**2. Compare Laptop Configurations Available Online**

| **Feature** | **Laptop 1**  **(Budget)**  **(50K)** **MSI Thin 15 B12UCX-1695IN Laptop** | **Laptop 2 (MidRange)**  **(100K)**  **ROG X13 FLOW GV301** | **Laptop 3 (HighPerformance)**  **(150K)** **MSI Crosshair 16 HX D14VFKG-206IN Laptop** |
| --- | --- | --- | --- |
| **CPU Model** | **Intel i5-12450H(12thGen)** | **Ryzen 6900HS** | **Intel i7-14700HX(14thGen)** |
| **CPU Cores/Threads** | 8 Cores 12 Threads | 8 Cores 16 Threads | Performance Cores: 8 Cores, 16 Threads, Efficient Cores: 12 Cores, 12 Threads, 1.5 GHz Base, 3.9 GHz Turbo |
| **Base/Turbo Clock Speed** | **2.0 GHz Base 4.4 GHz Turbo** | **3.3 GHz Base 4.8 GHz Turbo** | Performance Cores:2.1 GHz Base, 5.5 GHz Turbo,  Efficient Cores: 1.5 GHz Base, 3.9 GHz Turbo |
| **L3 Cache Size** | **12MB** | **16 MB** | **33MB** |
| **RAM Size** | **16GB** | **16GB** | **16GB** |
| **RAM Type & Speed** | **DDR4(3200 MHz)** | **DDR5(6400Mhz)** | **DDR5(6400MHz)** |
| **Storage Type**  **(HDD/SSD/NVMe)** | **SSD** | **NVMe** | **NVMe** |
| **Storage Capacity** | **512GB** | **512GB** | **1TB** |
| **Storage Interface (SATA/PCIe)** | **M.2(PCIe)** | **NVMe(M.2)** | **M.2(PCIe)** |
| **Price (Approximate)** | **52,000** | **89.900** | **134,400** |
| **GPU** | **RTX2050(4GB)** | **RTX 3050(4GB)** | **RTX 4060(8GB)** |

**Answer the Following Questions**

1. Which laptop has the best CPU for multitasking and data science workloads? Why?  
Best CPU: Laptop 3 – MSI Crosshair 16 HX (i7-14700HX)  
**CPU Architecture:** 20 cores (8 Performance + 12 Efficiency) and 28 threads, which is significantly better for parallel workloads like training ML models or running simultaneous processes (e.g., Jupyter + ETL + DB + browser).

**Base/Turbo Clocks:** Up to **5.5 GHz** turbo on performance cores.

**L3 Cache:** 33 MB — larger cache is great for data-heavy operations

2. Which laptop has the best memory configuration? Can it be upgraded?

**Memory: Tie between Laptop 2 and Laptop 3**

* **Both have:**
  + **16GB DDR5 @ 6400 MHz**, which is significantly faster than **DDR4 @ 3200 MHz** (Laptop 1).
  + DDR5 improves bandwidth and latency — crucial for large dataset operations and GPU memory paging.
* Laptop 3 has upgradable memory upto 32GB

3. Which storage type is the fastest among the three?

**Fastest Storage: Laptop 2 and Laptop 3** – Both use **NVMe (PCIe)** SSDs.

**Explanation:**

* NVMe (Non-Volatile Memory Express) over PCIe offers **much higher throughput** and **lower latency** than standard SATA SSDs or M.2 SATA.
* All three laptops use M.2 form factor, but **Laptop 2 and 3 explicitly use NVMe**, making them ideal for fast data access, large file loads (e.g., image/video datasets), and fast boot.

4. If you were to buy a laptop for data science, which one would you choose and why?

**Recommended: Laptop 3 – MSI Crosshair 16 HX D14VFKG-206IN**

**Why:**

* **CPU:** The i7-14700HX offers exceptional multicore performance with modern hybrid architecture (P+E cores).
* **High-speed DDR5 RAM** and upgrade potential.
* **1TB NVMe SSD** provides ample and fast storage.
* **Powerful GPU (RTX 4060 8GB)**: Useful for training models (CUDA), deep learning, and GPU-accelerated libraries.

**3. (Group Assignment) Compare Features:**

**Pre-requisite:**

• The participants have completed the Assignment #2 individually.

**Task:**

• Sit together and compare the 3 Laptops you just selected.

• Argument & justify why your chosen Laptop is better.

BUDGET:  
  
CPU:

The MSI Thin 15 uses the Intel Core i5-12450H, which is a more powerful processor designed for gaming and high-performance tasks. It has a higher TDP (45W) and better multi-core performance, which is beneficial for data science tasks that require heavy computation.

The Lenovo V14 uses the Intel Core i7-1255U, which is a low-power, efficient processor designed for thin and light laptops. While it is more power-efficient, it may not match the raw performance of the i5-12450H in demanding tasks.

STORAGE:

Both laptops have 512 GB SSDs, which provide fast read and write speeds, essential for data science workflows. The MSI Thin 15 has a PCIe Gen 4 SSD, which offers faster data transfer rates compared to the standard SSD in the Lenovo V14.

GRAPHICS:

The MSI Thin 15 has a dedicated NVIDIA GeForce RTX 2050 GPU, which can significantly speed up tasks that benefit from GPU acceleration, such as machine learning, deep learning, and data visualization.

The Lenovo V14 has Intel Iris Xe integrated graphics, which are suitable for basic tasks but may not provide the same level of performance for GPU-intensive tasks.

MIDRANGE:

CPU:

The i7-13620H offers more physical and logical cores than the Ryzen 9 6900HS, which is beneficial for parallel data processing and machine learning workloads.

GPU:

The RTX 4060 (8GB VRAM) in the TUF F15 is significantly more powerful than the RTX 3050Ti (4GB VRAM) in the Flow X13. For deep learning, GPU memory and compute power are critical, and the RTX 4060 nearly doubles the performance and VRAM.

HIGH-END:

CPU:

The HP Omen's i9-13900HX has more total cores (24 vs 20), which can be advantageous for Data science workloads.

UPGRADABILITY:

MSI supports up to 96GB RAM, which is a major advantage for handling large datasets or running memory-intensive workloads.

QUALITY OF LIFE:

MSI offers a higher resolution (2560x1600 vs 1920x1080), higher refresh rate (240Hz vs 165Hz), and a better 16:10 aspect ratio.

