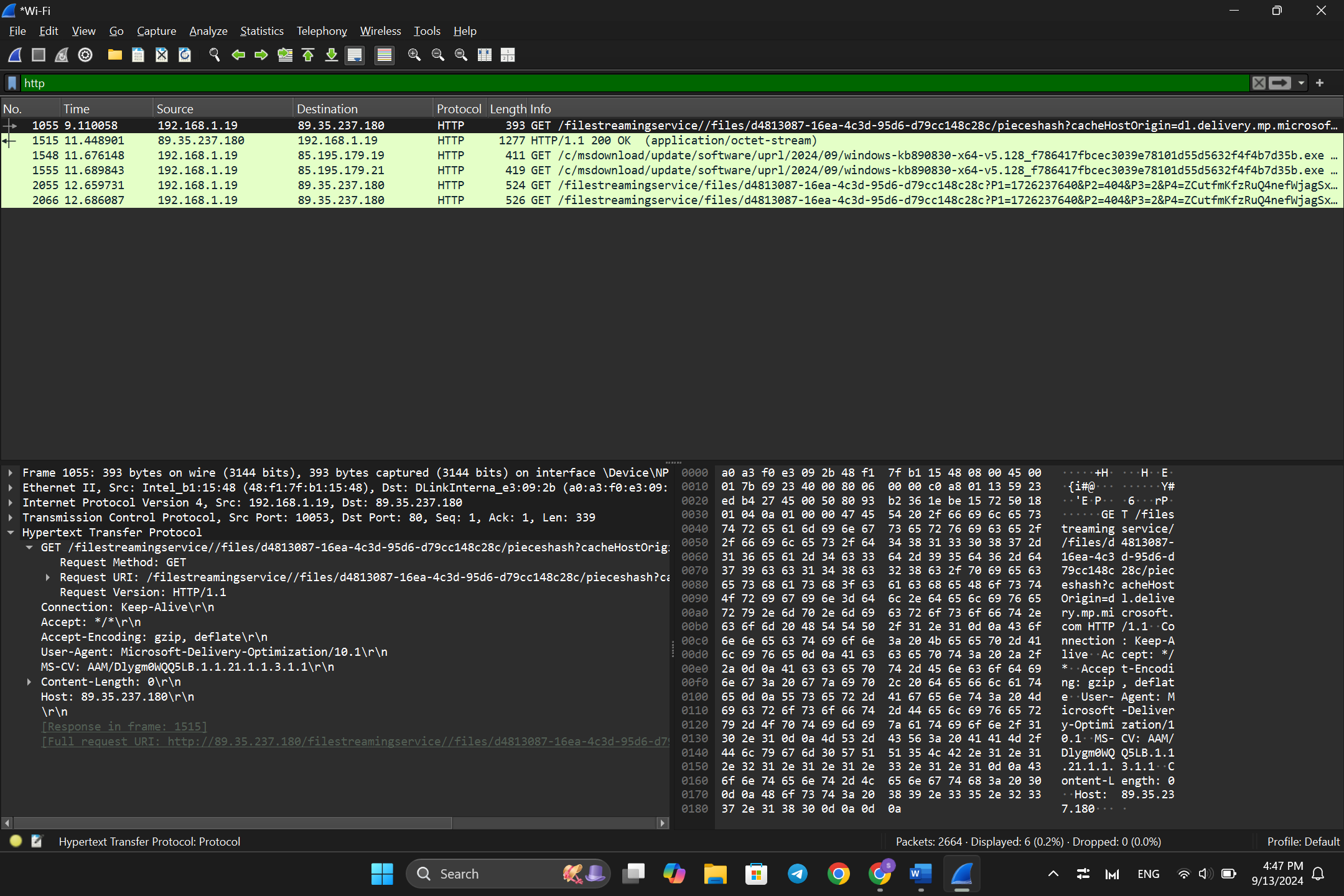
Lab Week 2

This lab session covers the usage of the Wireshark application to monitor and capture the outgoing and incoming packets from a network connection (WIFI, ethernet, etc.)

