

# **LAB 3**

# **PHÂN TÍCH HOẠT ĐỘNG**

# **GIAO THÚC TCP – UDP**

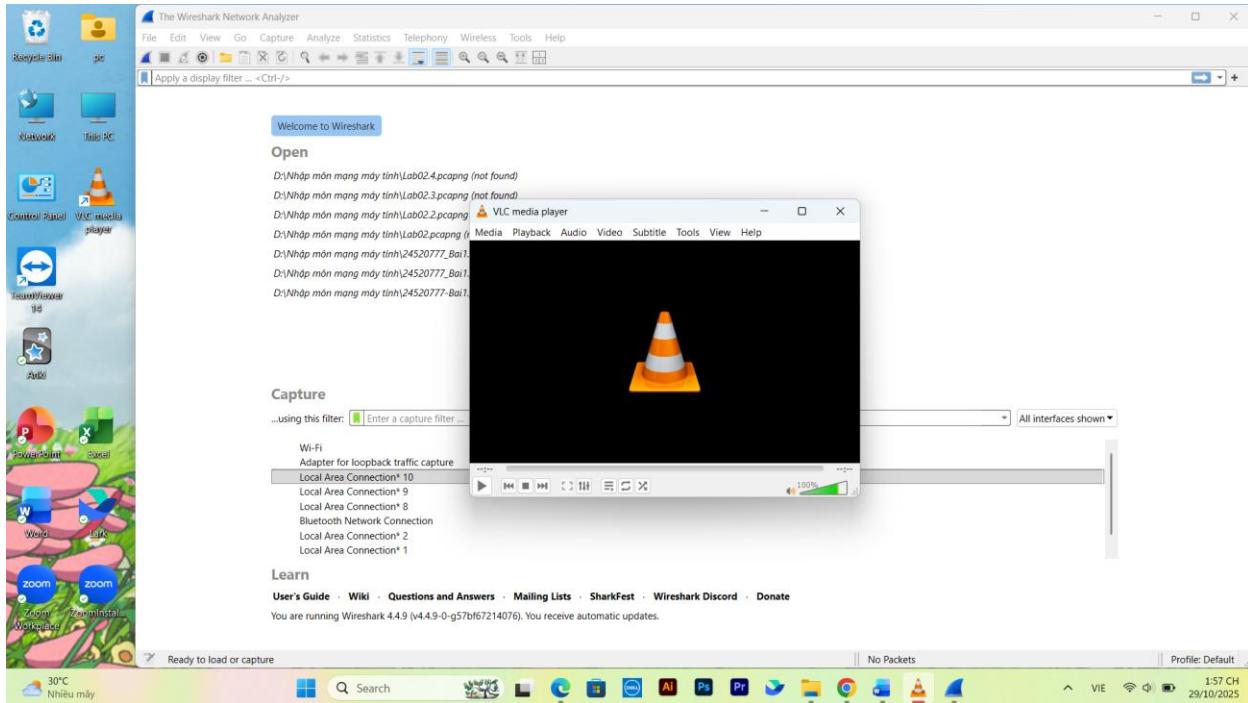
**Lớp thực hành: IT005.Q111.2**

**Giảng viên hướng dẫn: Nguyễn Thanh Nam**

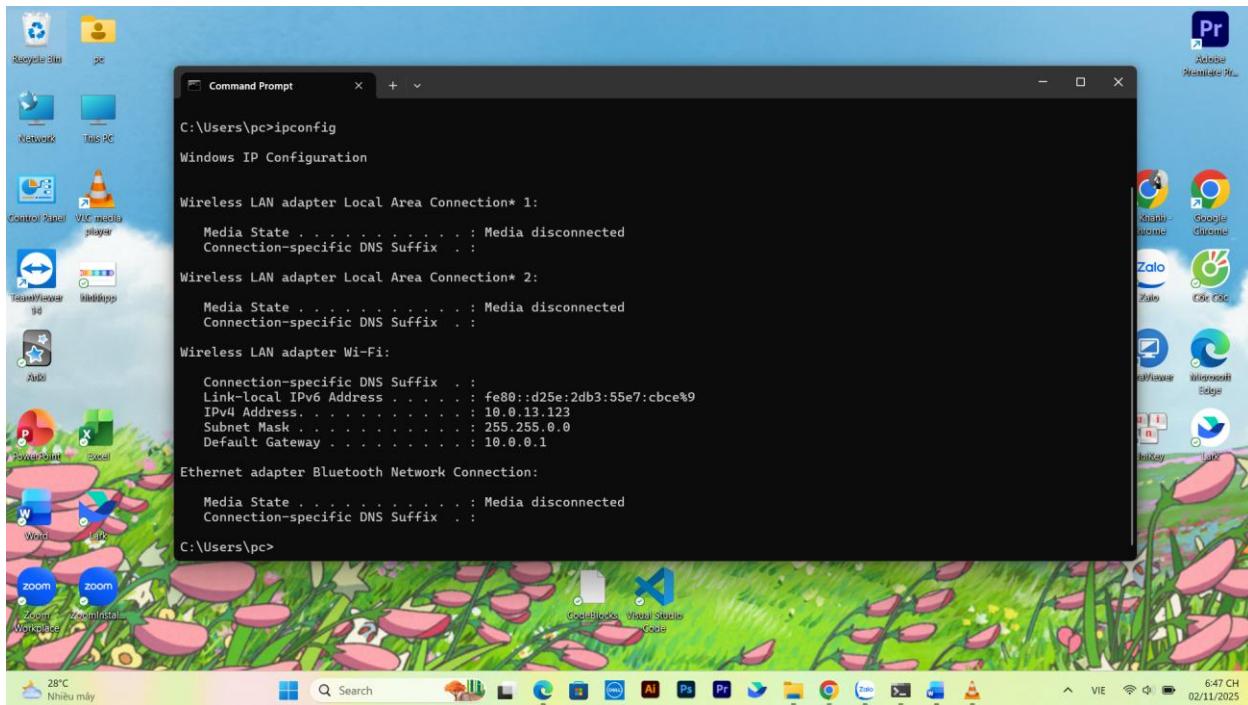
**Họ tên sinh viên: Đặng Văn Khánh**

**Mã số sinh viên: 24520777**

# I. CÀI ĐẶT



Cài đặt VLC media player



Máy bản thân

```

Command Prompt

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

Wireless LAN adapter Local Area Connection* 10:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . : fe80::abb2:1a86:f77c:10f4%6
IPv4 Address. . . . . : 10.0.13.125
Subnet Mask . . . . . : 255.255.0.0
Default Gateway . . . . . : 10.0.0.1

