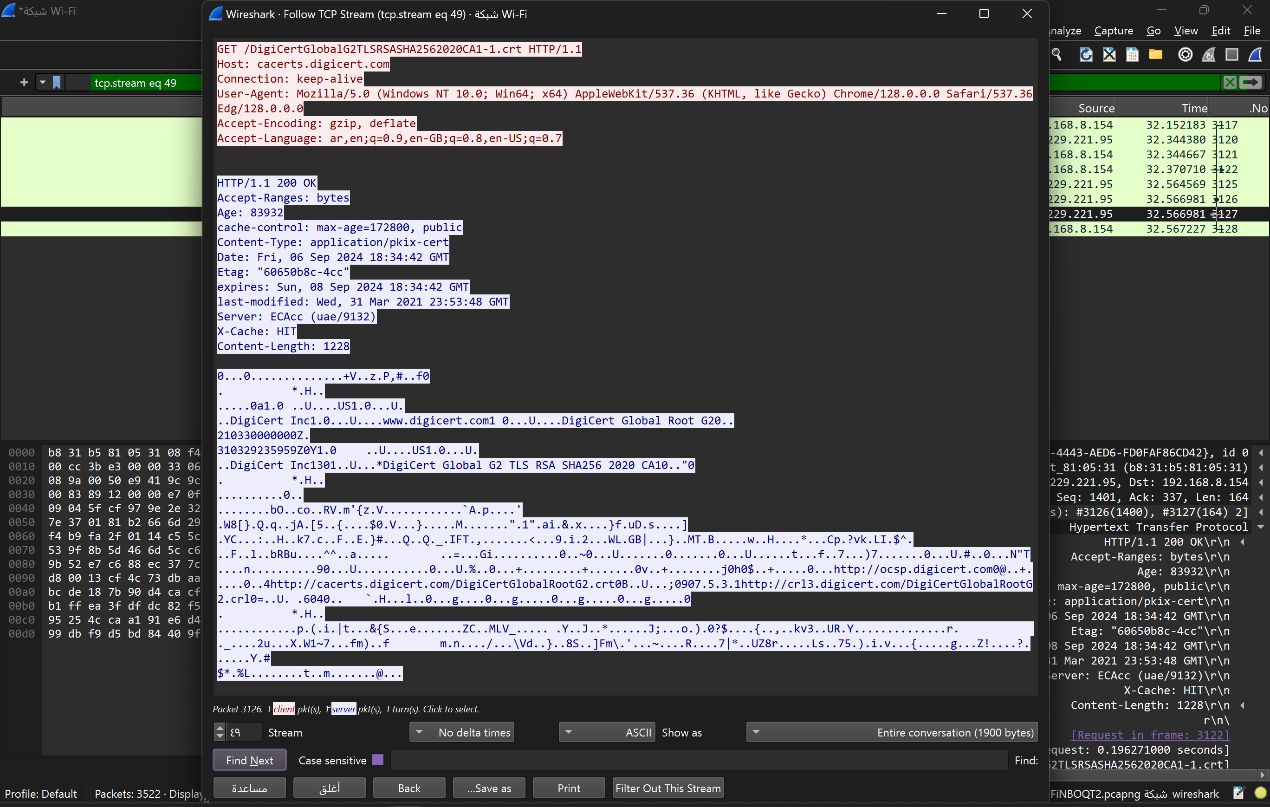
**Part2:**

