1. **Chọn một gói tin UDP, xác định các trường (field) có trong UDP header và giải thích ý nghĩa của mỗi trường đó? Gợi ý: Xem tại phần User Datagram Protocol.**

- Có 4 trường

+ **Source Port**: Trường này xác định cổng của người gửi thông tin và có ý nghĩa nếu muốn nhận thông tin phản hồi từ người nhận. Nếu không dùng đến thì đặt nó bằng 0.

+ **Destination port :** Trường xác định cổng nhận thông tin, và trường này là cần thiết

+ **Length:** Trường có độ dài 16 bit xác định chiều dài của toàn bộ datagram: phần header và dữ liệu. Chiều dài tối thiểu là 8 byte khi gói tin không có dữ liệu, chỉ có header.

+ **Checksum:** Trường checksum 16 bit dùng cho việc kiểm tra lỗi của phần header *và dữ liệu*.

```
> Frame 2: 158 bytes on wire (1264 bits), 158 bytes captured (1264 bits) on interface \Device\NPF_
> Ethernet II, Src: IntelCor_be:36:bf (64:bc:58:be:36:bf), Dst: DASANNet_0c:34:6f (d0:96:fb:0c:34
> Internet Protocol Version 4, Src: 192.168.1.9, Dst: 52.112.40.26
v User Datagram Protocol, Src Port: 50020, Dst Port: 3480
    Source Port: 50020
    Destination Port: 3480
    Length: 124
    Checksum: 0x1ec9 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
  > [Timestamps]
    UDP payload (116 bytes)
> Data (116 bytes)
```

( Trích : https://vi.wikipedia.org/wiki/UDP#C%E1%BB%95ng )

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

- Source Port: 2 bytes
- Destination Port: 2 bytes
- Length: 2 bytes
- Checksum: 2 bytes

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 được tính bằng byte của segment UDP, bao gồm cả header.

```
    Length: 124
    Checksum: 0x1ec9 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
  > [Timestamps]
    UDP payload (116 bytes)
v Data (116 bytes)
    Data: ff1000703b16c438623ffc5100
    [Length: 116]
```

- Giá trị của trường length là 124 là bao gồm 8 bytes của các trường trong header và 116 bytes của Data( segment UDP)

4. **Số bytes lớn nhất mà payload (phần chứa dữ liệu gốc, không tính UDP header) của UDP có thể chứa? Gợi ý: Dựa vào kích thước của trường Length trong UDP header và giá trị lớn nhất có thể thể hiện.**

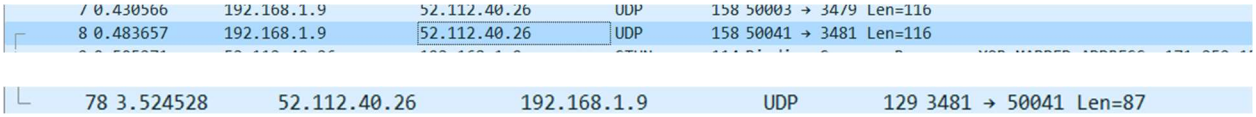
- Kích thước trường length là 2 bytes tương ứng có thể chứa $2^{16}$-1=65535 bytes vậy phần payload chứa 65535-8=65527 ( 8 bytes của phần header)

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

- Gía trị lớn nhất của port nguồn là 65535 ( $2^{16} – 1$)

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. Gợi ý: Có thể bắt gói tin UDP ở một tình huống khác để tìm được 1 cặp gói tin như trên.**

- cặp gói tin 8 và 78




Quan hệ:

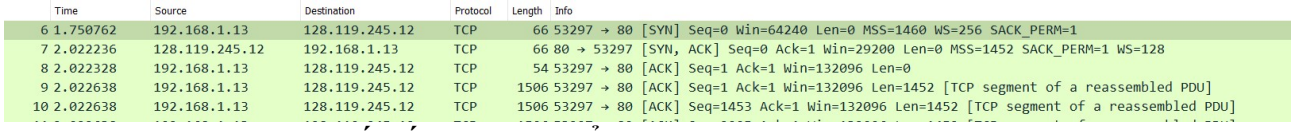
Gói 8 là từ port nguồn (từ máy) gửi đi với source port là 50041 đến destination port ( port đích ) là 3481

Còn gói 78 là từ port đích phản với destination port là 3481 đến source port ( port nguồn ) là 50041