Ethernet adapter Bluetooth Network Connection:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

Tunnel adapter Teredo Tunneling Pseudo-Interface:

Connection-specific DNS Suffix . :
IPv6 Address. . . . . : 2001:0:2851:fcb0:4e1:eda:d589:3cc2
Link-local IPv6 Address . . . . : fe80::4e1:eda:d589:3cc2%13
Default Gateway . . . . . : ::

C:\Users\Nhan>

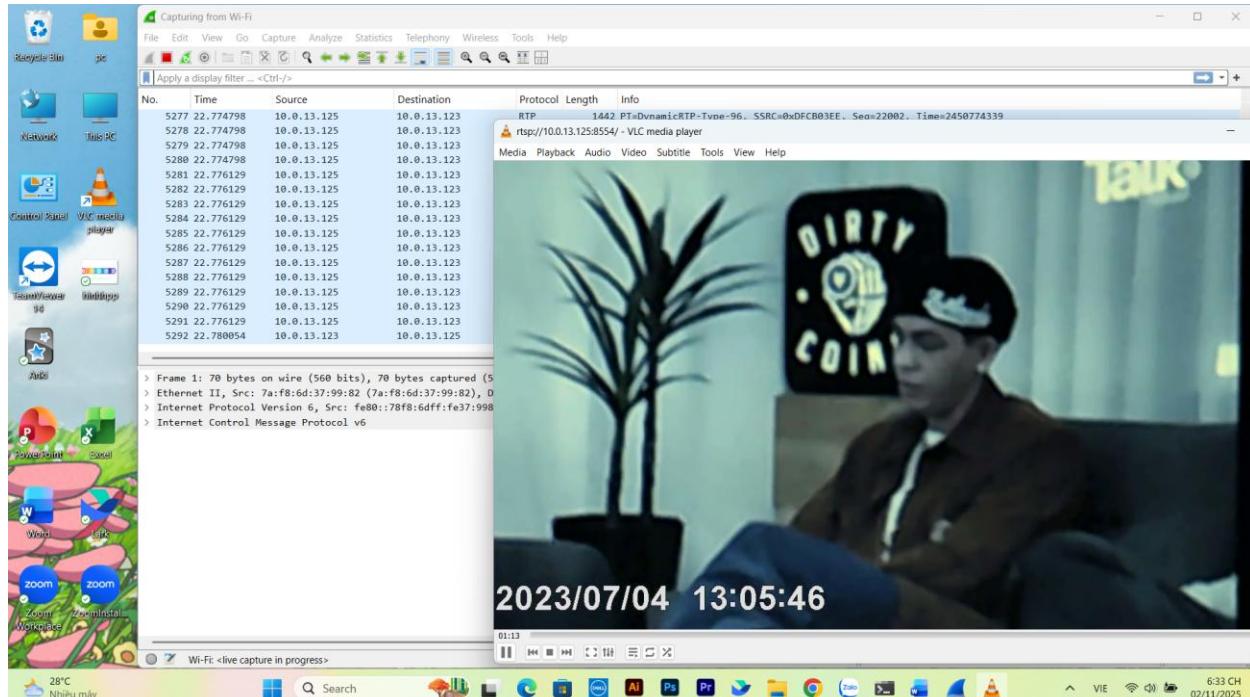
```

*Máy bạn*

## II. THỰC HÀNH

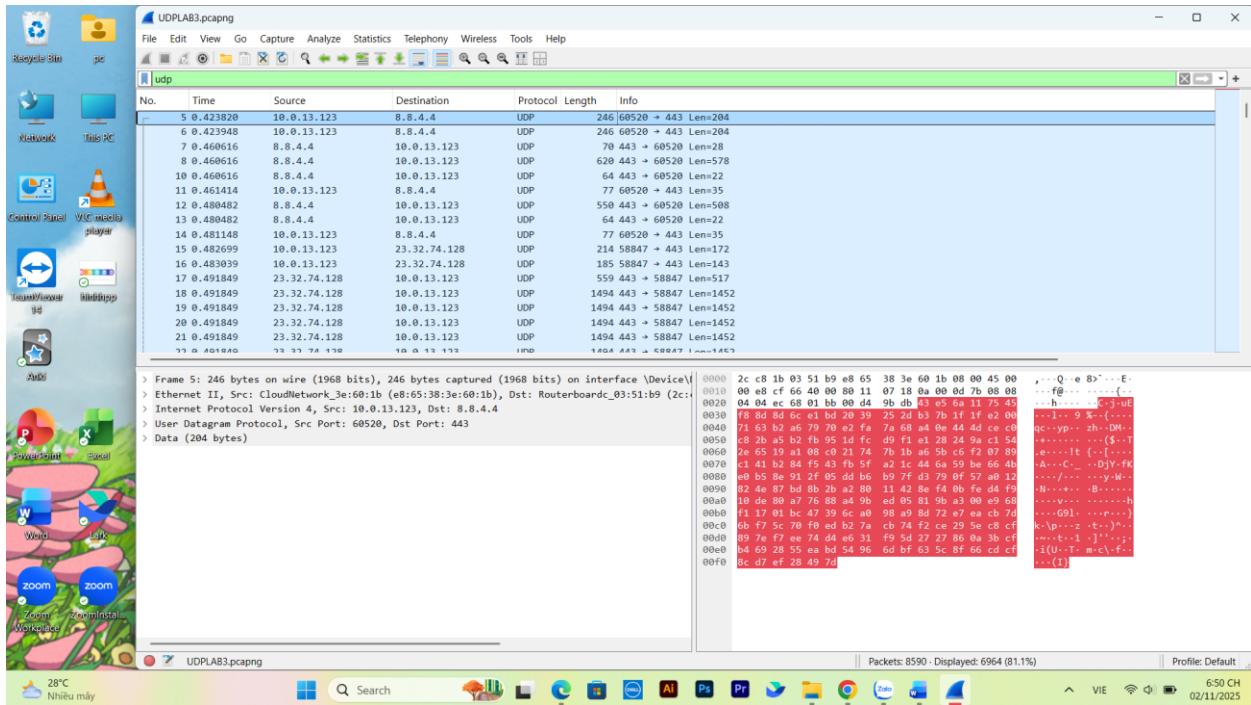
### TASK 1 - PHÂN TÍCH HOẠT ĐỘNG GIAO THỨC UDP

#### 1.1 Streaming video sử dụng UDP



*Hình ảnh stream video sử dụng UDP*

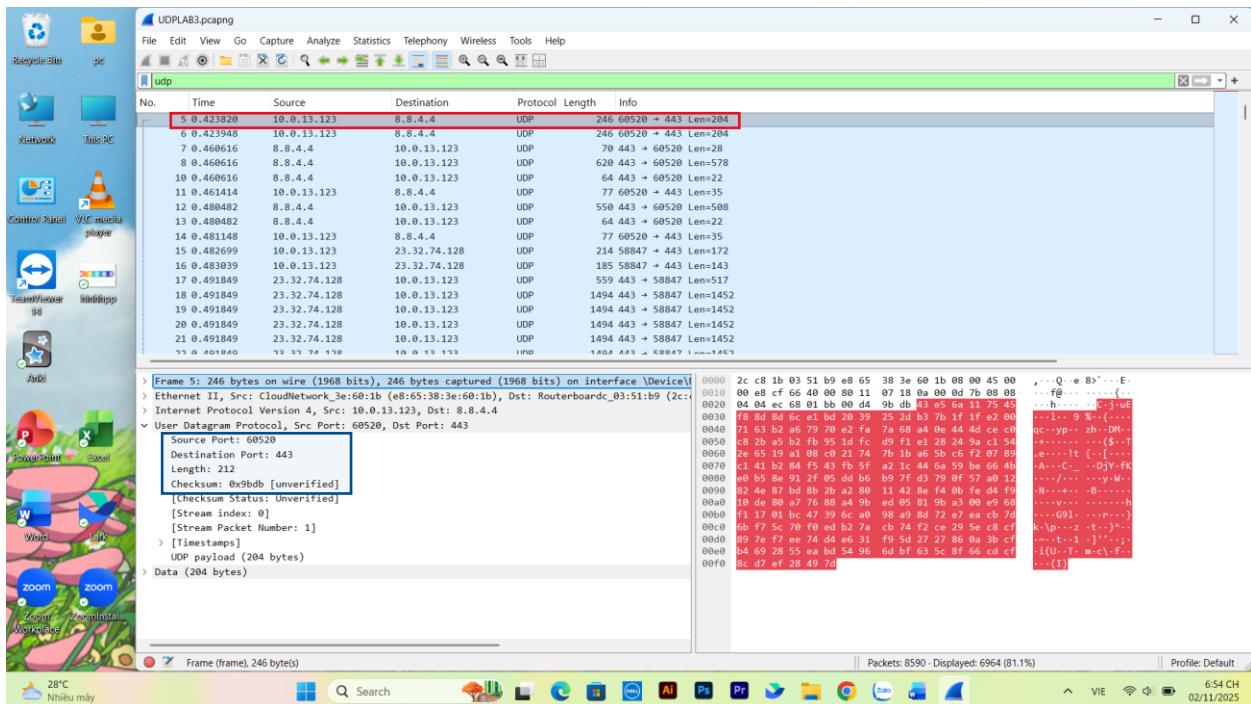
## 1.2 Tiến hành bắt gói tin UDP khi streaming video



