BÁO CÁO THỰC HÀNH

Môn học: Nhập môn mạng máy tính

Buổi báo cáo: Lab 03

Tên chủ đề: Phân tích hoạt động giao thức TCP - UDP

GVHD: Phan Trung Phát

Ngày thực hiện: 20/10/2022

Ngày nộp báo cáo: 30/10/2022

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: IT005.N11.KHTN.1

STT	Họ và tên	MSSV	Email
1	Trương Thanh Minh	21520064	21520064@gm.uit.edu.vn

2. ĐÁNH GIÁ KHÁC:

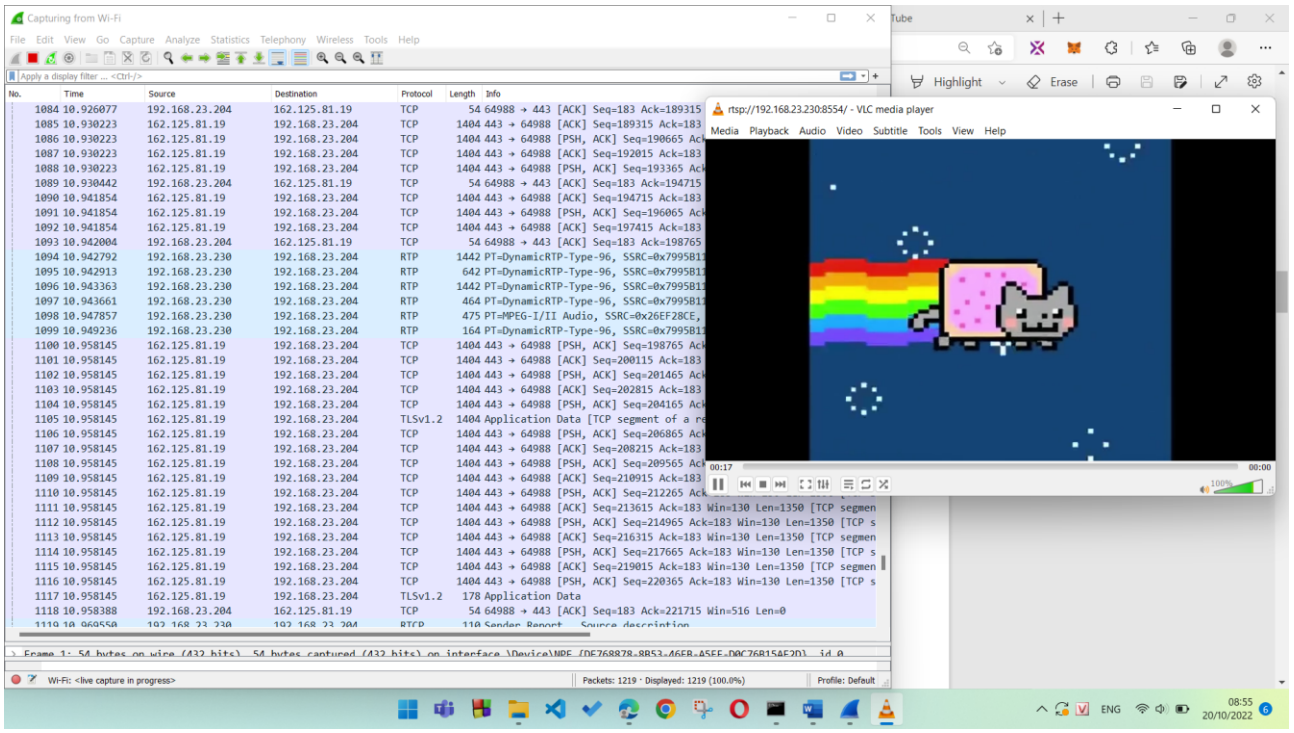
Nội dung	Kết quả
Tổng thời gian thực hiện bài thực hành trung bình	10 ngày
Link Video thực hiện *(nếu có)*	
Ý kiến *(nếu có)* + Khó khăn + Đề xuất …	
Điểm tự đánh giá	9.5

Phần bên dưới của báo cáo này là báo cáo chi tiết của nhóm thực hiện.

BÁO CÁO CHI TIẾT

---

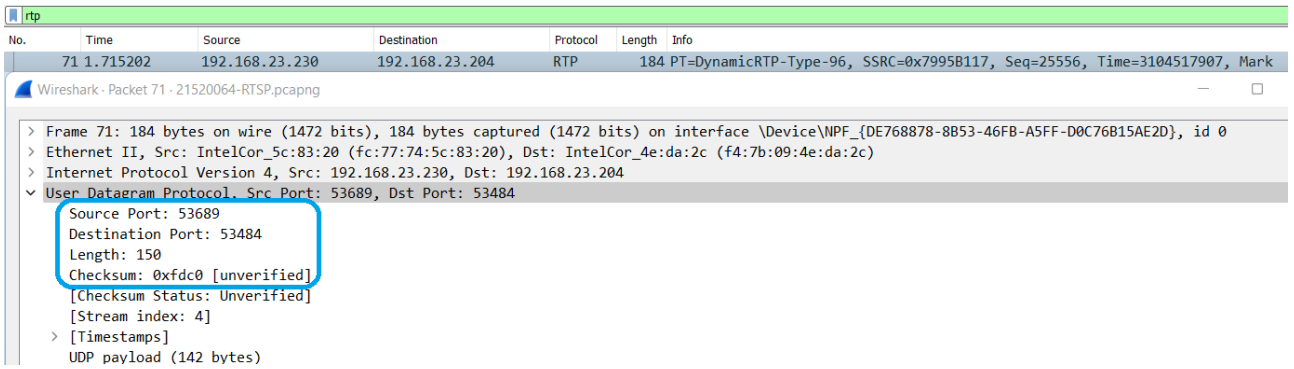
1. Task 1: Phân tích hoạt động giao thức UDP



Hình 1: Hình ảnh sau lúc xem video được stream từ máy của bạn

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

- Chọn RTP, ở user data gram: UDP



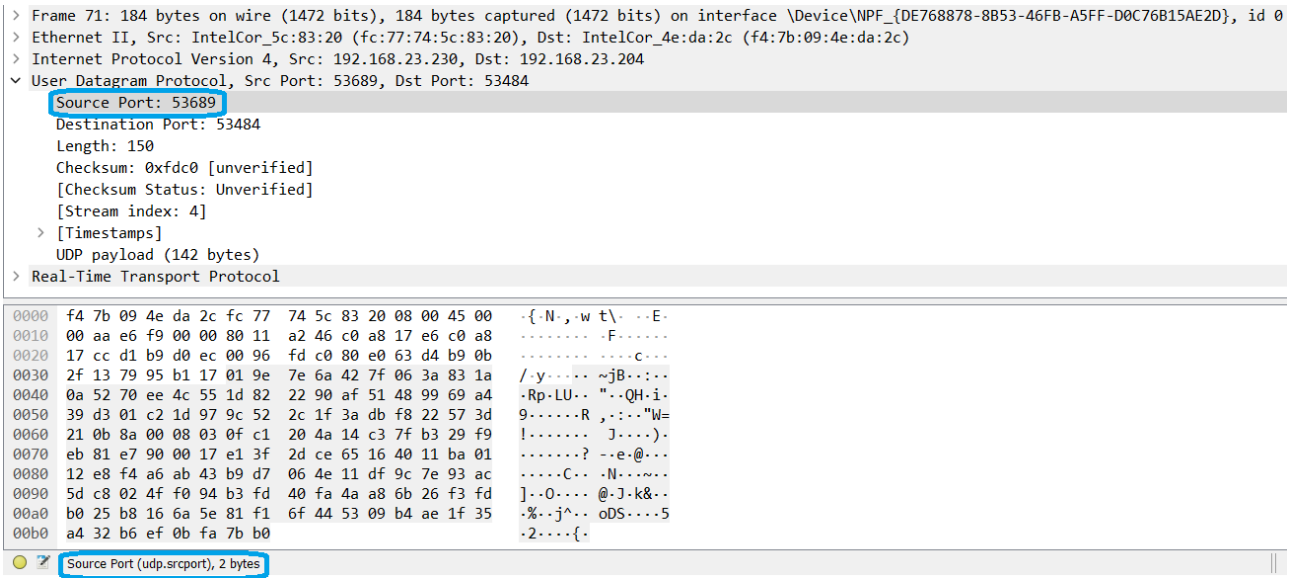
Hình 2: Các trường có trong UDP header

- Qua hình trên, ta thấy rằng trong gói tin đó có 4 trường:
  - Source Port: Địa chỉ của cổng gửi thông tin.
  - Destination Port: Địa chỉ của cổng nhận thông tin.
  - Length: Tổng độ dài (bytes) của gói tin.

- Checksum: Để xác minh rằng dữ liệu đầu cuối không bị hỏng bởi bộ định tuyến hoặc cầu nối trong mạng bởi quá trình xử lý trong hệ thống đầu cuối. Điều này cho phép người nhận để xác minh rằng đó có phải là đích dự kiến của gói tin hay không, vì nó bao gồm địa chỉ IP, số cổng và số giao thwusc và nó xác minh rằng gói tin không bị cắt ngắn hoặc độn, vì nó bao gồm trường kích thước.