• Decide, Agree and Come-up with one Laptop for each Category

• Present the arguments/justifications for your agreement on each Category

BUDGET:Srijan  
MIDDLE:Chattresh

HIGH END:Srijan

FINAL TABLE

| **Feature** | **Laptop 1**  **(Budget)**  **(50K)**  **(Srijan)** **MSI Thin 15 B12UCX-1695IN Laptop** | **Laptop 2 (MidRange)**  **(100K)**  **(Chhatresh)**  **ASUS TUF Gaming F15** | **Laptop 3 (HighPerformance)**  **(150K)**  **(Srijan)** **MSI Crosshair 16 HX D14VFKG-206IN Laptop** |
| --- | --- | --- | --- |
| **CPU Model** | **Intel i5-12450H(12thGen)** | **Intel Core i7 13th Gen 13620H** | **Intel i7-14700HX(14thGen)** |
| **CPU Cores/Threads** | 8 Cores 12 Threads | 10 Cores 16 Threads | Performance Cores: 8 Cores, 16 Threads, Efficient Cores: 12 Cores, 12 Threads, 1.5 GHz Base, 3.9 GHz Turbo |
| **Base/Turbo Clock Speed** | 2.0 GHz Base 4.4 GHz Turbo | Performance Cores: 2.4 GHz Base, 4.9 GHz Turbo.  Efficient Cores: 1.8 GHz Base, 3.6 GHz Turbo | Performance Cores:2.1 GHz Base, 5.5 GHz Turbo,  Efficient Cores: 1.5 GHz Base, 3.9 GHz Turbo |
| **L3 Cache Size** | **12MB** | **24 MB** | **33MB** |
| **RAM Size** | **16GB** | **16GB** | **16GB** |
| **RAM Type & Speed** | **DDR4(3200 MHz)** | **DDR5(4800Mhz)** | **DDR5(6400MHz)** |
| **Storage Type**  **(HDD/SSD/NVMe)** | **SSD** | **NVMe** | **NVMe** |
| **Storage Capacity** | **512GB** | **512GB** | **1TB** |
| **Storage Interface (SATA/PCIe)** | **M.2(PCIe)** | **NVMe(M.2)** | **M.2(PCIe)** |
| **Price (Approximate)** | **52,000** | **99.900** | **134,400** |
| **GPU** | **RTX2050(4GB)** | **RTX 4060(8GB)** | **RTX 4060(8GB)** |

**4. Real-World Network Analysis**

**Steps:**

**1. Install a Packet Sniffing Tool**

a. Use Wireshark (Windows/Linux/macOS) or tcpdump (Linux/macOS) to capture

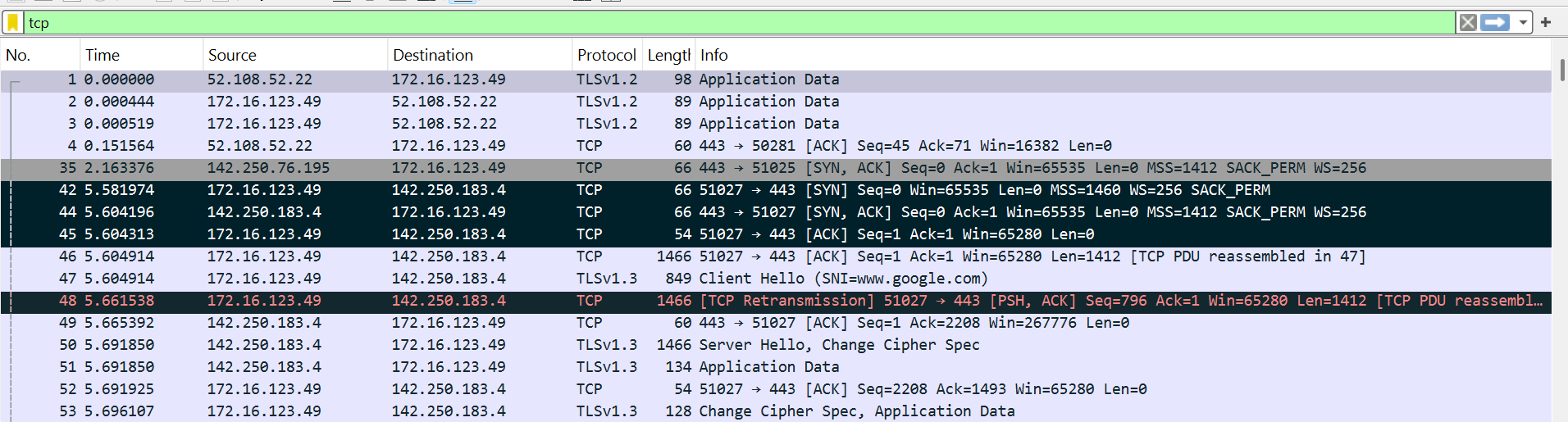
network traffic.

**2. Capture TCP/IP Traffic**

a. Open Wireshark and start capturing packets.

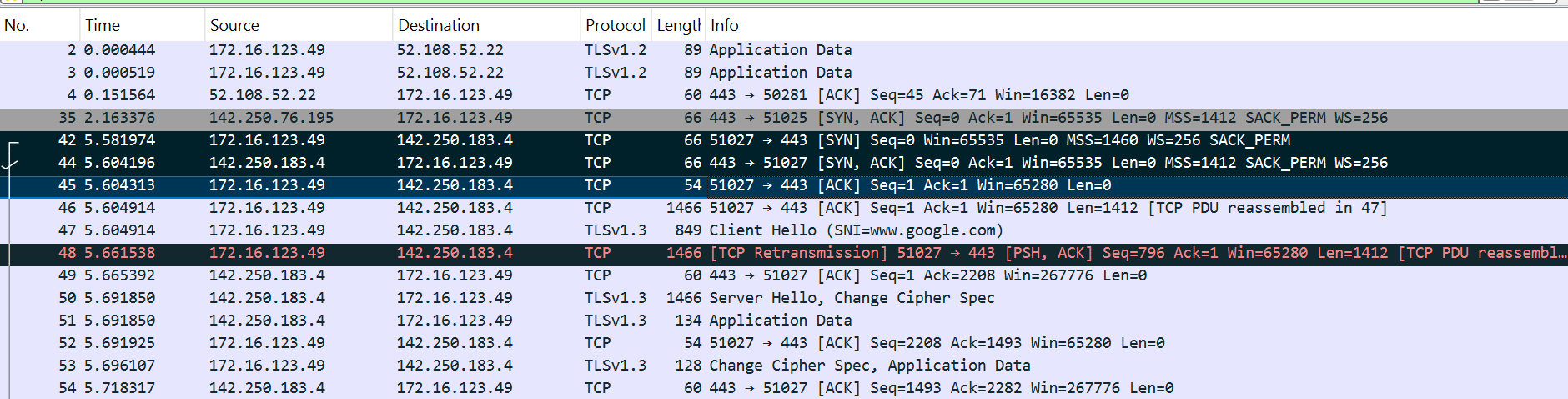
b. Visit any website (e.g., http://example.com).

c. Stop capturing packets and filter only TCP packets.

