

# 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: Nguyễn Văn Bảo

Ngày thực hiện: 27/10/2025

## THÔNG TIN CHUNG:

Lớp: <IT005.Q15.2>

STT	Họ và tên	MSSV	Email
1	Nguyễn Khoa Nguyên	24521190	24521190@gm.uit.edu.vn

## 1. ĐÁNH GIÁ KHÁC:

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

# BÁO CÁO CHI TIẾT

## 1. Task 1: Phân tích tổng quan giao thức TCP và UDP

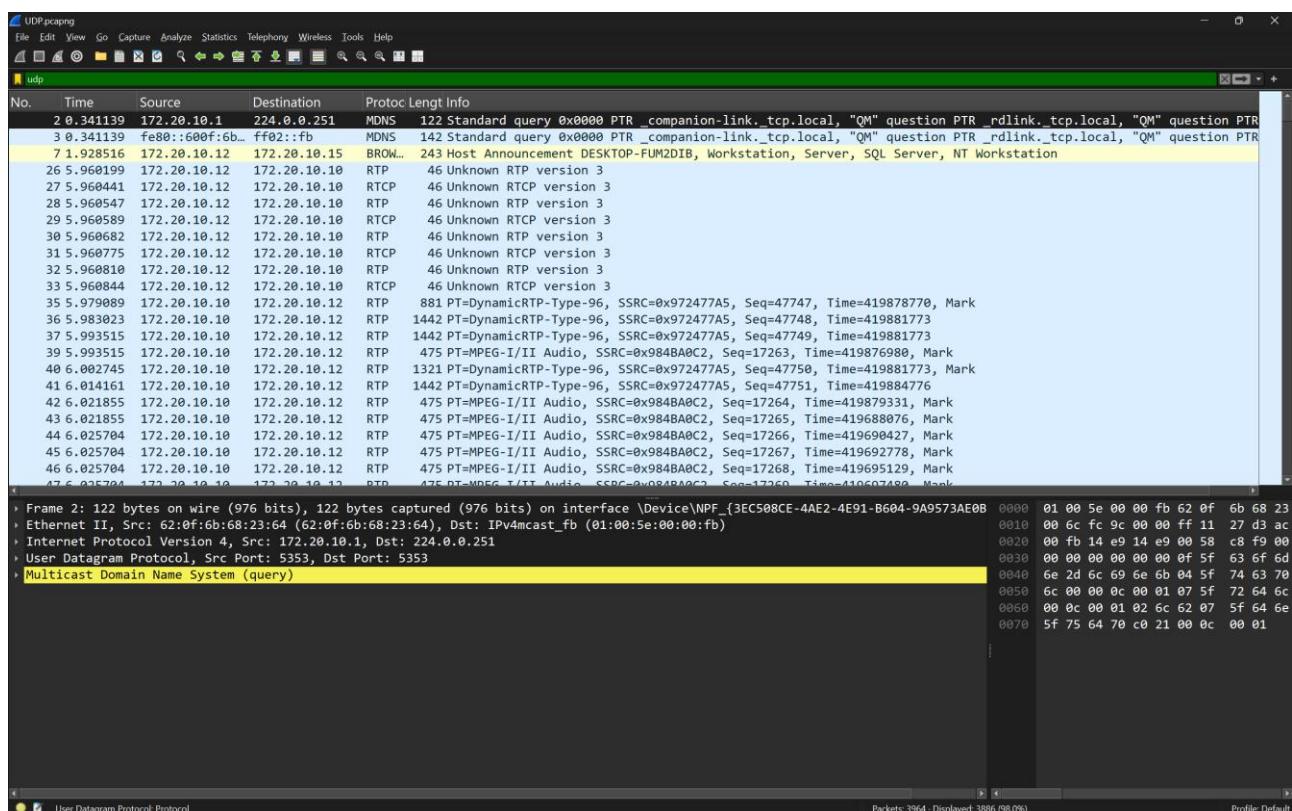
### 1.1. Phân tích, bắt gói tin khi xem streaming video lần 1

#### a) Tìm hiểu 1: Mô tả kết quả quan sát được khi xem Video khi xem streaming từ Server?

- Bởi vì sử dụng giao thức UDP – giao thức vận chuyển không tin cậy và gửi gói tin theo cơ chế best-effort, do đó khi streaming video, giao thức UDP sẽ không quan trọng có mất nội dung, gói tin hay không và cứ truyền bằng tất cả những gì có thể. Do đó khi coi video streaming ta sẽ thấy có những đoạn khung hình bị nhiễu và hầu như không thể thấy được nội dung nhưng video vẫn chạy và không dừng lại vì đây là cơ chế của giao thức TCP

#### b) Tìm hiểu 2: Tìm trong file Wireshark thu được, quá trình streaming ở kịch bản này sử dụng giao thức UDP hay TCP ?

- Quá trình streaming ở kịch bản này sử dụng giao thức UDP



- Server phát video có địa chỉ IP là 172.20.10.10 và client thực hiện coi video streaming có địa chỉ IP là 172.20.10.12. Hình trên là quá trình trao đổi gói tin giữa client và server khi coi video streaming

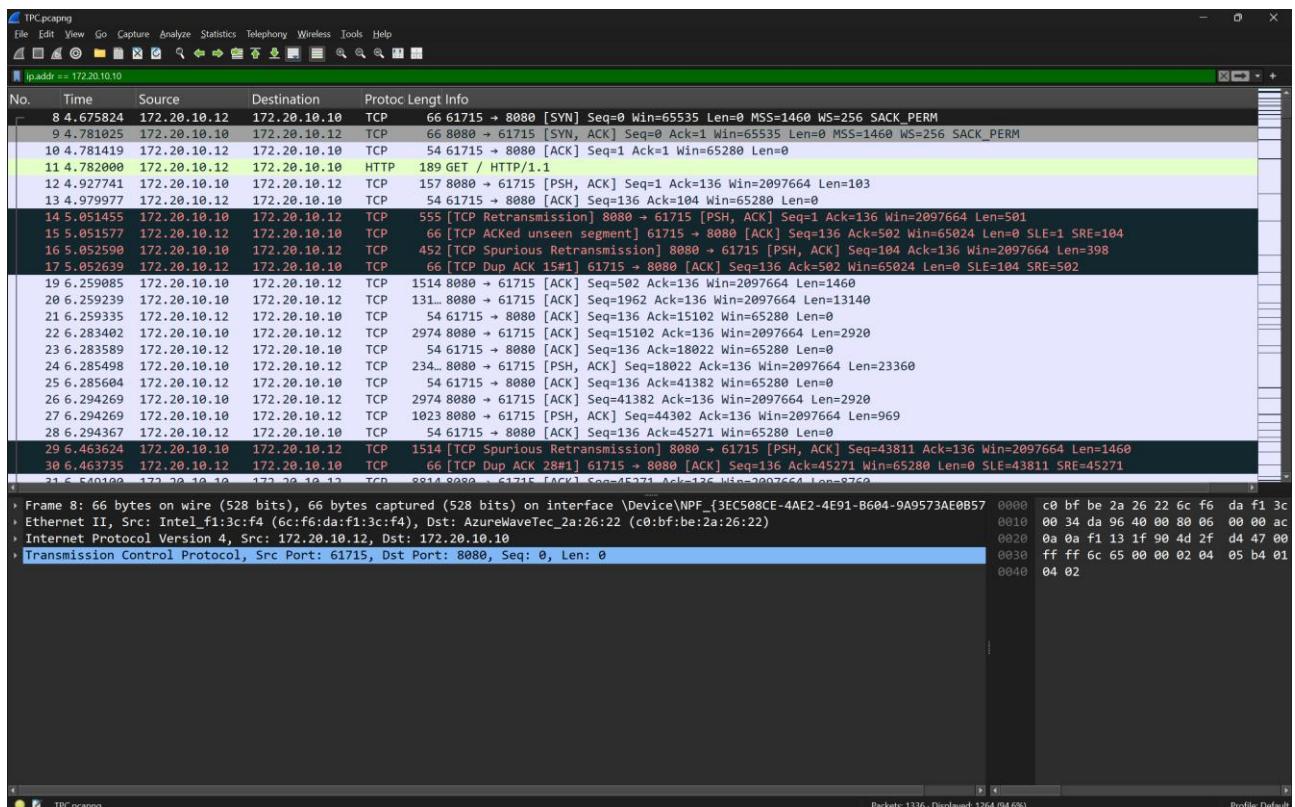
### 1.2. Phân tích, bắt gói tin khi xem streaming video lần 2

#### c) Tìm hiểu 3: Mô tả kết quả quan sát được khi xem Video streaming từ server ?

- Ở trường hợp này, bởi vì sử dụng giao thức TCP nên video sẽ mất một khoảng thời gian ban đầu lâu hơn khi ta coi lần 1 để thực hiện các quy trình của giao thức TCP gồm bắt tay 3 bước,... Trong quá trình coi thì sẽ có những đoạn khung lại khá lâu nhưng vẫn đảm bảo khung hình không bị mờ, nhòe nét, không thể thấy được nội dung vì đây chính là cơ chế của TCP

### **d) Tìm hiểu 4: Tìm trong file Wireshark thu được, quá trình streaming ở kịch bản này sử dụng giao thức UDP hay TCP?**