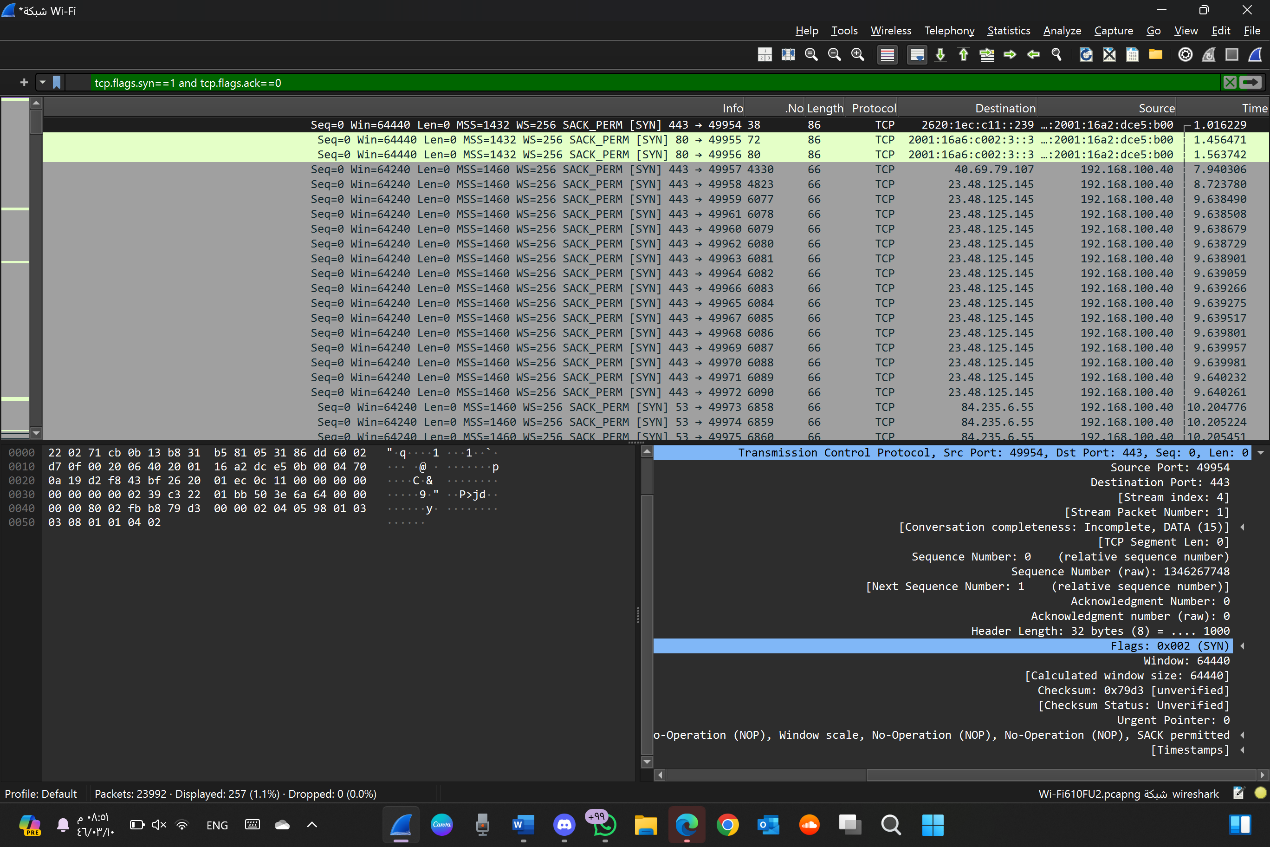
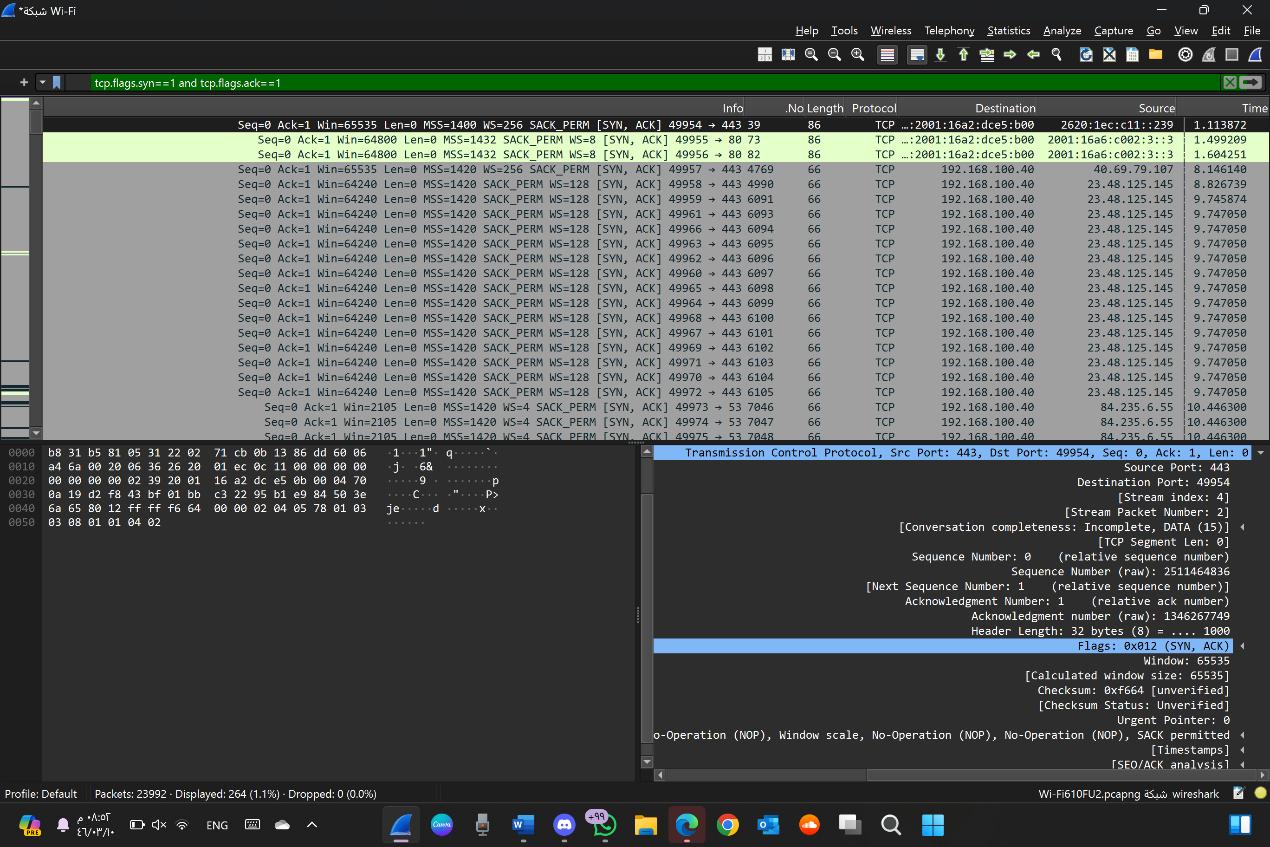
**صورة تحتوي على نص, لقطة شاشة, برمجيات, برامج الوسائط المتعددة