Hình ảnh sau khi lọc giao thức UDP của gói tin vừa bắt

## 1.3 Phân tích hoạt động giao thức UDP

- Chọn một gói tin UDP, xác định các trường (field) có trong UDP header và giải thích ý nghĩa của mỗi trường đó?



- Chọn gói tin UDP được khoanh vùng màu đỏ
- Các trường có trong UDP header là Source Port, Destination Port, Length, Checksum được khoanh vùng màu xanh
- Source Port: Cổng nguồn, số hiệu cổng của tiến trình gửi dữ liệu
- Destination Port: Cổng đích, xác định tiến trình nhận dữ liệu trên máy tính
- Length: Độ dài toàn bộ gói UDP bao gồm cả header và payload
- Checksum: Dùng để kiểm tra lỗi trong quá trình truyền

## 2. Qua thông tin hiển thị của Wireshark, xác định độ dài (tính theo byte) của mỗi trường trong UDP header?

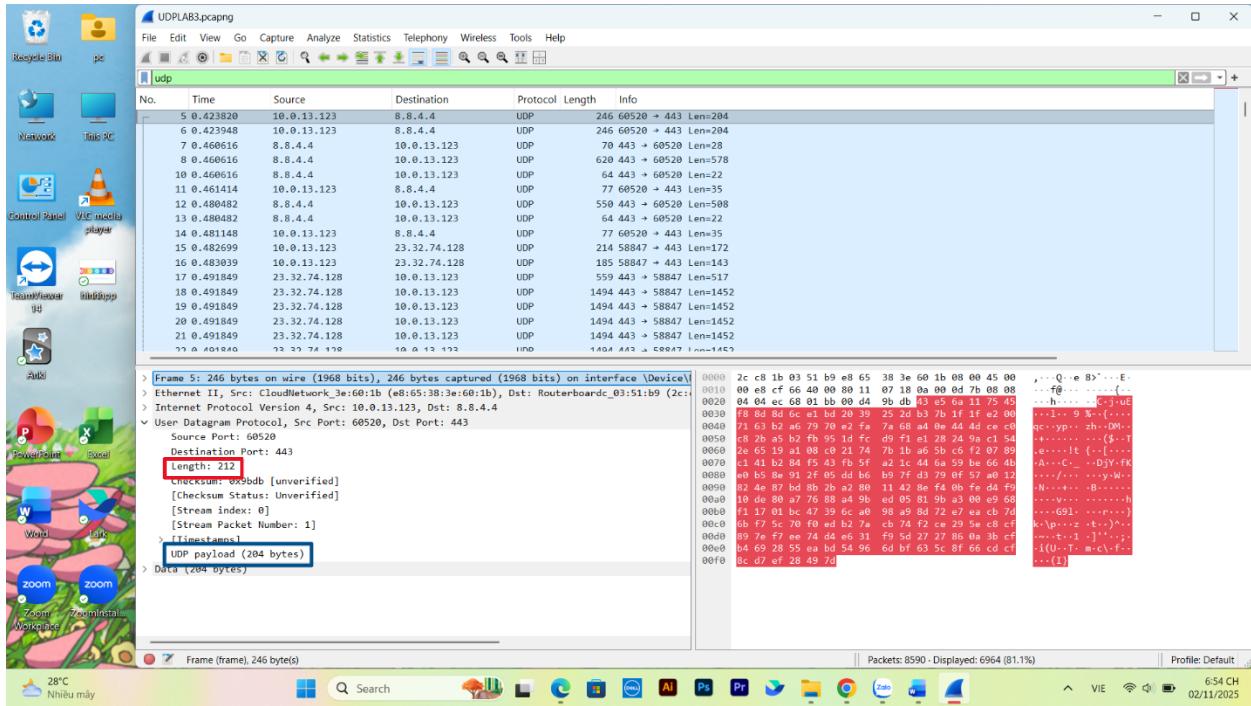
Xem thông tin được khoanh vùng màu xanh, ta nhận thấy độ dài của các trường trong UDP header lần lượt là:

- Source Port: 60520
- Destination Port: 443
- Length: 212
- Checksum: 0x9bdb

## 3. Giá trị của trường Length trong UDP header là độ dài của gì? Chứng minh nhận định này?

- Giá trị của trường Length trong UDP header là độ dài của 8 bytes header và N bytes payload
- Trong vùng khoanh màu đỏ, ta thấy trường Length trong UDP header có giá trị bằng 212 bytes
- Trong vùng khoanh màu xanh, ta thấy UDP Payload được bắt là 204 bytes
- $\text{Length} = 212 \text{ bytes} = 8 + 204$