- Quá trình streaming ở kịch bản này sử dụng giao thức TCP



- Server phát video có địa chỉ IP là 172.20.10.10 và client thực hiện coi video streaming có địa chỉ IP là 172.20.10.12. Hình trên là quá trình trao đổi gói tin giữa client và server khi coi video streaming

### **e) Tìm hiểu 5: Phân tích sự khác nhau giữa 2 lần xem streaming Video.**

- Lần 1: Video không có độ trễ mà liên tục được phát, nhưng trong lúc xem sẽ có những khung hình bị nhòe, mờ và không thể thấy nội dung mặc dù video không dừng lại
- Lần 2: Video có độ trễ, mất một khoảng thời gian ban đầu để load, sau đó video khi phát sẽ không có những khung hình bị nhòe, mờ như lần 1. Nhưng lúc này video sẽ có những lần bị dừng hẳn lại và không chạy nữa để load

## 2. Task 2: Phân tích hoạt động giao thức UDP

### 2.1. Thực hiện truy vấn DNS và bắt các gói tin UDP

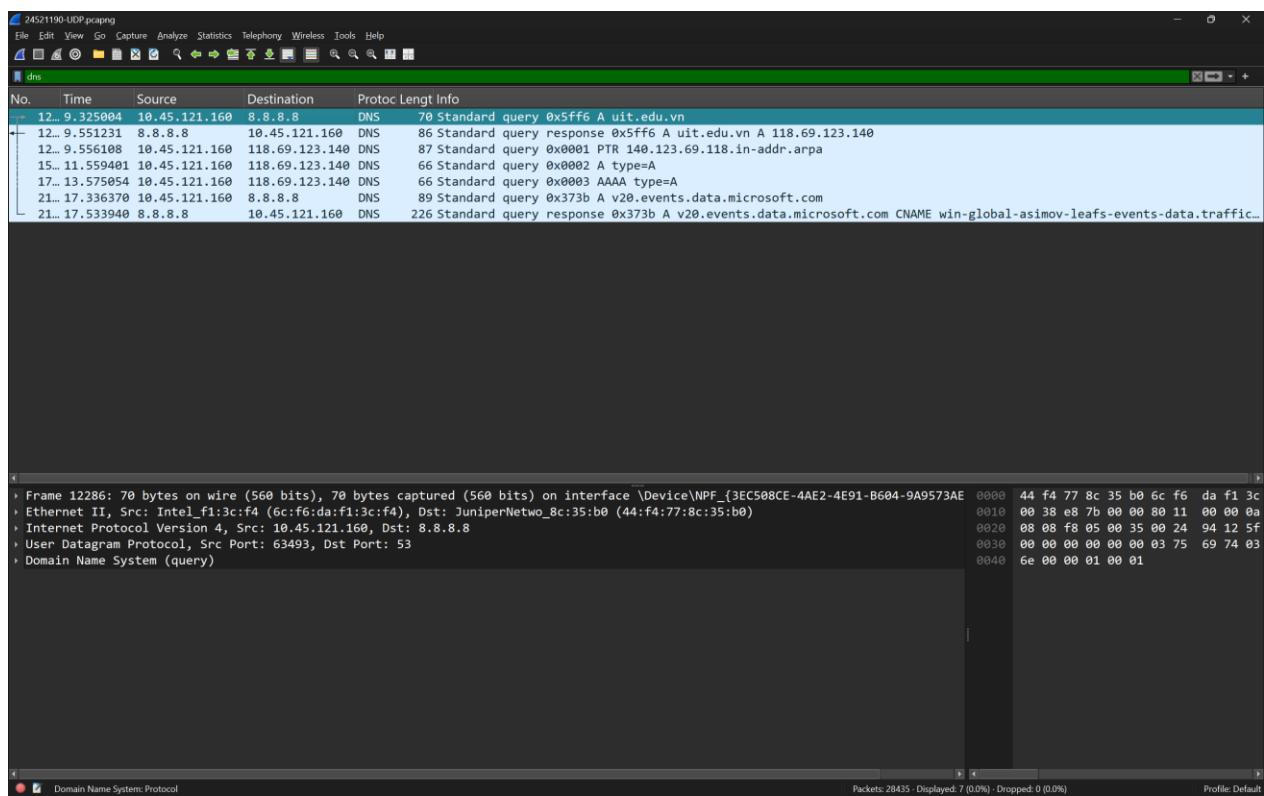
f) **Tìm hiểu 6: Thông qua hướng dẫn, tìm hiểu và trình bày lại về sự khác nhau cơ bản, ưu điểm, nhược điểm của UDP và TCP. Thông qua việc xem streaming Video, tìm hiểu và trình bày Tìm hiểu 1, 2, 3, 4, 5. Tìm và thực hiện Tìm hiểu 6 để sử dụng làm thông tin, góp phần hiểu rõ hơn các nội dung DNS và TCP ở bên dưới**

Tiêu chí	TCP	UDP
Loại giao thức	Hướng kết nối (Connection-oriented)	Không hướng kết nối (Connectionless)
Cách truyền dữ liệu	Thiết lập kết nối trước khi truyền, đảm bảo dữ liệu đến đúng thứ tự	Gửi các gói dữ liệu độc lập, không đảm bảo thứ tự
Độ tin cậy	Cao – có cơ chế kiểm soát lỗi, xác nhận, truyền lại gói bị mất	Thấp – không kiểm tra lỗi, không truyền lại
Tốc độ	Chậm hơn (do phải kiểm tra và đảm bảo độ tin cậy)	Nhanh hơn (ít kiểm soát hơn)
Ứng dụng điển hình	Web (HTTP/HTTPS), Email (SMTP), FTP,...	Truyền video, âm thanh trực tuyến, game online, DNS, VoIP

## 2.2. Phân tích hoạt động giao thức UDP

g) **Tại danh sách các gói tin bắt được, tìm gói tin truy vấn domain uit.edu.vn (hoặc domain đã chọn ở Task 2, bước 4 ở trên)**

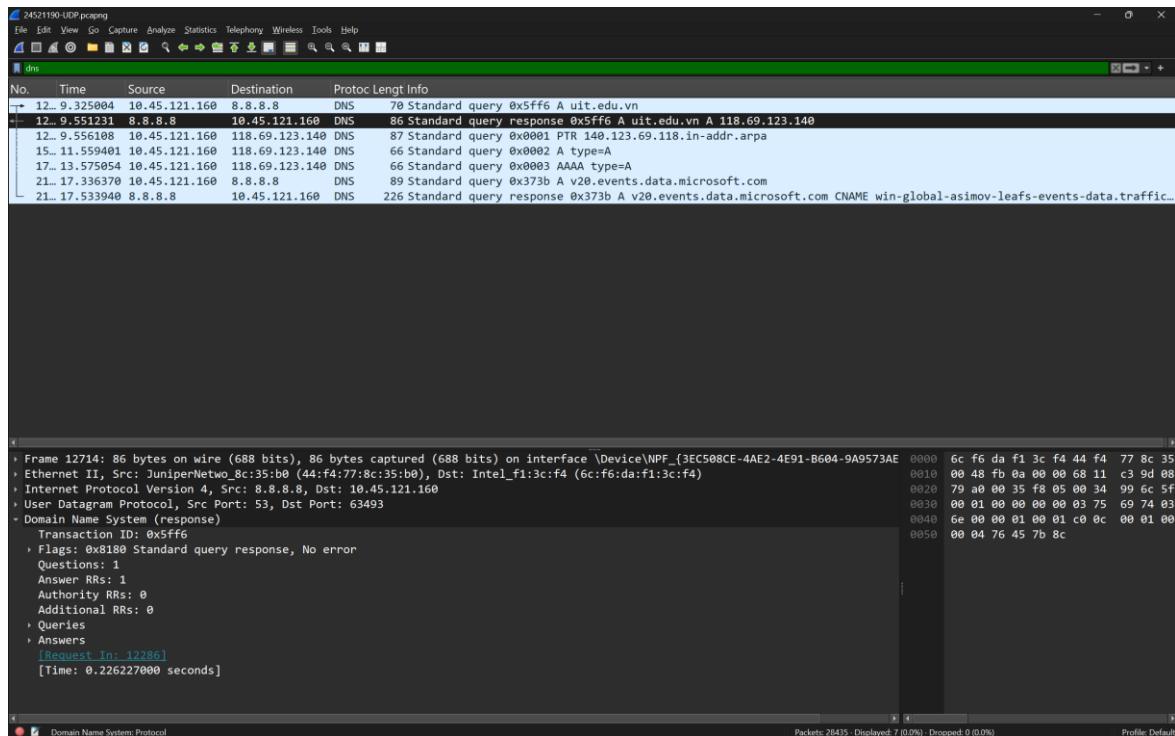
## Lab 03: Phân tích hoạt động giao thức TCP - UDP



