

ĐẠI HỌC QUỐC GIA TPHCM

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

Network Packet Analysis with Wireshark

Môn học: Mạng máy tính

Sinh viên thực hiện:

Nguyễn Quốc Nam - 24120098
Võ Hoàng Phúc - 24120123
Chế Nguyễn Thùy Trang - 24120469

Giáo viên hướng dẫn:

ThS. Chung Thuỳ Linh

Ngày 13 tháng 12 năm 2025



Danh sách thành viên

Đóng góp	MSSV	Họ và tên	Mã lớp	Nhiệm vụ
100%	24120123	Võ Hoàng Phúc	24CTT1	- Thực hiện các yêu cầu của part 1. - Trả lời và viết báo cáo part 1.
100%	24120098	Nguyễn Quốc Nam	24CTT1	- Thực hiện các yêu cầu của part 2. - Trả lời và viết báo cáo part 2. - Tổng hợp các báo cáo.
100%	24120469	Chế Nguyễn Thùy Trang	24CTT1	- Thực hiện các yêu cầu của part 3. - Trả lời và viết báo cáo part 3.

Bảng 1: Danh sách các thành viên và nhiệm vụ

Mục lục

1 Part 1: Analyzing HTTP Traffic (3.5 pt)	1
1.1 How many HTTP GET request messages were transmitted by the browser? To which Internet addresses were these requests directed?	1
1.2 Determine whether the browser retrieved the two images sequentially or concurrently from their respective web servers, and provide an explanation for your conclusion by examining the timing of the requests and the source IP addresses	2
1.3 Locate the HTTP response message containing the content of the initial HTML page (HTTP-wireshark-file4.html). What is the status code and status phrase provided by the server?	2
1.4 Based on your answer to Question 1, how many distinct TCP connections were established to fetch the HTML file and the two embedded images? Provide evidence by listing the unique Stream Index Numbers (e.g., tcp.stream eq X) that were used for these three objects	3
1.5 For the TCP connection that retrieved the initial HTML file, identify the three packets that form the TCP Three-Way Handshake. List the TCP flags set in each of these three packets in order	5
1.6 Select the largest data transfer packet (a packet with the PSH or ACK flag set and a large length) during the download of one of the image files. Examine the TCP Window Size Value in the packet details. What does this value represent, and why is it essential for the Transport Layer?	6
2 Part 2: Analyzing DHCP Traffic (3.5 pt)	8
2.1 Were any ARP packets sent or received during the DHCP packet-exchange period? If so, explain the purpose of those ARP packets	8
2.2 A host uses DHCP to obtain an IP address, among other configuration parameters. However, the host's IP address is not finalized until the completion of the four-message DHCP exchange. If the IP address is not assigned until the final message, what IP address values are used in the IP datagrams that carry these messages?	11

2.3 If, after receiving the ACK, the client sent an ARP probe for this new IP but received an ARP Reply from another device, what specific action is the client required to take according to the DHCP standard (RFC 2131)	12
2.4 Why does the DHCP Request message still use UDP Port 68 as the source port and UDP Port 67 as the destination port, even though a unique, high-numbered ephemeral port could technically be used? Explain how UDP's connectionless nature makes this fixed port usage simple and robust.	12
2.5 UDP checksum	13
2.5.1 In the DHCP Discover packet, expand the User Datagram Protocol (UDP) layer and examine the Checksum field. Does Wireshark display the message "[UDP checksum is good]" or "[UDP checksum is incorrect]"?	13
2.5.2 If the checksum had been calculated as incorrect, what action would the recipient's Transport Layer (UDP handler) have immediately taken, and why would this lead the DHCP process to stall?	13
3 Part 3: Network and Link Layer Analysis (3 pt)	13
3.1 Source/Destination IP: Locate the DNS Query packet sent from your host to the Google DNS server	13
3.1.1 What is the Source IP Address?	14
3.1.2 What is the Destination IP Address?	14
3.1.3 Explain why these Source and Destination IP addresses will remain unchanged as the packet travels across the internet to the Google DNS server.	14
3.2 Find the Time to Live (TTL) field in the IP header of the DNS Query. What does the initial value of the TTL represent, and how does your local router/gateway modify this value when forwarding the packet?	15
3.2.1 Find the Time to Live (TTL) field in the IP header of the DNS Query.	15
3.2.2 Giá trị TTL ban đầu đại diện cho điều gì?	16
3.2.3 Router điều chỉnh TTL như nào	16
3.3 Find the MAC Address of your local router/gateway (either by using arp -a in the command prompt or by finding the MAC address associated with the Gateway IP in your capture)	16