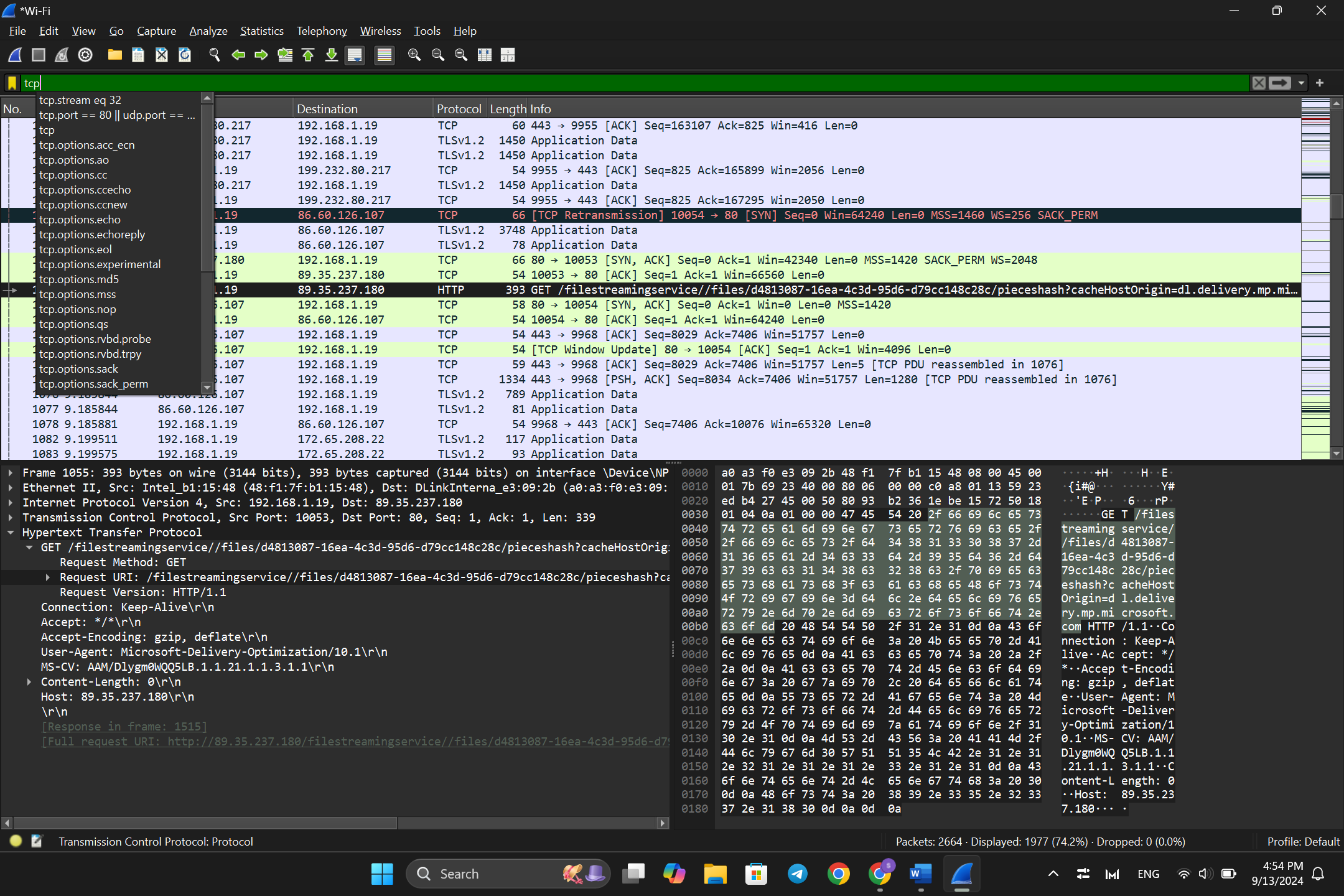
Part 1: Capturing HTTP Traffic.