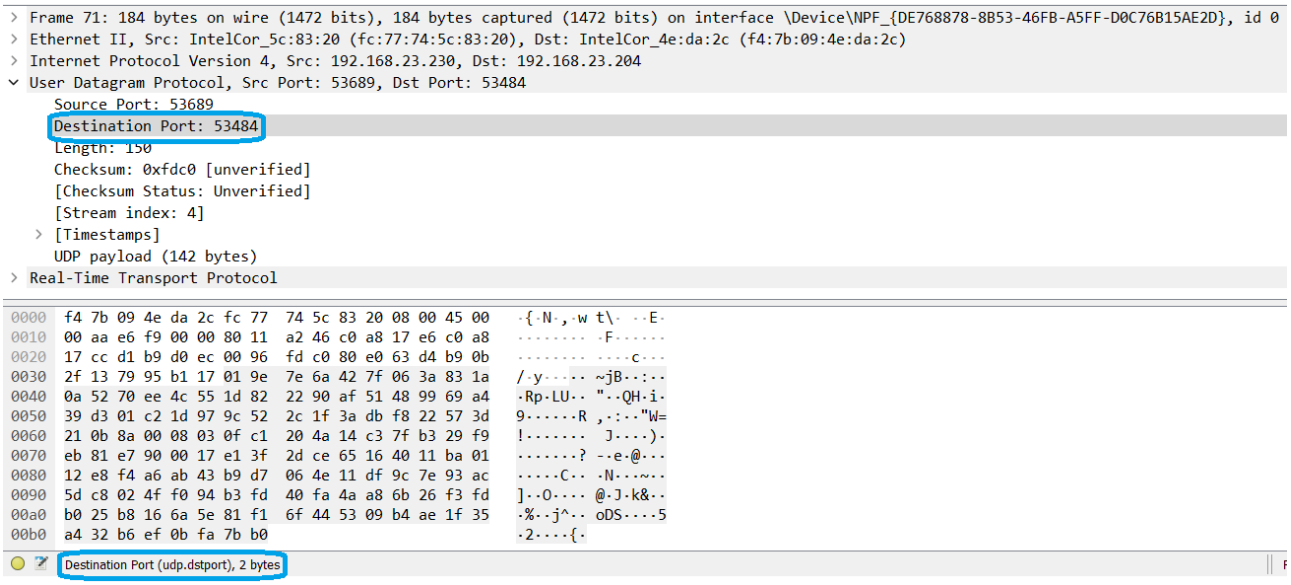
1.2. Qua thông tin hiển thị của Wireshark, xác định độ dài (tính theo byte) của mỗi trường trong UDP header?

- Ở trên, ta đã biết rằng có 4 trường UDP header
  - Source Port: có 2 bytes



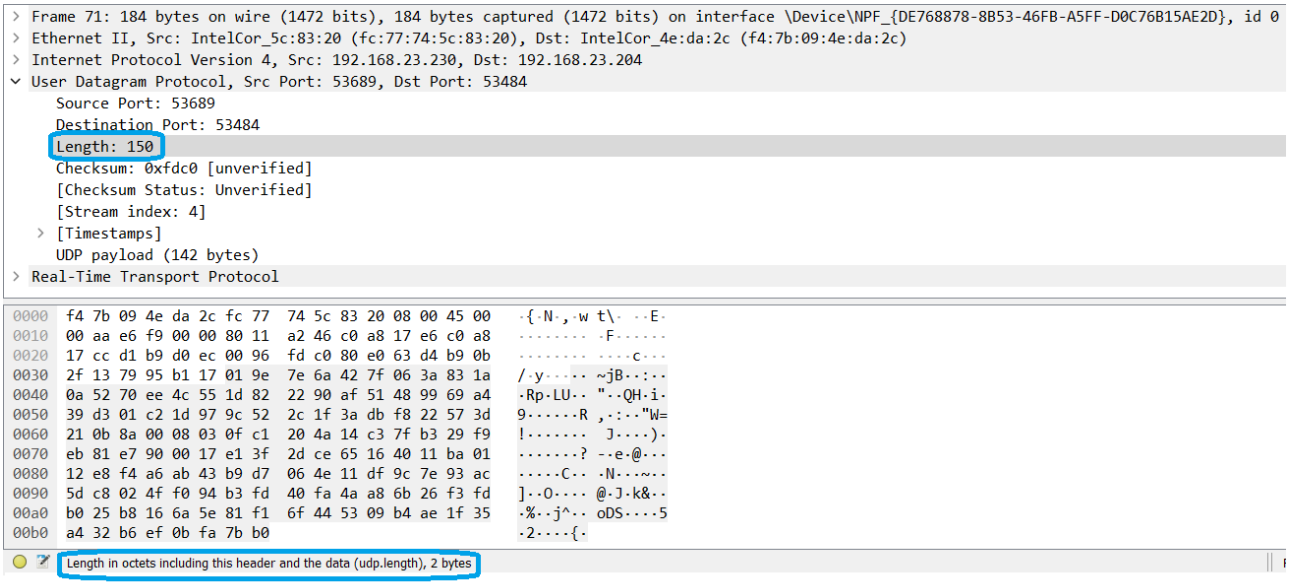
Hình 3: Độ dài byte của Source Port

  - Destination Port: 2 bytes



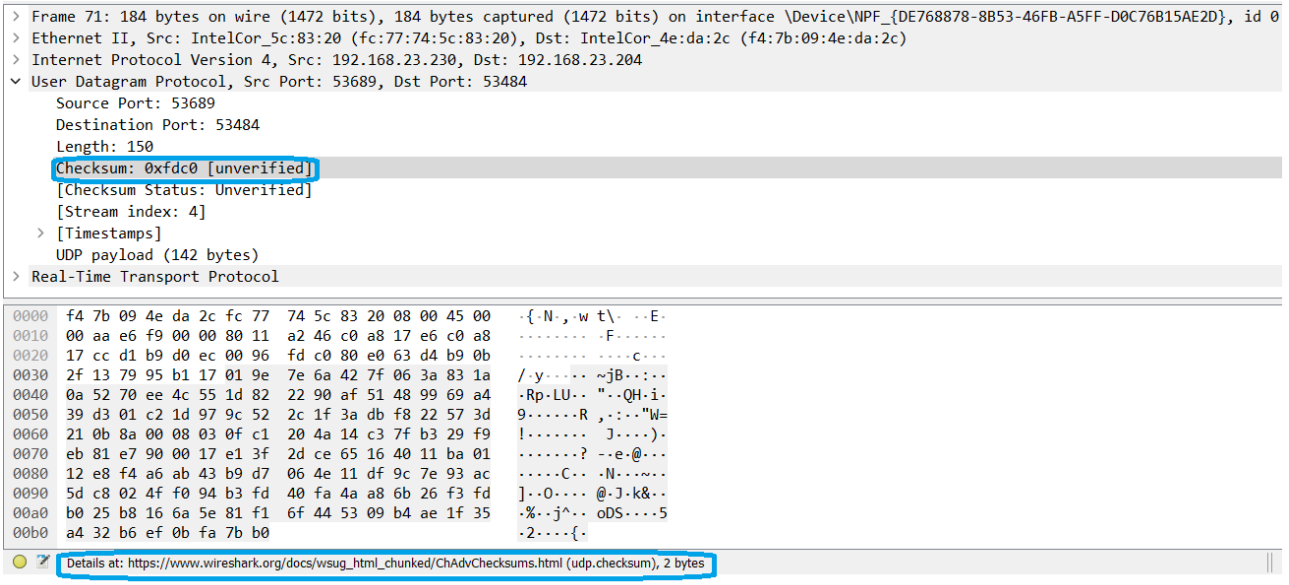
Hình 4: Độ dài bytes của Destination Port

- Length: 2 bytes



Hình 5: Độ dài bytes của Length

- Checksum: 2 bytes

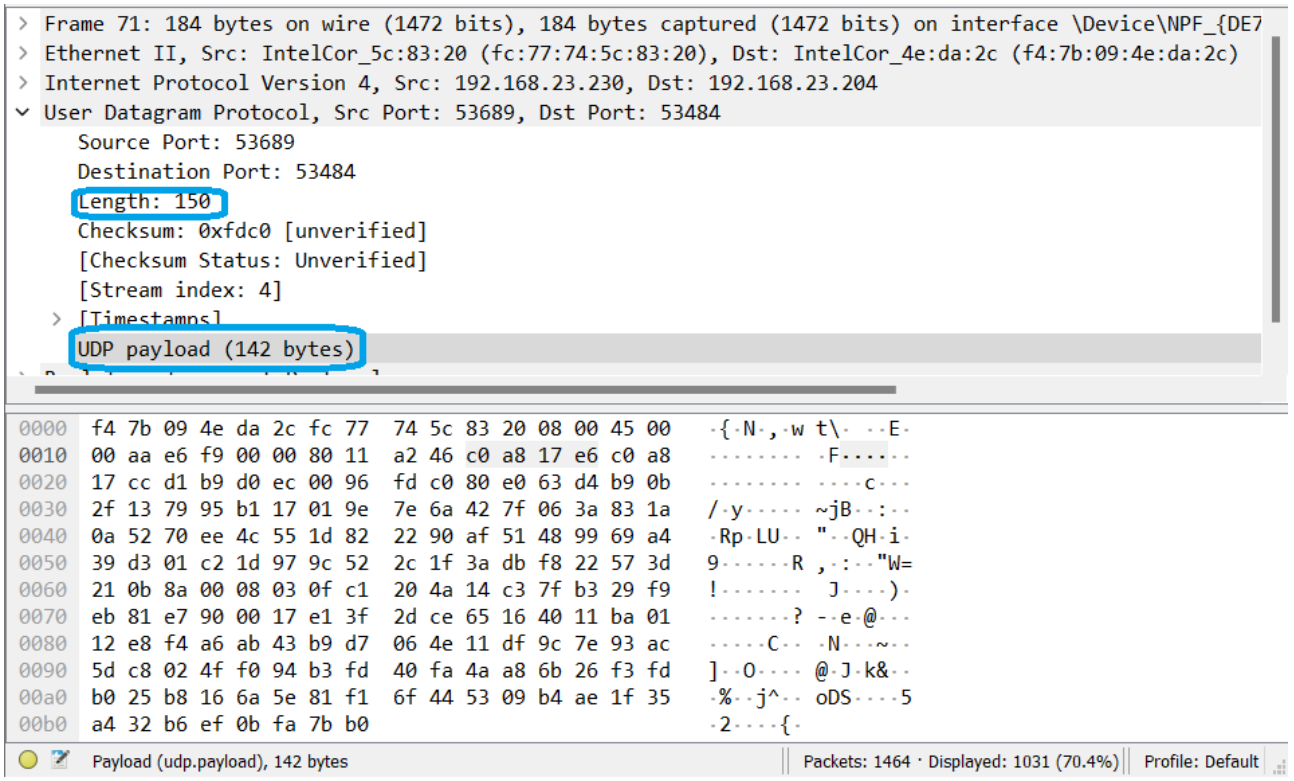


Hình 6: Độ dài bytes của Checksum

1.3. 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 tổng độ dài của các trường trong UDP header và UDP payload. Hay trình bày một cách tường minh hơn: *Length = UDP header + UDP payload*.
- Chứng minh: Ở các trường trong UDP header, như phần trên ta thấy, mỗi trường đều có độ dài là 2 bytes. Do đó, tổng độ dài của UDP header là **8**