3.4 What are the Source and Destination addresses in the Link Layer header (Ethernet or 802.11)? How are these addresses fundamentally different from the IP addresses you found?	17
3.4.1 What are the Source and Destination addresses in the Link Layer header (Ethernet or 802.11)?	17
3.4.2 How are these addresses fundamentally different from the IP addresses you found?	18
3.5 Examine the Link Layer header (Ethernet II or similar). What is the Type field's value, and what does it tell the receiving device (your router/gateway) about the protocol that follows?	18
3.6 Giá trị của trường Type là gì?	18
3.7 Ý nghĩa của Trường Type với thiết bị nhận (gateway/router)	18

Danh sách bảng

1	Danh sách các thành viên và nhiệm vụ	i
2	Các HTTP GET request được truyền bởi trình duyệt	1
3	Các tệp hình ảnh được yêu cầu bởi trình duyệt	2
4	Bảng địa chỉ IP dùng trong 4 thông điệp DHCP	11
5	So sánh địa chỉ MAC và địa chỉ IP	18

Danh sách hình vẽ

1	Lọc các gói tin HTTP GET trong Wireshark	1
2	Hai tệp hình ảnh từ hai yêu cầu GET	2
3	Status code và status phase của gói HTTP	3
4	TCP Stream cho file HTML	4
5	TCP Stream cho file pearson.png	5
6	TCP Stream cho file 8E_cover_small.jpg	5
7	TCP Stream cho file 8E_cover_small.jpg	6
8	TCP Flags của gói tin có TCP Segment Len lớn nhất	7
9	TCP Window Size Value	7
10	Lọc các gói tin DHCP trong Wireshark	9
11	Các gói tin ARP trong DHCP Exchange 1	9
12	Các gói tin ARP trong DHCP Exchange 2	10
13	Lọc gói tin DNS với ip của Google DNS server	14
14	Tính toàn vẹn end-to-end của địa chỉ IP	15
15	TTL Field in IP Header of DNS Query	15
16	MAC Address trong Ethernet Header của DNS Query	16
17	MAC Address trong Command Prompt	17
18	Source and Destination MAC Address in Link Layer Header	17
19	Trường Type trong Link Layer Header	18

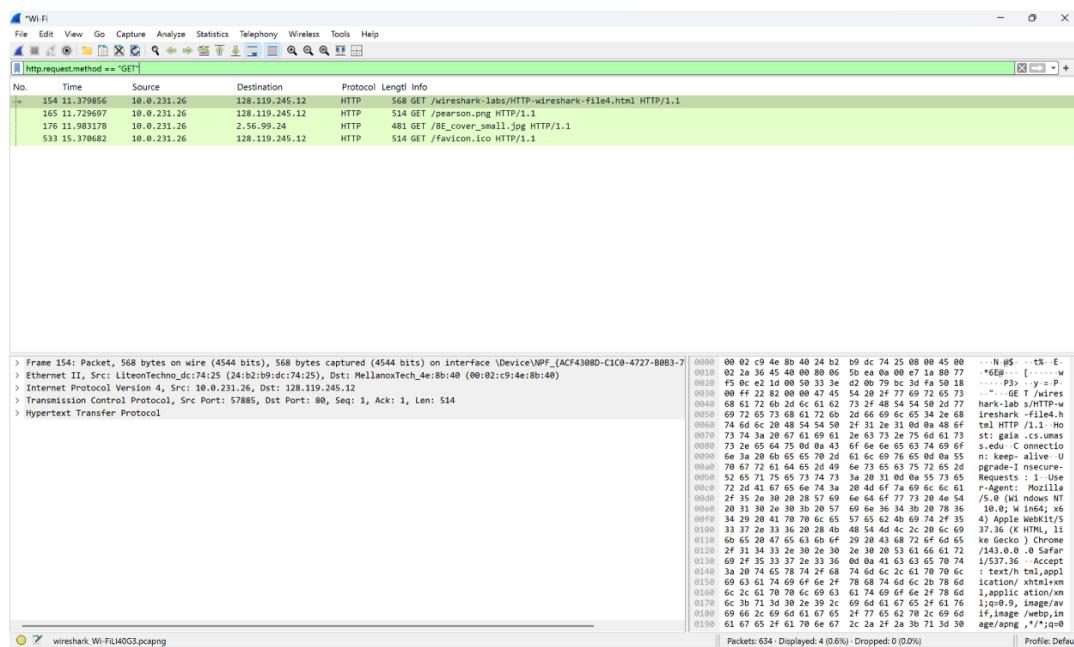
1 Part 1: Analyzing HTTP Traffic (3.5 pt)

