Họ và tên: Bùi Lê Nhật Tri

MSSV: 23521634

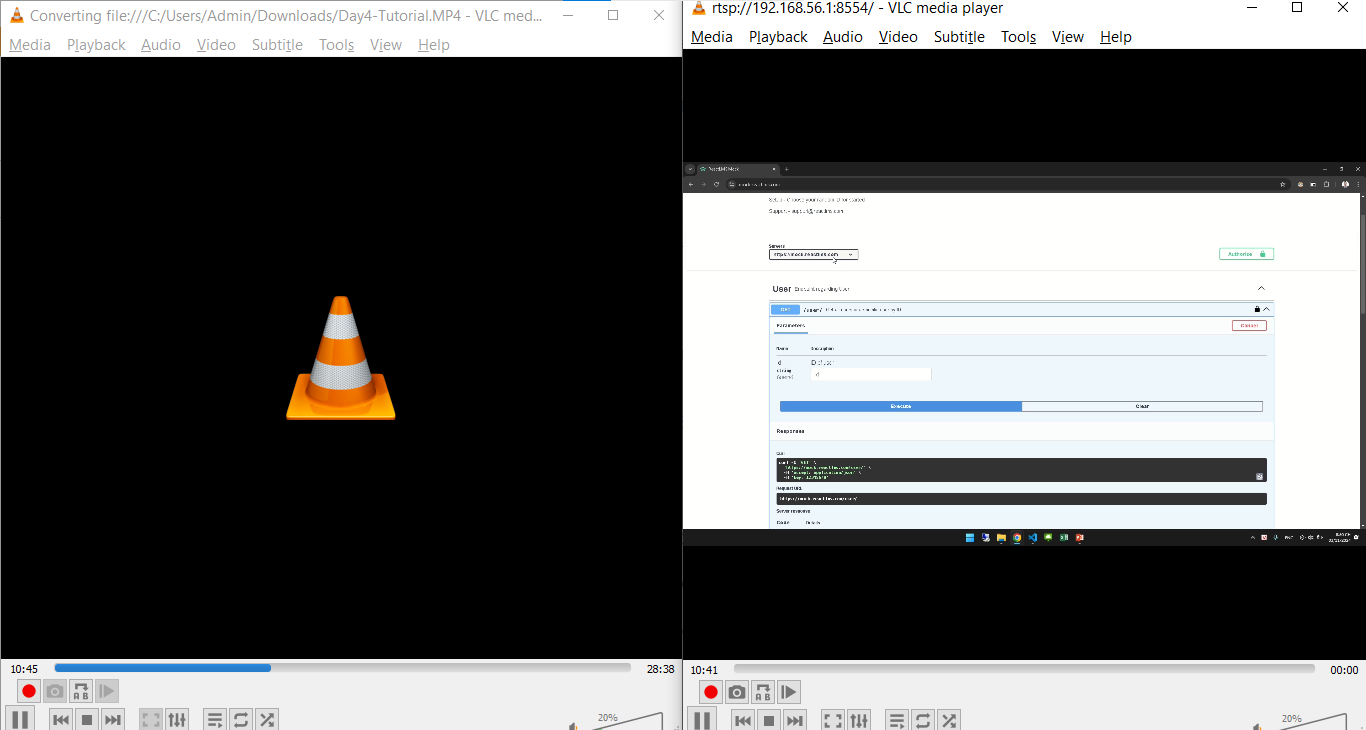
Lớp: IT005.P119

BÁO CÁO THỰC HÀNH LAB 3:

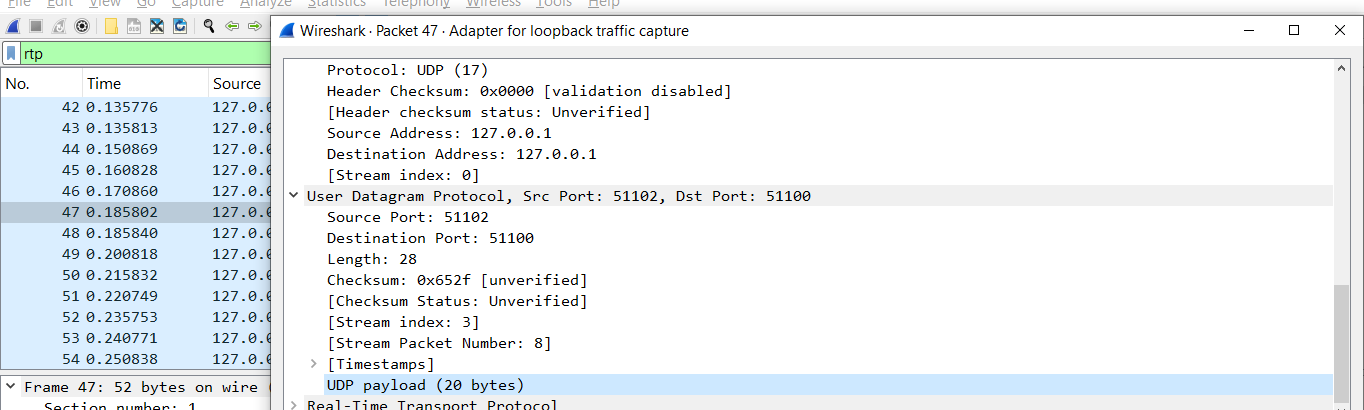
PHÂN TÍCH HOẠT ĐỘNG GIAO THỨC

TCP - UDP

Task 1: Phân tích hoạt động giao thức UDP



Câu 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?*



- Source port: Số hiệu cổng nơi đã gửi gói dữ liệu (datagram).

- Destination port: Số hiệu cổng nơi datagram được chuyển tới.

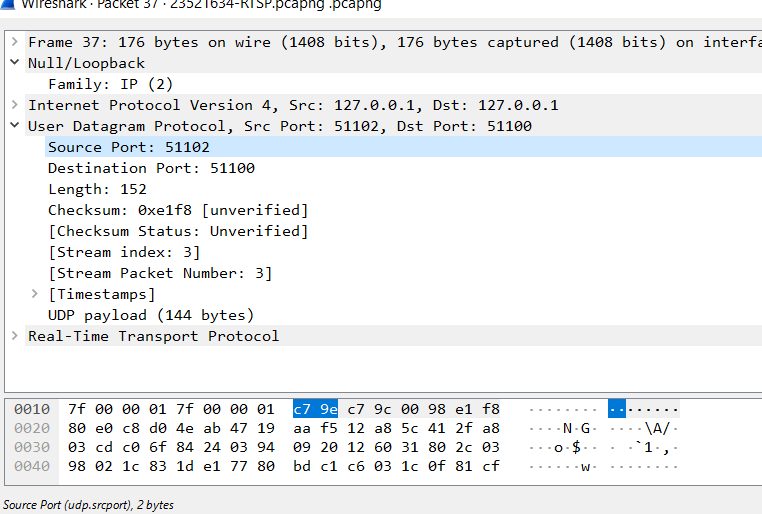
- Length: Độ dài tổng cộng kể cả phần header của gói UDP datagram.