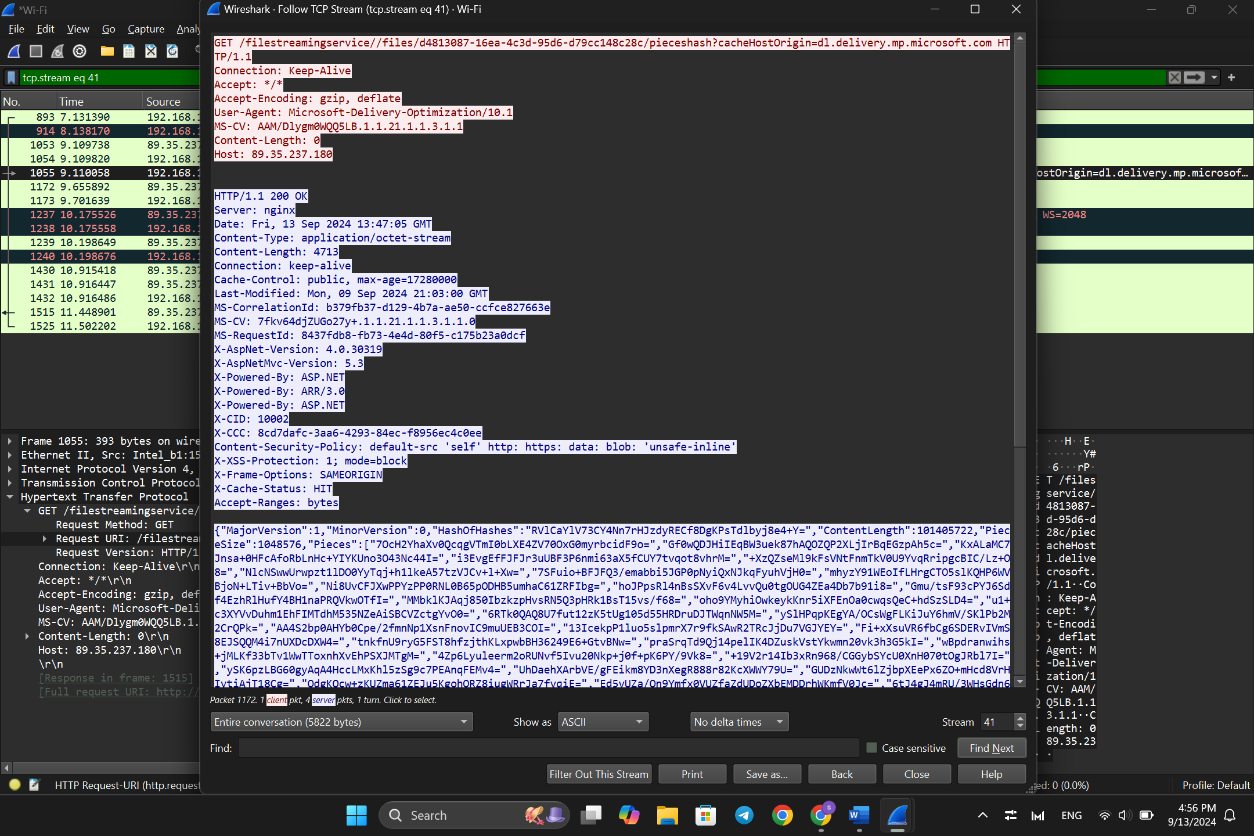
* Task1: Start Wireshark and capture packets.
* Task2: Filter HTTP packets and analyze them.