1.1 How many HTTP GET request messages were transmitted by the browser? To which Internet addresses were these requests directed?

- Để lọc các HTTP GET request, tại ô Display Filter của Wireshark, nhập:

```
1 http.request.method == "GET"
```

- Wireshark lúc này chỉ hiển thị các gói tin HTTP GET. Tiến hành đếm số gói tin trong danh sách.



Hình 1: Lọc các gói tin HTTP GET trong Wireshark

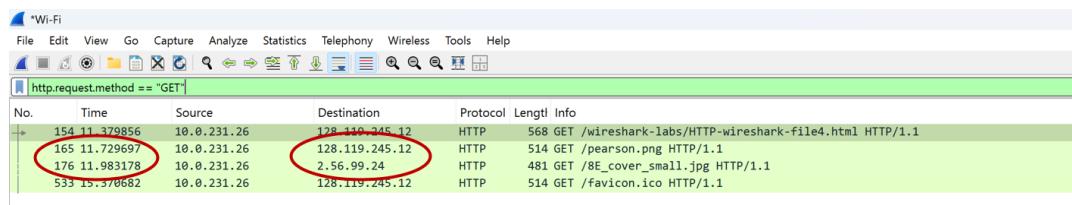
- Kết quả thu được 4 gói tin HTTP GET:

STT	Dòng	Địa chỉ IP đích	Thông tin
1	154	128.119.245.12	GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
2	165	128.119.245.12	GET /pearson.png HTTP/1.1
3	176	2.56.99.24	GET /8E_cover_small.jpg HTTP/1.1
4	533	128.119.245.12	GET /favicon.ico HTTP/1.1

Bảng 2: Các HTTP GET request được truyền bởi trình duyệt

1.2 Determine whether the browser retrieved the two images sequentially or concurrently from their respective web servers, and provide an explanation for your conclusion by examining the timing of the requests and the source IP addresses

- Sau khi lọc các gói tin HTTP GET, thu được 4 yêu cầu GET với 2 yêu cầu GET tương ứng với hai tệp hình ảnh:



Hình 2: Hai tệp hình ảnh từ hai yêu cầu GET

STT	Dòng	Time	Địa chỉ IP đích	File
1	165	11.729697	128.119.245.12	pearson.png
2	176	11.983178	2.56.99.24	8E_cover_small.jpg