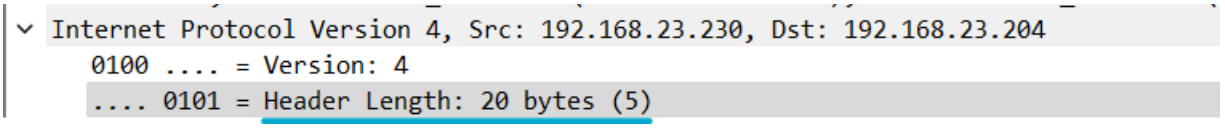
bytes. Ngoài ra, độ dài của UDP payload là **142 bytes.** Do đó, ở trên ta thấy rằng trường **Length** trong UDP header là **142 + 8 = 150 bytes**.



Hình 7: Giá trị trường Length và số bytes UDP payload

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

- Số bytes lớn nhất mà payload của UDP có thể chứa (tính luôn cả UDP header và IP header) là $\mathbf{2^{16}}$ – 1 = 65535 (bytes). (Gọi tắt là *Length*)
- Mà Length = Data Length + UDP header Length + IP header Length. Trong đó:
  - UDP header của ta cố định là 8 bytes (đã trình bày ở trên)
  - Vì ta dùng IPv4, do đó IP header của ta là 20 bytes.



Hình 8: IP header length

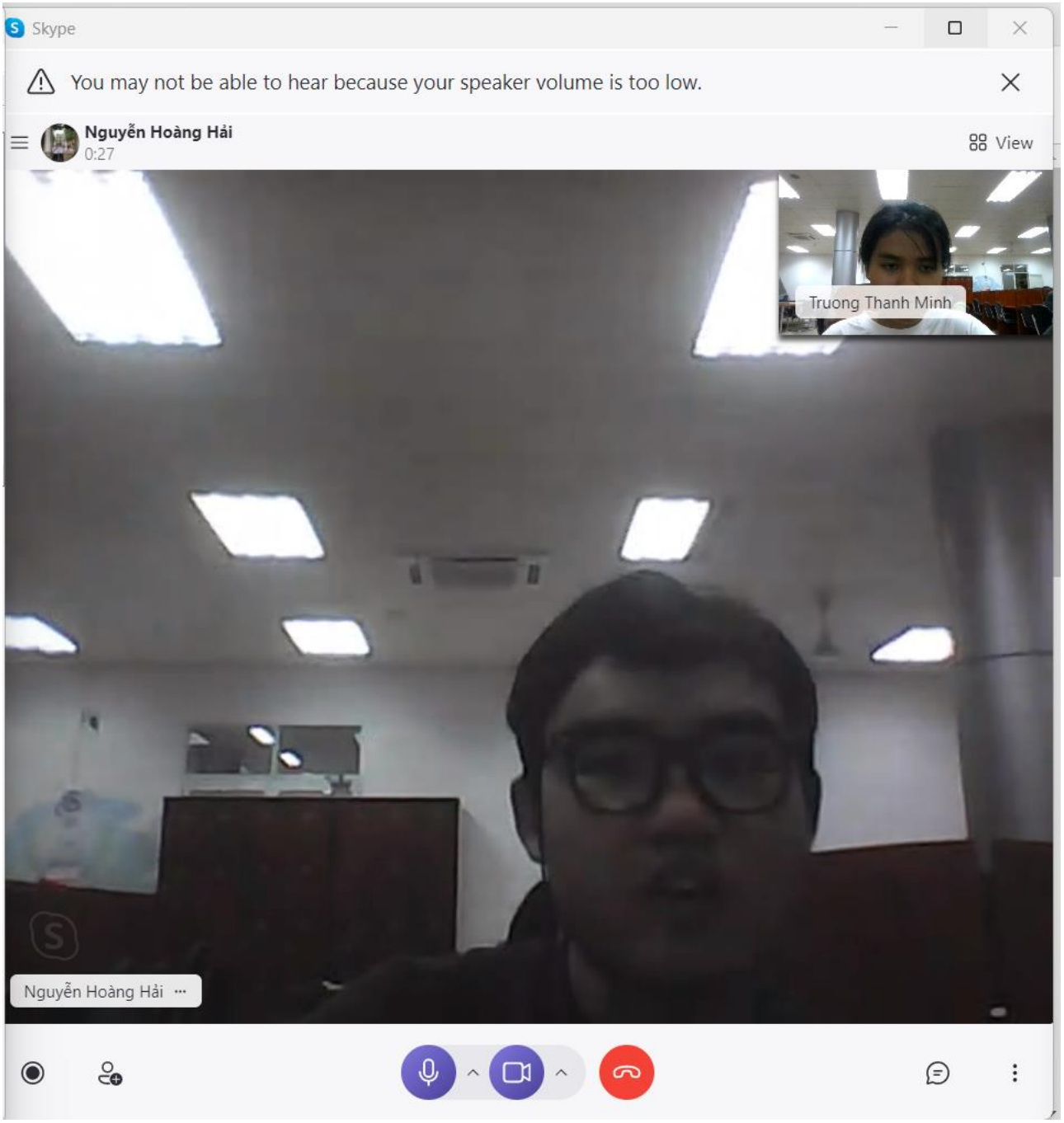
⇨ Data Length của ta là: 65535 – 8 – 20 = 65507 (bytes).

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

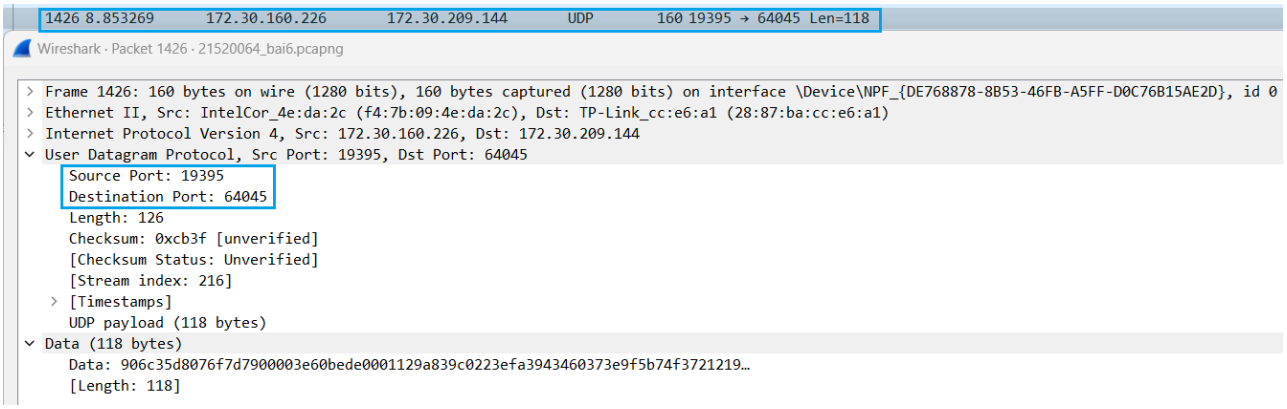
- Như đã đề cập ở trên, Source port có độ dài là 2 bytes (16 bits). Do đó, giá trị lớn nhất có thể có của nó là $\mathbf{2^{16}}$ – 1 = 65535.

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

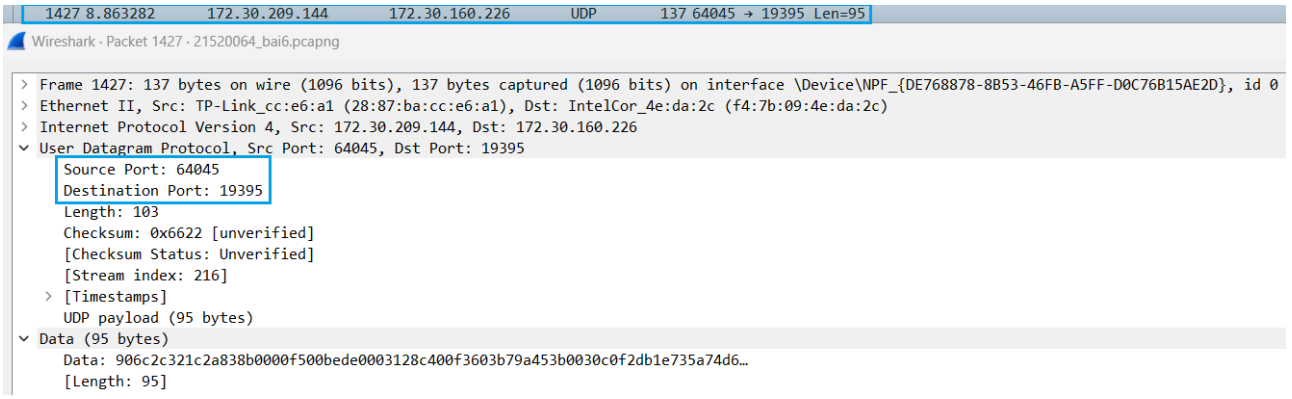
- Ở bài này, em thực hiện việc gọi video call bằng Skype vì Skype có sử dụng giao thức UDP.