⇒ Nhận định **Length = 8 bytes UDP header + N bytes payload** là đúng



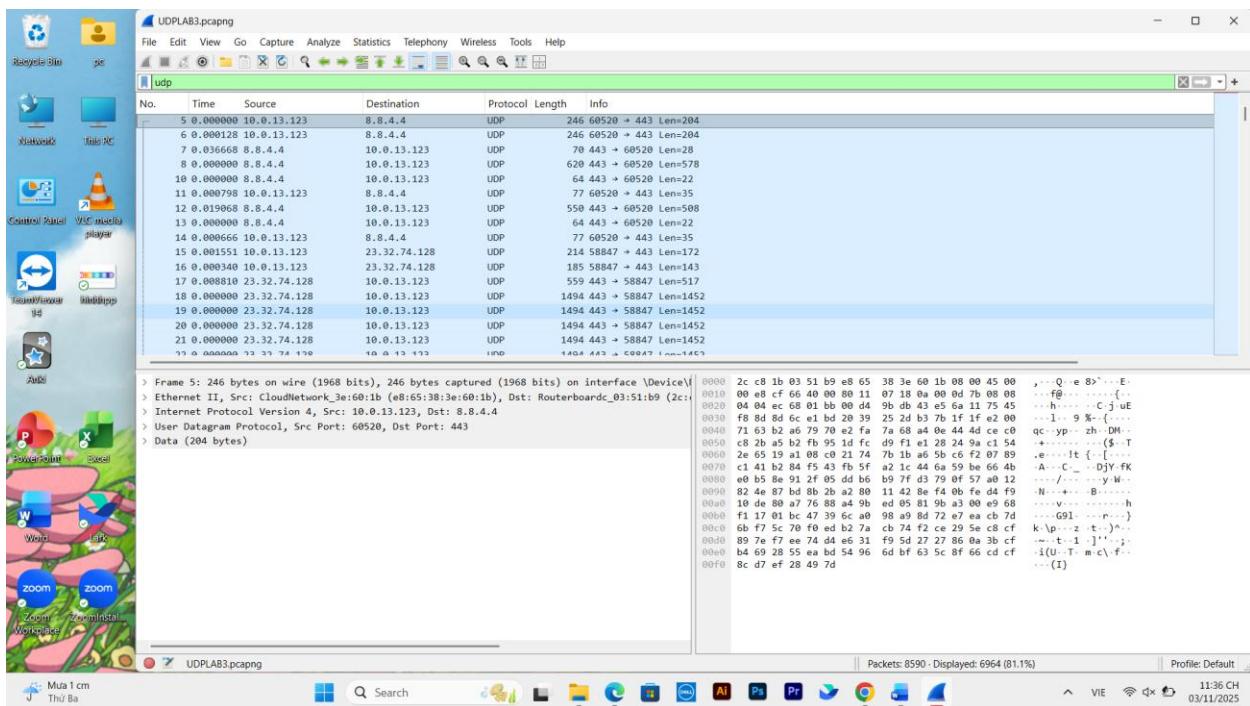
#### 4. Số bytes lớn nhất mà payload (phần chứa dữ liệu gốc, không tính UDP header và IP header) của UDP có thể chứa?

- Trường Length của UDP có thể chứa  $2^{16} - 1$  (bytes)
  - Length = UDP header + payload
- ⇒ UDP payload = Length – UDP header =  $2^{16} - 1 - 8$  (bytes)
- IP gói tin tối đa (Ipv4) trường Total Length trong IP header tối thiểu là 16 bits
  - IP header tối thiểu là 20 bytes
- ⇒ UDP payload tối đa không tính UDP header và IP header =  $2^{16} - 1 - 20 - 8$

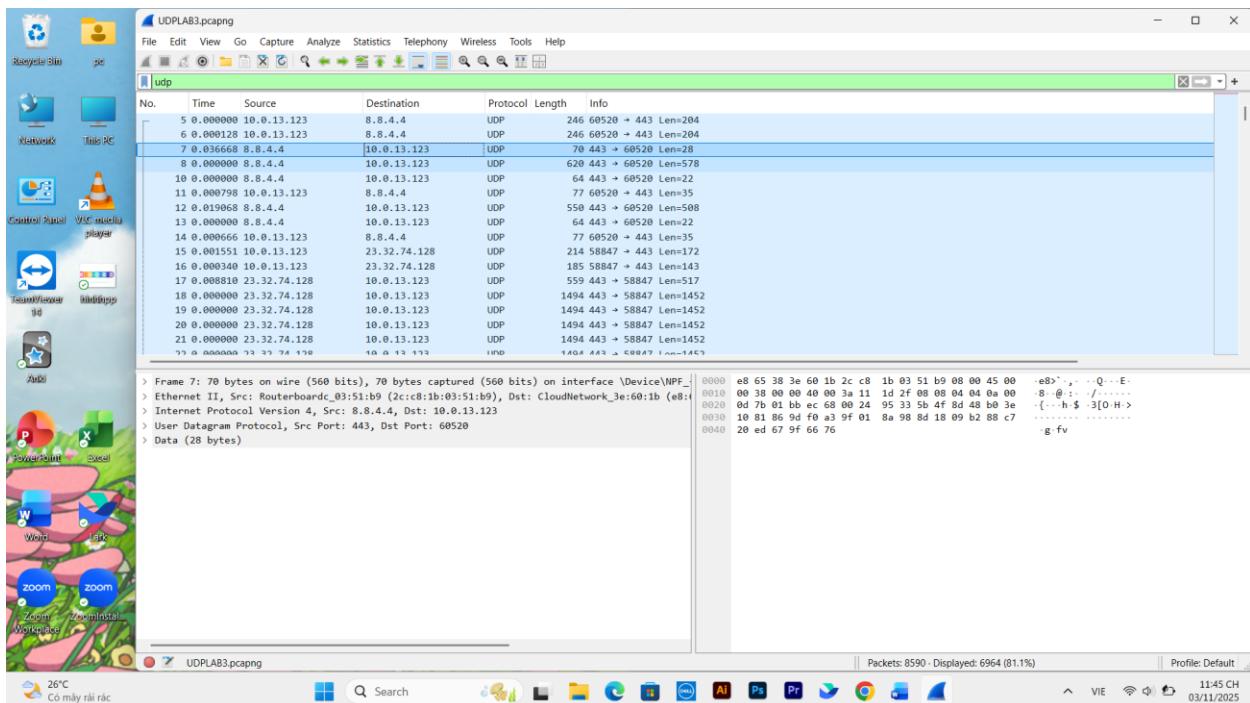
#### 5. Giá trị lớn nhất có thể có của port nguồn (Source port)?

- Giá trị lớn nhất có thể có của port nguồn là  $2^{16} - 1$

#### 6. \* Tìm và kiểm tra một cặp gói tin sử dụng giao thức UDP gồm: gói tin do máy mình gửi và gói tin phản hồi của gói tin đó. Miêu tả mối quan hệ về port number của 2 gói tin này.