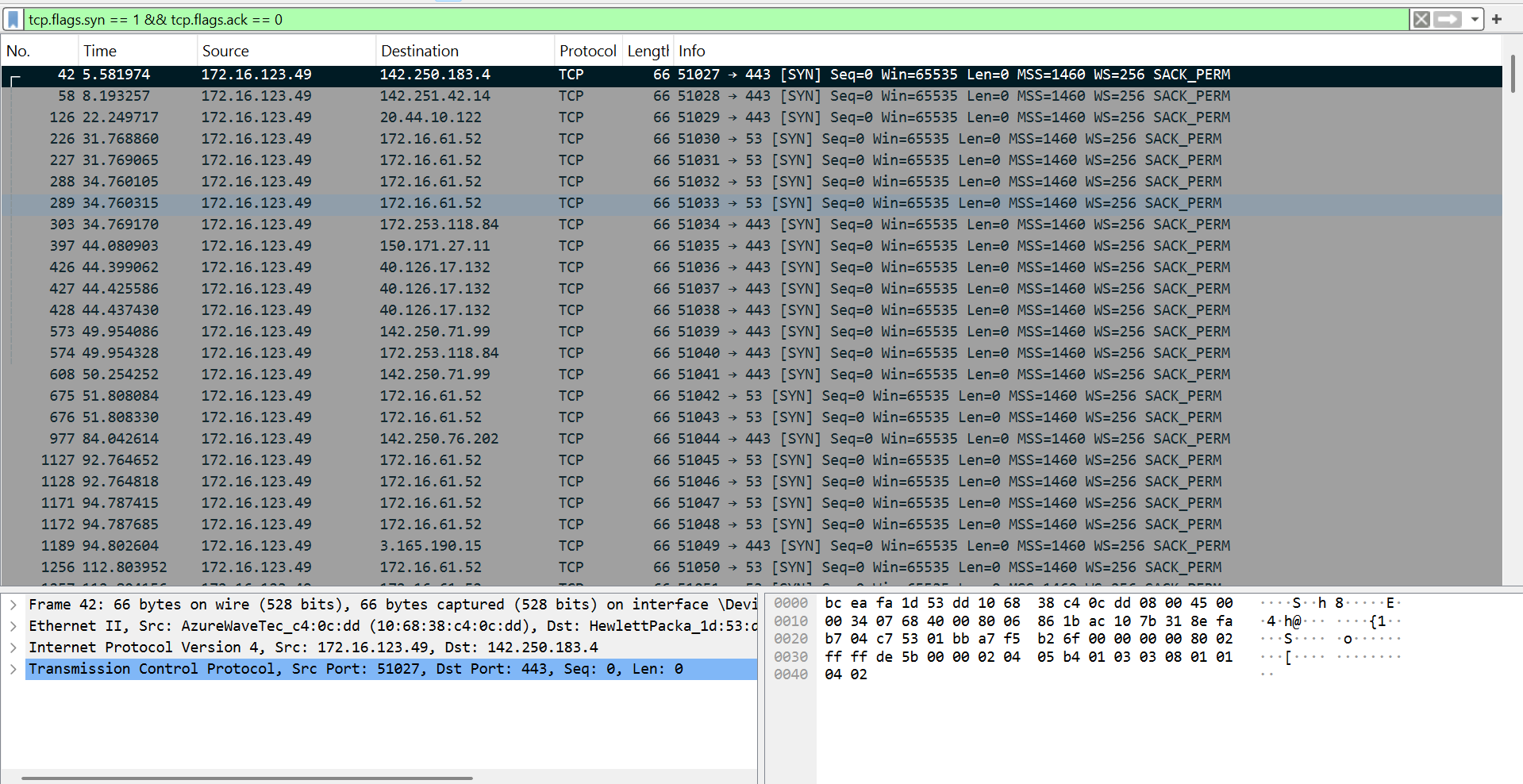


**3. Analyze the TCP Connection**

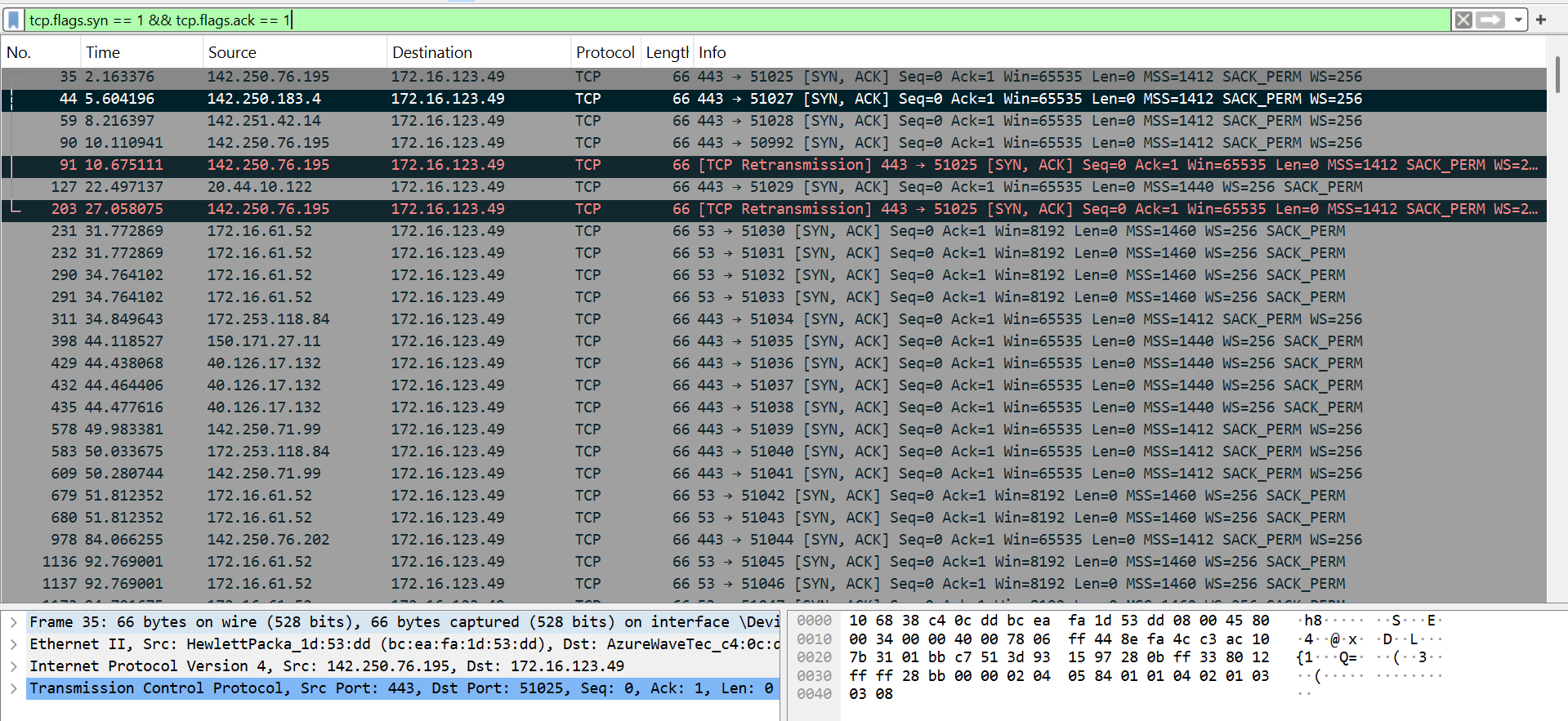
a. Identify the three-way handshake (SYN, SYN-ACK, ACK).