- Checksum: Trường checksum dùng cho việc kiểm tra lỗi của phần header và dữ

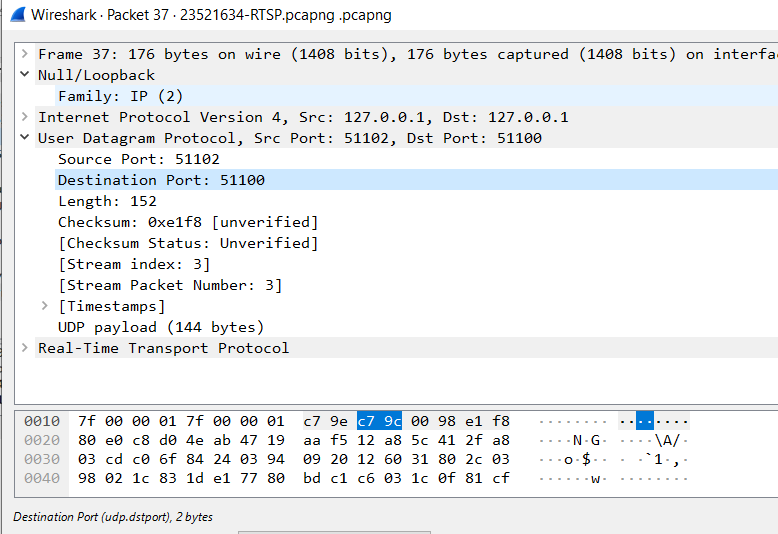
liệu, nếu phát hiện lỗi thì UDP datagram sẽ bị loại bỏ mà không có thông báo trả về nơi gửi.

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

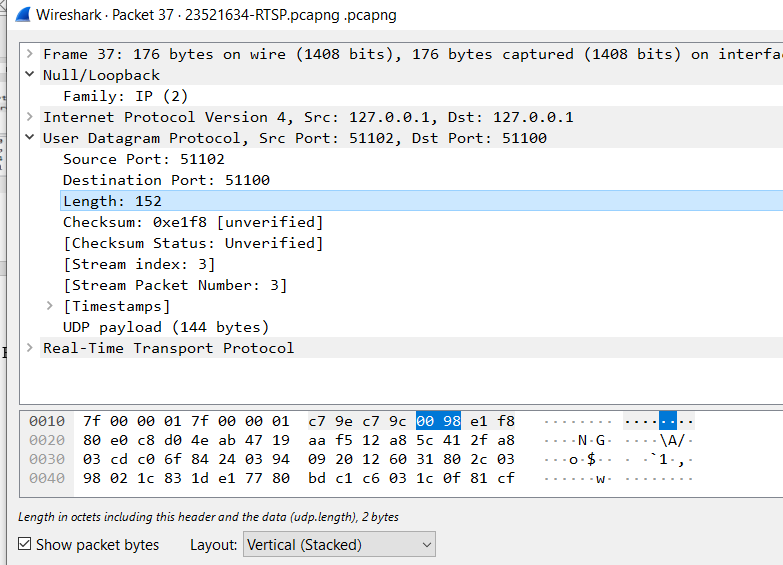
Độ dài của Source Port: 2 Bytes



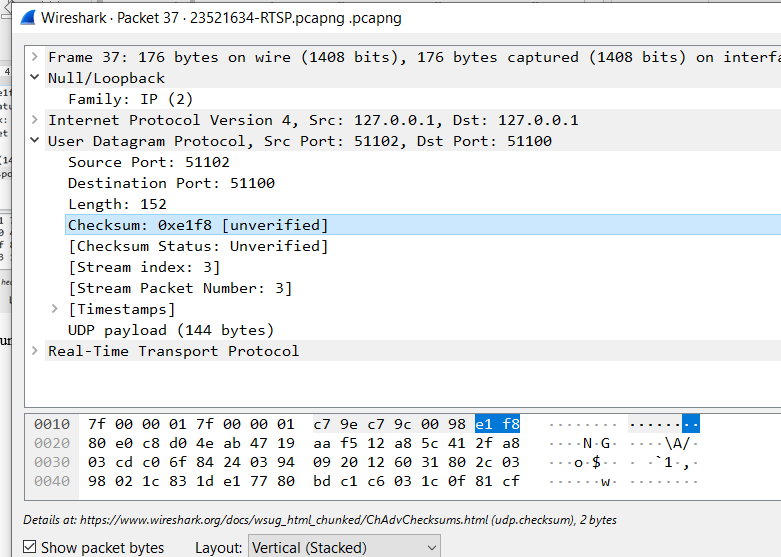
Độ dài của Destination Port: 2 Bytes



Độ dài của Length: 2 Bytes



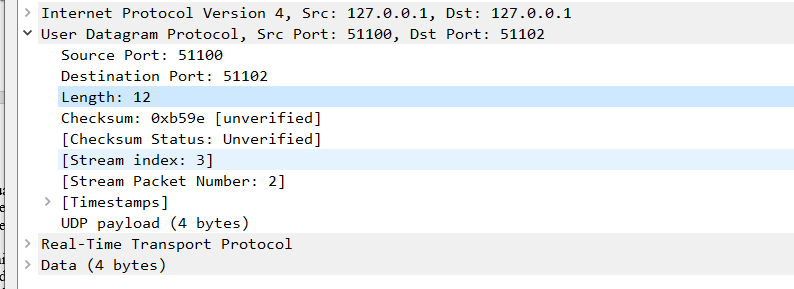
Độ dài củaChecksum: 2 Bytes



Câu 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 4 bytes. Do đó, ở trên ta thấy rằng trường Length trong UDP header là 4 + 8 = 12 bytes.



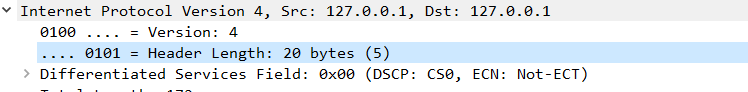
Câu 4:  *Số bytes lớn nhất mà payload (phần chứa dữ liệu gốc, không tính UDP header và IP header) của UDP có thể chứa?*

- Số bytes lớn nhất mà payload của UDP có thể chứa (tính luôn cả UDP header và IP header) là 216– 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.