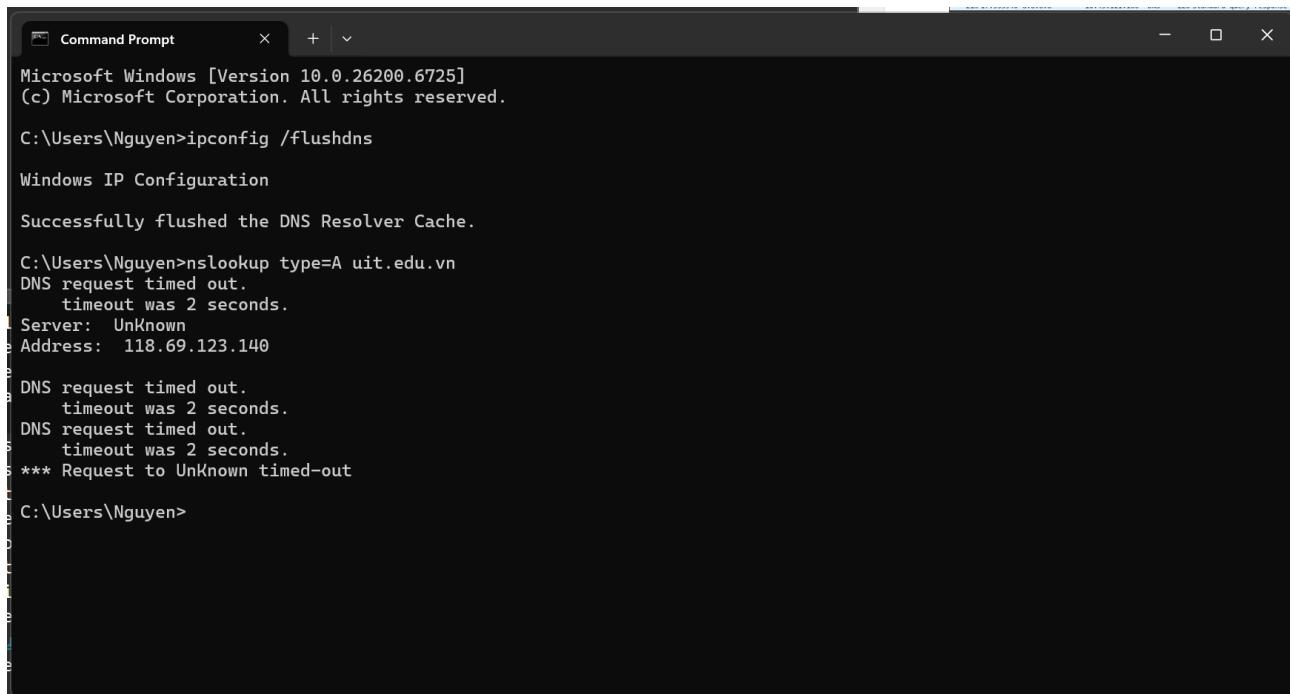
- Vì ta thực hiện truy vấn DNS nên các gói tin yêu cầu miền uit.edu.vn sẽ được xuất hiện trong mục sử dụng giao thức DNS

**h) Xác định gói tin phản hồi của truy vấn trên? Từ thông điệp phản hồi, ghi lại địa chỉ IP của domain uit.edu.vn**

- Gói tin phản hồi của truy vấn trên.



- Lúc này server có địa chỉ IP là 8.8.8.8 sẽ trả về kết quả cho client có địa chỉ IP là 10.45.121.160 kết quả địa chỉ IP của uit.edu.vn là 118.69.123.140 như trong cmd trả về



```

Command Prompt

Microsoft Windows [Version 10.0.26200.6725]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Nguyen>ipconfig /flushdns

Windows IP Configuration

Successfully flushed the DNS Resolver Cache.

C:\Users\Nguyen>nslookup type=A uit.edu.vn
DNS request timed out.
    timeout was 2 seconds.
Server: Unknown
Address: 118.69.123.140

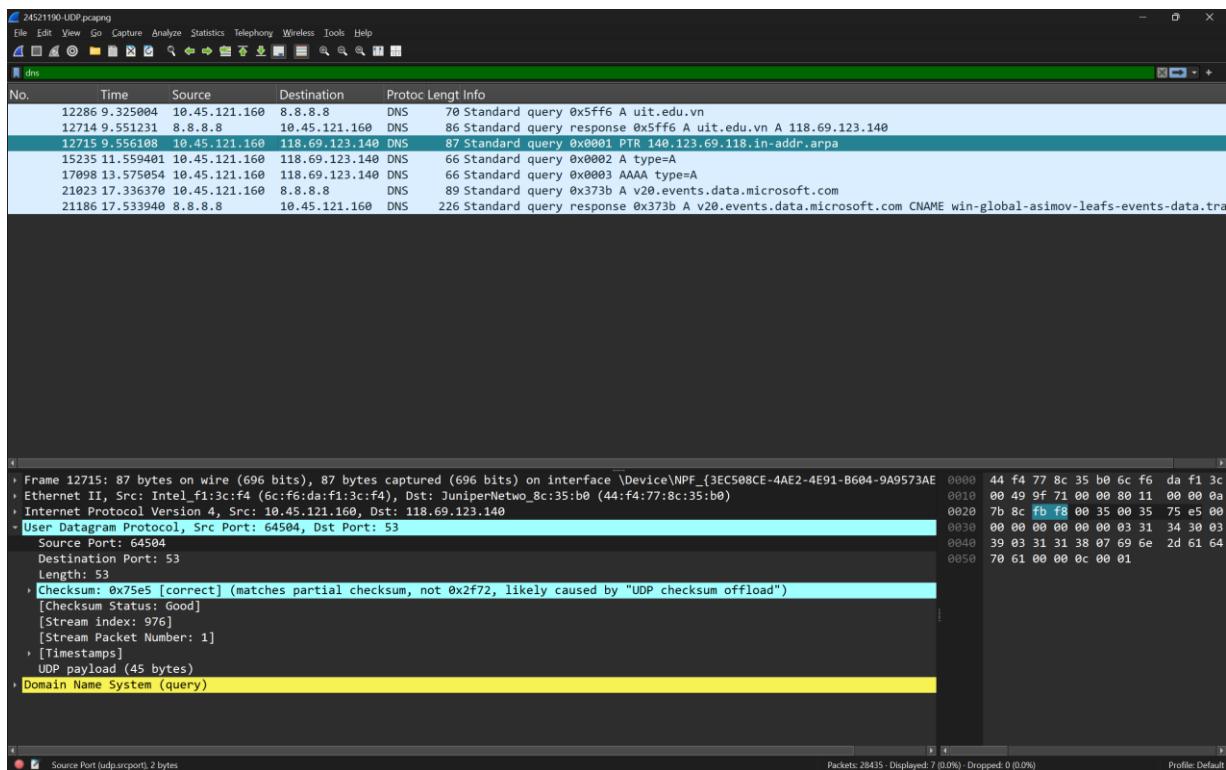
DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
*** Request to Unknown timed-out

C:\Users\Nguyen>

```

i) 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.

- Ta sẽ chọn gói tin thứ 3 trong hình dưới đây để giải thích





- Source Port: 64504: Đây chính là trường đầu tiên của UDP header, có độ dài 2 bytes có ý nghĩa là số cổng (port) của nơi yêu cầu gói tin UDP
- Destination Port: 53: Đây chính là trường thứ hai của UDP header, có độ dài 2 bytes có ý nghĩa là số cổng (port) của server nhận yêu cầu và trả về phản hồi
- Length: 53: Đây chính là trường thứ ba của UDP header, có độ dài 2 bytes có ý nghĩa là tổng độ dài của UDP header (8 bytes) và payload (45 bytes)
- Checksum: 0x75e5: Đây là trường kiểm tra lỗi của UDP, nếu ghi chú [correct] thì chưa chắc gói tin có lỗi hay không nhưng nếu không có [correct] thì chắc chắn gói tin UDP xảy ra lỗi. Phần này có độ dài 2 bytes

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

Trường	Độ dài
Source Port	<pre>Source Port: 64504 Destination Port: 53 Length: 53 ▶ Checksum: 0x75e5 [correct] [Checksum Status: Good] [Stream index: 976] [Stream Packet Number: 1] ▶ [Timestamps] UDP payload (45 bytes) ▶ Domain Name System (query)</pre> <p>Source Port (udp.srcport), 2 bytes</p>
Destination Port	<pre>Source Port: 64504 Destination Port: 53 Length: 53 ▶ Checksum: 0x75e5 [correct] [Checksum Status: Good] [Stream index: 976] [Stream Packet Number: 1] ▶ [Timestamps] UDP payload (45 bytes) ▶ Domain Name System (query)</pre> <p>Destination Port (udp.dstport), 2 bytes</p>



Length	Source Port: 64504 Destination Port: 53 Length: 53 <ul style="list-style-type: none"> <li>‣ Checksum: 0x75e5 [correct] (matches partial checksum, not 0x2f72) [Checksum Status: Good] [Stream index: 976] [Stream Packet Number: 1]</li> <li>‣ [Timestamps]</li> <li>‣ UDP payload (45 bytes)</li> <li>‣ Domain Name System (query)</li> </ul> <p>Length in octets including this header and the data (udp.length), 2 bytes</p>
Checksum	Source Port: 64504 Destination Port: 53 Length: 53 <ul style="list-style-type: none"> <li>‣ Checksum: 0x75e5 [correct] (matches partial checksum, not 0x2f72) [Checksum Status: Good] [Stream index: 976] [Stream Packet Number: 1]</li> <li>‣ [Timestamps]</li> <li>‣ UDP payload (45 bytes)</li> <li>‣ Domain Name System (query)</li> </ul> <p>Details at: <a href="https://www.wireshark.org/docs/wsug_html_chunked/ChAdvChecksums.html">https://www.wireshark.org/docs/wsug_html_chunked/ChAdvChecksums.html</a> (udp.checksum), 2 bytes</p>

k) 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 UDP header (8 bytes) cộng với độ dài của payload (data) như trong hình dưới ta thấy trường Length có giá trị 53 (bao gồm 8 bytes UDP header cộng với 45 bytes payload)