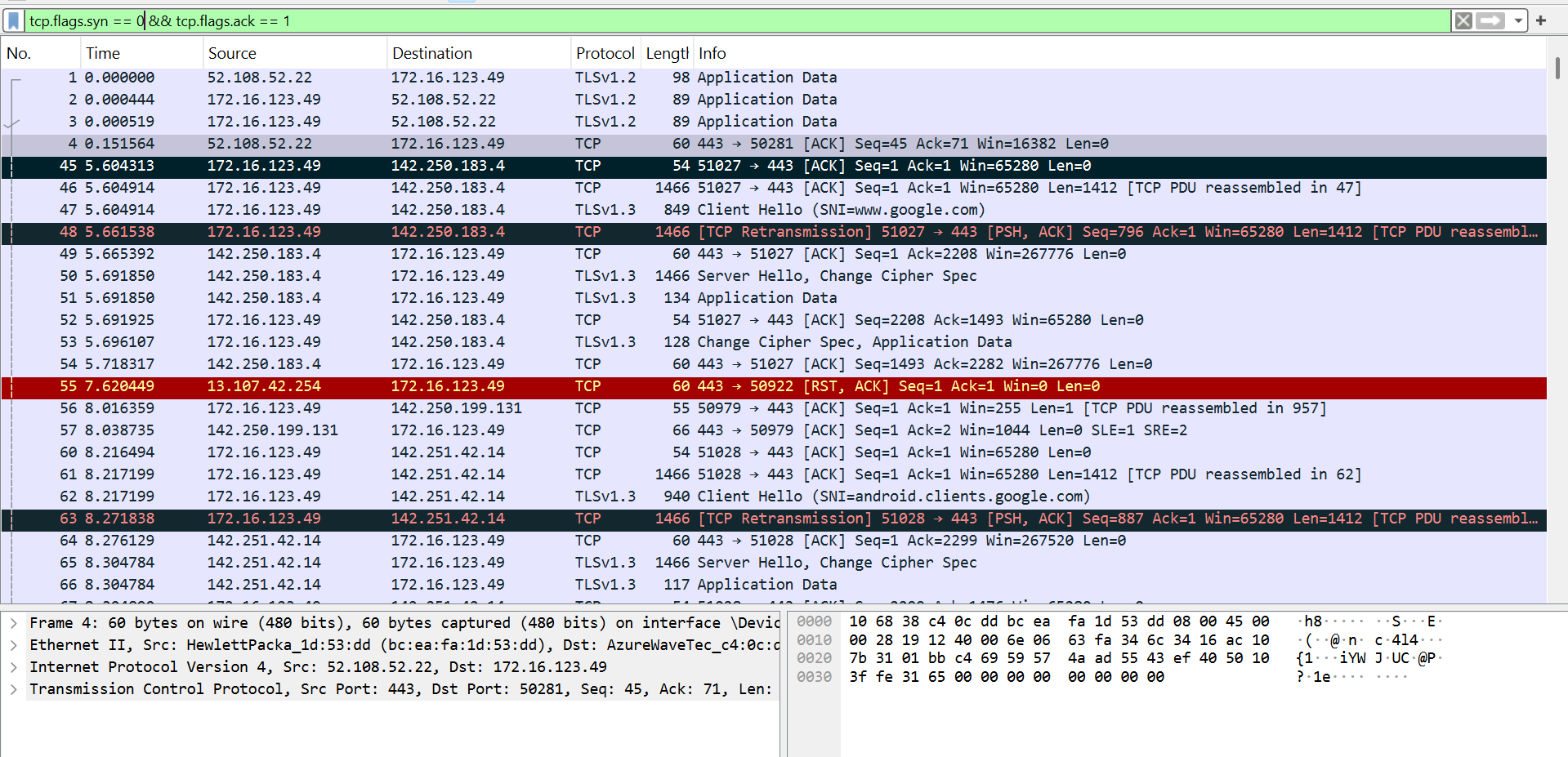
Filter for SYN packets only



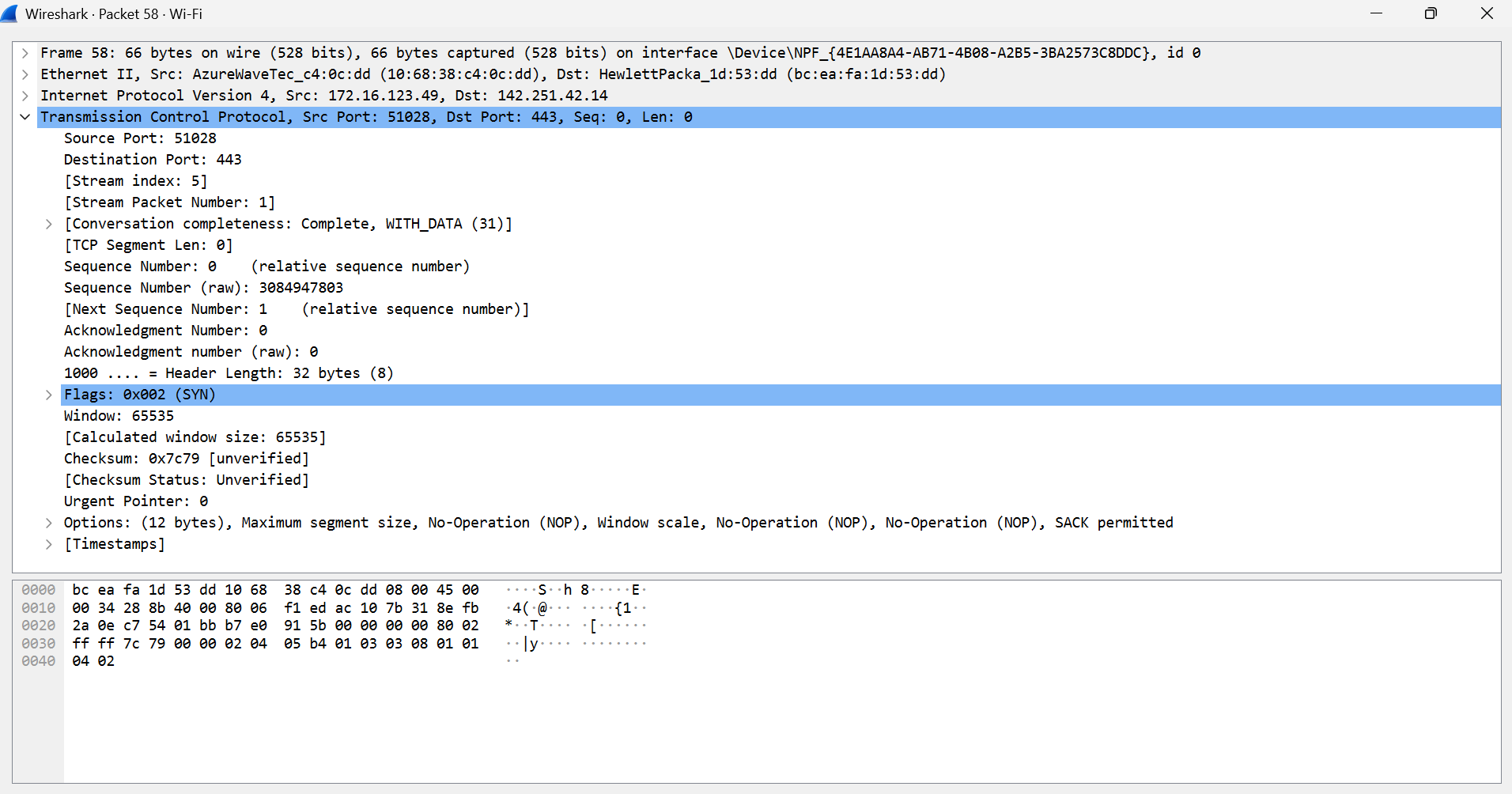
FILTER for SYN,ACK



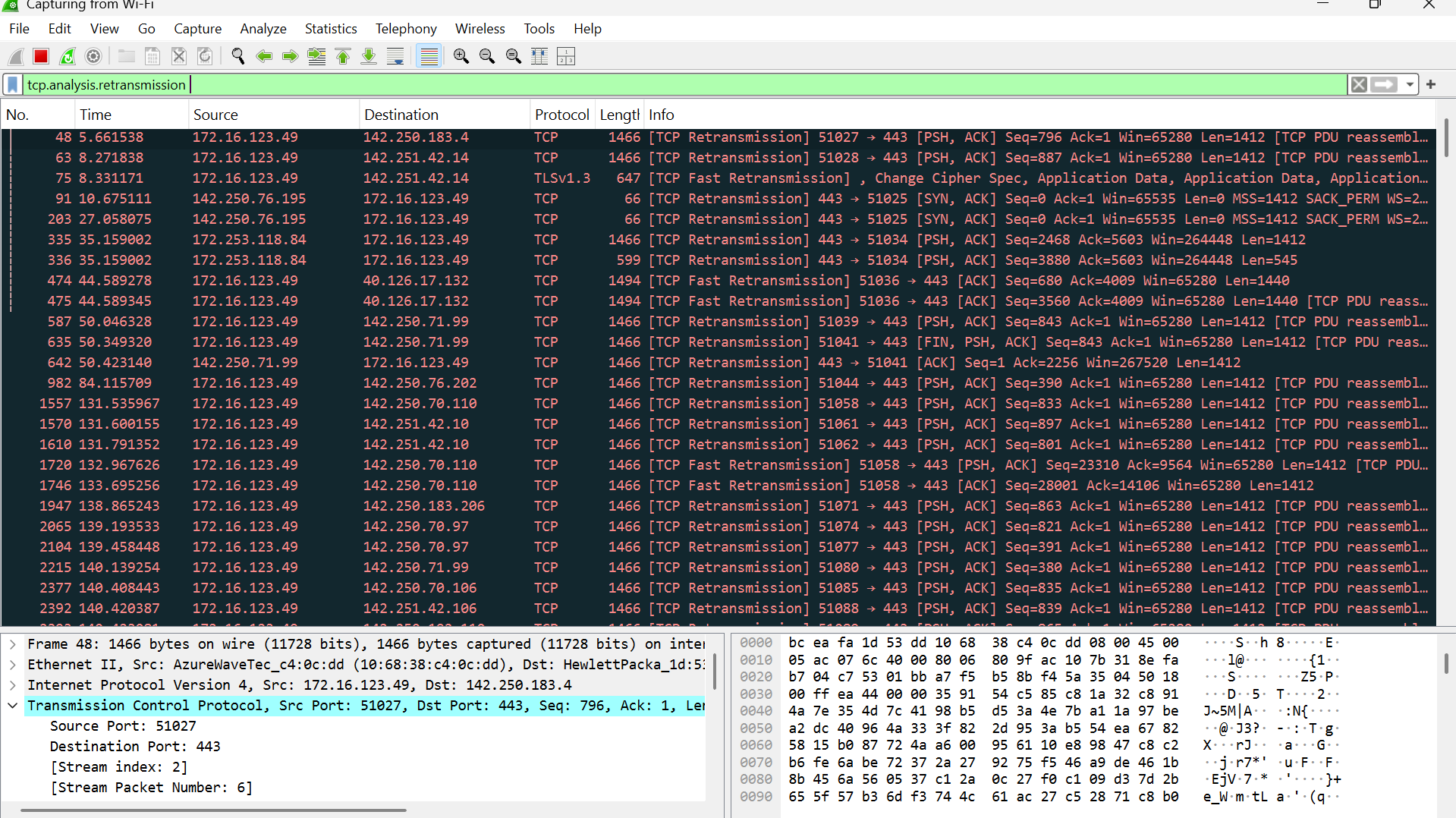
FILTER for ACK