Hình 9: Hình ảnh lúc gọi skype



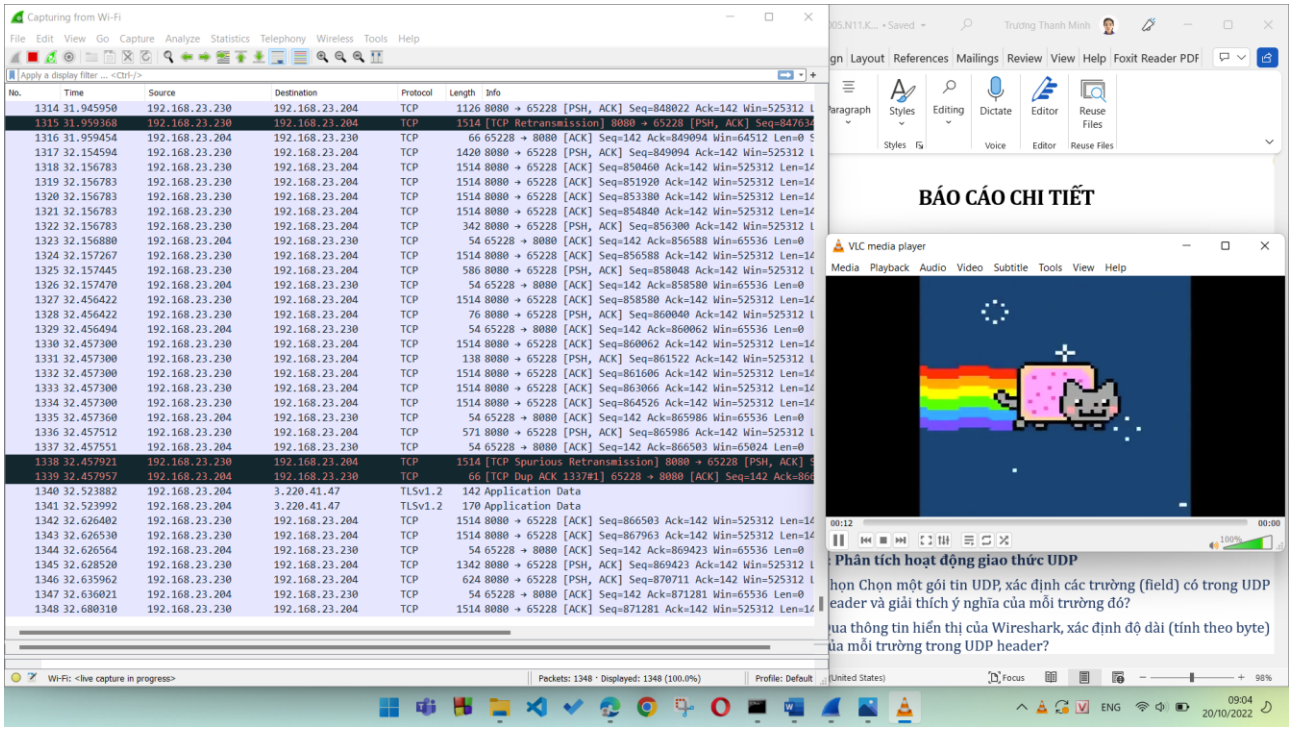
Hình 10: Gói tin được gửi từ máy mình đến máy bạn



Hình 11: Gói tin được gửi từ máy bạn đến máy mình

- Mối quan hệ giữa port number của 2 gói tin này là: Khi ta gửi request đến máy khác, port của máy ta sẽ trở thành Source Port (19395), port máy bạn sẽ trở thành Destination Port (64045). Trong trường hợp ngược lại, khi máy bạn gửi phản hồi về request trên, port của máy ta sẽ trở thành Destination Port (19395), và port của máy bạn sẽ là Source Port (64045).

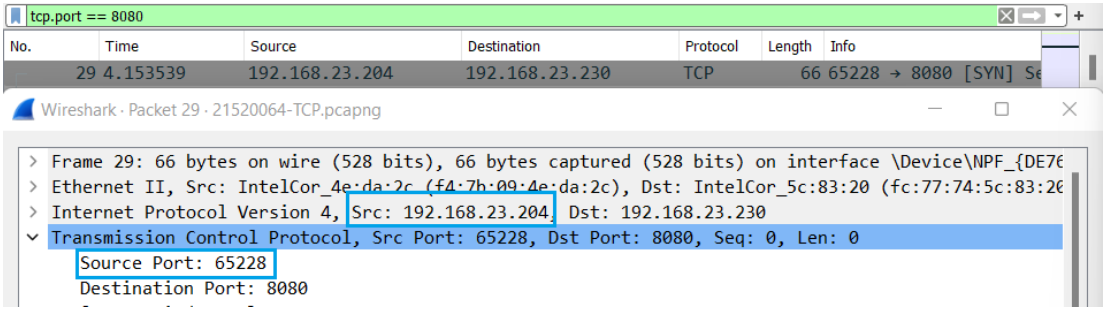
2. Task 2: Phân tích hoạt động giao thức TCP



Hình 12: Hình ảnh sau khi coi video được stream từ máy của bạn

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

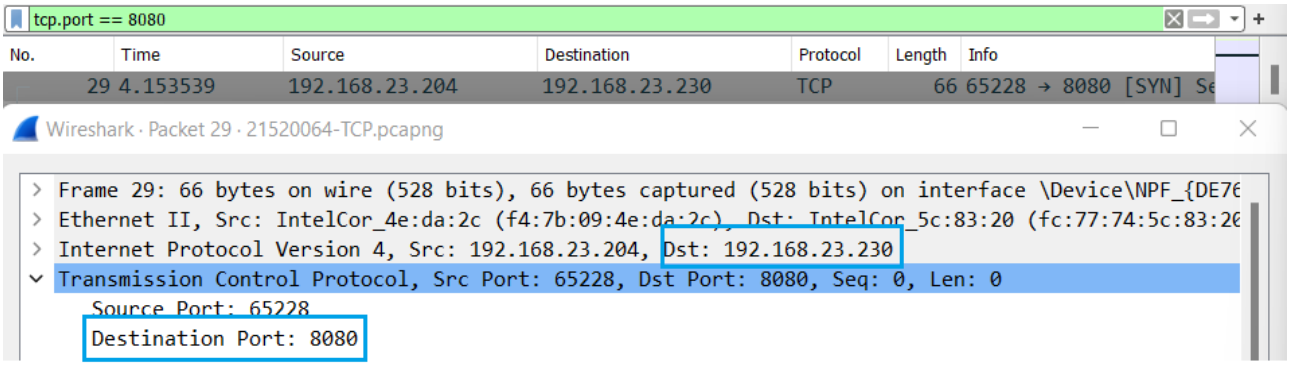
- Máy Client:
  - Địa chỉ IP: 192.168.23.204
  - TCP port: 65228



Hình 13: Địa chỉ IP của TCP port của Client

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

- Server: Cùng ở gói tin trên, địa chỉ Source là của Client, còn địa chỉ Destination là của Server.
  - Địa chỉ IP: 192.168.23.230
  - Port: 8080

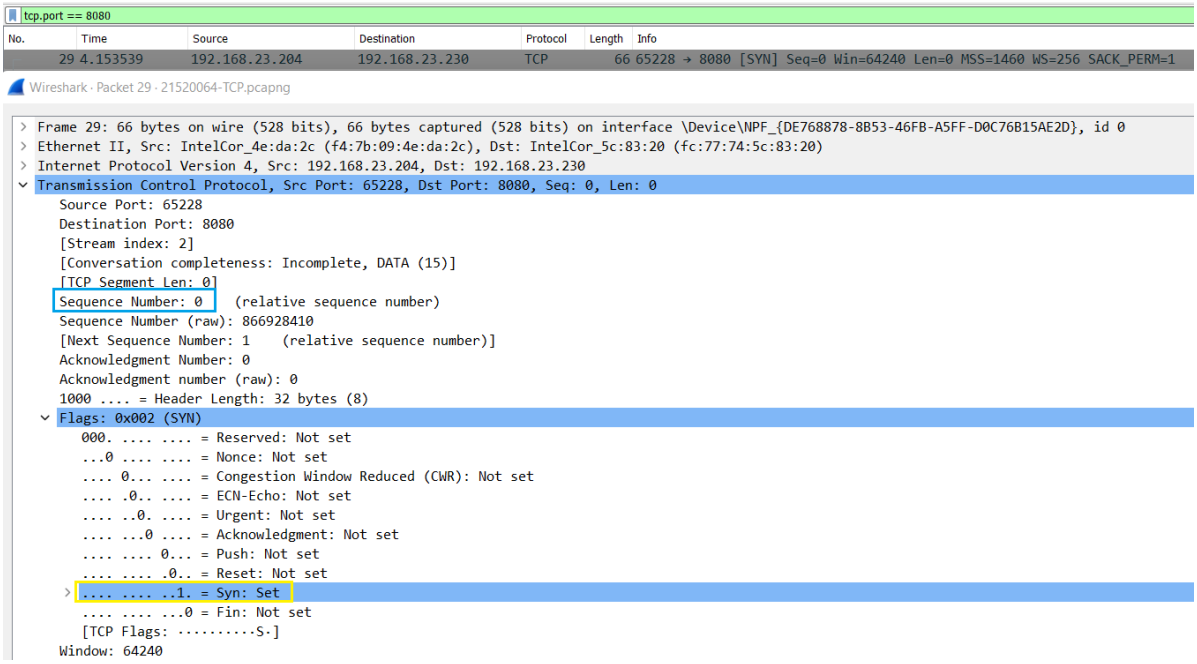


Hình 14: Địa chỉ IP của Server

- Vì TCP là một kết nối 2 chiều. Do đó, một kết nối TCP thường được sử dụng cho cả việc gửi và nhận các segments. Và các việc đó được sử dụng ở trong cùng một cổng. Và cổng đó chính là cổng ở trên: 8080.