Length: 53 <ul style="list-style-type: none"> <li>‣ Checksum: 0x75e5 [correct] (matches partial checksum, not 0x2f72, likely caused by "UDP checksum offload") [Checksum Status: Good] [Stream index: 976] [Stream Packet Number: 1]</li> <li>‣ [Timestamps]</li> <li>‣ UDP payload (45 bytes)</li> <li>‣ Domain Name System (query)</li> </ul>
--

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

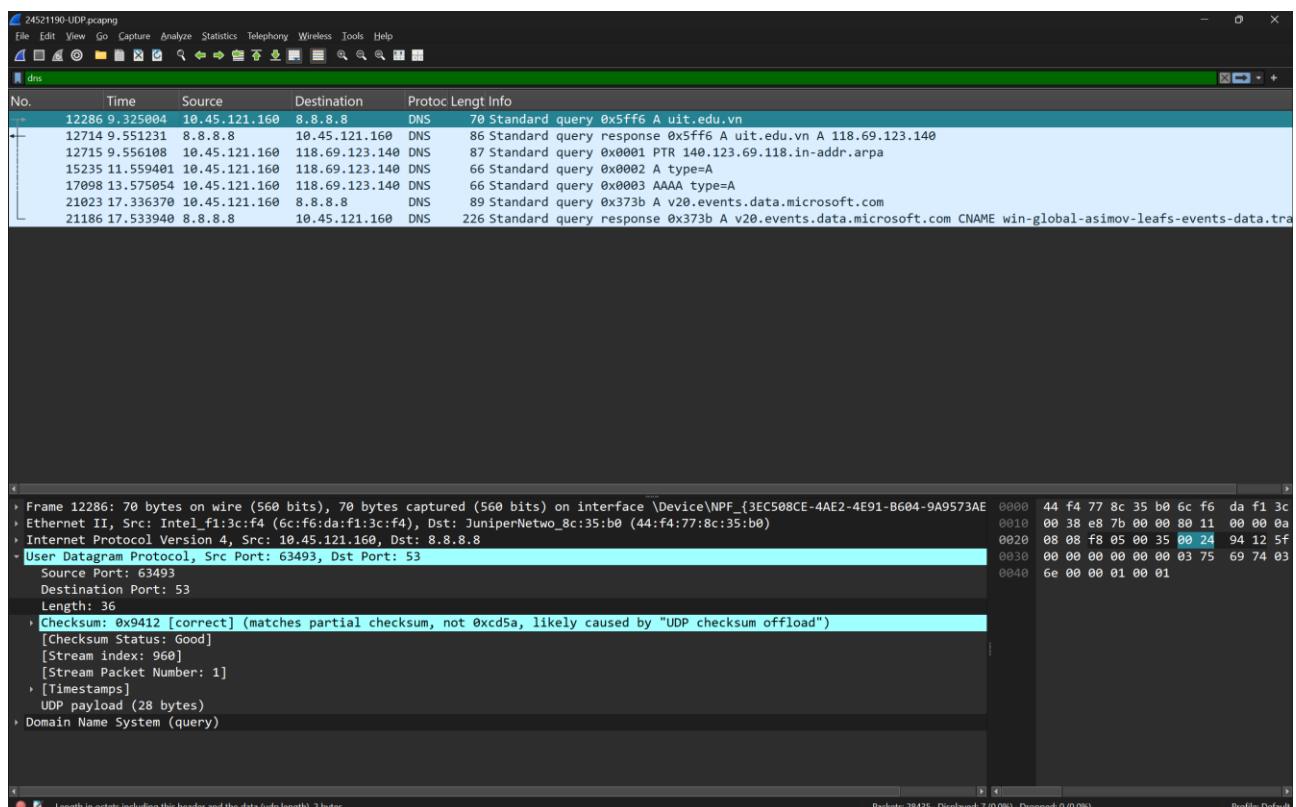
- Ta được biết trường Length trong UDP header có độ dài là 2 bytes (16 bits), do đó số payload lớn nhất của UDP có thể chứa là 65.527 bytes. Kích thước này được tính bằng cách lấy kích thước tối đa của gói tin UDP (65.535 bytes – do length được biểu diễn bằng số 16 bit nên có giá trị tối đa là  $2^{16} - 1$ ) trừ đi 8 byte của UDP header và 20 byte của IP header

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

- Giá trị lớn nhất có thể có của port nguồn (Source port) là 65535. Con số này được tính từ việc port được biểu diễn bằng 2 byte (16 bit), với giá trị từ  $2^0$  đến  $2^{16}-1$

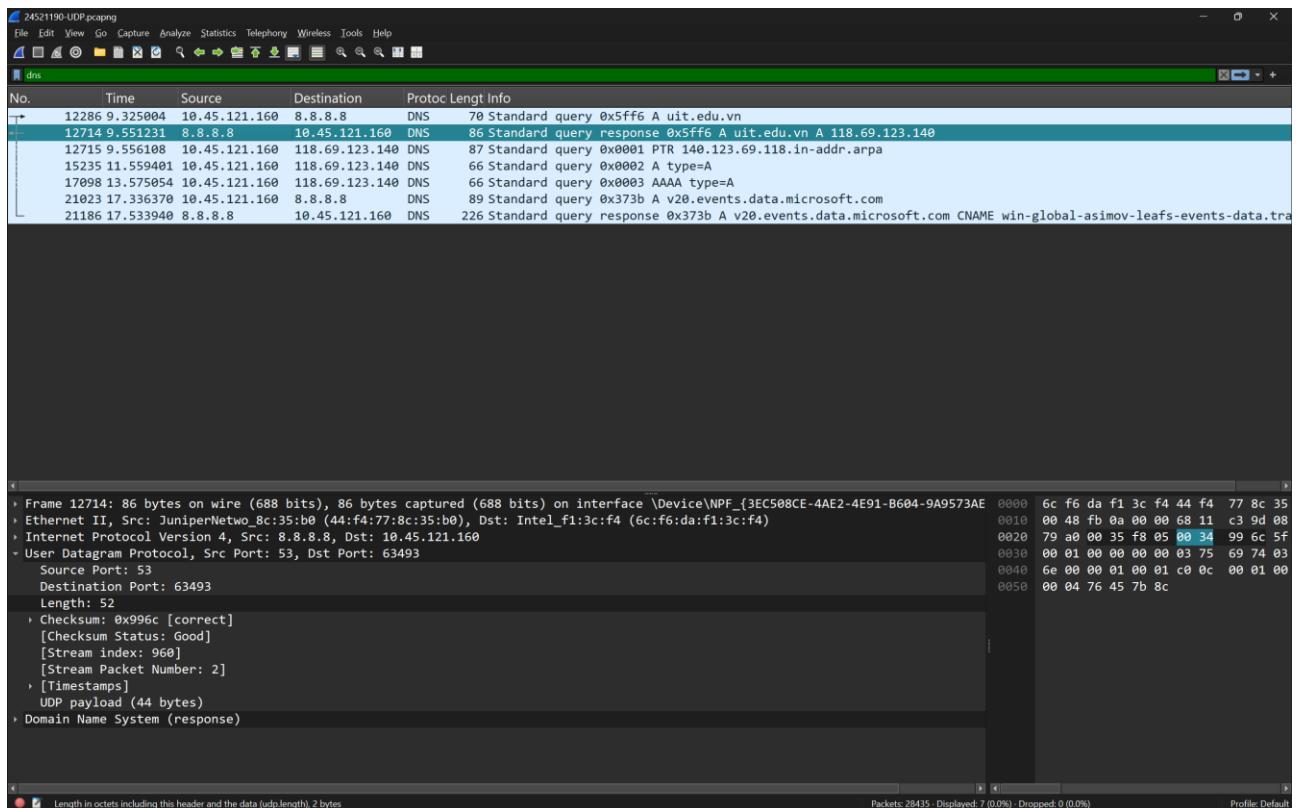
**n) Quan sát 2 gói tin tìm được ở câu 2 và 3, mô tả mối quan hệ giữa các địa chỉ IP và các port của 2 gói tin này.**

- Gói gửi từ client đến server



- Trong gói tin này: Source (IP, Port) = (10.45.121.160, 63493) và Destination (IP, Port) = (8.8.8.8, 53)
- Gói gửi từ server đến client