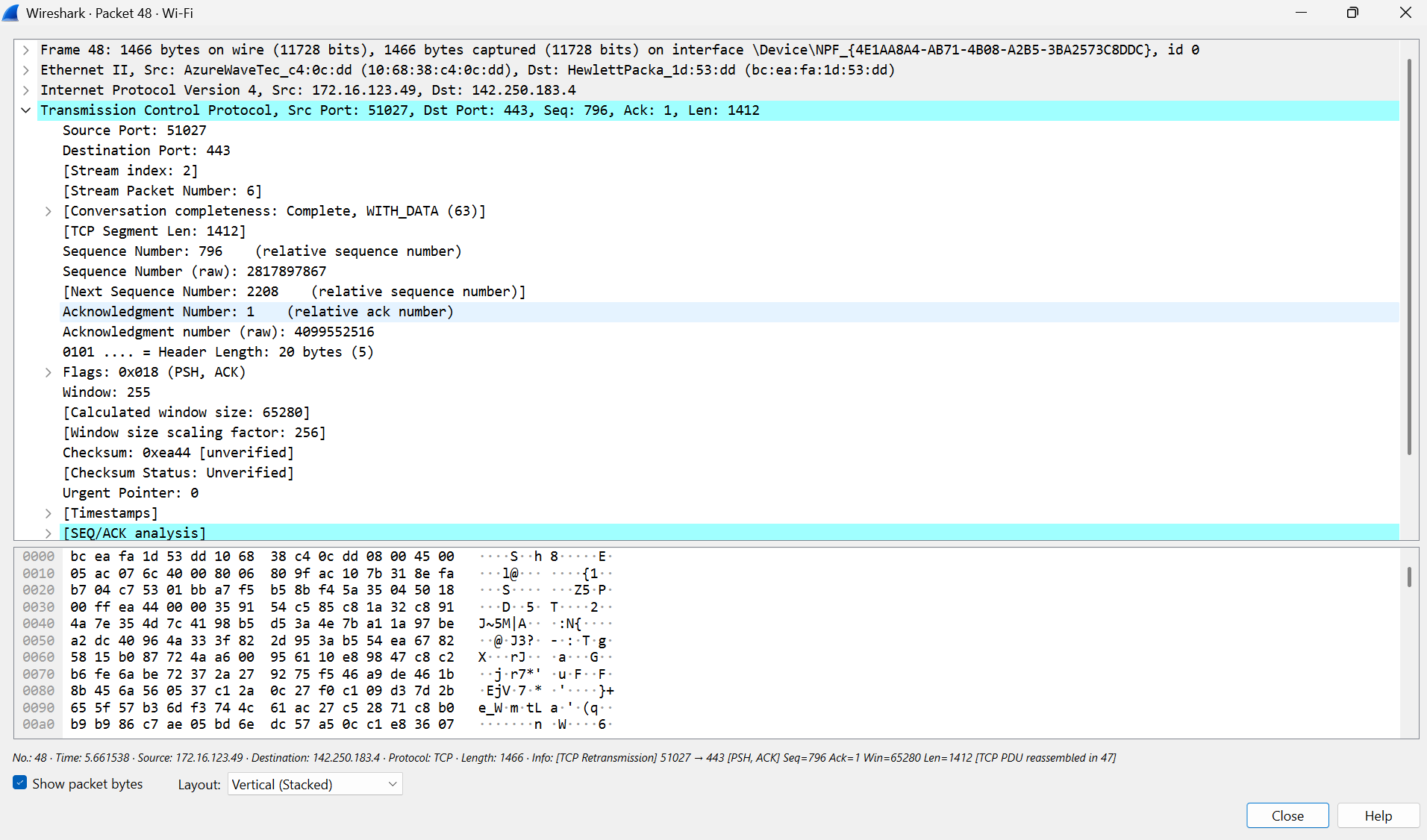


b. Observe the packet sequence numbers.



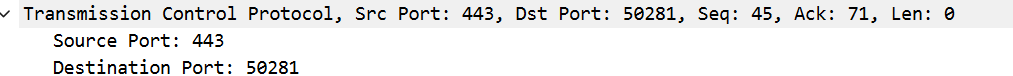
c. Find a TCP retransmission or dropped packet.