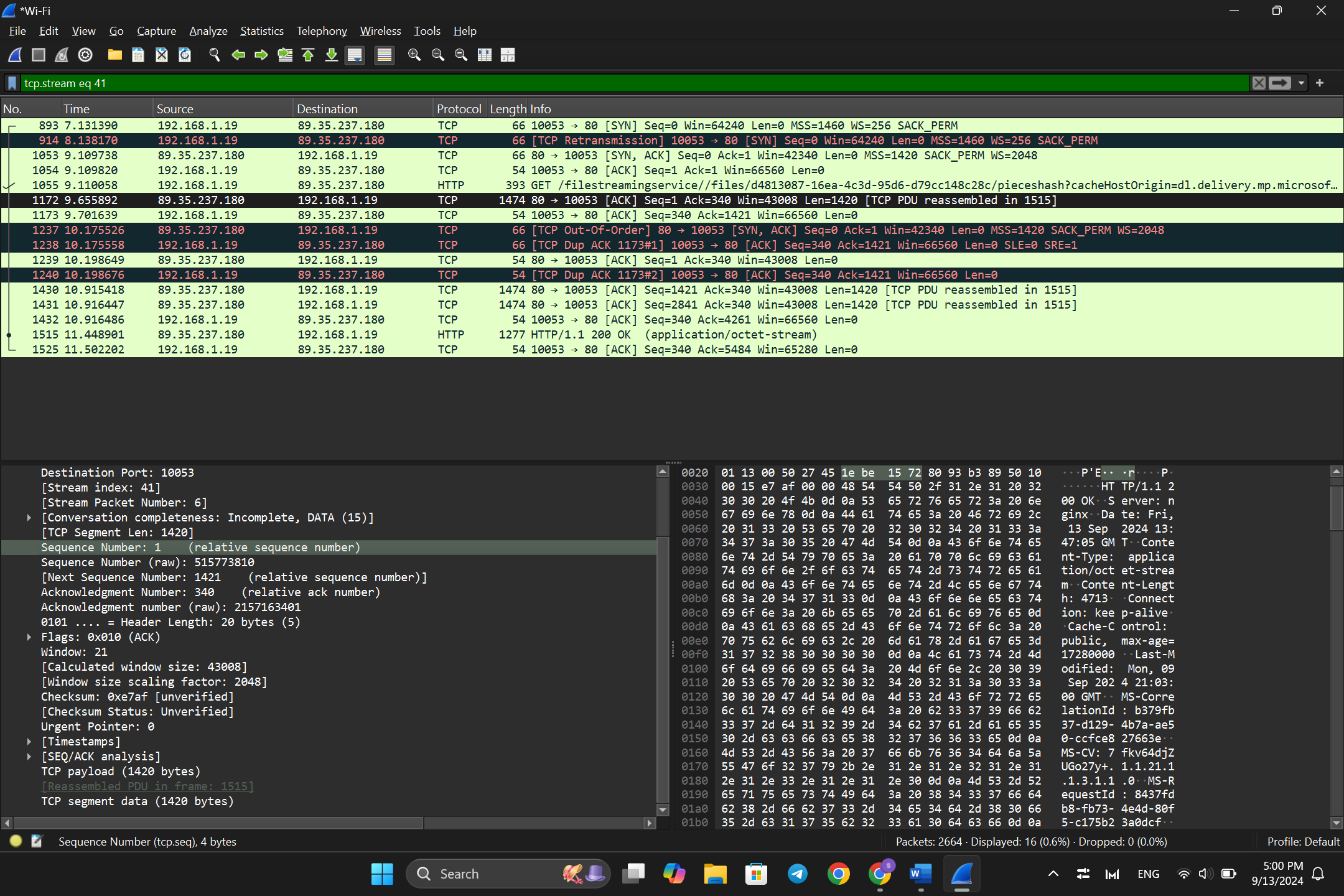
Observe the HTTP request and response messages:

* THE METHOD: GET
* THE URL: "/filestreamingservice//files/d4813087-16ea-4c3d-95d6-d79cc148c28c/pieceshash?cacheHostOrigin=dl.delivery.mp.microsoft.com"
* THE RESPONSE CODE: 200 OK