### Gói tin do máy mình gửi



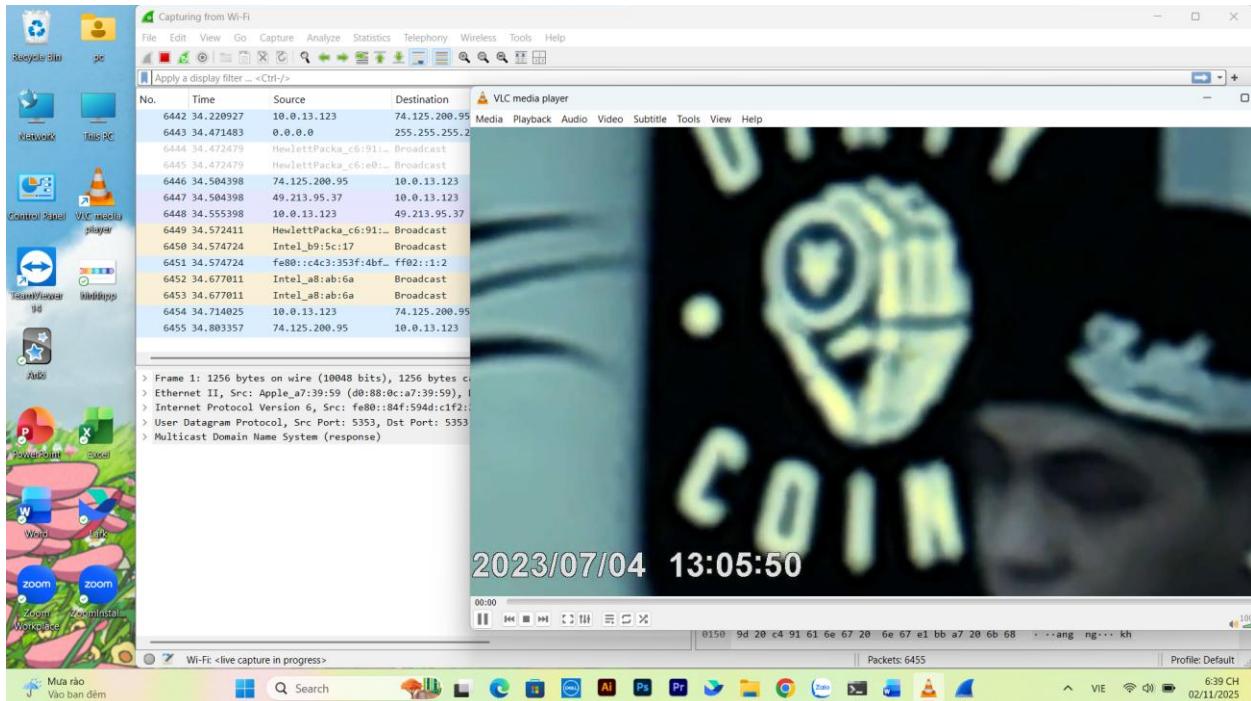
### Gói tin phản hồi của tin đó

- Gói tin do máy mình gửi có Src Port là 60520, Des Port là 443
- Gói tin phản hồi của tin đó có Src Port là 443, Des Port là 60520

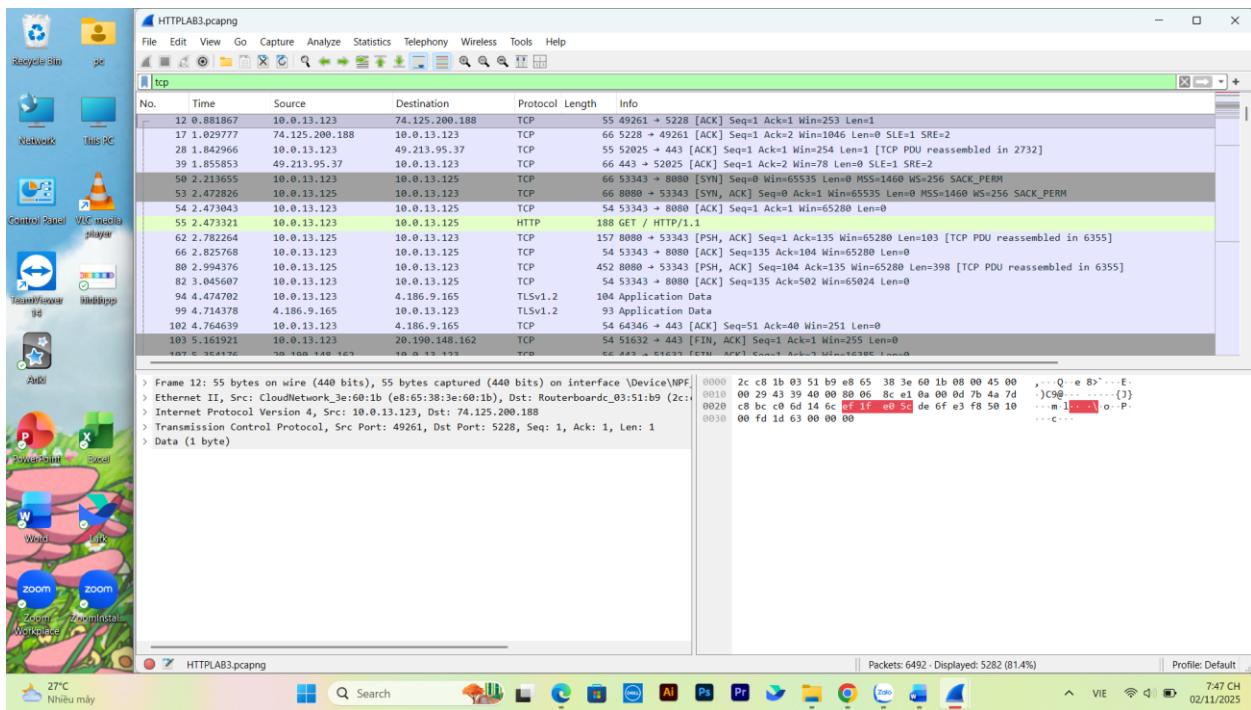
⇒ Trong một cặp gói tin UDP gồm gói gửi và gói phản hồi, Src Port và Des Port của 2 gói tin ngược nhau, có mối quan hệ nghịch đảo.

## TASK 2 – PHÂN TÍCH HOẠT ĐỘNG GIAO THỨC TCP

### 2.1 Streaming video sử dụng HTTP và bắt gói tin TCP



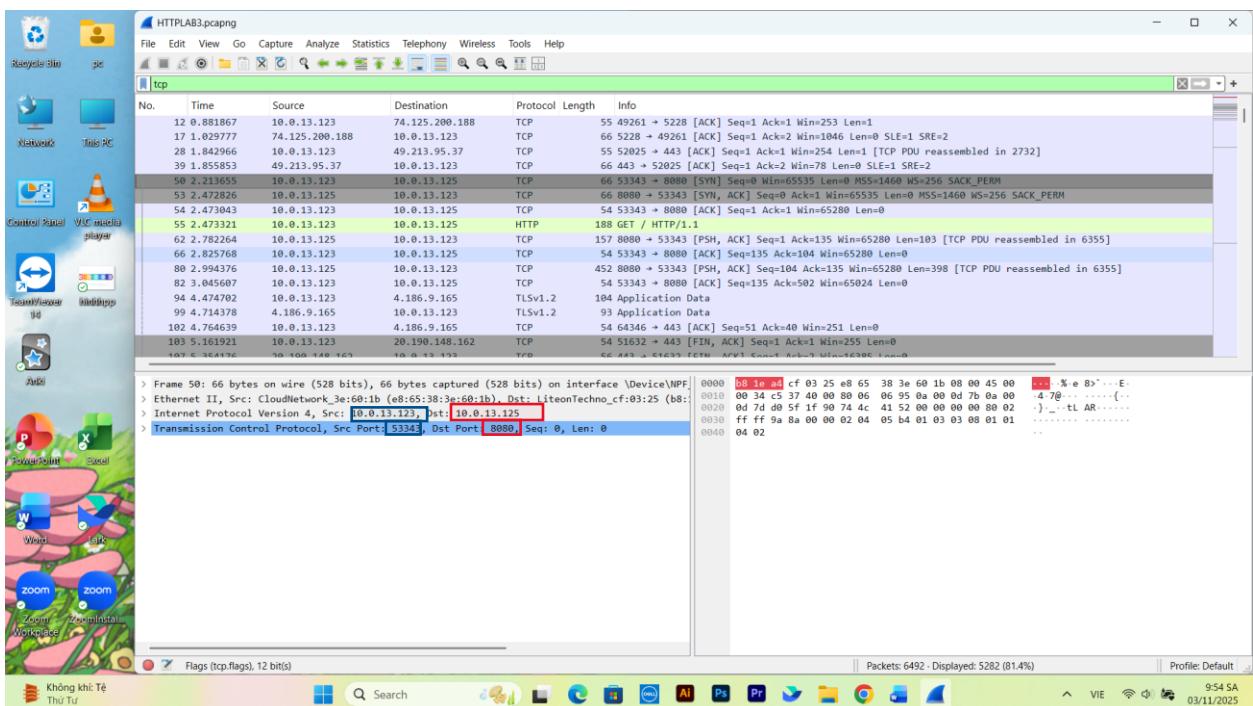
Hình ảnh sau khi stream video sử dụng HTTP