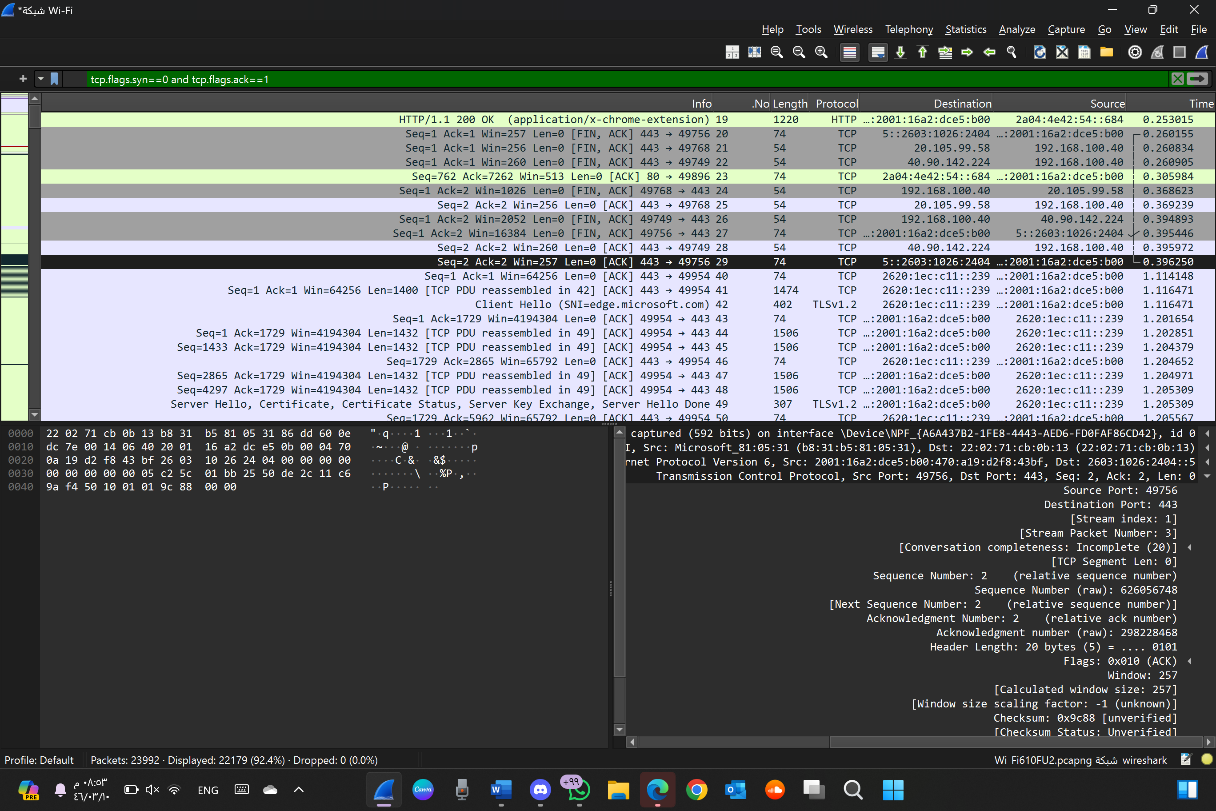
تم إنشاء الوصف تلقائياً**

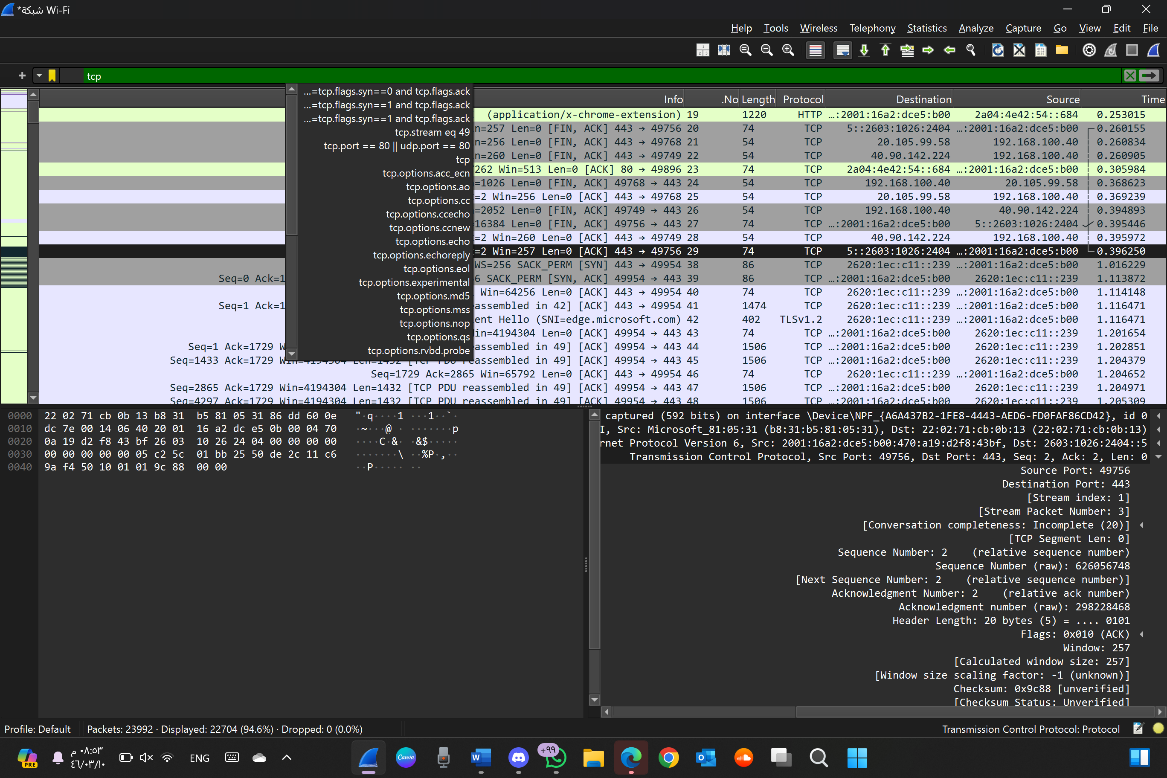
**صورة تحتوي على نص, الإلكترونيات, لقطة شاشة, برمجيات