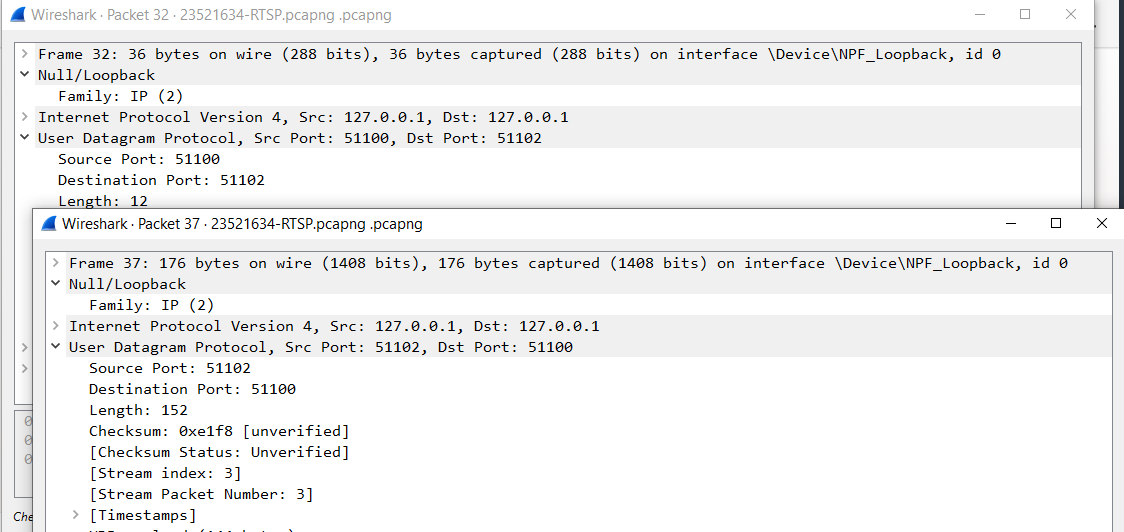
=> Data Length của ta là: 65535 – 8 – 20 = 65507 (bytes).



Câu 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 216– 1 = 65535.

Câu 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*



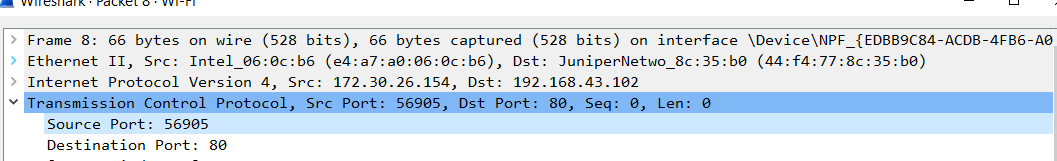
Mối quan hệ giữa port number của 2 gói tin này là: Khi ta gửi request đến máy khác, port của máy ta sẽ trở thành Source Port (19395), port máy bạn sẽ trở thành Destination Port (64045). Trong trường hợp ngược lại, khi máy bạn gửi phản hồi về request trên, port của máy ta sẽ trở thành Destination Port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

Task 2: Phân tích hoạt động giao thức TCP

Câu 7: *Tìm địa chỉ IP và TCP port của máy Client?*

IP Client: 172.30.26.154

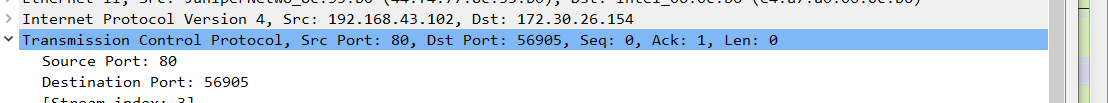
TCP Port Client: 56905

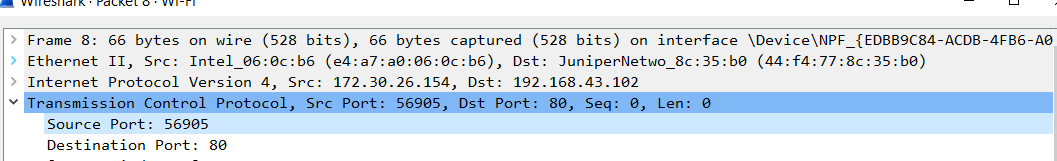


Câu 8: *Tìm địa chỉ IP của Server? Kết nối TCP dùng để gửi và nhận các segments sử dụng port nào?*