Hình ảnh sau khi lọc các giao thức TCP

## 2.2 Phân tích hoạt động giao thức TCP

### 7. Tìm địa chỉ IP và TCP port của máy Client?



- Địa chỉ IP của máy client: 10.0.13.123

- TCP port của máy client: **Src: 53343** là source port tạm thời do client tự chọn được khoanh vùng màu xanh

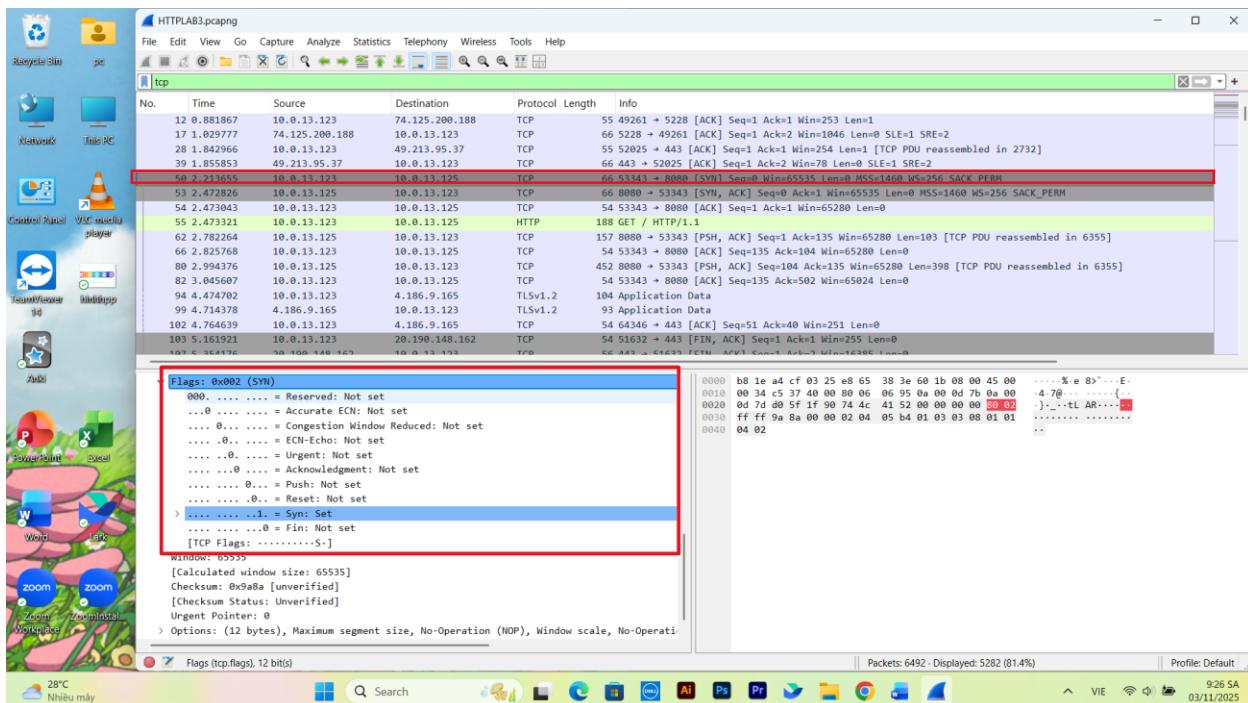
## 8. Tìm địa chỉ IP của Server? Kết nối TCP dùng để gửi và nhận các segments sử dụng port nào?

- Địa chỉ IP: 10.0.13.125
- Kết nối TCP dùng để gửi và nhận các segments sử dụng port: **Src Port: 53343** và **Dst port: 8080**

## 9. TCP SYN segment (gói tin TCP có cờ SYN) sử dụng sequence number nào để khởi tạo kết nối TCP giữa client và server? Thành phần nào trong segment cho ta biết segment đó là TCP SYN segment? Gợi ý: Quan sát trường Flags.

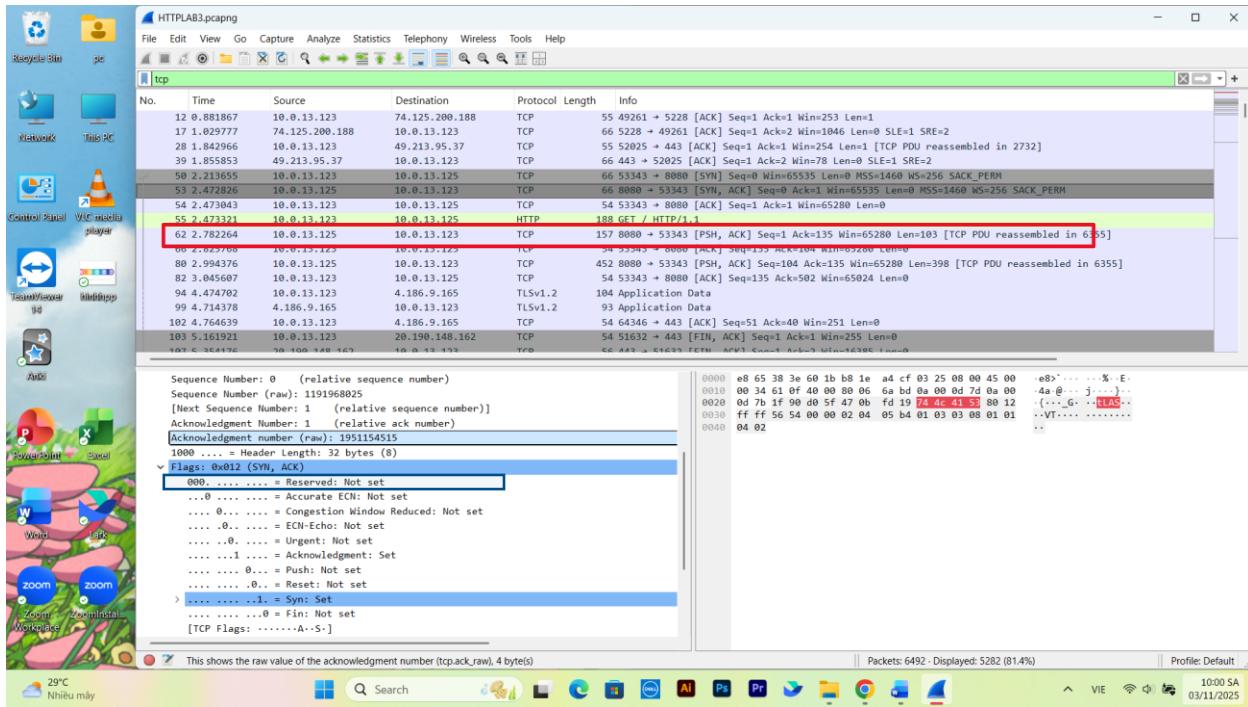
- Chọn gói tin được khoanh vùng màu đỏ như hình dưới
- TCP SYN segment sử dụng Initial sequence number (ISN) – số thứ tự ban đầu để khởi tạo kết nối TCP giữ client và server
- Trong trường Flags cho thấy: .....1.. = Syn: Set, các gói tin khác ACK, FIN, RST,... đều bằng 0.