Bảng 3: Các tệp hình ảnh được yêu cầu bởi trình duyệt

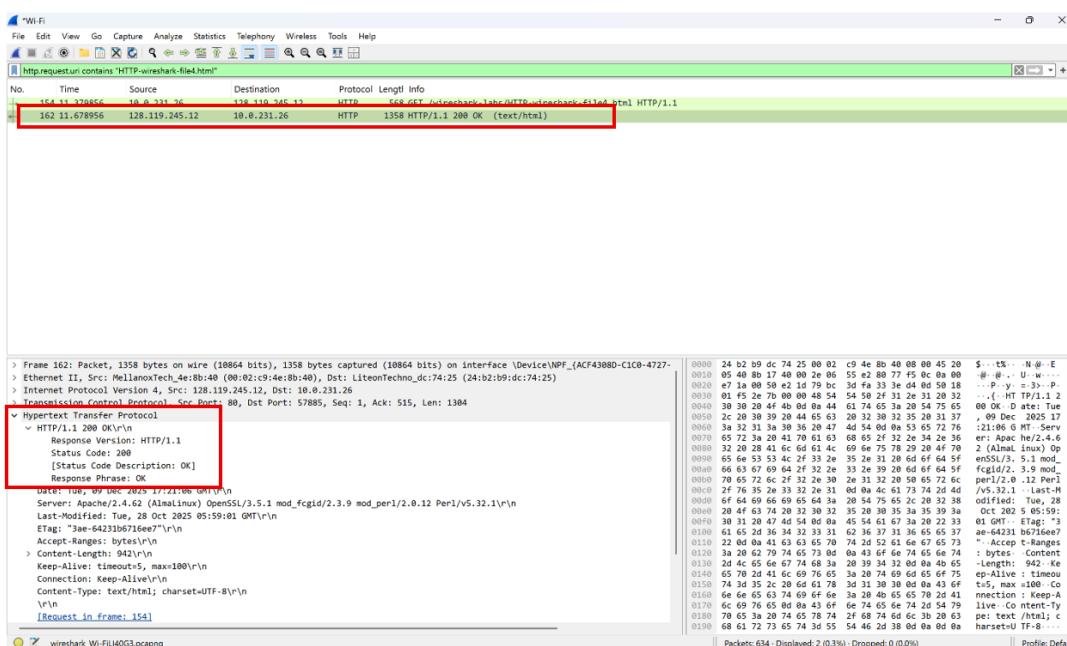
- Hai ảnh được lấy từ 2 địa chỉ IP đích khác nhau
- Chênh lệch thời gian giữa 2 yêu cầu GET: $11.983178 - 11.729697 = 0.253481$ giây → rất nhỏ
- Do hai ảnh được gửi trên hai địa chỉ IP đích khác nhau và hai yêu cầu GET được gửi cách nhau rất gần nên trình duyệt gửi hai hình ảnh đồng thời.

1.3 Locate the HTTP response message containing the content of the initial HTML page (HTTP-wireshark-file4.html). What is the status code and status phrase provided by the server?

- Để xác định gói HTTP Response chứa nội dung của trang HTML, tại ô Display Filter của Wireshark, nhập:

1 http.request.uri contains "HTTP-wireshark-file4.html"

- Trong Packet List, tìm gói có Info: HTTP/1.1 200 OK (đây là HTTP Response message từ server trả về nội dung HTML).
 - Nhấp vào gói HTTP Response đã chọn để mở mục Packet Details → Hypertext Transfer Protocol.
- Status Code: 200
 - Status Phrase: OK



Hình 3: Status code và status phase của gói HTTP

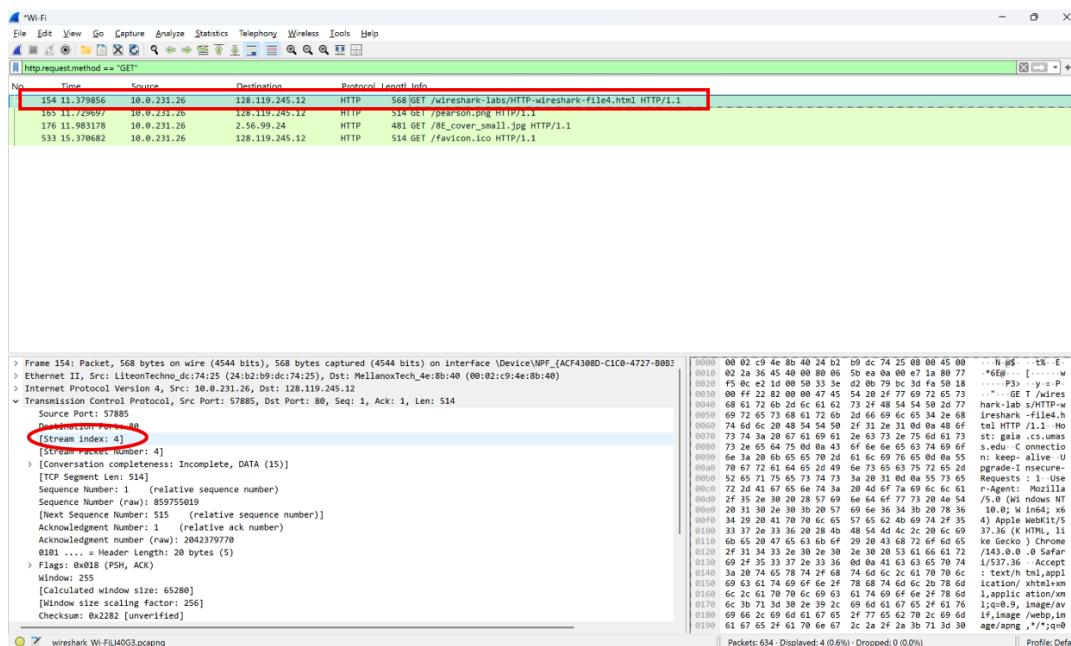
1.4 Based on your answer to Question 1, how many distinct TCP connections were established to fetch the HTML file and the two embedded images? Provide evidence by listing the unique Stream Index Numbers (e.g., tcp.stream eq X) that were used for these three objects