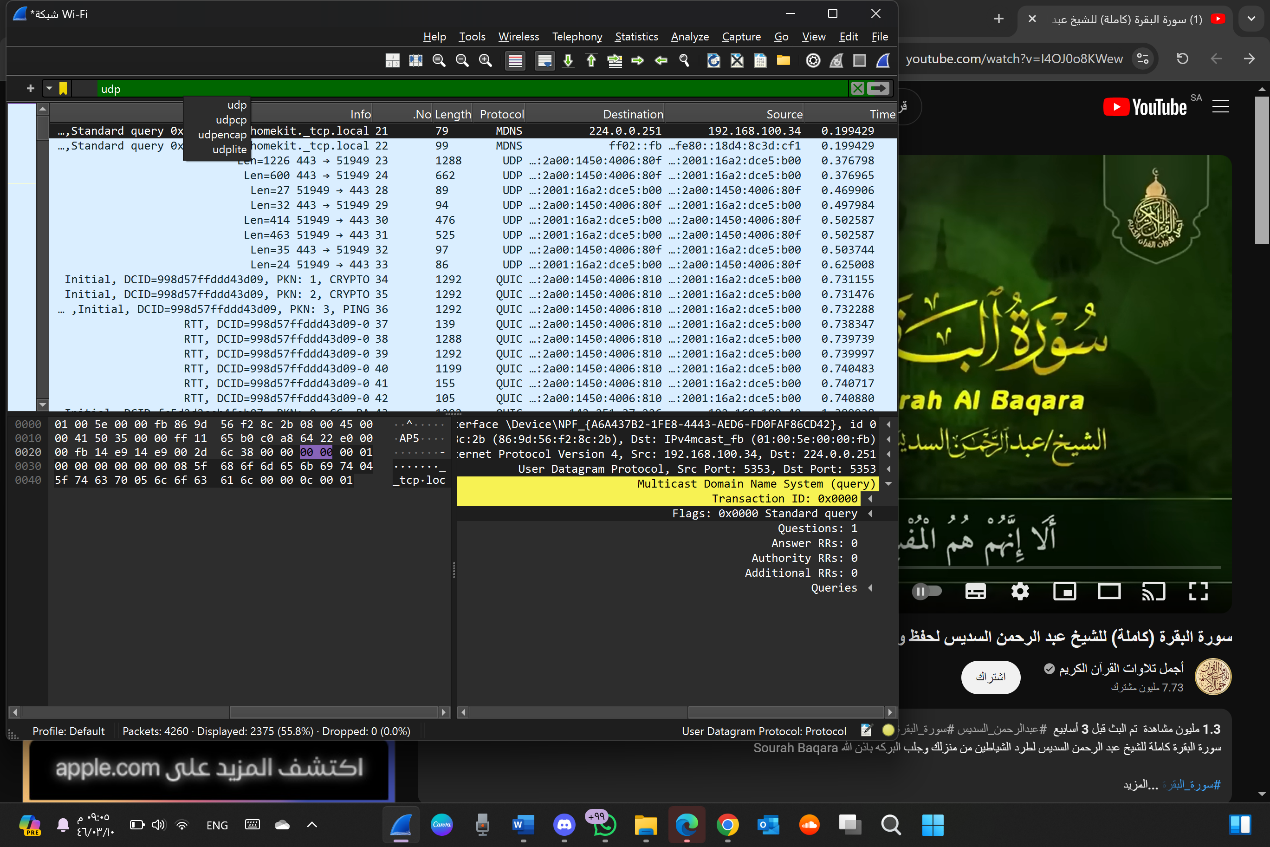