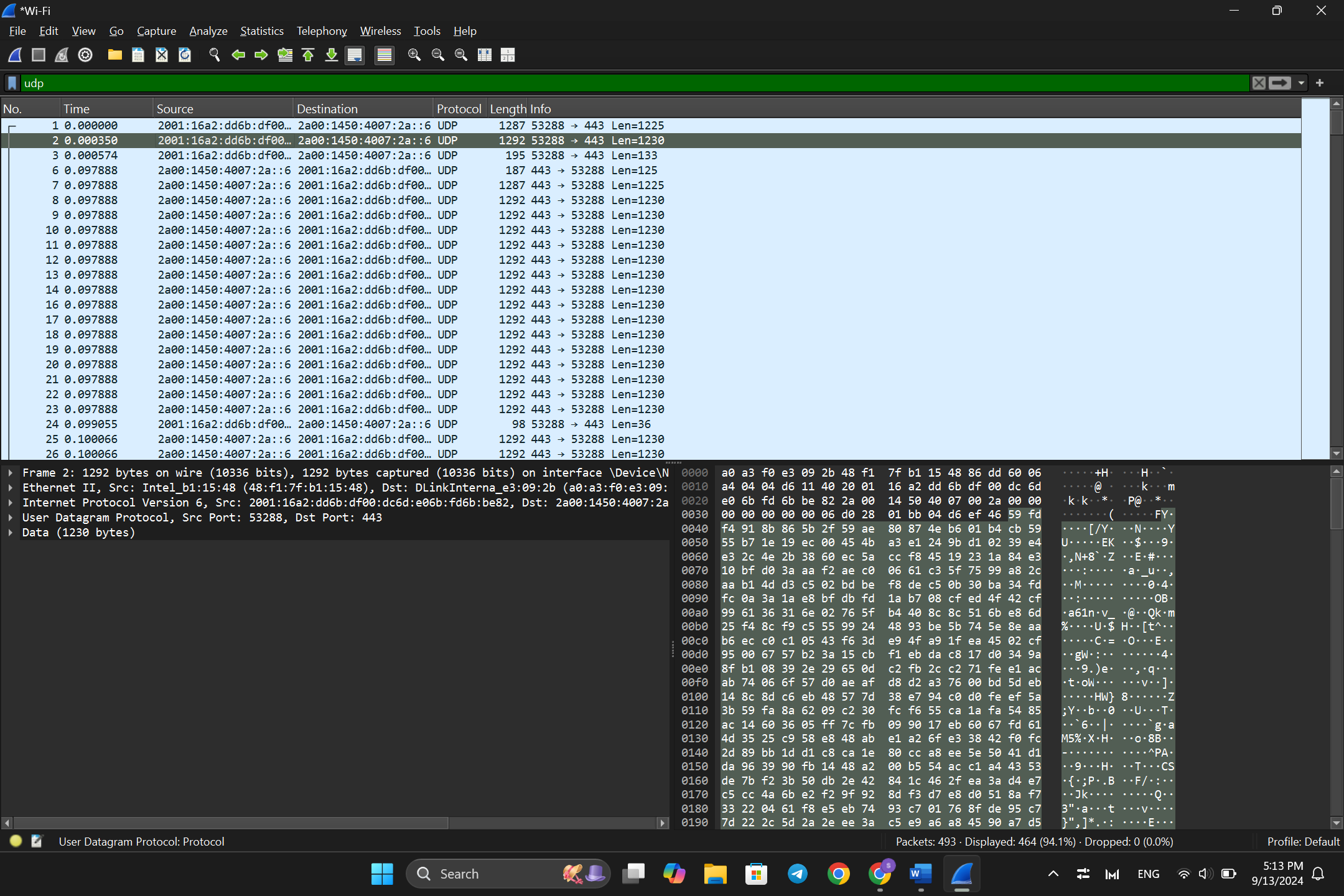
Part 2: Analyzing TCP/IP Traffic.