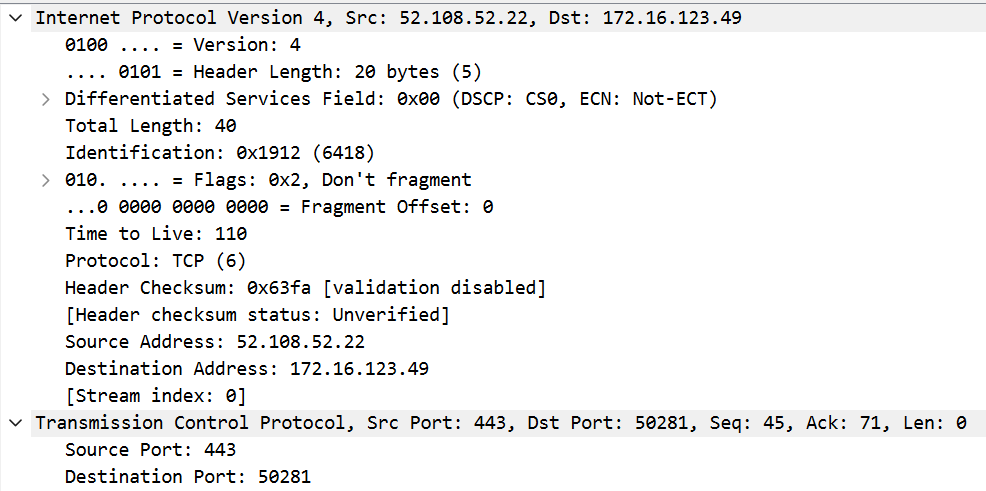


**4. Answer the following questions:**

a. What are the source and destination IP addresses of the captured packets?



b. What is the port number used for the connection?



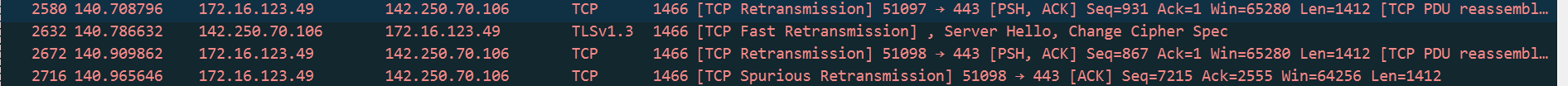
c. What happens when a packet is lost? How does TCP handle it?

TCP detects loss (due to missing ACK or timeout).

It **retransmits** the lost packet automatically.

Mechanisms used:

* **Timeouts**
* **Duplicate ACKs**
* **Fast Retransmit**



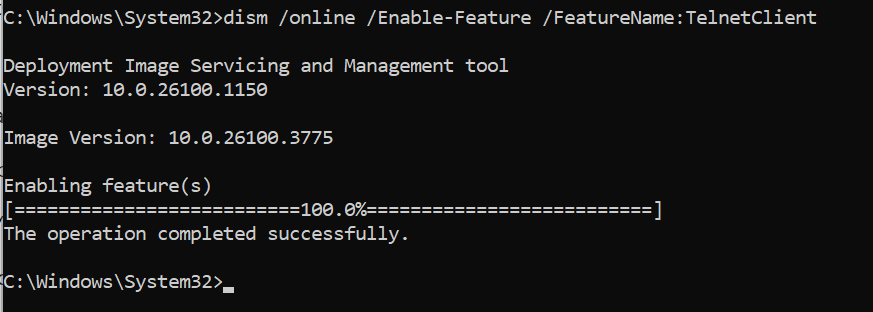
**5. HTTP Request using TELNET**

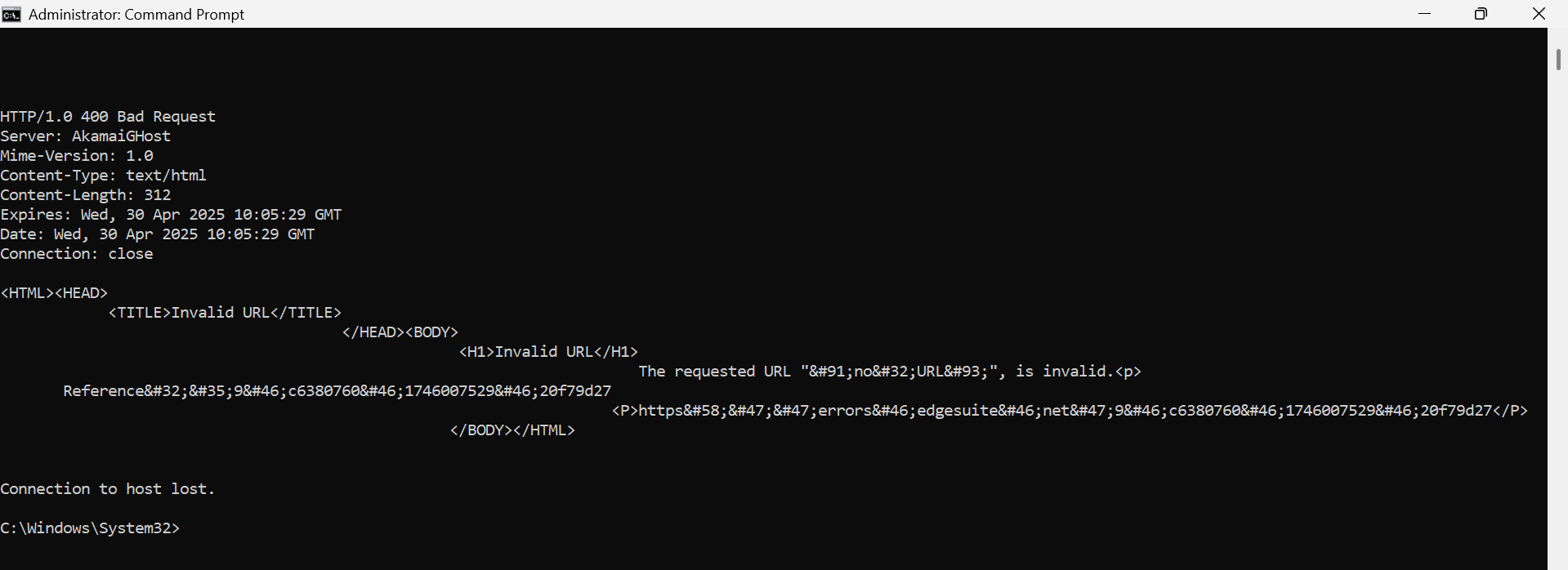
Objective:

Understand how HTTP works at a low level by manually sending an HTTP request.

Steps:

1. Open Command Line (Windows/Linux/macOS)



a. Run the command: telnet example.com 80  


b. If telnet is not installed, install it first.

2. Send an HTTP Request Manually

a. Type the following HTTP request and press Enter twice:

i. GET / HTTP/1.1

ii. Host: example.com  

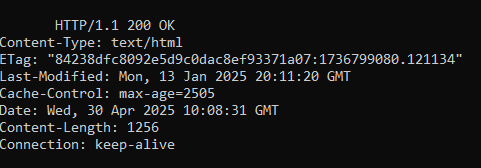

b. Observe the HTTP response headers and content.

3. Answer the following questions:

a. What is the HTTP status code returned? What does it mean?  
HTTP/1.1 200 OK

* The server successfully processed the HTTP request and returned the requested resource (in this case, the HTML page of example.com).

b. What are the HTTP headers received?



Content-Type: Specifies that the content is HTML.