IP Server: 192.168.43.102

TCP dùng port 80 để cho gửi và nhận các segments

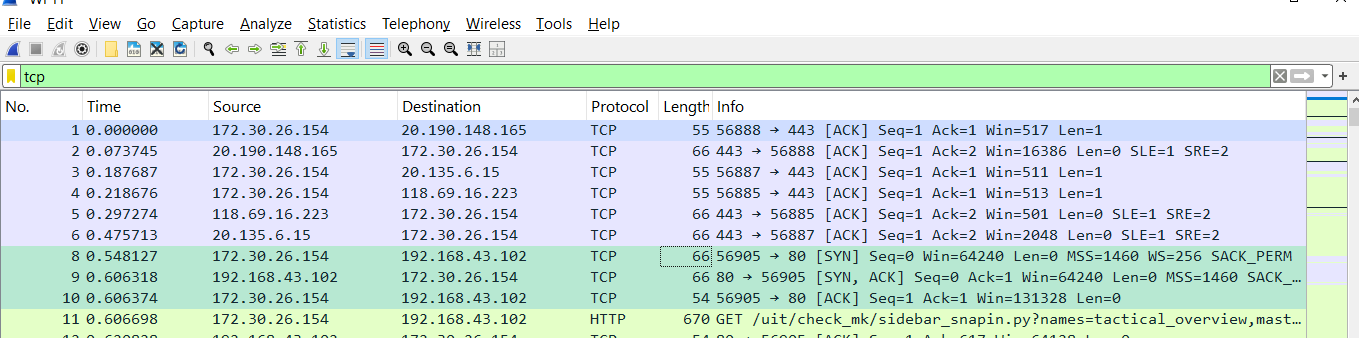


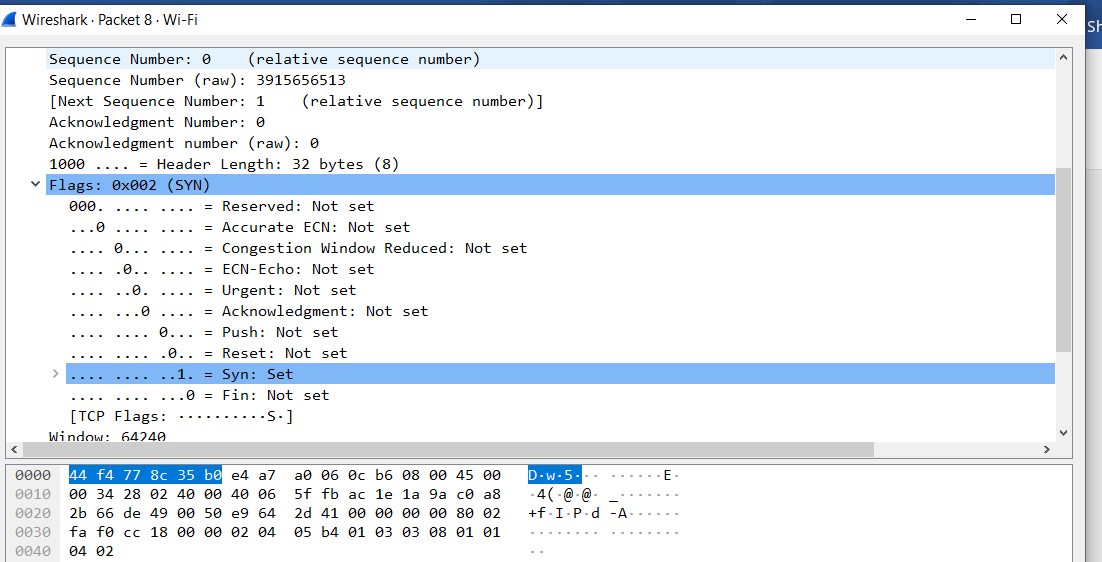


Câu 9: *TCP SYN segment (gói tin TCP có cờ SYN) sử dụng sequence number nào để khởi tạo kết nối TCP giữa client và server? Thành phần nào trong segment cho ta biết segment đó là TCP SYN segment?*

- Để gửi và nhận gói tin qua mạng, client và server cần phải khởi tạo kết nối với nhau thông qua 3 bước bắt tay:

* 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.





- 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.

Câu 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?*

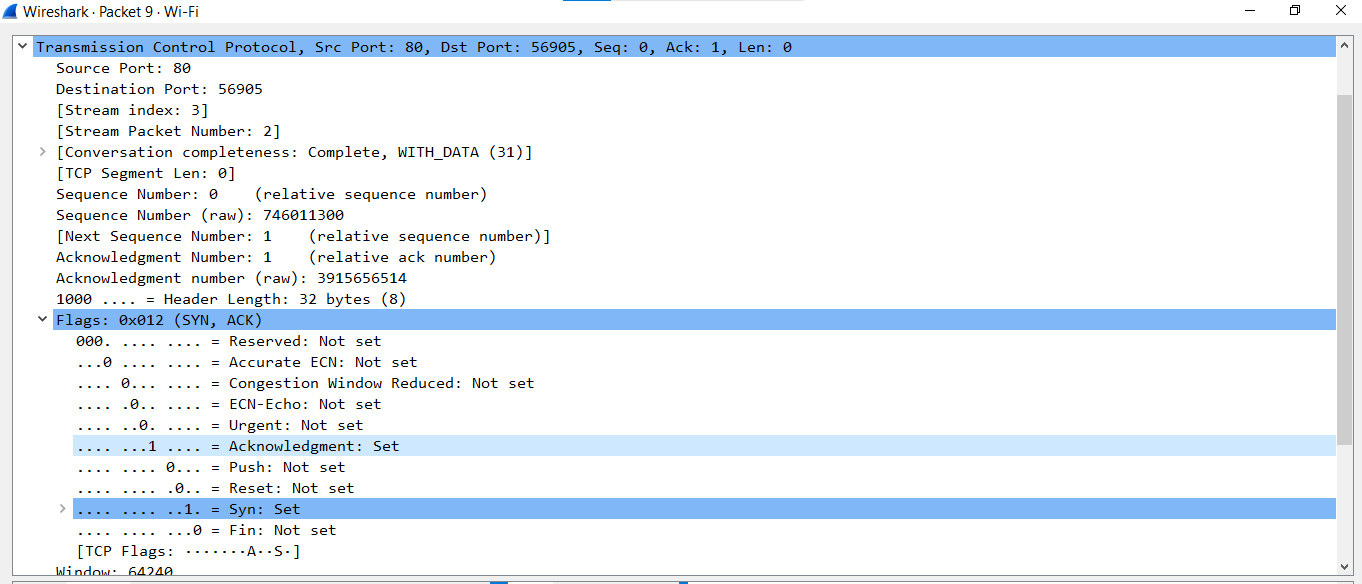
- Từ dưới trên, ta thấy rằng Sequence number của gói tin SYN/ACK segment được gửi bởi server đến client để trả lời cho SYN segment ở trên là 0

- Từ ảnh dưới, ta thấy rằng giá trị Acknowledgement trong SYN/ACK là 1.

- Cách server xác định được giá trị Acknowledgement:

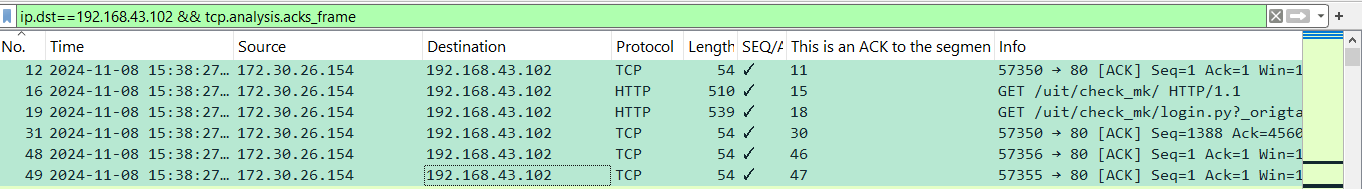
* Ban đầu, ở SYN, Client sẽ khởi tạo giá trị Sequence number của SYN là 0.
* Sau đó, ở bước thứ 2, giá trị Acknowledgement tính gói tin [SYN, ACK] sẽ được Server bằng giá trị Sequence + 1. Ở ví dụ này, nó sẽ giá trị Acknowledgement = 1. (Sequence number ở bước trước là 0 + 1).
* Tương tự ở gói tin [ACK] nó cũng được tính bằng giá trị Sequence ở bước trước + 1. Tức khi này giá trị ACK = 1.