⇒ Là TCP SYN segment **SYN = 1**, gói tin khởi tạo kết nối TCP

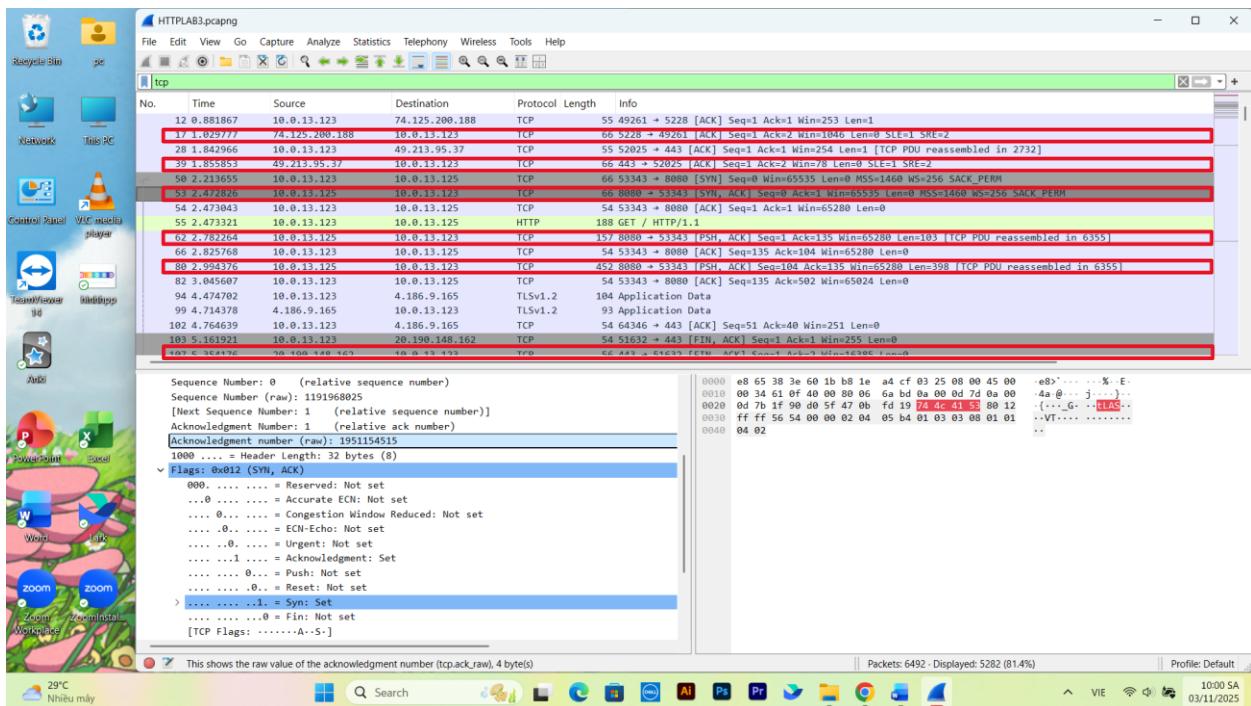


10. Tìm sequence number của gói tin SYN/ACK segment được gửi bởi server đến client để trả lời cho SYN segment? Tìm giá trị của Acknowledgement trong SYN/ACK segment? Làm sao server có thể xác định giá trị đó? Thành phần nào trong segment cho ta biết segment đó là SYN/ACK segment?

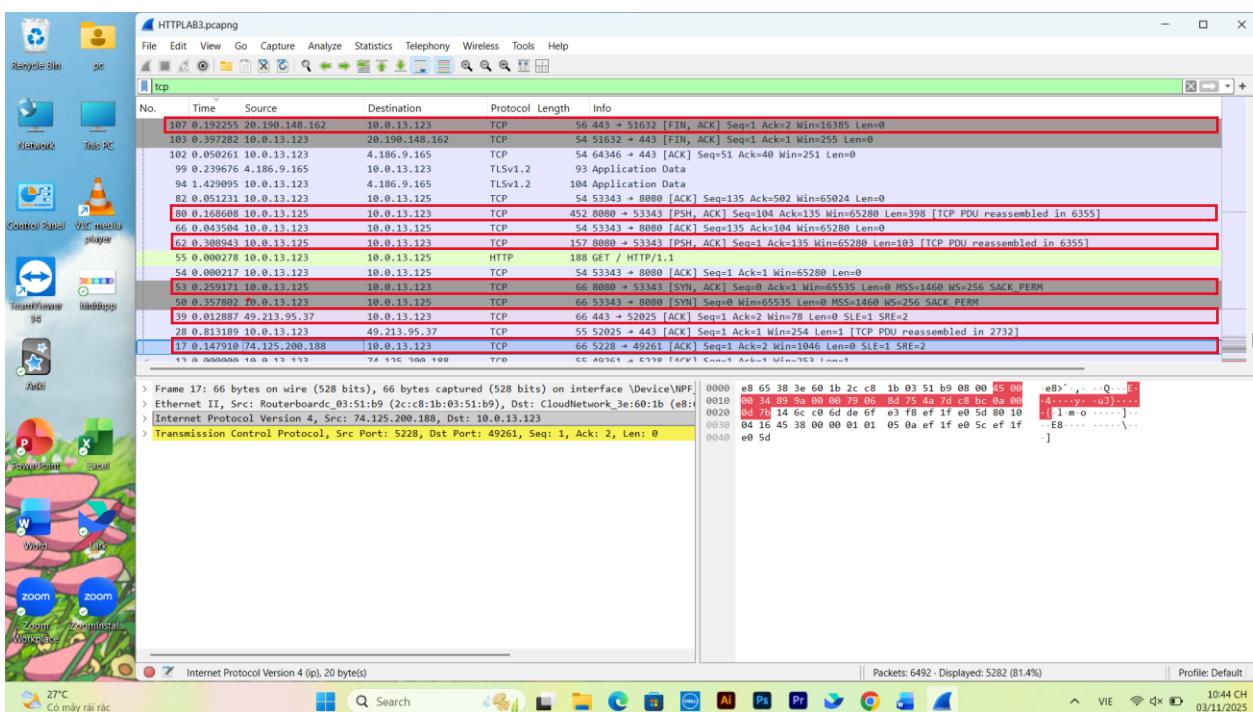
- SYN/ACK segment được gửi bởi server đến client để trả lời cho SYN segment được khoanh vùng màu đỏ
- Trong trường Flags thấy: 1 = Syn: Set và 1 = Acknowledgment: Set  
⇒ SYN = 1, ACK = 1 server trả lời lời mời kết nối của client
- Acknowledgement trong SYN/ACK segment là 1951154515 được khoanh vùng màu xanh
- Server xác định giá trị ACK bằng sequence number của gói SYN từ client + 1
- Dựa vào trường Flags có cả SYN = 1 và ACK = 1 ta xác định được segment đó là SYN/ACK segment



11. Chỉ ra 6 segment đầu tiên mà server gửi cho Client (dựa vào Số thứ tự gói – No)



- 6 segments đầu tiên mà server gửi cho client là No.17, No.39, No.53, No.62, No.80, No.107
- Sequence number của gói tin thứ tự No.17 là 1, No.39 là 1 No.53 là 0, No.62 là 1, No.80 là 104, No.107 là 1