## Lab 03: Phân tích hoạt động giao thức TCP - UDP

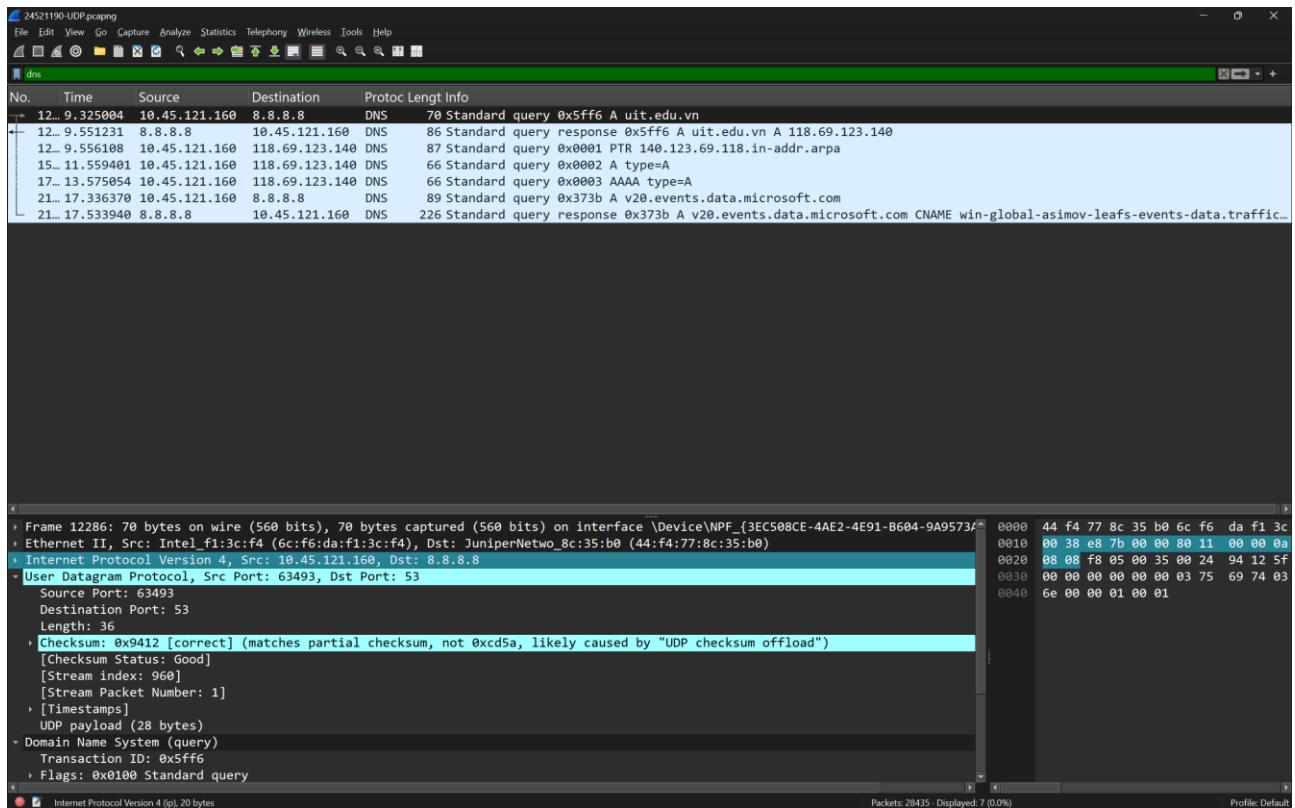


- Trong gói tin này: Source (IP, Port) = (8.8.8.8, 53) và Destination (IP, Port) = (10.45.121.160, 63493)
- Mỗi quan hệ: Trong gói tin request từ client đến server ta sẽ có source ip và source port chính là của client đang yêu cầu, và destination ip và destination port là của server. Nhưng trong gói tin response từ server trả về client, ta sẽ có 2 địa chỉ source và destination sẽ được đổi ngược lại so với gói request. Ta biết rằng địa chỉ IP để biết được địa chỉ thiết bị đầu cuối, còn số cổng (port) cho ta biết tiến trình, do đó các gói tin phải lưu lại các giá trị này để gửi đúng thiết bị và đúng tiến trình đang yêu cầu

**o) Chọn 1 gói tin UDP, dựa trên các thông tin của gói tin này và tính UDP Checksum. So sánh kết quả tự tính toán và trường Checksum của gói tin UDP. Giải thích cách tính.**

- Ta sẽ chọn gói như hình dưới:
  - + IP nguồn: 10.45.121.160 -> bytes: 0A 2D 79 A0
  - + IP đích: 8.8.8.8 -> bytes: 08 08 08 08
  - + Pseudo-header: 0A 2D 79 A0 08 08 08 08 00 11 00 24
  - + UDP header: F8 05 00 35 00 24 00 00
    - source port 63943: F8 05
    - destination port 53: 00 35
    - length 36: 00 24
    - checksum = 0: 00 00
  - + UDP payload (hex): 5f f6 01 00 00 01 00 00 00 00 00 00 03 75 69 74 03 65 64 75 02 76 6e 00 00 01 00 01

- Toàn bộ vùng dữ liệu để cộng (pseudo + UDP header(checksum=0) + payload) dài 48 bytes



### 3. Task 3: Phân tích hoạt động giao thức TCP

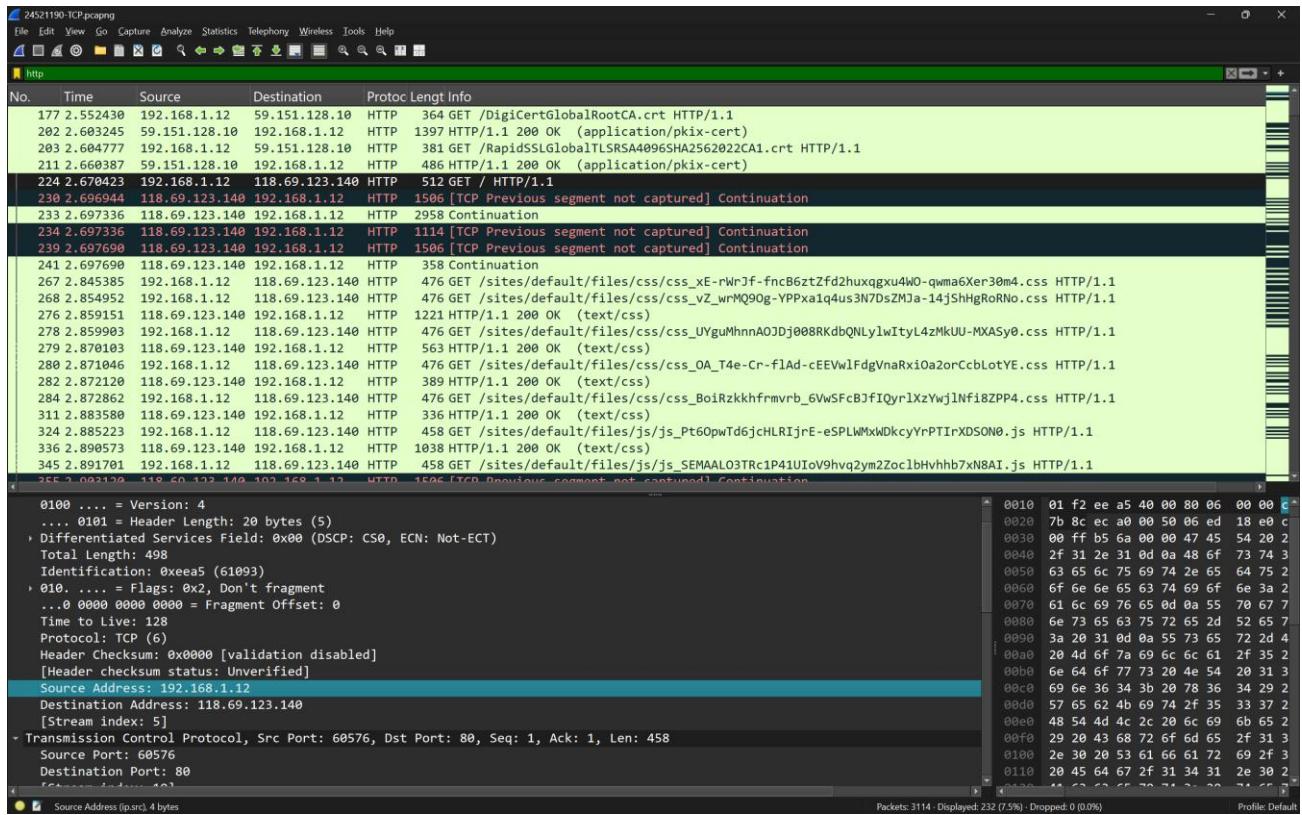
#### 3.1. Truy cập website và bắt các gói tin TCP