2.9. 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**?

- Để gửi và nhận gói tin qua mạng, client và server cần phải khởi tạo kết nối với nhau thông qua 3-way handshark:
  - **Bước 1 (SYN):** Trong bước đầu tiên, client thiết lập kết nối với máy chủ. Nó gửi một phân đoạn với SYN và thông báo cho máy chủ về việc client sẽ bắt đầu giao tiếp và với số thứ tự của nó. Client sẽ gửi một message yêu cầu kết nối với Server.
  - **Bước 2 (SYN + ACK):** Trong bước này, máy chủ trả lời yêu cầu của client với bộ tín hiệu SYN-ACK (nếu đồng ý kết nối). Trong đó, ACK biểu thị phản hồi của phân đoạn được nhận và SYN biểu thị số thứ tự mà nó có thể bắt đầu với phân đoạn.
  - **Bước 3 (ACK):** Ở bước cuối cùng, Client nhận phản hồi của máy chủ và thông báo tới máy chủ bằng một đoạn tin nhắn với ACK và thông báo với máy chủ là đã nhận được phản hồi. Sau đó, cả 2 thiết lập một kết nối đáng tin cậy mà chúng sẽ bắt đầu truyền dữ liệu.

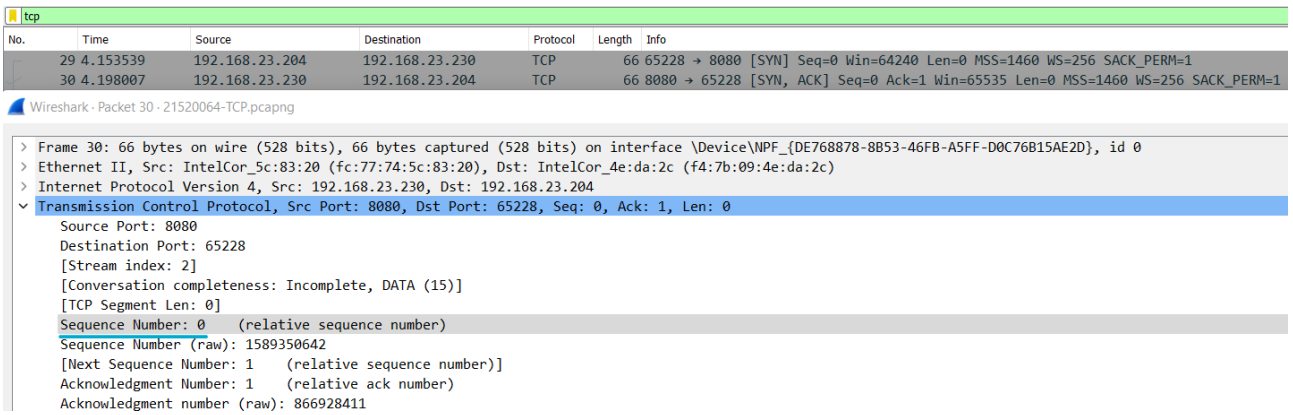


Hình 15: Sequence number của gói tin TCP SYN

- Ở hình trên, ta thấy rằng gói tin của ta được server gán cho Sequence number là 0.
- Ta thấy rằng Flags (SYN) được set bằng 1 → TCP segment.

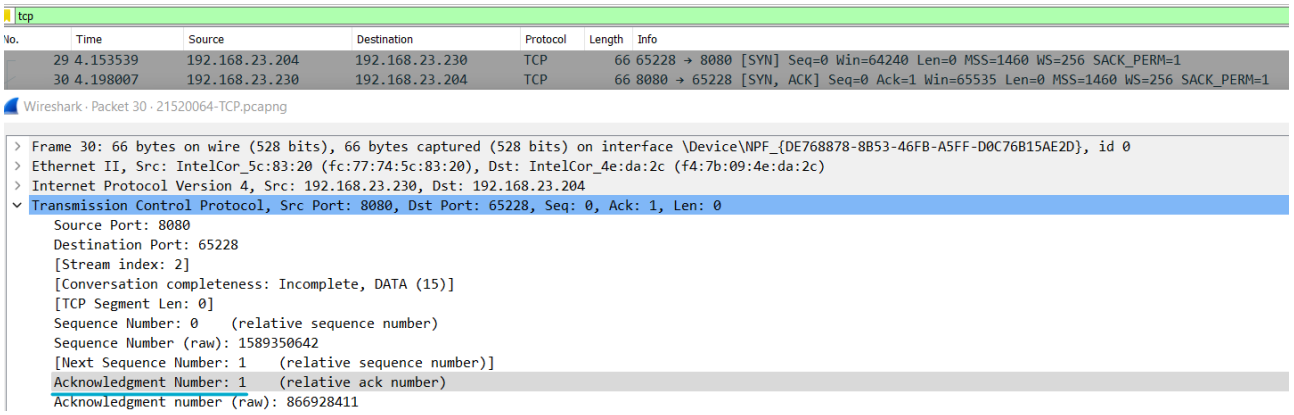
2.10. Tìm **sequence number** của gói tin **SYN/ACK segment** được gửi bởi server đến client để trả lời cho SYN segment?

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?



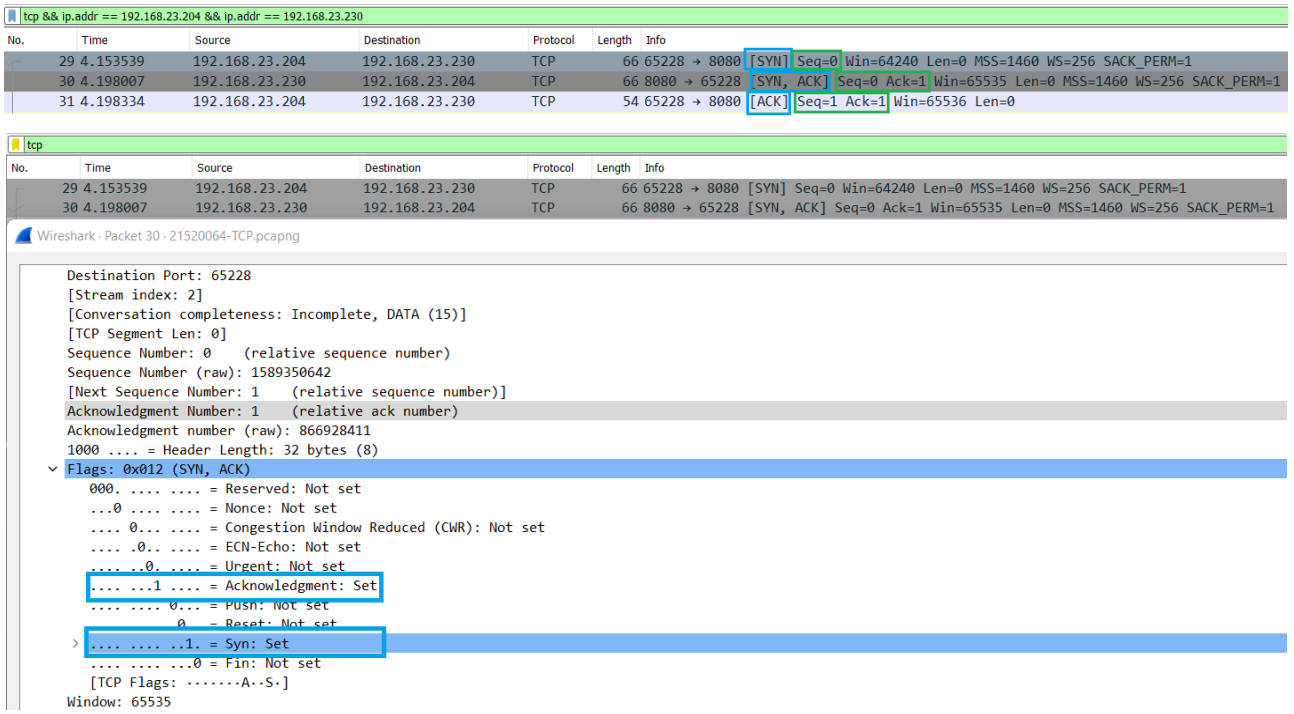
Hình 16: Chi tiết gói tin [SYN, ACK]

- Từ ảnh trên, ta thấy rằng Sequence number của gói tin SYN/ACK segment được gửi bởi server đến client để trả lời cho SYN segment ở trên là 0.



Hình 17: Giá trị Acknowledgement

- Từ ảnh trên, ta thấy rằng giá trị Acknowledgement trong SYN/ACK là 1.
- Cách server xác định được giá trị Acknowledgement:
  - Ban đầu, ở SYN, Client sẽ khởi tạo giá trị Sequence number của SYN là 0.
  - Sau đó, ở bước thứ 2, giá trị Acknowledgement tính gói tin [SYN, ACK] sẽ được Server bằng giá trị Sequence + 1. Ở ví dụ này, nó sẽ giá trị Acknowledgement = 1. (Sequence number ở bước trước là 0 + 1).
  - Tương tự ở gói tin [ACK] nó cũng được tính bằng giá trị Sequence ở bước trước + 1. Tức khi này giá trị ACK = 1.

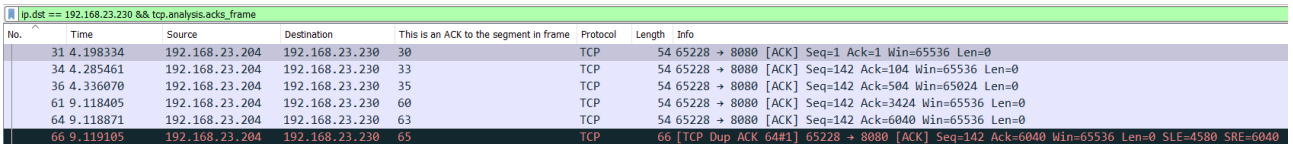


Hình 18: Cách nhận biết SYN/ACK segment

- Ta thấy rằng, trong ảnh trên, ở trường Flags, SYN và Acknowledgment được gán giá trị bằng 1. Do đó, ta có thể nhận biết được rằng đây là SYN/ACK segments.