ETag: A unique identifier for the version of the resource.

Last-Modified: When the content was last changed.

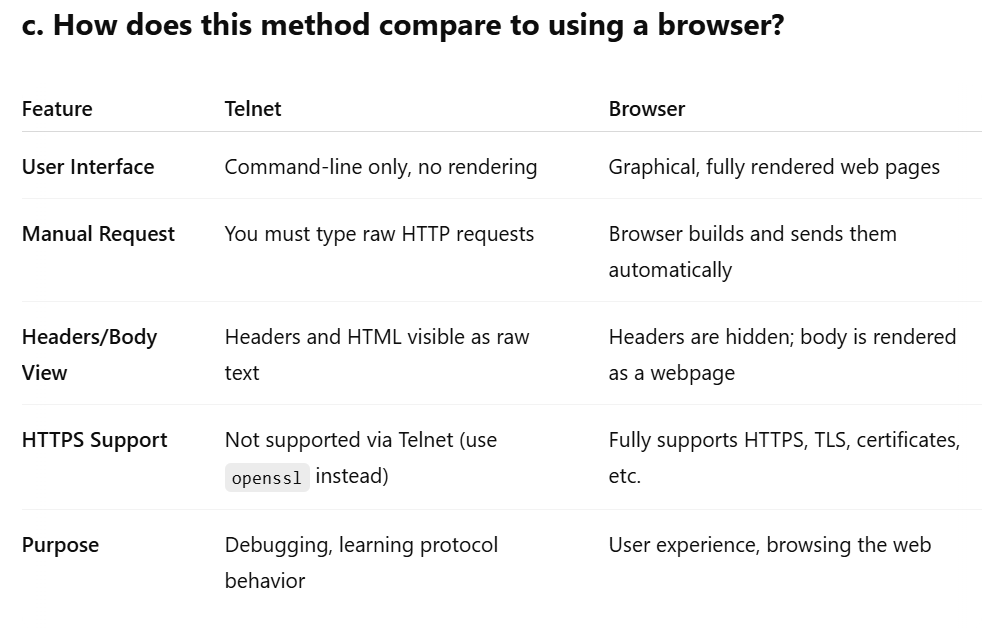
Cache-Control: How long (in seconds) clients can cache the response.

Date: The date and time the response was sent.

Content-Length: The size of the response body in bytes.

Connection: keep-alive: Keeps the connection open for further requests.

c. How does this method compare to using a browser?



**6. Compare HTTP vs. HTTPS Request**

**Objective:**

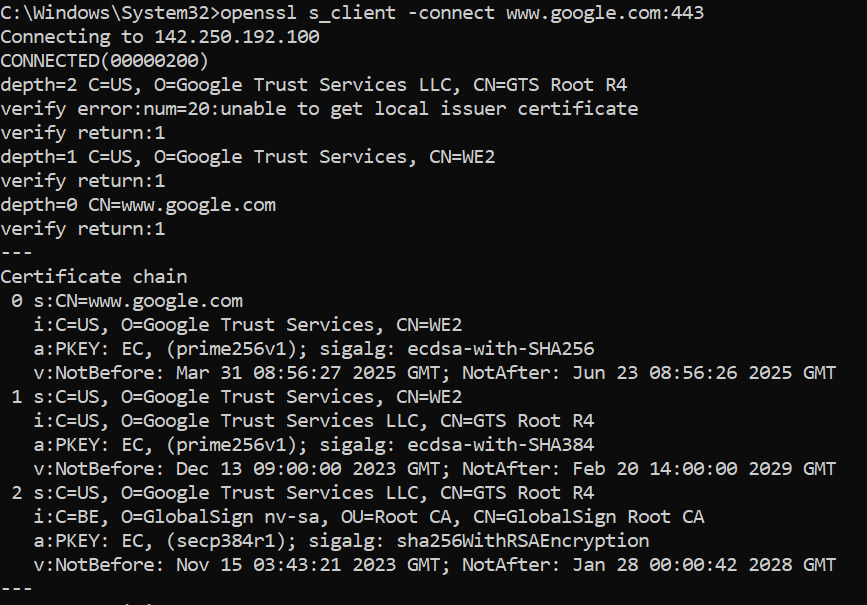
Understand how HTTPS encrypts communication compared to HTTP.

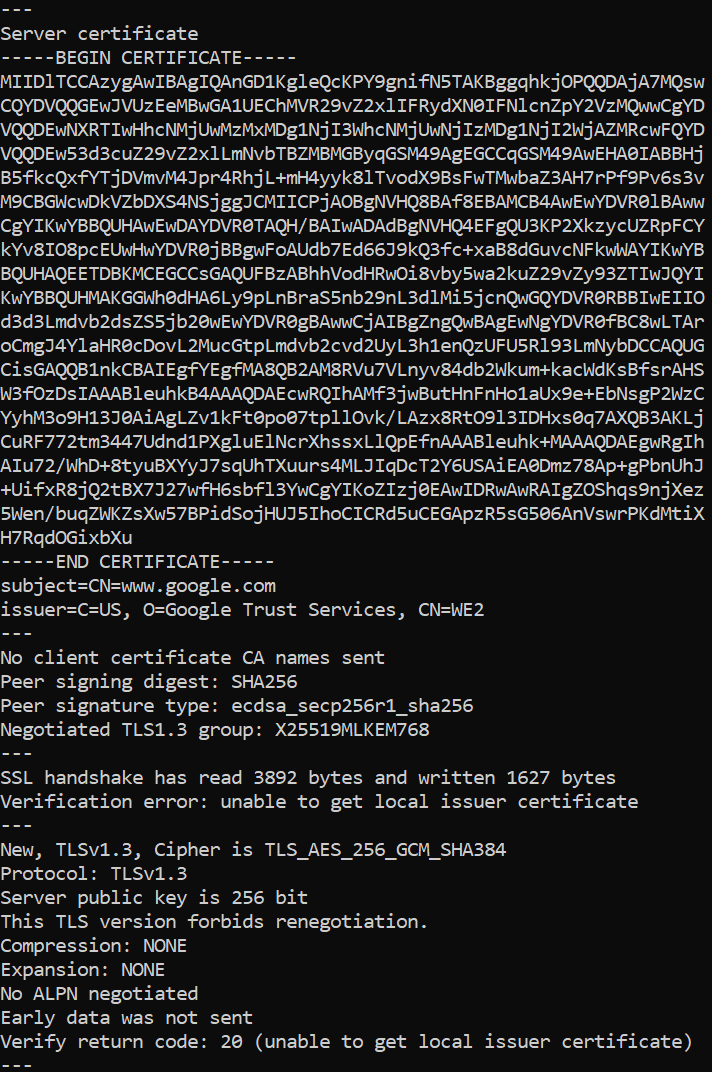
**Steps:**

1. Use OpenSSL to Connect to an HTTPS Server

a. Run the following command:

openssl s\_client -connect [www.google.com:443](http://www.google.com:443)





b. Observe the TLS handshake and certificate details.

2. Answer the following questions:

a. What TLS version is being used?

Protocol: TLSv1.3

b. What is the certificate authority (CA) issuing the SSL certificate?

Google Trust Services LLC

c. What encryption algorithm is used for securing communication?