#### 3.2. Phân tích hoạt động giao thức TCP

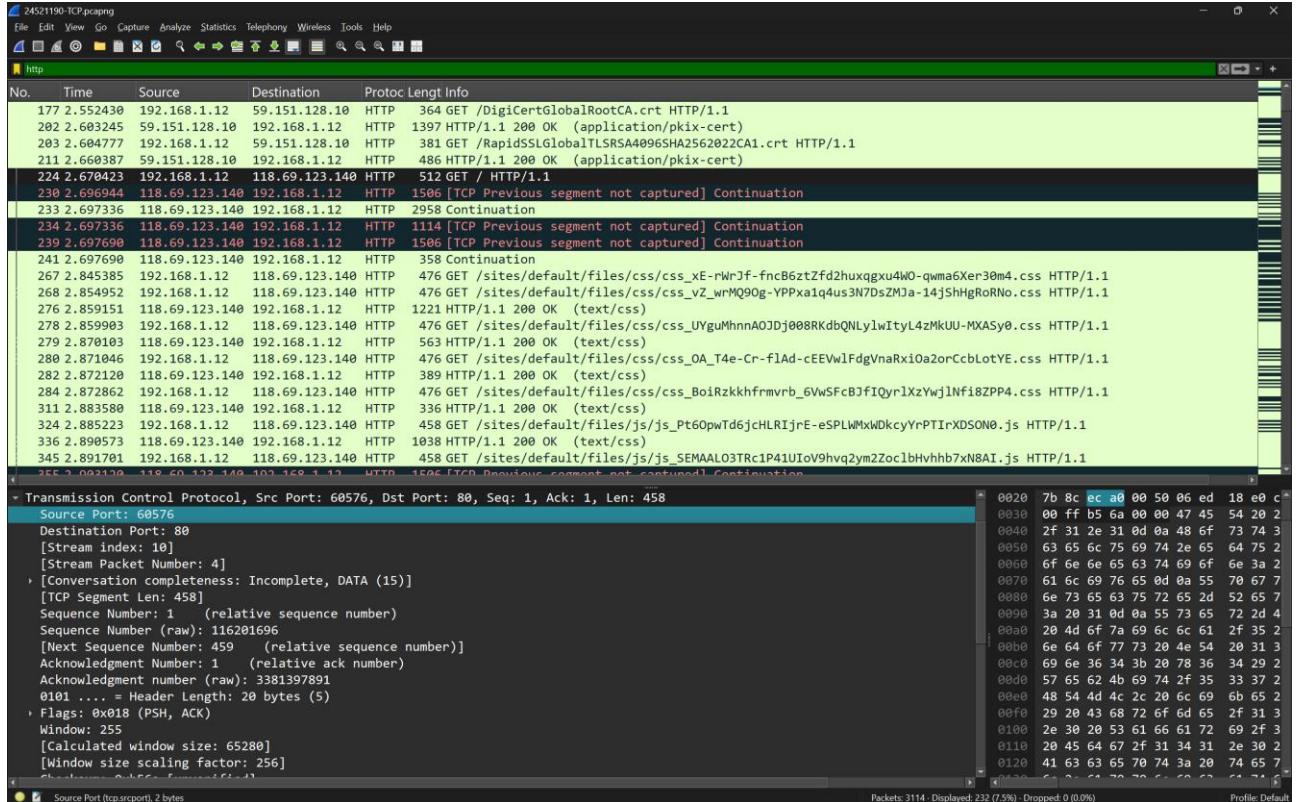
p) *Tìm địa chỉ IP và TCP port của máy client ?*

- Địa chỉ IP: 192.168.1.12

## Lab 03: Phân tích hoạt động giao thức TCP - UDP



- TCP port: 60576



q) Tìm địa chỉ IP của Server ? Nó sử dụng port nào để nhận các segments ?

- Địa chỉ IP của server: 118.69.123.140

```

- Internet Protocol Version 4, Src: 192.168.1.12, Dst: 118.69.123.140
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 498
  Identification: 0xeea5 (61093)
  ▶ 010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 128
  Protocol: TCP (6)
  Header Checksum: 0x0000 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 192.168.1.12
  Destination Address: 118.69.123.140
  [Stream index: 5]
  ▶ Transmission Control Protocol, Src Port: 60576, Dst Port: 80, Seq: 1, Ack: 1, Len: 458
  ▶ Hypertext Transfer Protocol

```

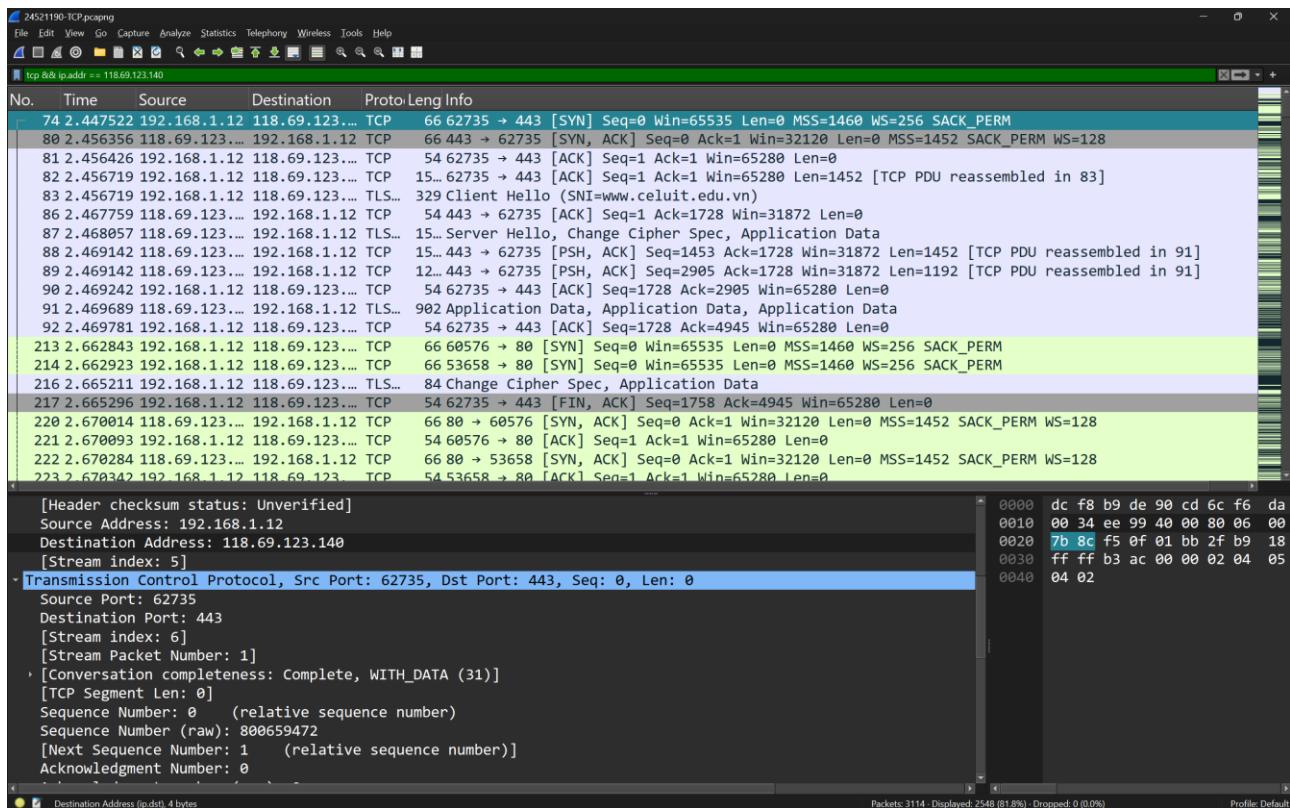
- Server sử dụng port 80 để nhận các segments

```

- Frame 224: 512 bytes on wire (4096 bits), 512 bytes captured (4096 bits) on interface \Device\NPF_{3EC508CE-4AE2-4E91-B604-9A95^
- Ethernet II, Src: Intel_f1:3c:f4 (6c:f6:da:f1:3c:f4), Dst: zte_de:90:cd (dc:f8:b9:de:90:cd)
- Internet Protocol Version 4, Src: 192.168.1.12, Dst: 118.69.123.140
- ▶ Transmission Control Protocol, Src Port: 60576, Dst Port: 80, Seq: 1, Ack: 1, Len: 458
  Source Port: 60576
  Destination Port: 80
  [Stream index: 10]
  [Stream Packet Number: 4]
  ▶ [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 458]
  Sequence Number: 1 (relative sequence number)
  Sequence Number (raw): 116201696
  [Next Sequence Number: 459 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 3381397891
  0101 .... = Header Length: 20 bytes (5)
  ▶ Flags: 0x018 (PSH, ACK)

```

r) Tìm TCP SYN segment (gói tin TCP có cờ SYN) khởi tạo kết nối TCP giữa client và server? Thành phần nào trong segment cho ta biết segment đó là TCP SYN segment?



- Gói tin thứ 74 chính là gói TCP SYN segment khởi tạo kết nối giữa client và server
- Thành phần cho ta biết segment đó là TCP SYN segment là cờ SYN trong trường flags được bật lên 1 và đây cũng là gói đầu tiên mà client gửi đến server. Từ đó cho ta biết đây chính là gói TCP SYN segment khởi tạo kết nối giữa client và server

```
Acknowledgment number (raw): 0
1000 .... = Header Length: 32 bytes (8)
Flags: 0x002 (SYN)
    000. .... .... = Reserved: Not set
    ...0 .... .... = Accurate ECN: Not set
    ....0.... .... = Congestion Window Reduced: Not set
    ....00.... .... = ECN-Echo: Not set
    ....00.... .... = Urgent: Not set
    ....00.... .... = Acknowledgment: Not set
    ....00.... .... = Push: Not set
    ....00.... .... = Reset: Not set
    ....00....0.... = Syn: Set
    ....00....0.... = Fin: Not set
[TCP Flags: .....S..]
Window: 65535
```

**s) 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 được giá trị đó ? Thành phần nào trong segment cho ta biết segment đó là SYN/ACK segment ?**