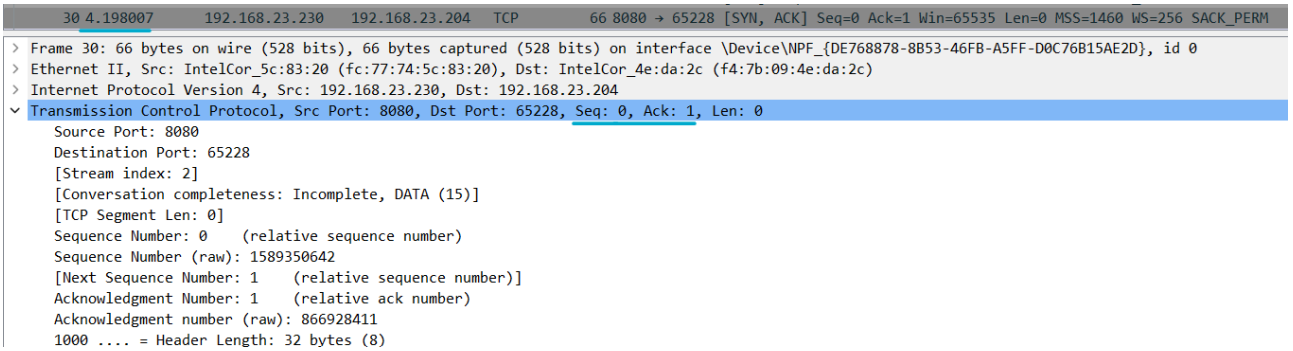
2.11. Chỉ ra 6 segment đầu tiên mà server gửi cho Client (dựa vào Số thứ tự gói – No)

- Để biết được 6 segments đầu tiền mà Server gửi cho Client, ta sử dụng filter như hình sau (trong đó, *ip.dst* của ta chính là địa chỉ ip của Server, *tcp.analysis.acks_frame* để lấy những gói tin TCP mà nó phản hồi lại gói tin ở trước nó):



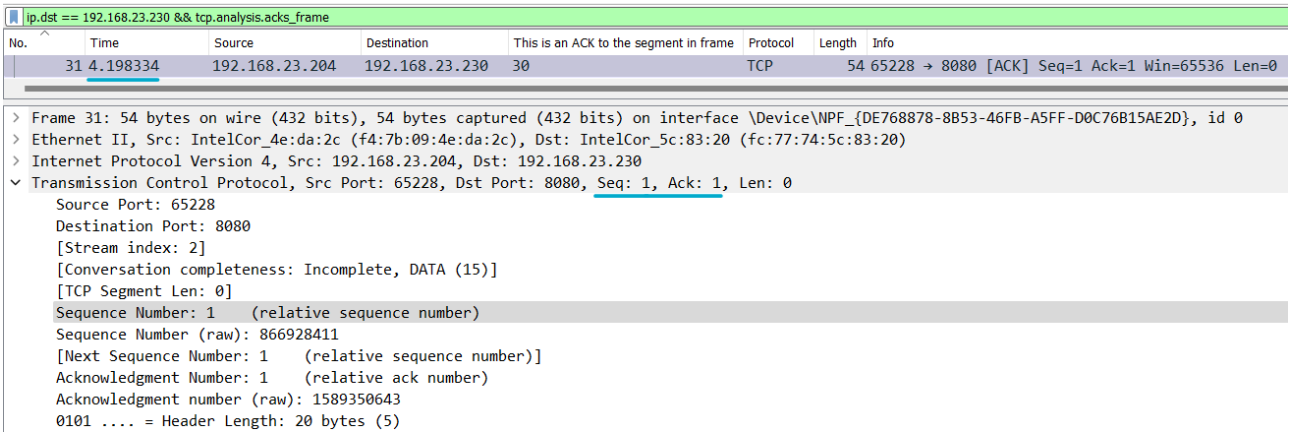
Hình 19: Các gói tin Client phản hồi Server

- Trong hình trên, trường *This is an ACK to the segment in frame* có ý nghĩa là đây là gói tin phải hồi cho gói tin nào. Ví dụ như gói tin số 31, trường đó là gói tin số 30, thì có nghĩa là gói tin số 31 là gói tin phản hồi của gói tin số 30. Và trường đó nó cũng chính là gói tin mà Server gửi về cho Client.
- Sau khi thực hiện lọc xong, ta suy ra được 6 gói tin đầu tiên Server gửi cho Client có ACK:
    - **Segment 30**: Thời gian segment được gửi là 4.198007. Seq = 0, Ack = 1.



Hình 20: Gói tin 30

ACK phản hồi của segment này là gói tin số 31. Thời điểm: 4.198334. Seq = 1, Ack = 1.



Hình 21: Gói tin phản hồi gói tin 30

- **Segment 33**: Thời gian gửi segment là: 4.234682. Seq = 1, Ack = 142

```
33 4.234682    192.168.23.230   192.168.23.204   TCP      157 8080 → 65228 [PSH, ACK] Seq=1 Ack=142 Win=525312 Len=103 [TCP segment of a reassembled PDU]
> Frame 33: 157 bytes on wire (1256 bits), 157 bytes captured (1256 bits) on interface \Device\NPF_{DE768878-8B53-46FB-A5FF-D0C76B15AE2D}, id 0
> Ethernet II, Src: IntelCor_5c:83:20 (fc:77:74:5c:83:20), Dst: IntelCor_4e:da:2c (f4:7b:09:4e:da:2c)
> Internet Protocol Version 4, Src: 192.168.23.230, Dst: 192.168.23.204
v Transmission Control Protocol, Src Port: 8080, Dst Port: 65228, Seq: 1, Ack: 142, Len: 103
    Source Port: 8080
    Destination Port: 65228
    [Stream index: 2]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 103]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 1589350643
    [Next Sequence Number: 104    (relative sequence number)]
    Acknowledgment Number: 142    (relative ack number)
    Acknowledgment number (raw): 866928552
    0101 .... = Header Length: 20 bytes (5)
```

Hình 22: Gói tin 33

ACK phản hồi của segment này là gói tin số 34. Thời điểm: 4.285461. Seq = 142, Ack = 104.

```
ip.dst == 192.168.23.230 && tcp.analysis.acks_frame
No.  Time      Source          Destination     This is an ACK to the segment in frame  Protocol  Length  Info
  31 4.198334  192.168.23.204  192.168.23.230  30                                      TCP         54 65228 → 8080 [ACK] Seq=1 Ack=1 Win=65536 Len=0
  34 4.285461  192.168.23.204  192.168.23.230  33                                      TCP         54 65228 → 8080 [ACK] Seq=142 Ack=104 Win=65536 Len=0
> Frame 34: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{DE768878-8B53-46FB-A5FF-D0C76B15AE2D}, id 0
> Ethernet II, Src: IntelCor_4e:da:2c (f4:7b:09:4e:da:2c), Dst: IntelCor_5c:83:20 (fc:77:74:5c:83:20)
> Internet Protocol Version 4, Src: 192.168.23.204, Dst: 192.168.23.230
v Transmission Control Protocol, Src Port: 65228, Dst Port: 8080, Seq: 142, Ack: 104, Len: 0
    Source Port: 65228
    Destination Port: 8080
    [Stream index: 2]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 0]
    Sequence Number: 142    (relative sequence number)
    Sequence Number (raw): 866928552
    [Next Sequence Number: 142    (relative sequence number)]
    Acknowledgment Number: 104    (relative ack number)
    Acknowledgment number (raw): 1589350746
    0101 .... = Header Length: 20 bytes (5)
```

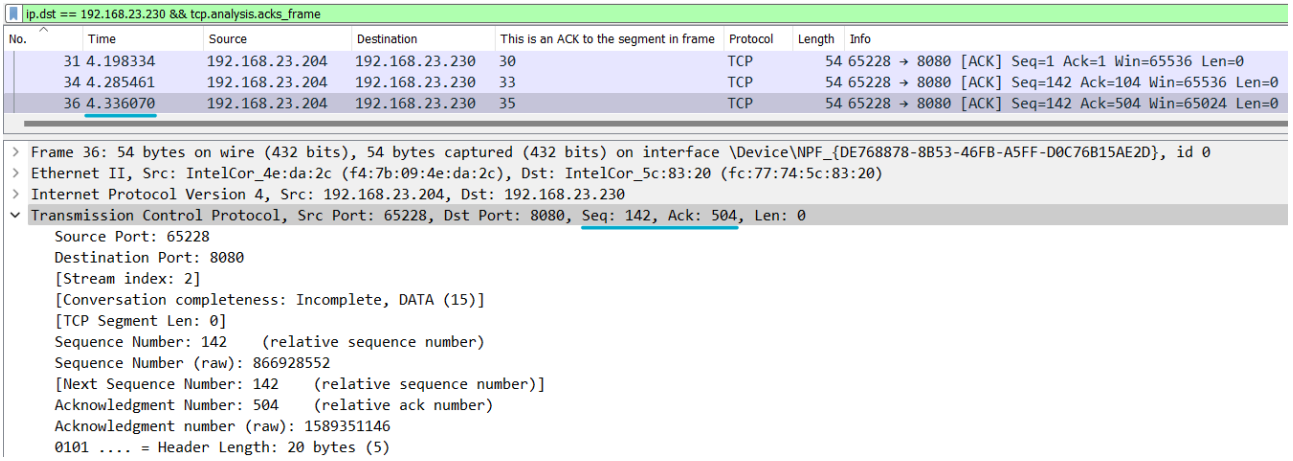
Hình 23: Gói tin phản hồi gói tin 33

- **Segment 35**: Thời gian segment được gửi đi là 4.295216. Seq = 104, Ack = 142.

```
35 4.295216    192.168.23.230   192.168.23.204   TCP      454 8080 → 65228 [PSH, ACK] Seq=104 Ack=142 Win=525312 Len=400 [TCP segment of a reassembled PDU]
> Frame 35: 454 bytes on wire (3632 bits), 454 bytes captured (3632 bits) on interface \Device\NPF_{DE768878-8B53-46FB-A5FF-D0C76B15AE2D}, id 0
> Ethernet II, Src: IntelCor_5c:83:20 (fc:77:74:5c:83:20), Dst: IntelCor_4e:da:2c (f4:7b:09:4e:da:2c)
> Internet Protocol Version 4, Src: 192.168.23.230, Dst: 192.168.23.204
v Transmission Control Protocol, Src Port: 8080, Dst Port: 65228, Seq: 104, Ack: 142, Len: 400
    Source Port: 8080
    Destination Port: 65228
    [Stream index: 2]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 400]
    Sequence Number: 104    (relative sequence number)
    Sequence Number (raw): 1589350746
    [Next Sequence Number: 504    (relative sequence number)]
    Acknowledgment Number: 142    (relative ack number)
    Acknowledgment number (raw): 866928552
    0101 .... = Header Length: 20 bytes (5)
```