- Ta thấy rằng, 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.

**

Câu 11: *Chỉ ra 6 segment đầu tiên mà server gửi cho Client (dựa vào Số thứ tự gói – No)  
- Tìm sequence number của 6 segments đầu tiên đó?  
- Xác định thời gian mà mỗi segment được gửi, thời gian ACK cho mỗi  
segment được nhận?  
- Đưa ra sự khác nhau giữa thời gian mà mỗi segment được gửi và thời gian ACK cho mỗi segment được nhận bằng cách tính RTT (Round Trip Time) cho 6 segments này?*

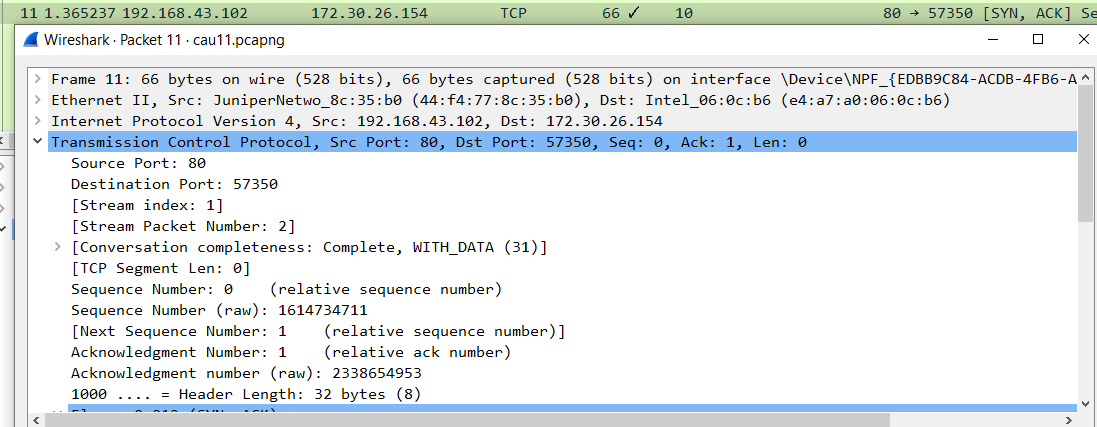
- Để biết được 6 segments đầu tiền mà Server gửi cho Client, ta sử dụng filter như hình sau (trong đó, *ip.dst* của ta chính là địa chỉ ip của Server, *tcp.analysis.acks\_frame* để lấy những gói tin TCP mà nó phản hồi lại gói tin ở trước nó): ip.dst==192.168.43.102 && tcp.analysis.acks\_frame



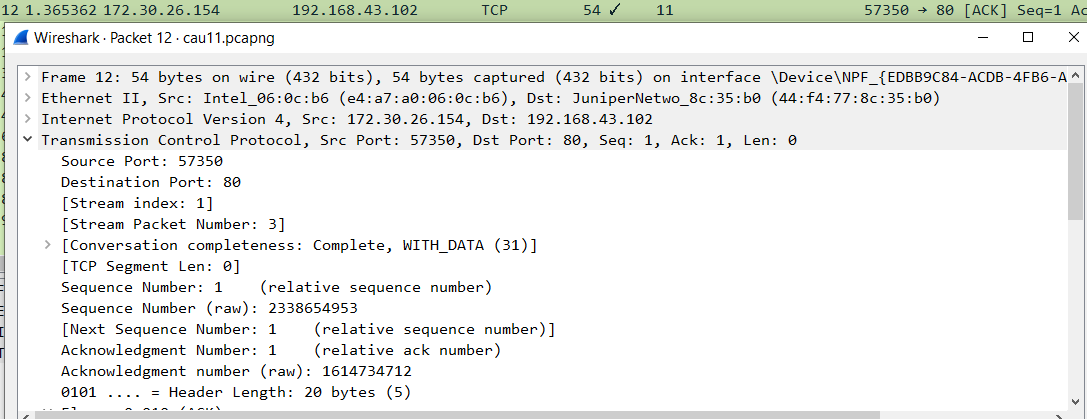
- Trong hình trên, trường This is an ACK to the segment in frame có ý nghĩa là đây là gói tin phải hồi cho gói tin nào. Ví dụ như gói tin số 12, trường đó là gói tin số 11, thì có nghĩa là gói tin số 12 là gói tin phản hồi của gói tin số 11. Và trường đó nó cũng chính là gói tin mà Server gửi về cho Client.

- Sau khi thực hiện lọc xong, ta suy ra được 6 gói tin đầu tiên Server gửi cho Client có ACK:

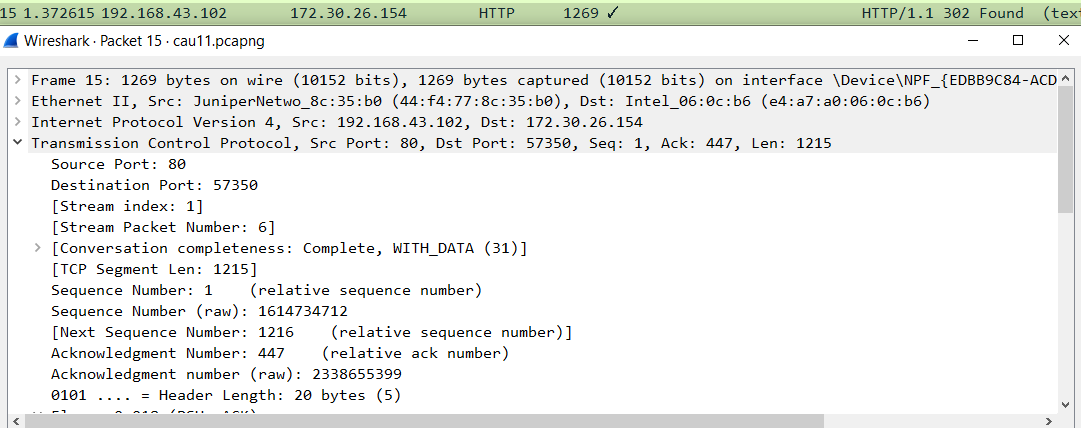
1. Segment 11: Thời gian segment được gửi là 1.365237. Seq = 0, Ack = 1.