## Lab 03: Phân tích hoạt động giao thức TCP - UDP

74 2.447522 192.168.1.12 118.69.123... TCP	66 62735 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
80 2.456356 118.69.123... 192.168.1.12 TCP	66 443 → 62735 [SYN, ACK] Seq=0 Ack=1 Win=32120 Len=0 MSS=1452 SACK_PERM WS=128
81 2.456426 192.168.1.12 118.69.123... TCP	54 62735 → 443 [ACK] Seq=1 Ack=1 Win=65280 Len=0
82 2.456719 192.168.1.12 118.69.123... TCP	15... 62735 → 443 [ACK] Seq=1 Ack=1 Win=65280 Len=1452 [TCP PDU reassembled in 83]
83 2.456719 192.168.1.12 118.69.123... TLS...	329 Client Hello (SNI=www.celuit.edu.vn)
86 2.467759 118.69.123... 192.168.1.12 TCP	54 443 → 62735 [ACK] Seq=1 Ack=1728 Win=31872 Len=0
87 2.468057 118.69.123... 192.168.1.12 TLS...	15... Server Hello, Change Cipher Spec, Application Data
88 2.469142 118.69.123... 192.168.1.12 TCP	15... 443 → 62735 [PSH, ACK] Seq=1453 Ack=1728 Win=31872 Len=1452 [TCP PDU reassembled in 91]
89 2.469142 118.69.123... 192.168.1.12 TCP	12... 443 → 62735 [PSH, ACK] Seq=2905 Ack=1728 Win=31872 Len=1192 [TCP PDU reassembled in 91]
90 2.469242 192.168.1.12 118.69.123... TCP	54 62735 → 443 [ACK] Seq=1728 Ack=2905 Win=65280 Len=0
91 2.469689 118.69.123... 192.168.1.12 TLS...	982 Application Data, Application Data, Application Data
92 2.469781 192.168.1.12 118.69.123... TCP	54 62735 → 443 [ACK] Seq=1728 Ack=4945 Win=65280 Len=0
213 2.6662843 192.168.1.12 118.69.123... TCP	66 60576 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
214 2.6662923 192.168.1.12 118.69.123... TCP	66 53658 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
216 2.665211 192.168.1.12 118.69.123... TLS...	84 Change Cipher Spec, Application Data
217 2.665296 192.168.1.12 118.69.123... TCP	54 62735 → 443 [FIN, ACK] Seq=1758 Ack=4945 Win=65280 Len=0
220 2.670014 118.69.123... 192.168.1.12 TCP	66 80 → 60576 [SYN, ACK] Seq=0 Ack=1 Win=32120 Len=0 MSS=1452 SACK_PERM WS=128
221 2.670093 192.168.1.12 118.69.123... TCP	54 60576 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0
222 2.670284 118.69.123... 192.168.1.12 TCP	66 80 → 53658 [SYN, ACK] Seq=0 Ack=1 Win=32120 Len=0 MSS=1452 SACK_PERM WS=128
223 2.670284 192.168.1.12 118.69.123... TCF	54 53658 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0

- Gói tin thứ 80 chính là gói SYN/ACK segment được gửi bởi server đến client để trả lời cho SYN segment vì ta có thể thấy rõ ràng gói này được gửi bởi server (địa chỉ IP = 118.69.123.140) và nằm ngay sau gói tin SYN do client gửi đến server
- Giá trị Acknowledgement trong SYN/ACK segment là: 800659473 và còn có thêm giá trị 1 cũng chính là con số này nhưng được mặc định để dễ nhận biết

Destination Address: 192.168.1.12
[Stream index: 5]
Transmission Control Protocol, Src Port: 443, Dst Port: 62735, Seq: 0, Ack: 1, Len: 0
Source Port: 443
Destination Port: 62735
[Stream index: 6]
[Stream Packet Number: 2]
Conversation completeness: Complete, WITH_DATA (31)
[TCP Segment Len: 0]
Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 633704080
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 800659473
1000 .... = Header Length: 32 bytes (8)

- Server xác định được giá trị đó nhờ vào trường sequence number của gói SYN được gửi ngay trước đó cộng thêm 1. Ta có thể thấy trong hình thì sequence number của gói SYN chính là 800659472 (cũng được khởi tạo ngẫu nhiên) và khi ta cộng thêm 1 ta sẽ có được giá trị ACK của gói SYN/ACK

Destination Address: 118.69.123.140
[Stream index: 5]
Transmission Control Protocol, Src Port: 62735, Dst Port: 443, Seq: 0, Len: 0
Source Port: 62735
Destination Port: 443
[Stream index: 6]
[Stream Packet Number: 1]
Conversation completeness: Complete, WITH_DATA (31)
[TCP Segment Len: 0]
Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 800659472
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 0
Acknowledgment number (raw): 0
1000 .... = Header Length: 32 bytes (8)

- Thành phần cho ta biết đây chính là gói SYN/ACK là trường flags trong gói tin khi có cờ ACK và cờ SYN được bật lên 1

```

Acknowledgment number (raw): 800659473
1000 .... = Header Length: 32 bytes (8)
Flags: 0x012 (SYN, ACK)
  000. .... .... = Reserved: Not set
  ...0 .... .... = Accurate ECN: Not set
  .... 0.... .... = Congestion Window Reduced: Not set
  .... .0.... .... = ECN-Echo: Not set
  .... ..0.... .... = Urgent: Not set
  .... .0.... .... = Acknowledgment: Set
  .... .... 0.... = Push: Not set
  .... .... .0.... = Reset: Not set
  .... .... ..1.... = Syn: Set
  .... .... ...0.... = Fin: Not set
[TCP Flags: .....A..S..]
Window: 32120

```

t) 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. Dưa 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 cho 6 segments này

- 6 segment đầu tiên mà server gửi cho client có số thứ tự: 80, 86, 220, 222, 225, 228

No.	Time	Source	Destination	Protocol	Length	Info
80 2.456356	118.69.123.140	192.168.1.12	TCP	66 443 → 62735 [SYN, ACK]	Seq=0 Ack=1 Win=32120 Len=0 MSS=1452 SACK_PERM WS=128	
86 2.467759	118.69.123.140	192.168.1.12	TCP	54 443 → 62735 [ACK]	Seq=1 Ack=1728 Win=31872 Len=0	
220 2.670014	118.69.123.140	192.168.1.12	TCP	66 80 → 60576 [SYN, ACK]	Seq=0 Ack=1 Win=32120 Len=0 MSS=1452 SACK_PERM WS=128	
222 2.670284	118.69.123.140	192.168.1.12	TCP	66 80 → 53658 [SYN, ACK]	Seq=0 Ack=1 Win=32120 Len=0 MSS=1452 SACK_PERM WS=128	
225 2.671864	118.69.123.140	192.168.1.12	TCP	54 443 → 62735 [FIN, ACK]	Seq=4945 Ack=1758 Win=31872 Len=0	
228 2.673643	118.69.123.140	192.168.1.12	TCP	54 443 → 62735 [ACK]	Seq=4946 Ack=1759 Win=31872 Len=0	
230 2.696944	118.69.123.140	192.168.1.12	HTTP	1506 [TCP Previous segment not captured] Continuation		
275 2.859151	118.69.123.140	192.168.1.12	TCP	1506 80 → 60576 [ACK]	Seq=11529 Ack=881 Win=31872 Len=1452 [TCP PDU reassembled in 276]	
279 2.870103	118.69.123.140	192.168.1.12	HTTP	563 HTTP/1.1 200 OK	(text/css)	
281 2.872120	118.69.123.140	192.168.1.12	TCP	2958 80 → 60576 [PSH, ACK]	Seq=14148 Ack=1303 Win=31872 Len=2904 [TCP PDU reassembled in 282]	
285 2.881247	118.69.123.140	192.168.1.12	TCP	1506 80 → 53658 [ACK]	Seq=516 Ack=845 Win=31872 Len=1452 [TCP PDU reassembled in 336]	
300 2.883580	118.69.123.140	192.168.1.12	TCP	1506 80 → 60576 [ACK]	Seq=17387 Ack=1725 Win=31872 Len=1452 [TCP PDU reassembled in 311]	
355 2.983120	118.69.123.140	192.168.1.12	HTTP	1506 [TCP Previous segment not captured] Continuation		
379 2.984470	118.69.123.140	192.168.1.12	TCP	4410 80 → 53658 [ACK]	Seq=31927 Ack=1249 Win=31872 Len=4356 [TCP PDU reassembled in 399]	
421 2.917077	118.69.123.140	192.168.1.12	HTTP	1496 HTTP/1.1 200 OK	(text/javascript)	
423 2.926048	118.69.123.140	192.168.1.12	TCP	1506 80 → 60576 [ACK]	Seq=80578 Ack=2533 Win=31872 Len=1452 [TCP PDU reassembled in 428]	
435 2.929598	118.69.123.140	192.168.1.12	HTTP	1506 [TCP Previous segment not captured] Continuation		
472 2.935484	118.69.123.140	192.168.1.12	TCP	1506 80 → 60576 [ACK]	Seq=97635 Ack=2937 Win=31872 Len=1452 [TCP PDU reassembled in 474]	
481 2.940676	118.69.123.140	192.168.1.12	TCP	1506 80 → 53658 [ACK]	Seq=73688 Ack=2482 Win=31872 Len=1452 [TCP PDU reassembled in 529]	
546 2.950984	118.69.123.140	192.168.1.12	HTTP	658 HTTP/1.1 200 OK	(PNG)	
550 2.955716	118.69.123.140	192.168.1.12	TCP	1506 80 → 53658 [ACK]	Seq=128473 Ack=2957 Win=31872 Len=1452 [TCP PDU reassembled in 988]	
621 2.962265	118.69.123.140	192.168.1.12	TCP	1506 80 → 60576 [ACK]	Seq=101534 Ack=3861 Win=31872 Len=1452 [TCP PDU reassembled in 1066]	
873 2.976486	118.69.123.140	192.168.1.12	TCP	66 80 → 65184 [SYN, ACK]	Seq=0 Ack=1 Win=32120 Len=0 MSS=1452 SACK_PERM WS=128	
875 2.976754	118.69.123.140	192.168.1.12	TCP	66 80 → 57455 [SYN, ACK]	Seq=0 Ack=1 Win=32120 Len=0 MSS=1452 SACK_PERM WS=128	