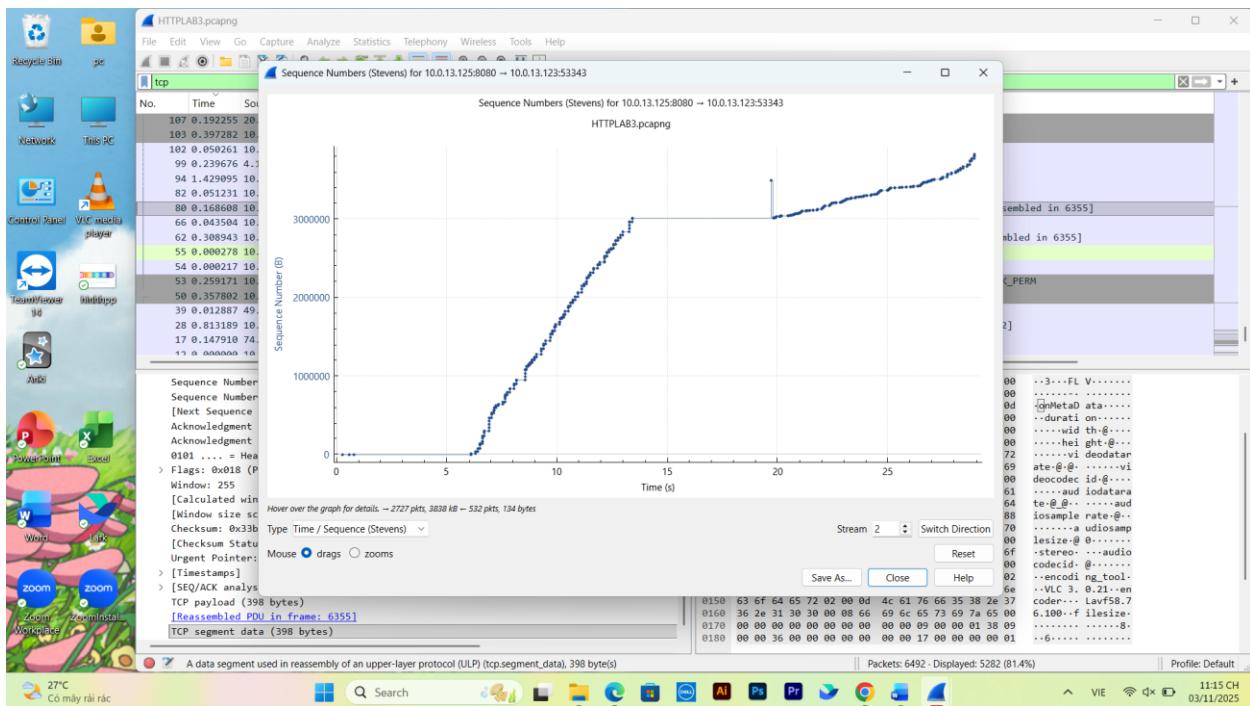


Thao tác: View – Time Display Format – Second Since Previous Display Packet  
để Wireshark hiện thời gian gửi gói tin.

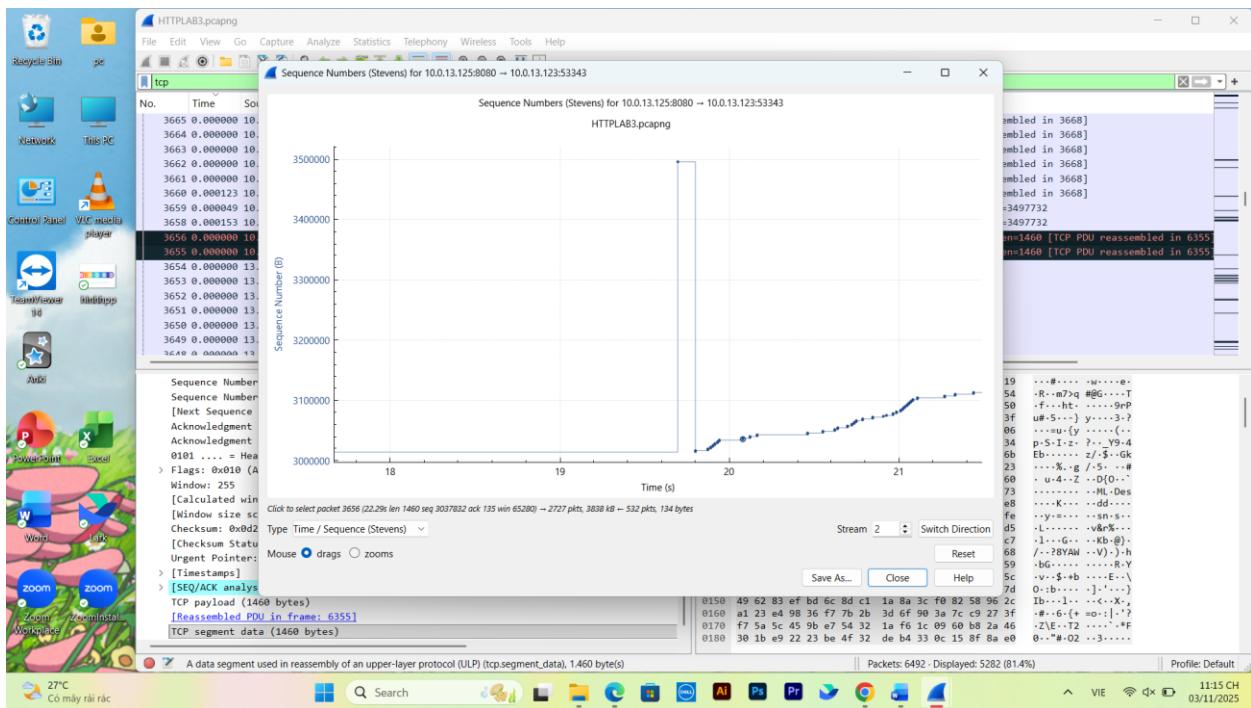
RTT = Thời gian nhận ACK – Thời gian gửi gói tin

| STT | Thời gian gửi | Thời gian nhận ACK | RTT(Round trip time) |
|-----|---------------|--------------------|----------------------|
| 17  | 0.147910      | 1.029777           | 0.881867             |
| 39  | 0.012887      | 1.855853           | 1.842966             |
| 53  | 0.259171      | 2.472826           | 2.213655             |
| 62  | 0.308943      | 2.782264           | 2.473321             |
| 80  | 0.168608      | 2.994376           | 2.825768             |
| 107 | 0.192255      | 5.354176           | 5.161921             |

12. Có segment nào được gửi lại hay không? Thông tin nào trong quá trình truyền tin cho chúng ta biết điều đó?



- Trong đồ thị trục X – Y với X là thời gian, Y là Sequence Number. Trước giây thứ 14 khi X tăng, Y có xu hướng tăng, tức là thời gian càng lâu, khi TCP gửi dữ liệu mới thì số Sequence Number tăng lên.



- Có segment được gửi lại, bởi trong hình, tại khoảng thời gian 18 – 19s, đồ thị đã có đường đi xuống đó là do TCP đã gửi lại những gói đã gửi trước đó dẫn đến hiện tượng Sequence Number quay lại giá trị cũ và đường đồ thị đi xuống.

⇒ **Có segment được gửi lại**