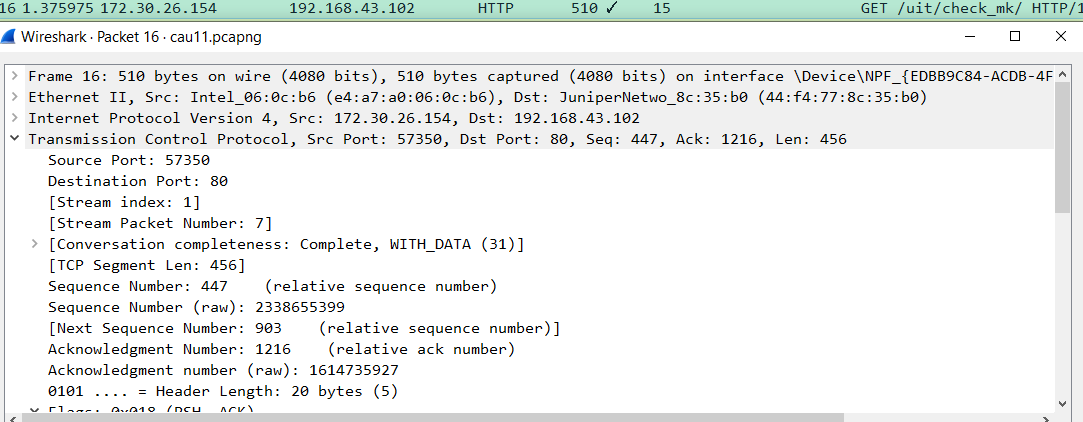
ACK phản hồi của segment này là gói tin số 12. Thời điểm: 1.365362. Seq = 1, Ack = 1.



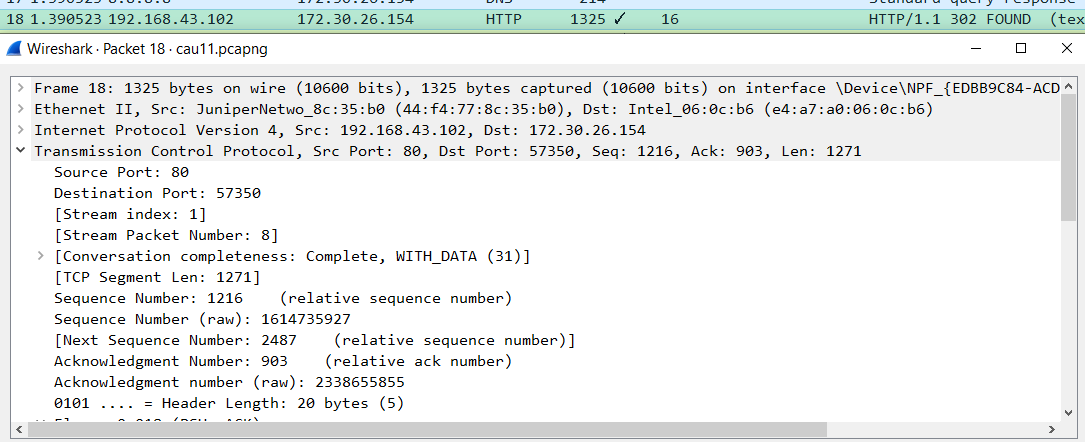
1. Segment 15: Thời gian segment được gửi là 1.372615. Seq = 1, Ack = 447.



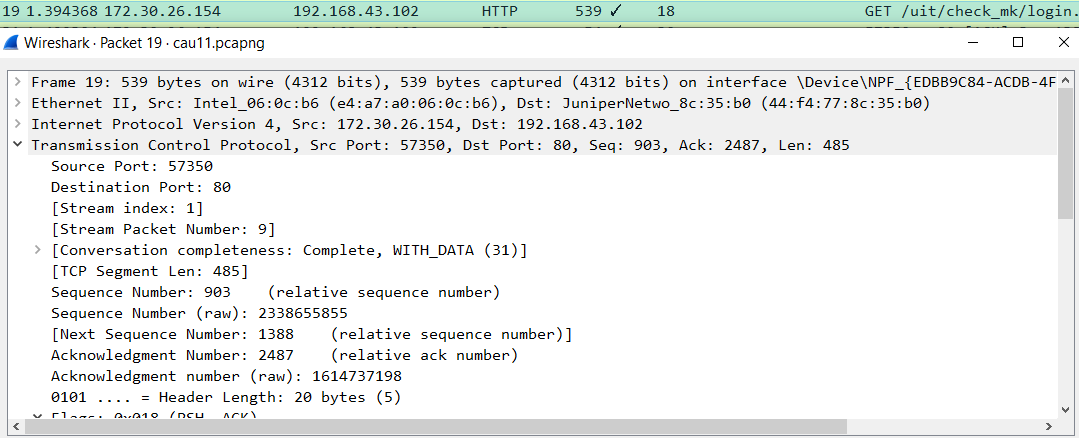
ACK phản hồi của segment này là gói tin số 16. Thời điểm: 1.375975. Seq = 447, Ack = 1216.



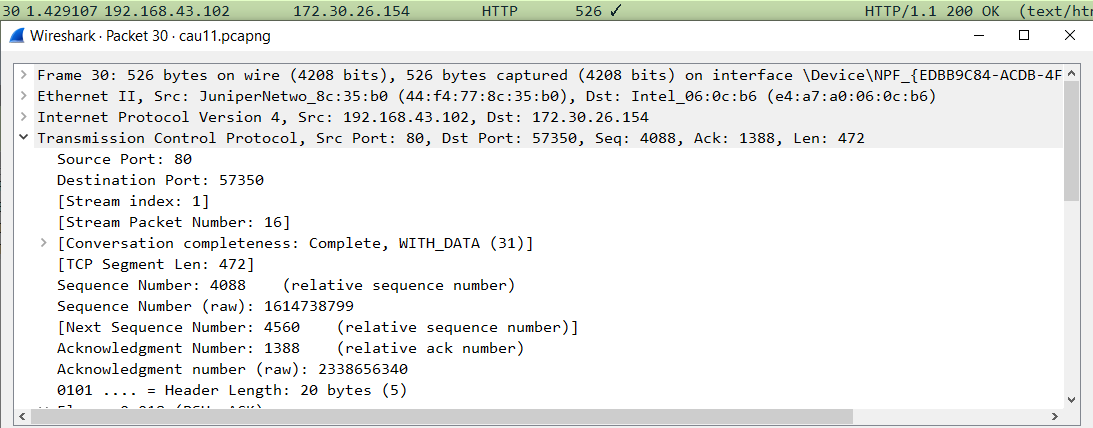
1. Segment 18: Thời gian segment được gửi là 1.390523. Seq = 1216, Ack = 903.