# TCP

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

- Địa chỉ IP của máy là 192.168.1.13
- TCP port là 53297




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

- Địa chỉ IP của Server là 128.119.245.12
- Sử dụng TCP port là 80

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.**

- TCP SYN segment sử dụng sequence number là 1 số ngẫu nhiên , trong trường hợp này là: 3174299413

```
Transmission Control Protocol, Src Port: 53297, Dst Port: 80, Seq: 0, Len:
    Source Port: 53297
    Destination Port: 80
    [Stream index: 1]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 3174299413
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    1000 .... = Header Length: 32 bytes (8)
```

```
    Acknowledgment number (raw): 0
    1000 .... = Header Length: 32 bytes (8)
  Flags: 0x002 (SYN)
    000. .... .... = Reserved: Not set
    ...0 .... .... = Nonce: Not set
    .... 0... .... = Congestion Window Reduced (CWR): Not set
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
    .... ...0 .... = Acknowledgment: Not set
    .... .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
  > .... .... ..1. = Syn: Set
    .... .... ...0 = Fin: Not set
```

- Trong FLAG cờ Syn đang được bật ( 1 )

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?**

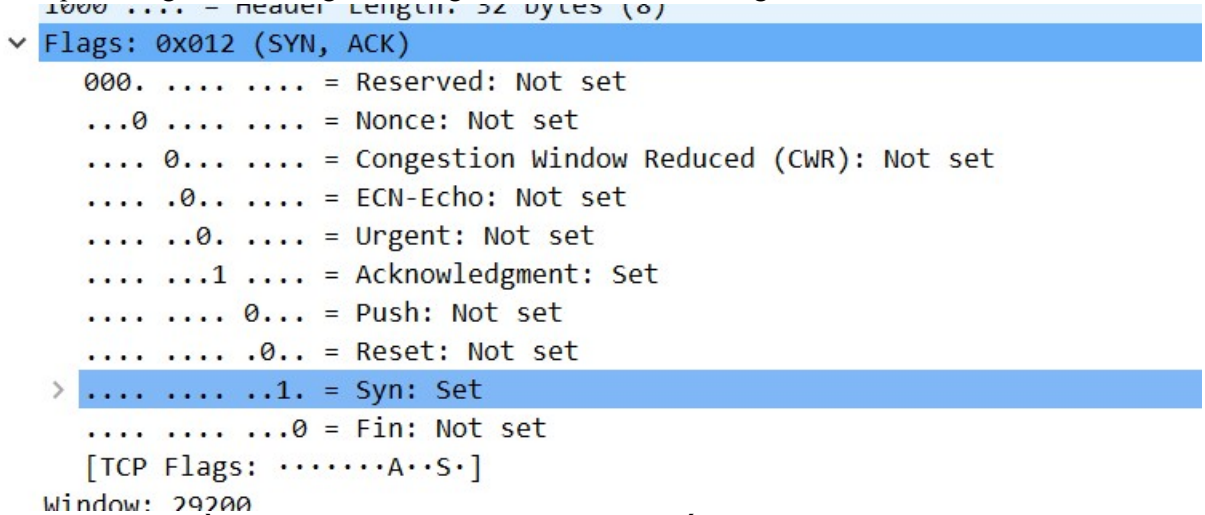
- Sequece number của gói SYN/ACK là segment là 32759905
- Gía trị của Acknowledgement trong SYN/ACK segment là 3174299414

```
Source Port: 80
Destination Port: 53297
[Stream index: 1]
[TCP Segment Len: 0]
Sequence Number: 0    (relative sequence number)
Sequence Number (raw): 32759905
[Next Sequence Number: 1    (relative sequence number)]
Acknowledgment Number: 1    (relative ack number)
Acknowledgment number (raw): 3174299414
1000 .... = Header Length: 32 bytes (8)
> Flags: 0x012 (SYN, ACK)
Window: 29200
[Calculated window size: 29200]
```

- Ta dễ thấy rằng sequence number của TCP Syn là 3174299413 là 1 số ngẫu nhiên để tạo kết nối , khi nhận được gói tin có seq là 3174299413 thì server sẽ phản hồi lại gói tin có

Acknowledgment có ACKnum là 3174299414 ( thể hiện vị trí mong đợi tiếp theo ) và trong trường hợp này chính xác là 3174299414 (3174299413+ 1 )