- Sequence number của 6 segment này:
  - + Gói 80: Seq = 0
  - + Gói 86: Seq = 1
  - + Gói 220: Seq = 0
  - + Gói 222: Seq = 0
  - + Gói 225: Seq = 4945
  - + Gói 228: Seq = 4946
- Thời gian mà mỗi segment được gửi, thời gian ACK cho mỗi segment được nhận
- Gói 80:
  - + Thời gian gửi: 2.456356

80 2.456356 118.69.123.140 192.168.1.12 TCP 66 443 → 62735 [SYN, ACK] Seq=0 Ack=1 Win=32120 Len=0 MSS=1452 SACK\_PERM WS=128

+ Thời gian ACK: 2.456426

81 2.456426 192.168.1.12 118.69.123.140 TCP 54 62735 → 443 [ACK] Seq=1 Ack=1 Win=65280 Len=0

## Lab 03: Phân tích hoạt động giao thức TCP - UDP

- Gói 86:
- + Thời gian gửi: 2.467759

```
1 86 2.467759 118.69.123.140 192.168.1.12 TCP 54 443 → 62735 [ACK] Seq=1 Ack=1728 Win=31872 Len=0
```

- + Thời gian ACK: 2.469242

```
1 90 2.469242 192.168.1.12 118.69.123.140 TCP 54 62735 → 443 [ACK] Seq=1728 Ack=2905 Win=65280 Len=0
```

- Gói 220:
- + Thời gian gửi: 2.670014

```
1 220 2.670014 118.69.123.140 192.168.1.12 TCP 66 80 → 60576 [SYN, ACK] Seq=0 Ack=1 Win=32120 Len=0 MSS=1452 SACK_PERM WS=128
```

- + Thời gian ACK: 2.670093

```
1 221 2.670093 192.168.1.12 118.69.123.140 TCP 54 60576 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0
```

- Gói 222:
- + Thời gian gửi: 2.670284

```
1 222 2.670284 118.69.123.140 192.168.1.12 TCP 66 80 → 53658 [SYN, ACK] Seq=0 Ack=1 Win=32120 Len=0 MSS=1452 SACK_PERM WS=128
```

- + Thời gian ACK: 2.670342

```
1 223 2.670342 192.168.1.12 118.69.123.140 TCP 54 53658 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0
```

- Gói 225:
- + Thời gian gửi: 2.671864

```
1 225 2.671864 118.69.123.140 192.168.1.12 TCP 54 443 → 62735 [FIN, ACK] Seq=4945 Ack=1758 Win=31872 Len=0
```

- + Thời gian ACK: 2.671911

```
1 226 2.671911 192.168.1.12 118.69.123.140 TCP 54 62735 → 443 [ACK] Seq=1759 Ack=4946 Win=65280 Len=0
```

- Gói 228:
- + Thời gian gửi: 2.673643

```
1 228 2.673643 118.69.123.140 192.168.1.12 TCP 54 443 → 62735 [ACK] Seq=4946 Ack=1759 Win=31872 Len=0
```

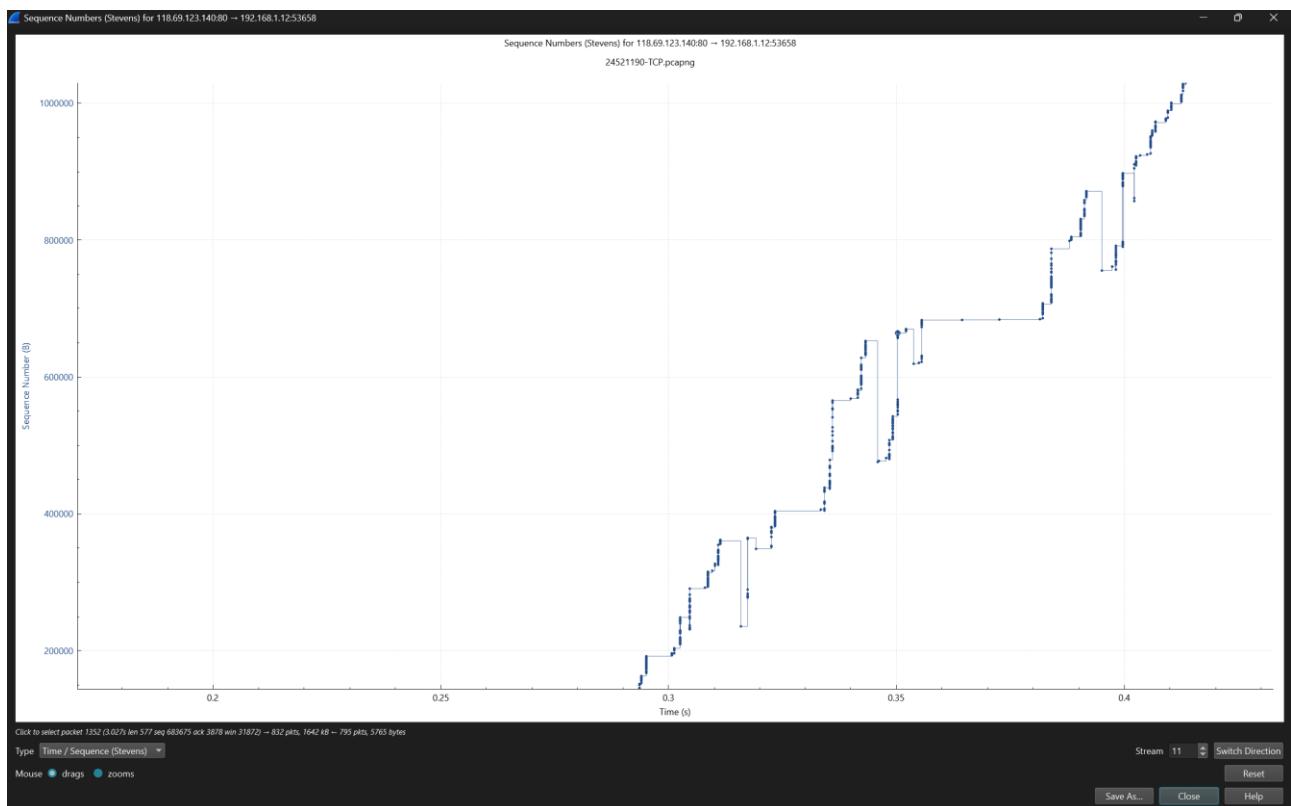
- + Thời gian ACK: 2.697406

```
1 235 2.697406 192.168.1.12 118.69.123.140 TCP 54 60576 → 80 [ACK] Seq=459 Ack=2905 Win=65280 Len=0
```

STT	Thời gian	RTT	SEQ number	ACK number
80	2.456356	0.000007	0	1
86	2.467759	0.001483	1	1728
220	2.670014	0.000079	0	1
222	2.670284	0.000058	0	1
225	2.671864	0.000047	4945	1758
228	2.673643	0.023763	4946	1759

*u) 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 (ta chọn gói tin 222)



- Ta biết được gói tin được gửi lại vì trong biểu đồ trên, seq của gói tin 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 (trước) + độ dài của gói tin trước

- Sequence number ở cùng một bên gửi sẽ tăng dần. Tuy nhiên ở hình trên nó lại giảm so với gói tin trước. Từ đó ta có thể biết rằng đây là một gói tin được gửi lại

## YÊU CẦU CHUNG

### 1) Đánh giá

- Chuẩn bị tốt các yêu cầu đặt ra trong bài thực hành.
- Sinh viên hiểu và tự thực hiện được bài thực hành, trả lời đầy đủ các yêu cầu đặt ra.
- Nộp báo cáo kết quả chi tiết những đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (*nếu có*); giải thích cho quan sát (*nếu có*).
- Sinh viên báo cáo kết quả thực hiện và nộp bài.

### 2) Báo cáo

- File **.PDF** hoặc **.docx**. Tập trung vào nội dung, giải thích.
- Nội dung trình bày bằng Font chữ **Times New Romans/** hoặc font chữ của mẫu báo cáo này (**UTM Avo**) – **cỡ chữ 13**. **Canh đều (Justify)** cho văn bản. **Canh giữa (Center)** cho ảnh chụp.
- Đặt tên theo định dạng: LabX-MSSV1. (trong đó X là Thứ tự buổi Thực hành).  
Ví dụ: Lab01-21520001
- Nộp file báo cáo trên theo thời gian đã thống nhất tại [courses.uit.edu.vn](http://courses.uit.edu.vn).

**Bài sao chép, trẽ, ... sẽ được xử lý tùy mức độ vi phạm.**

**HẾT**