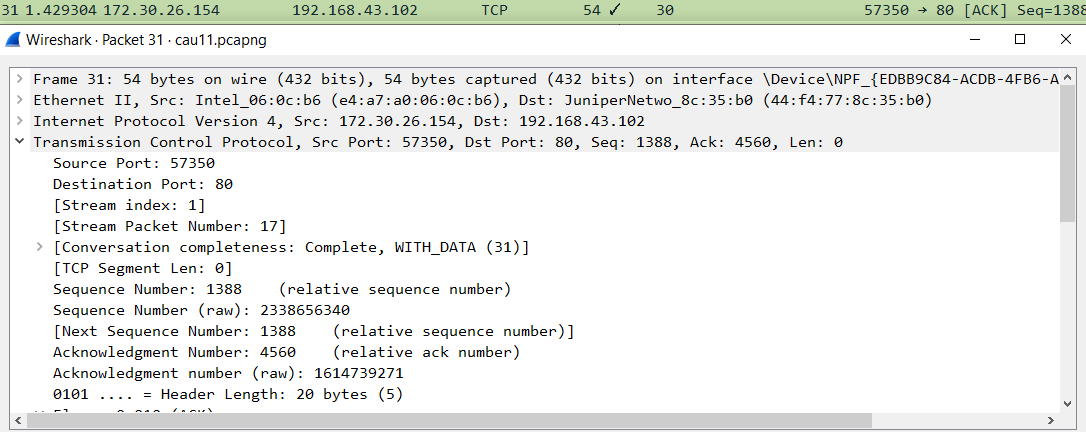
ACK phản hồi của segment này là gói tin số 19. Thời điểm: 1.394368. Seq = 903, Ack = 2487.



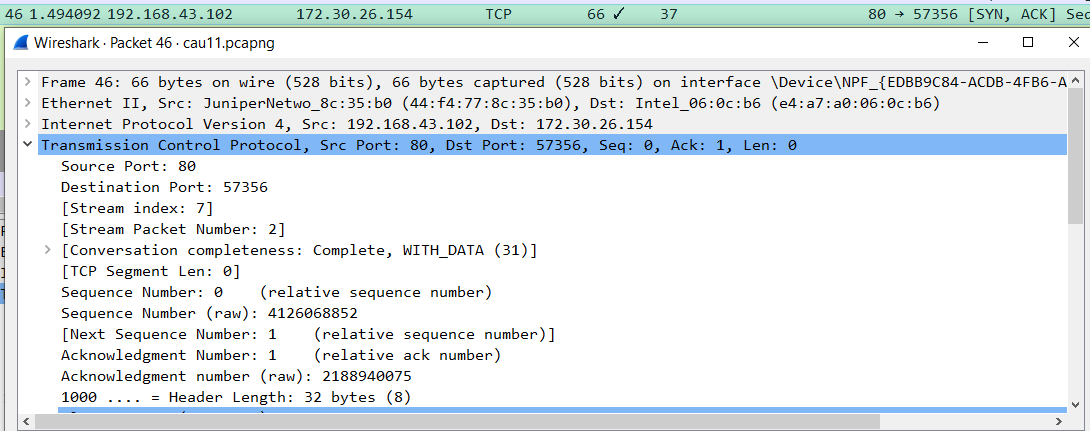
1. Segment 30: Thời gian segment được gửi là 1.429107. Seq = 4088, Ack = 1388.



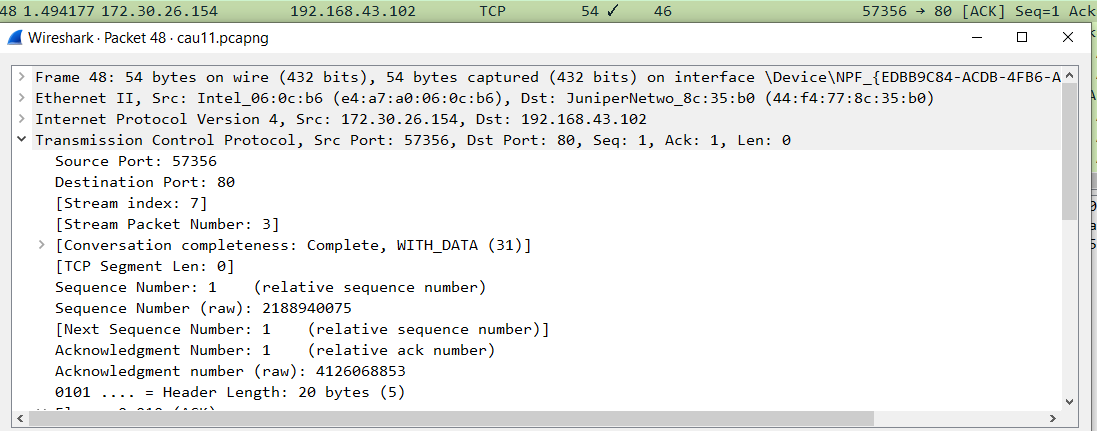
ACK phản hồi của segment này là gói tin số 31. Thời điểm: 1.429304. Seq = 1388, Ack = 4560.



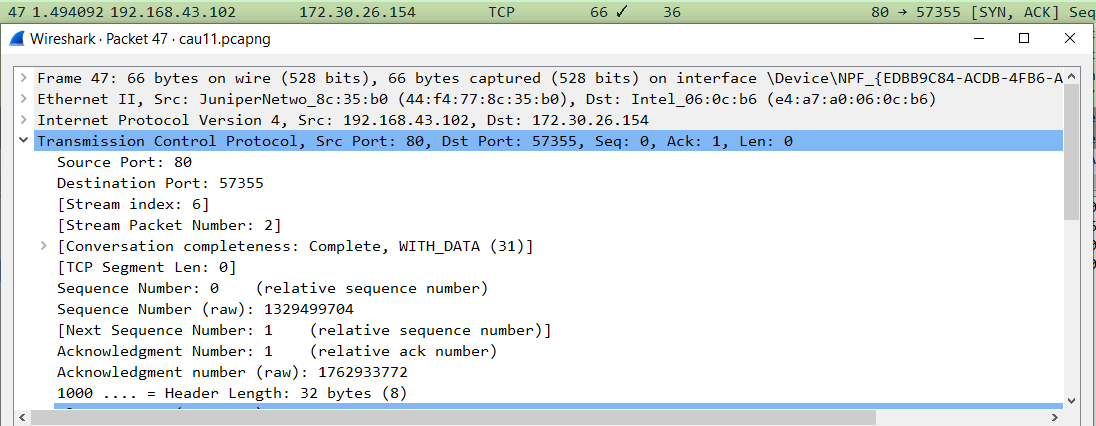
1. Segment 46: Thời gian segment được gửi là 1.494092. Seq = 0, Ack = 1.