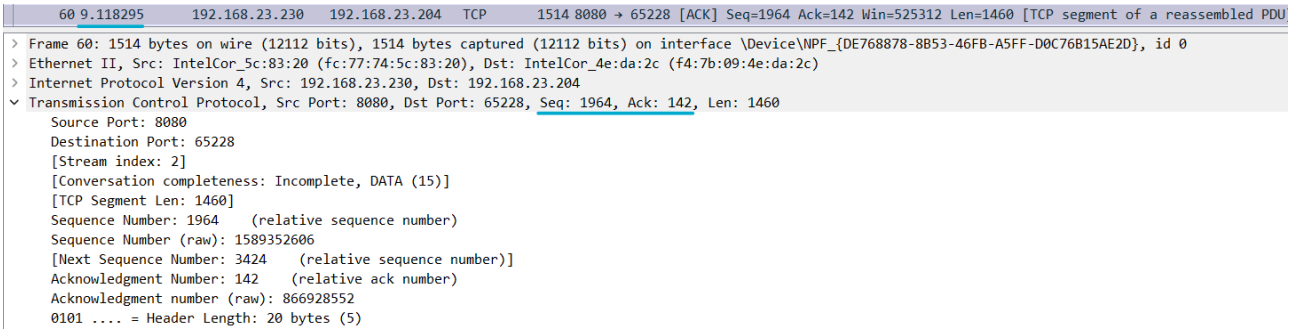
Hình 24: Gói tin 35

ACK phản hồi của segment này là gói tin số 36. Thời điểm: 4.336070. Seq = 142, Ack = 504.



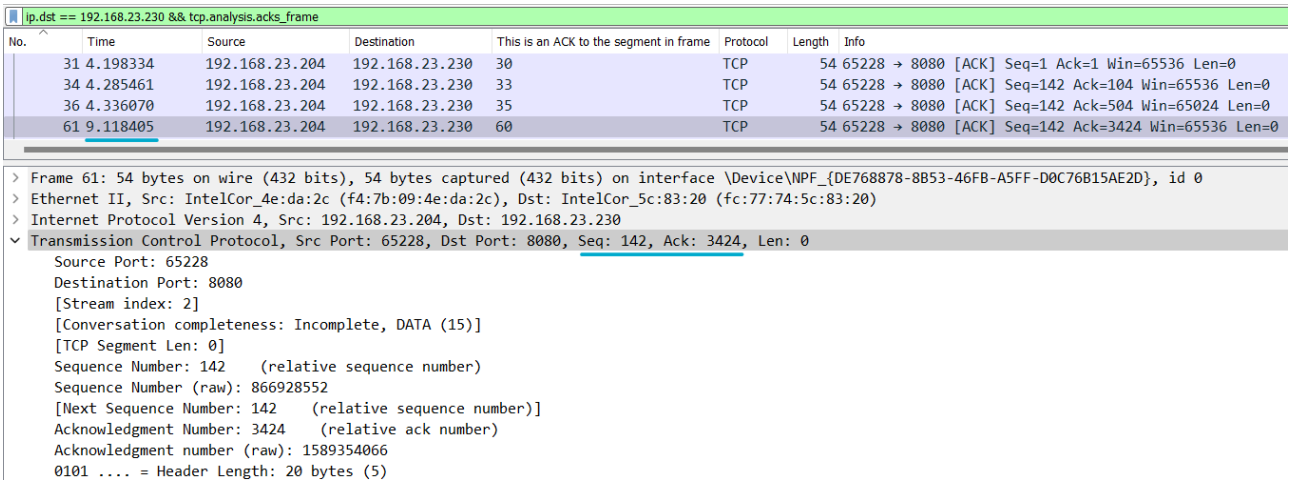
Hình 25: Gói tin phản hồi gói tin số 35

- **Segment 60**: Thời điểm gói tin được gửi đi là: 9.118295. Seq = 1964, Ack = 142.



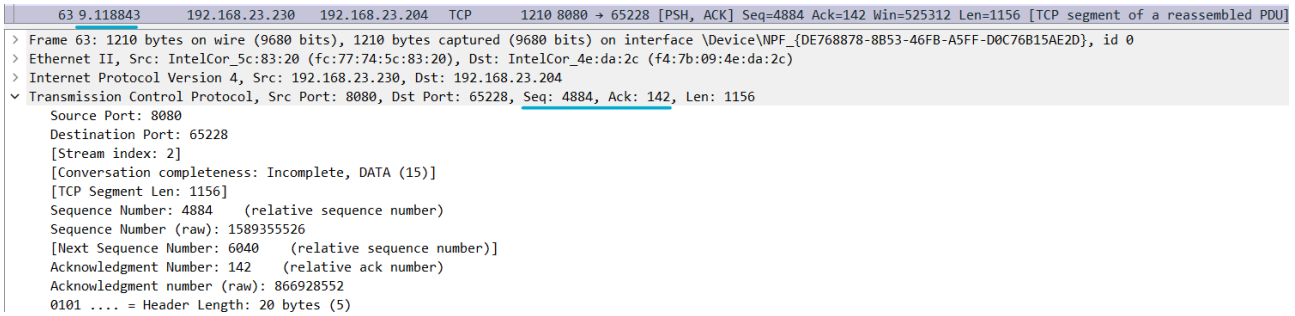
Hình 26: Gói tin 60

ACK phản hồi của segment này là gói tin số 61. Thời điểm: 9.118405. Seq = 142, Ack = 3424.



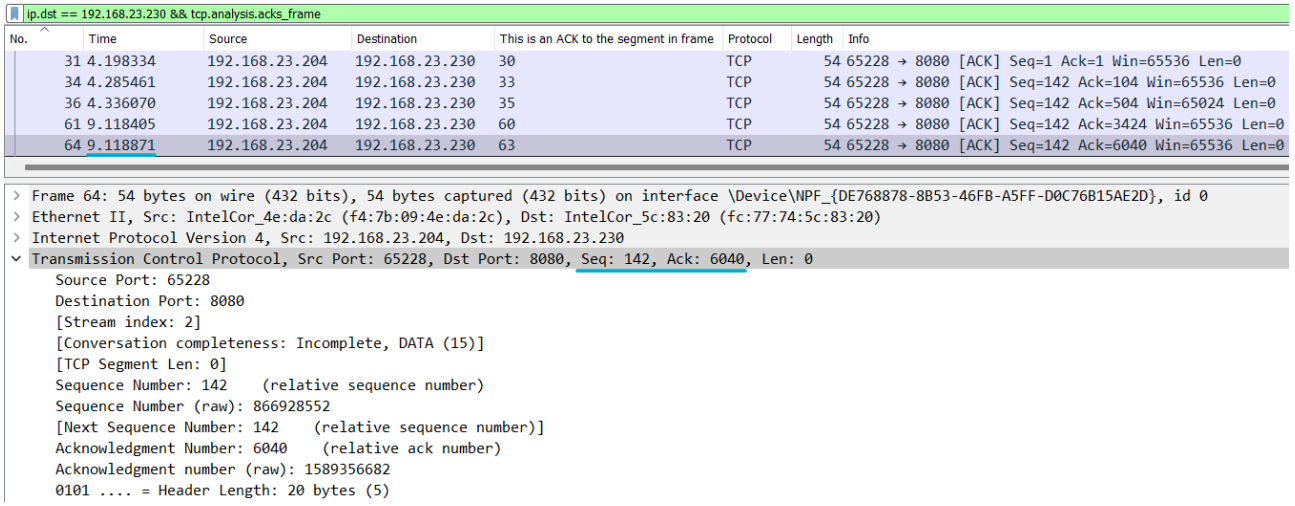
Hình 27: Gói tin phản hồi gói tin số 60

- **Segment 63**: Thời gian gói tin được gửi đi là 9.118843. Seq = 4884, Ack = 142.



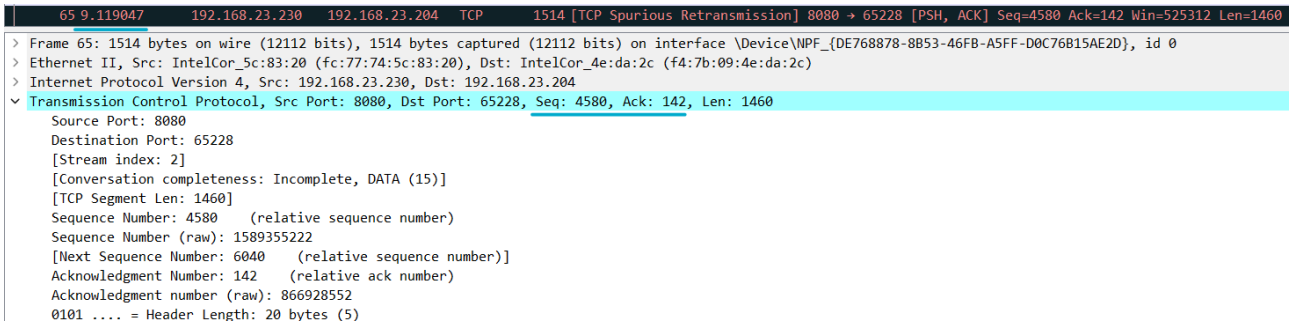
Hình 28: Gói tin 63

ACK phản hồi của segment này là gói tin số 64. Thời điểm: 9.118871. Seq = 142, Ack = 6040.