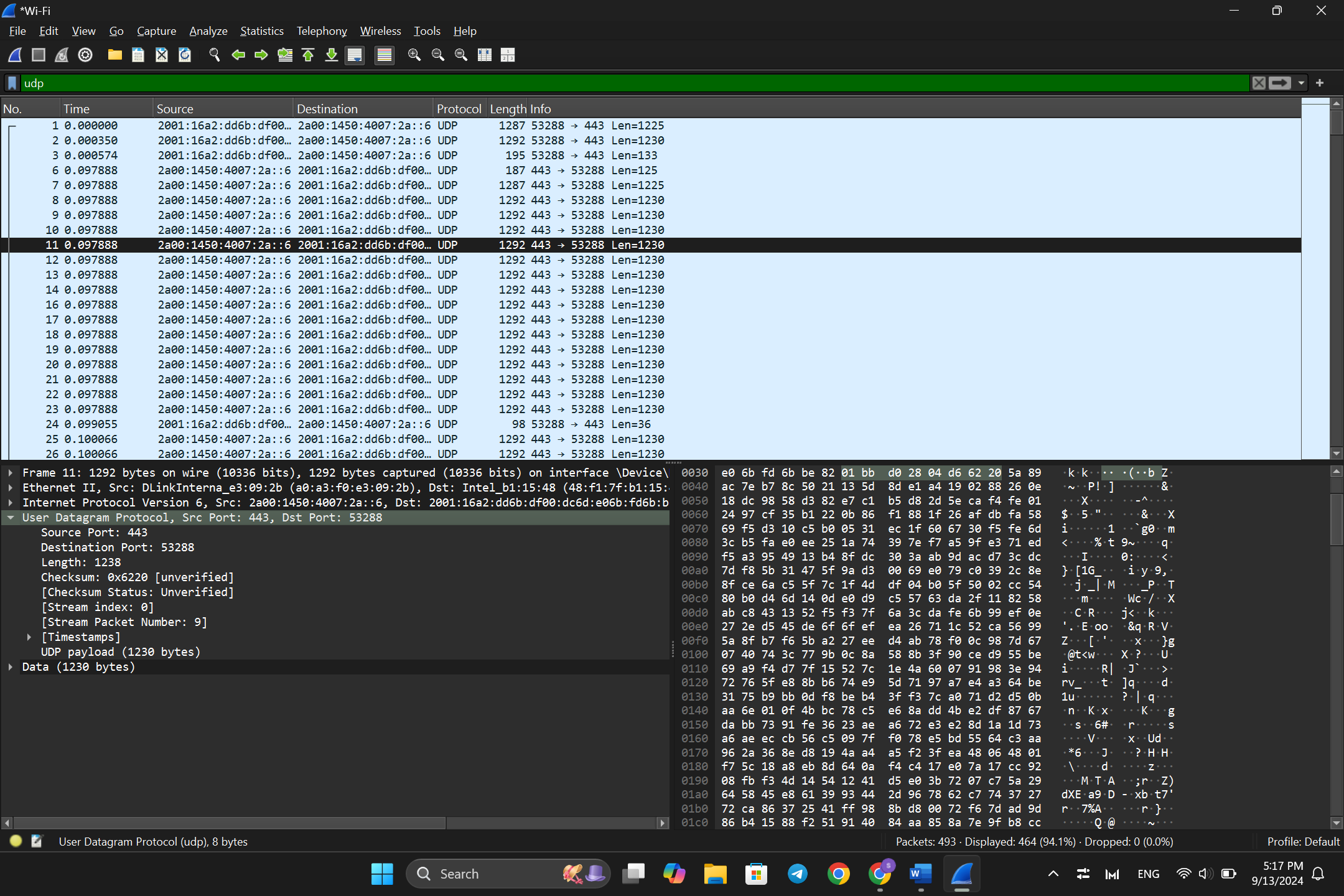
* Task 1: Filter TCP packets
* Task 2: Analyze TCP handshake and investigate Data Transfer and Termination