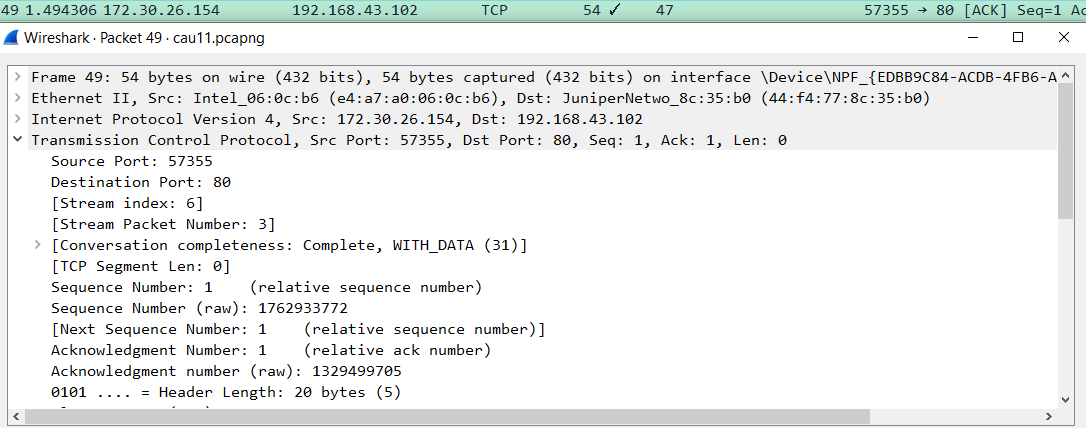
ACK phản hồi của segment này là gói tin số 48. Thời điểm: 1.494177. Seq = 1, Ack = 1.



1. Segment 47: Thời gian segment được gửi là 1.494092. Seq = 0, Ack = 1.



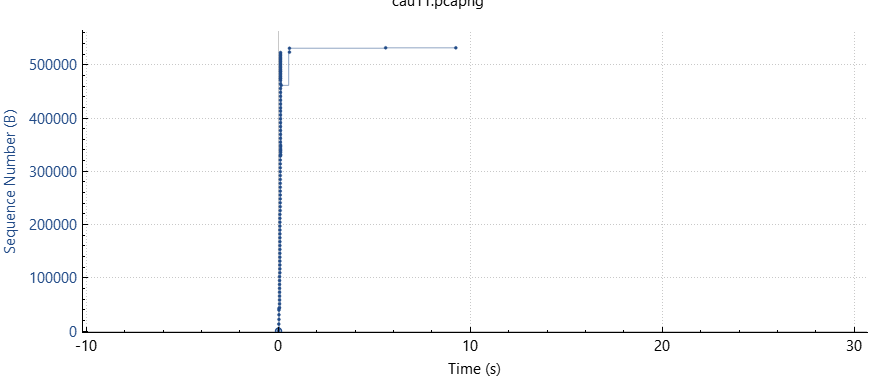
ACK phản hồi của segment này là gói tin số 49. Thời điểm: 1.494306. Seq = 1, Ack = 1.



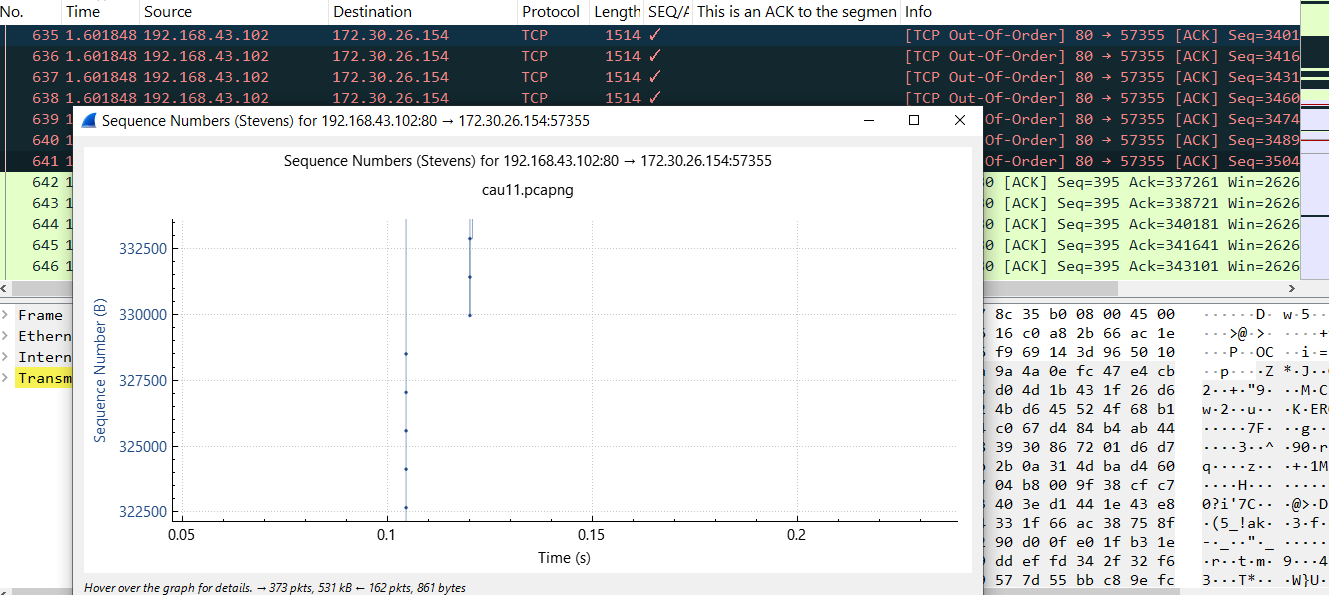
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Thời gian gửi | Thời gian nhận ACK | RTT (s) | SEQ number | ACK number |
| 1 | 1.365237 | 1.365362 | 0.000125 | 0 | 1 |
| 15 | 1.372615 | 1.375975 | 0.003360 | 1 | 447 |
| 18 | 1.390523 | 1.394368 | 0.003845 | 1216 | 903 |
| 30 | 1.429107 | 1.429304 | 0.000197 | 4088 | 1388 |
| 46 | 1.494092 | 1.494177 | 0.000085 | 0 | 1 |
| 47 | 1.494092 | 1.494306 | 0.000214 | 0 | 1 |

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

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



Dễ thấy Segment được gửi lại là segment 635.



* Ta biết nó được gửi lại vì trong biểu đồ trên, seq của gói tin 635 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 635 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.