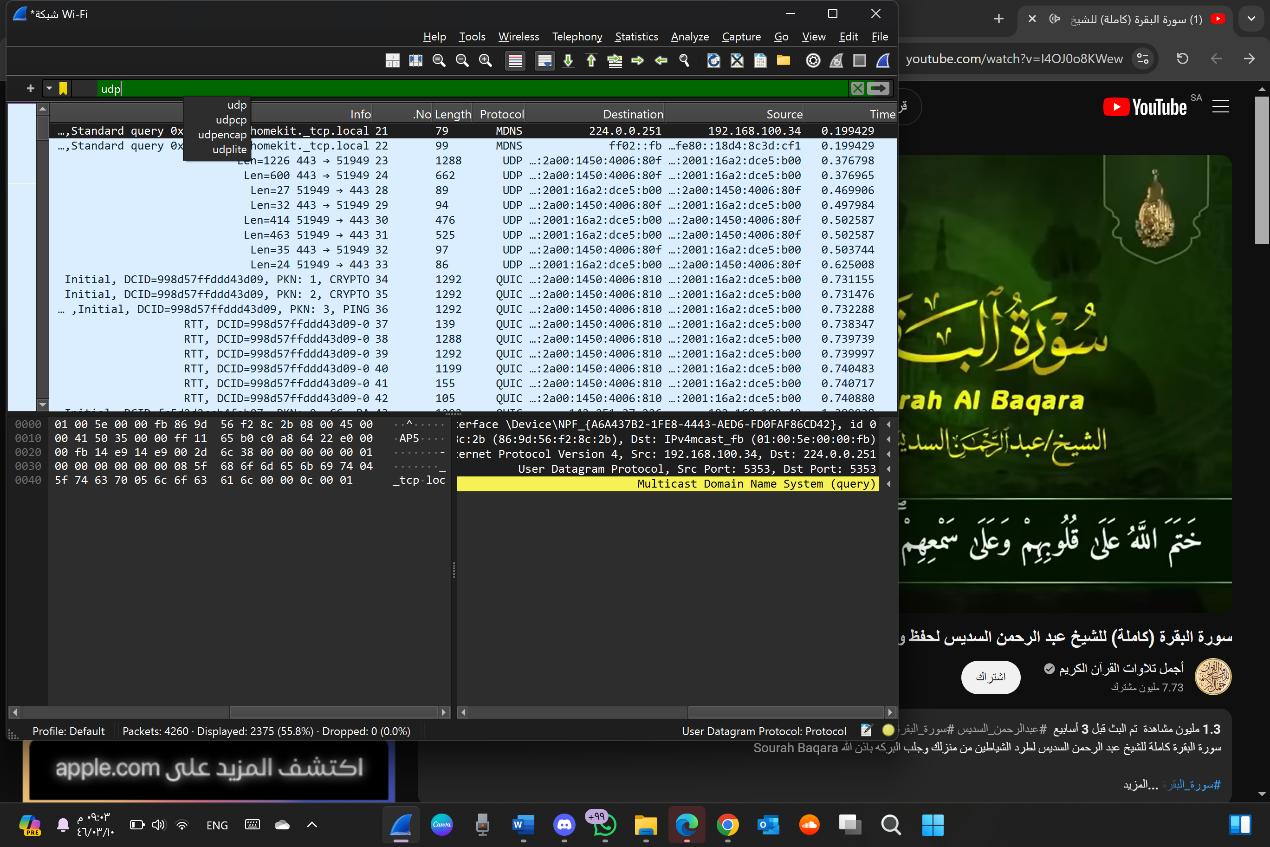
تم إنشاء الوصف تلقائياً**