- Thành phần Flag sẽ cho chúng ta biết segment đó là SYN/ACK segment

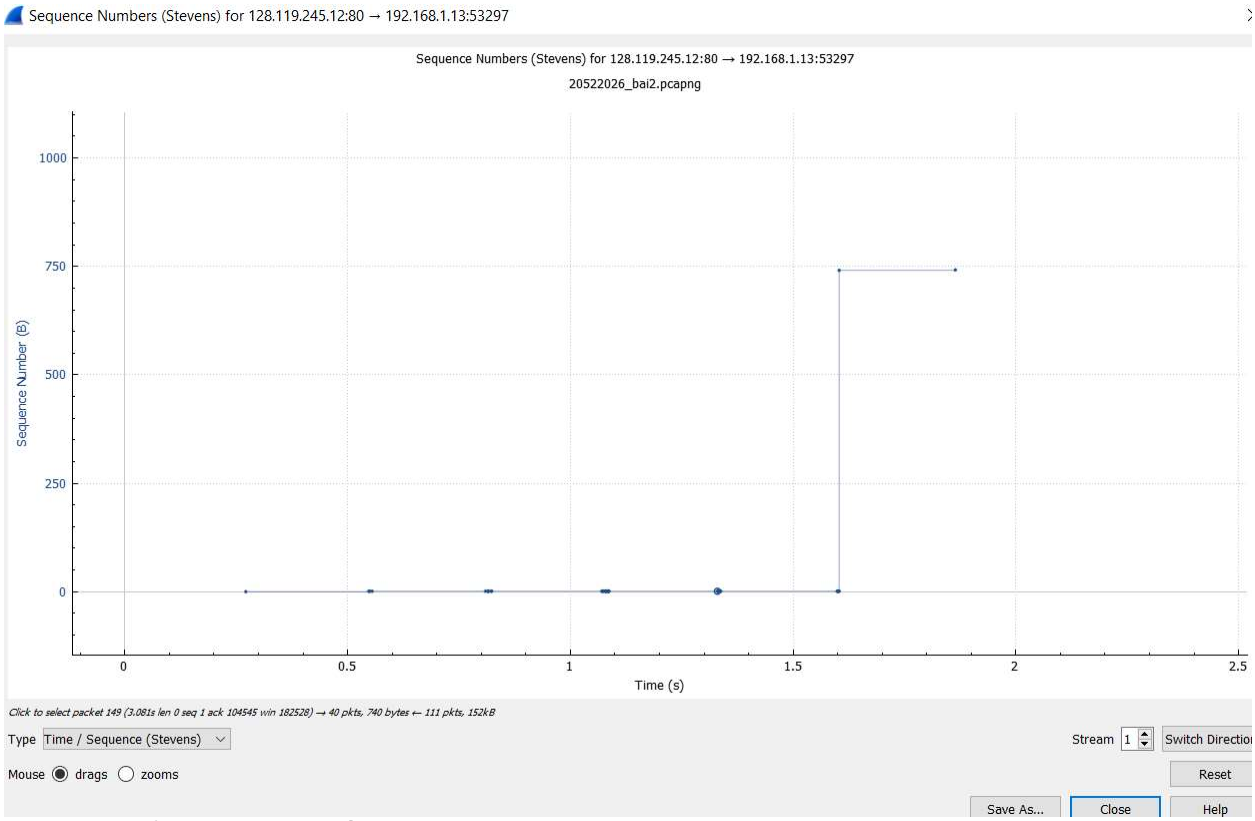



**11. Chỉ ra 6 segment đầu tiên mà Client gửi cho sẻ(dựa vào Số thứ tự gói – No) - Tìm sequence number của 6 segments đầu tiên đó? - Xác định thời gian mà mỗi segment được gửi, thời gian ACK cho mỗi segment được nhận? - Đưa ra sự khác nhau giữa thời gian mà mỗi segment được gửi và thời gian ACK cho mỗi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

| STT | Sequence number | Thời gian gửi | Thời gian nhận ACK | RTT |
|---|---|---|---|---|
| 12 | 3174303770 | 2.022638 | 2.298188 | 0.27555(s) |
| 15 | 3174308126 | 2.022638 | 2.300575 | 0.277937(s) |
| 18 | 3174312482 | 2.022638 | 2.305771 | 0.283133(s) |
| 24 | 3174319742 | 2.298260 | 2.560272 | 0.262012(s) |
| 26 | 3174322646 | 2.298260 | 2.565669 | 0.267409(s) |
| 31 | 3174328454 | 2.300628 | 2.566626 | 0.265998(s) |

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

- Không có gói tin nào phải gửi lại

- 

- Không có điểm nào bị rớt xuống chứng tỏ cho không có gói tin nào phải gửi lại