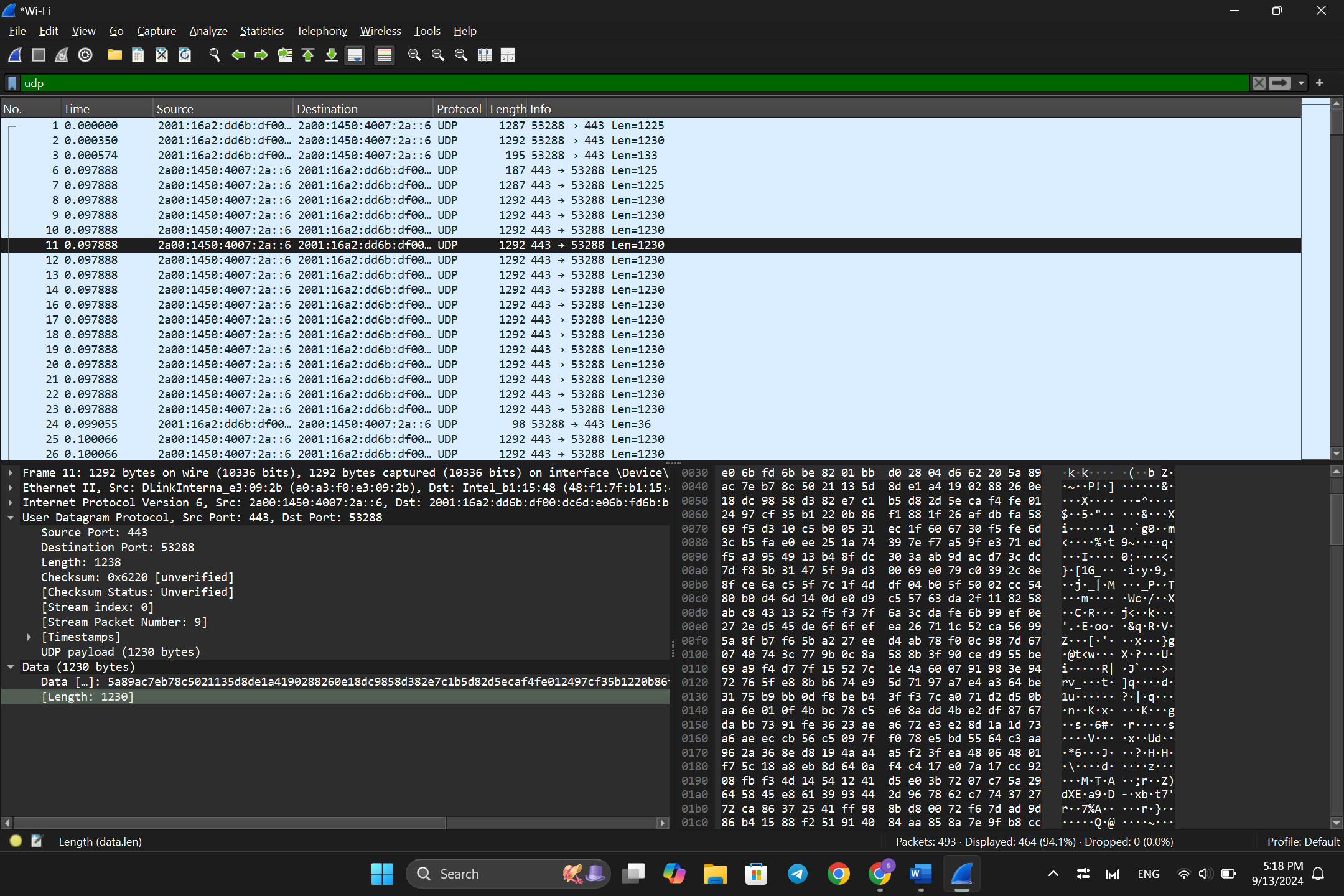


Part 3: Capturing and Analyzing UDP Traffic

* Task 1: Generate UDP traffic and capture packets
* Task 2: Filter and analysis UDP Packets



Observe the source and destination ports, length, and data.

* THE SOURCE PORT: 443
* THE DESTINATION PORT: 53288
* THE LENGTH:
* THE DATA:

Part 4: Comparing TCP and UDP by filling in the following tables:

|  |  |  |
| --- | --- | --- |
|  | TCP OR UDP | REASONS |
| Reliability and Connection Establishment | TCP | * IT’S A CONNECTION -ORINTED PROTOCOL * DATA TRANSMISSION IS GUARANTED WITH ERROR CHECKING, ACKNOWLEDGMENT AND RETRANSMISSION IF NEEDED. |
| Data Integrity and Ordering | TCP | * TCP ENSURE DATA IS DELEVIRED IN ORDER AND GUARANTEES DATA INTEGRITY BY USING SEQUENCE NUMBERS AND ACK PACKETES AND SO ON. |

|  |  |  |
| --- | --- | --- |
|  | TCP | UDP |
| Use cases | * FILE TRANSFERS * EMAILS (SMTP) * WEB BROWSING (HTTP/HTTPS) | * VIDEO STREAMING * ONLINE GAMING * VOICE OVER IP (VoIP) |
| Performance | * SLOWER (Due to connection establishment, error checking, flow control, and acknowledgment features) | * FASTER (Minimal overhead with no connection establishment, error checking, or retransmission, which makes it ideal for speed-critical applications) |