TLS\_AES\_256\_GCM\_SHA384

d. How is HTTPS different from HTTP in terms of security?

| **Feature** | **HTTP** | **HTTPS (HTTP over TLS)** |
| --- | --- | --- |
| **Encryption** | No | Yes (TLS/SSL encryption) |
| **Data Privacy** | Plaintext (can be sniffed) | Encrypted end-to-end |
| **Authentication** | No identity verification | Verified via CA certificates |
| **Data Integrity** | Can be modified | Protected from tampering |
| **Port** | 80 | 443 |

**7. API Communication using HTTP Methods**

Objective:

Understand different HTTP methods (GET, POST, PUT, DELETE) using a REST API.

Steps:

**1. Use an API Testing Tool (cURL)**

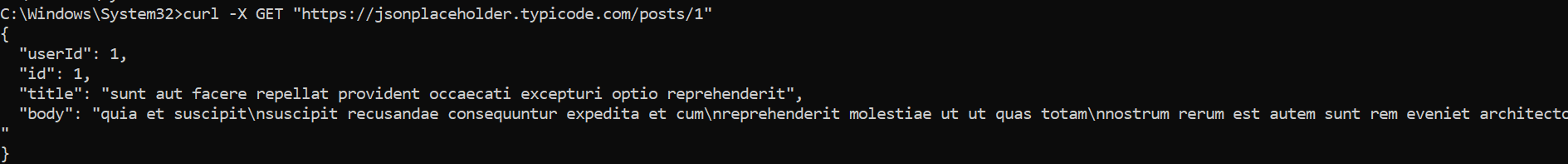
a. Try the following HTTP requests on a public API (e.g., JSONPlaceholder).

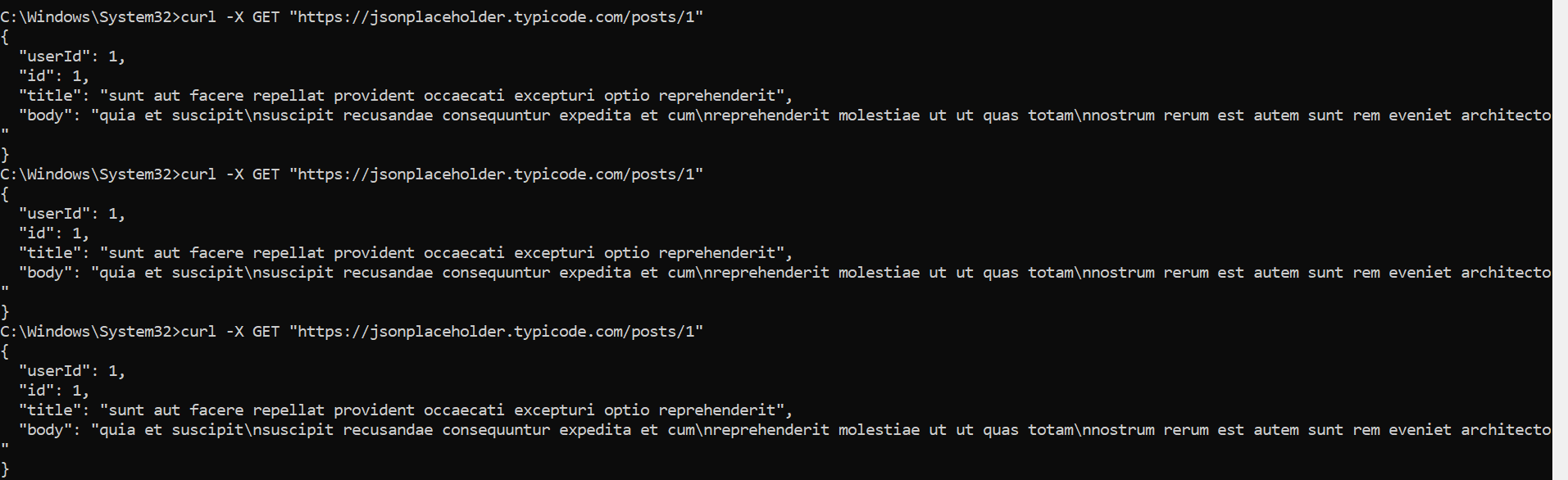
**2. Make API Calls:**

a. GET Request:

curl -X GET

"<https://jsonplaceholder.typicode.com/posts/1>"



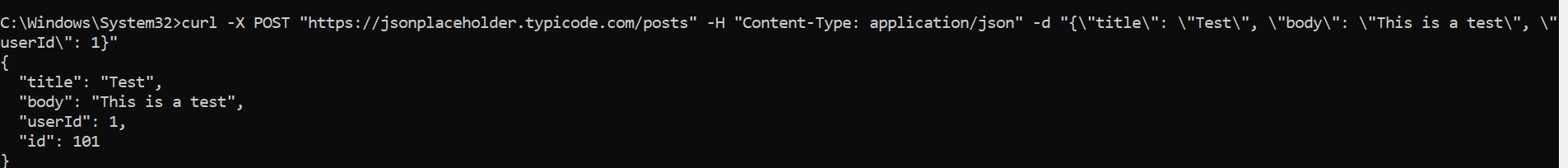


b. POST Request (Create New Data):

curl -X POST "https://jsonplaceholder.typicode.com/posts"

-H "Content-Type: application/json" -d '{"title": "Test",

"body": "This is a test", "userId": 1}'



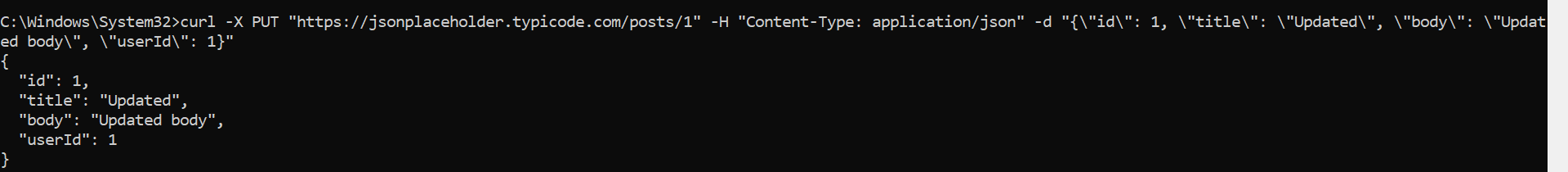
c. PUT Request (Update Data):

curl -X PUT

"https://jsonplaceholder.typicode.com/posts/1" -H

"Content-Type: application/json" -d '{"id":1, "title":

"Updated", "body": "Updated body", "userId": 1}'



d. DELETE Request:

curl -X DELETE

<https://jsonplaceholder.typicode.com/posts/1>



3. Answer the following questions:

a. What response do you get for each HTTP method?

| **Method** | **Response Example (JSON)** |
| --- | --- |
| **GET** | Fetched resource data |
| **POST** | Echoes created object with new id |
| **PUT** | Echoes updated object |
| **DELETE** | No content returned (empty body) |

b. What HTTP status codes are returned?

| **Method** | **Status Code** | **Meaning** |
| --- | --- | --- |
| **GET** | 200 OK | Successful fetch |
| **POST** | 201 Created | Resource created |
| **PUT** | 200 OK | Resource updated |
| **DELETE** | 200 OK | Resource deleted |

c. What is the difference between PUT and POST?

| **Feature** | **POST** | **PUT** |
| --- | --- | --- |
| Purpose | Create a new resource | Update an existing resource (or create if not exists) |
|  |  |  |
| URL Target | Collection endpoint (/posts) | Specific resource (/posts/1) |
| Response | New id created | Existing resource overwritten |