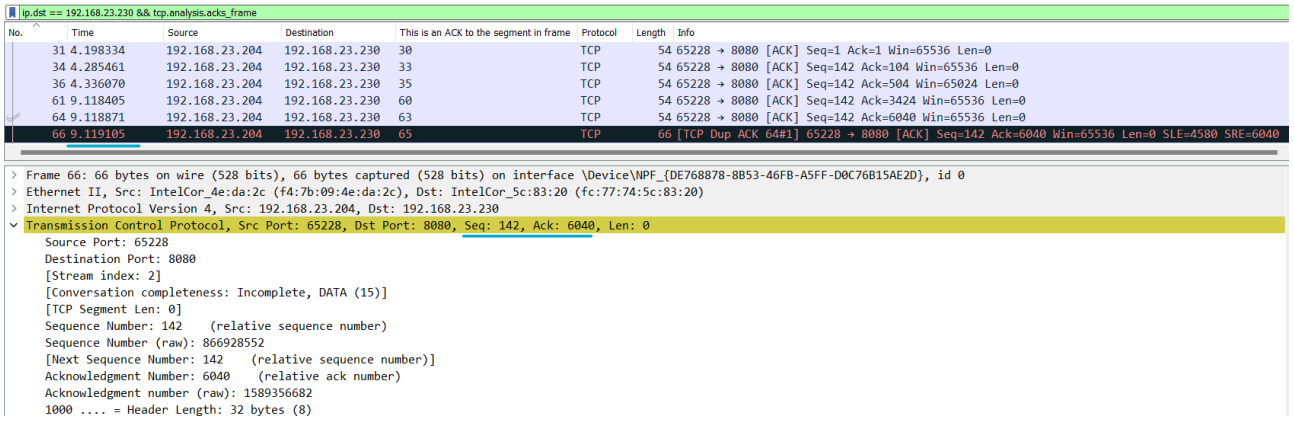


Hình 29: Gói tin phản hồi gói tin số 63

- **Segment 65**: Thời gian gói tin được gửi đi: 9.119047. Seq = 4580, Ack = 142.



ACK phản hồi của segment này là gói tin số 66. Thời điểm: 9.119105. Seq = 142, Ack = 6040.



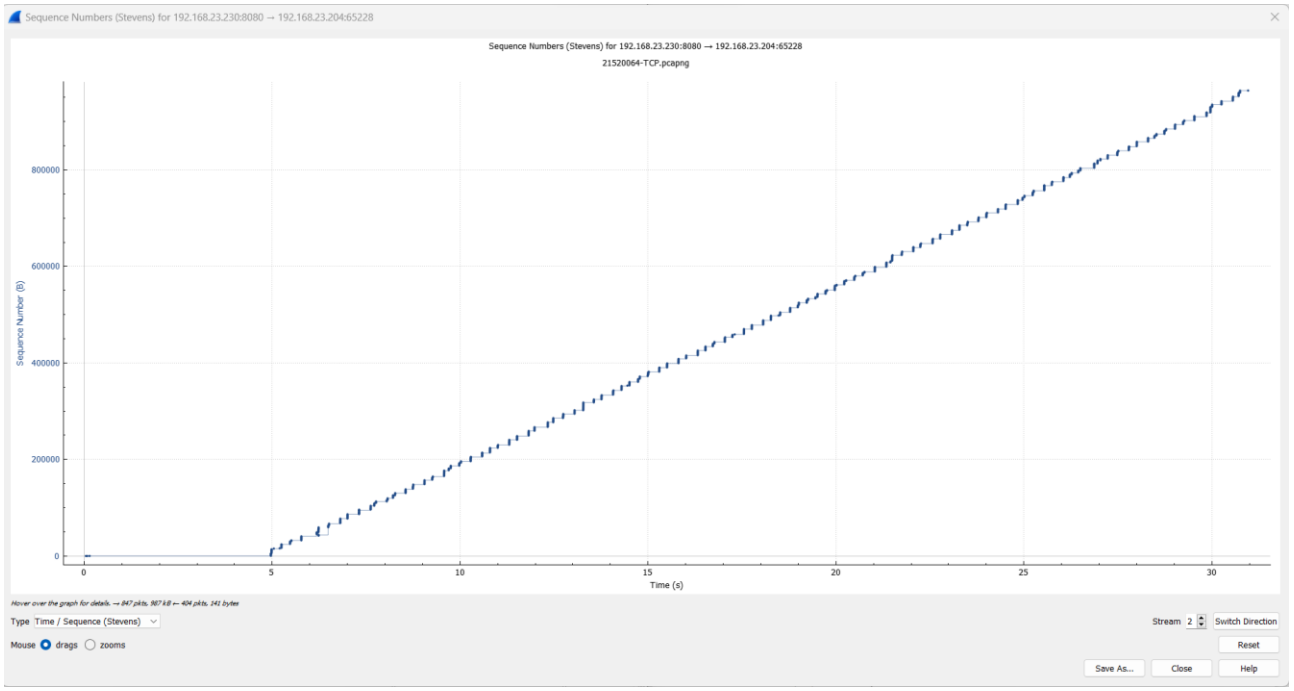
Hình 30: Gói tin phản hồi gói tin 65

No.	Thời gian gửi	Thời gian nhận ACK	RTT (s)	SEQ number	ACK number
30	4.198007	4.198334	0.000327	0	1
33	4.234682	4.285461	0.050779	1	142
35	4.295216	4.336070	0.040854	104	142
60	9.118295	9.118405	0.000110	1964	142
63	9.118843	9.118871	0.000028	4884	142
65	9.119047	9.119105	0.000121	4580	142

Bảng 1: Bảng thống kê và tính các giá trị thời gian

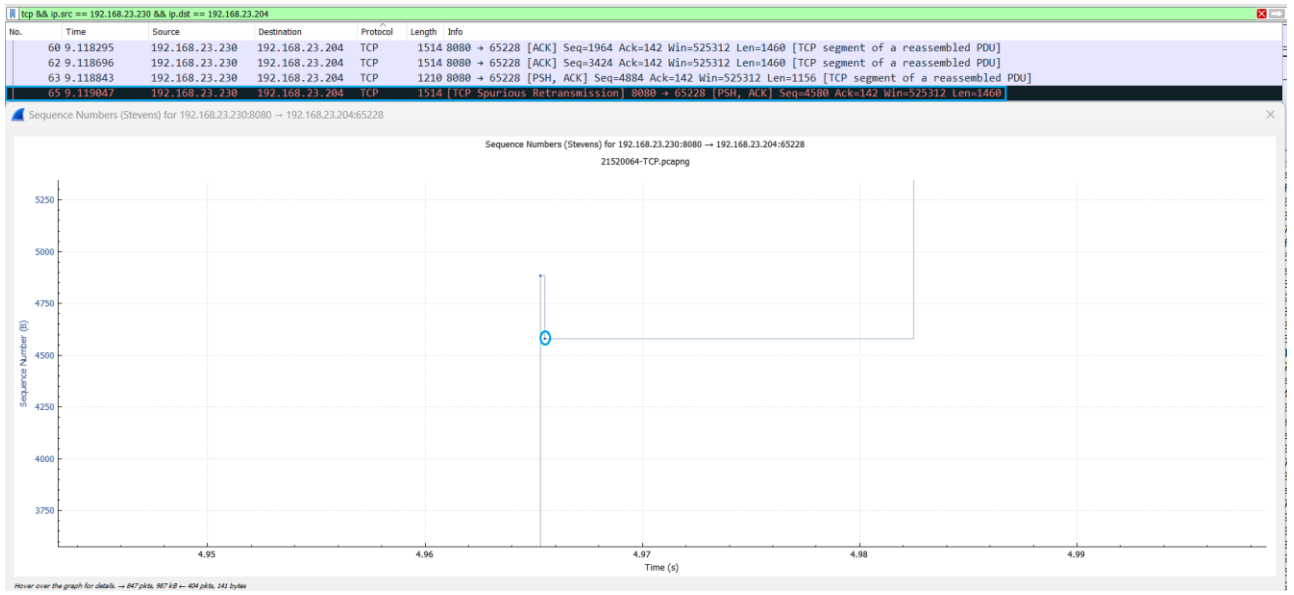
2.12. Có segment nào được gửi lại hay không? Thông tin nào trong quá trình truyền tin cho chúng ta biết điều đó? Giải thích.

- Để biết được rằng có segment nào được gửi lại hay không ta quan sát biểu đồ về Sequence Number như bên dưới:



Hình 31: Biểu đồ Sequence Number (Stevens)

- Segment được gửi lại là segment 65.



Hình 32: Hình ảnh một gói tin được gửi lại (được phóng ta từ biểu đồ trên)

- Ta biết nó được gửi lại vì trong biểu đồ trên, seq của gói tin 65 đột ngột giảm xuống. Mà ta biết rằng cùng một bên gửi, số sequence number của một segment sẽ được tính như sau:

Sequence number (current) = sequence number (liền trước) + độ dài của gói tin trước.

⇨ Sequence number ở cùng một bên gửi sẽ tăng dần. Tuy nhiên, ở gói tin 65 nó lại giảm so với gói tin trước. Do đó, ta có thể biết được rằng, đây là một gói tin được gửi lại từ một gói tin nào đó ở trên.