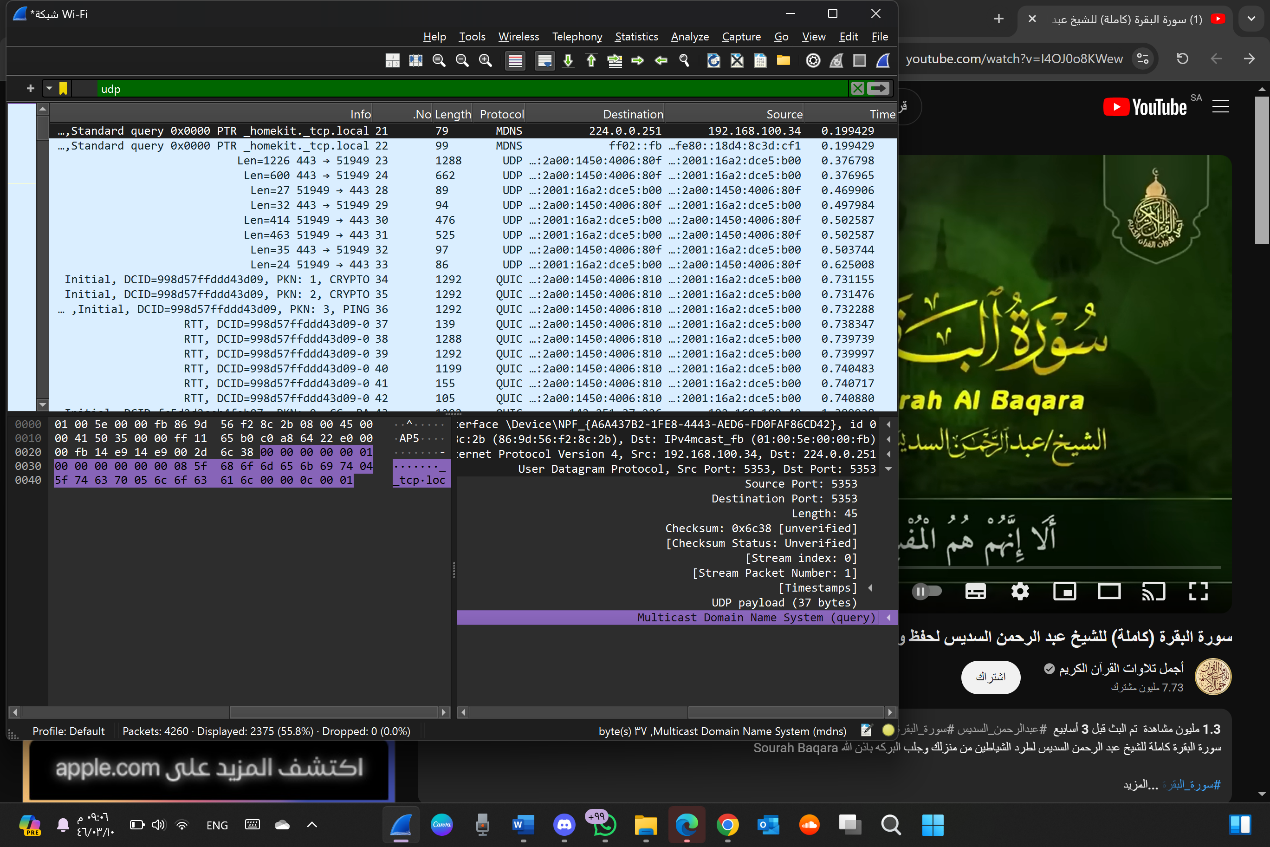
****











part 3:

Comparing the Simplicity of UDP Headers with TCP Headers

**UDP Header:** UDP headers are much simpler, consisting of only four fields: Source Port, Destination Port, Length, and Checksum. This simplicity makes UDP lightweight, with minimal processing overhead, which is ideal for scenarios requiring fast, low-latency communication without the need for reliability or ordered delivery.

**TCP Header**: TCP headers are more complex, containing many fields such as Source Port, Destination Port, Sequence Number, Acknowledgment Number, Data Offset, Flags (e.g., SYN, ACK, FIN), Window Size, Checksum, Urgent Pointer, and optional fields. These fields support connection establishment, flow control, error recovery, and reliable, ordered data transfer, but they also add significant overhead compared to UDP.

**Part4**:

|  |  |  |
| --- | --- | --- |
| Reasons | Tcp or udp |  |
| Reliable, connection-oriented protocol that establishes a connection with a three-way handshake before data transfer begins. It guarantees that all data is delivered without errors and in the correct order. UDP: Unreliable, connectionless protocol that does not establish a connection before sending data. There is no guarantee of data delivery, no ordering, and no error correction, making it faster but less reliable. | TCP | Reliability and Connection Establishment |
| Ensures data integrity and correct sequencing through acknowledgment and retransmission mechanisms. It maintains the order of packets, making it suitable for applications where data consistency is crucial. UDP: Does not ensure data integrity or order. Packets may arrive out of order, be duplicated, or get lost, but this reduces overhead and latency, making it suitable for real-time applications where speed is more critical than reliability | TCP | Data integrity and ordering |

**Task2:**

|  |  |
| --- | --- |
| UDP | TCP |
| File transfers (FTP), web browsing (HTTP/HTTPS), email (SMTP), and database synchronization where reliable communication is essential | Use case |
| Higher overhead due to features like error correction, flow control, and connection establishment. This ensures reliability but can result in slower performance. Ideal for applications that require guaranteed  delivery and data integrity. | Performance |