- Tại ô Display Filter của Wireshark, nhập:

```
1 http.request.method == "GET"
```

- Xác định TCP Stream cho file HTML:

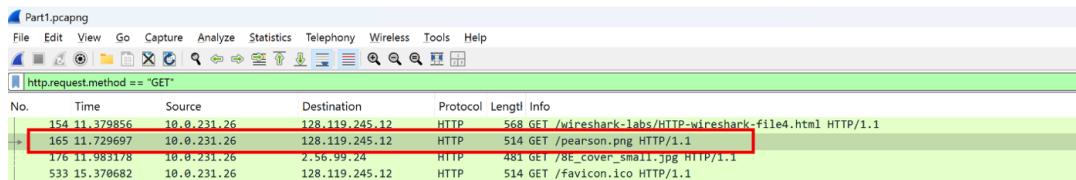
- Trong Packet List, tìm gói HTTP GET của file HTML.
- Nhấp vào gói HTTP Response đã chọn để mở mục Packet Details → Hypertext Transfer Protocol.
- Tìm mục Stream index để xác định TCP Stream của file HTML: `tcp.stream eq 4`.



Hình 4: TCP Stream cho file HTML

- Xác định TCP Stream cho 2 file ảnh:

- Trong Packet List, tìm gói HTTP GET của hai file hình ảnh.
- Nhấp vào gói HTTP Response đã chọn để mở mục Packet Details → Hypertext Transfer Protocol.
- Tìm mục Stream index để xác định TCP Stream của 2 file hình ảnh:
 - `pearson.png`: `tcp.stream eq 4`

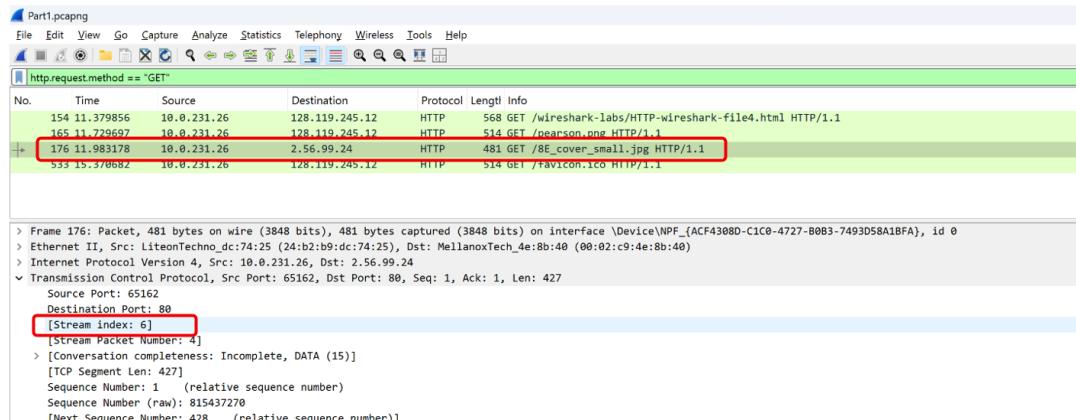


```
> Frame 165: Packet, 514 bytes on wire (4112 bits), 514 bytes captured (4112 bits) on interface \Device\NPF_{ACF4308D-C1C0-4727-B0B3-7493D58A1BFA}, id 0
> Ethernet II, Src: LiteonTechno_dc:74:25 (24:b2:b9:dc:74:25), Dst: MellanoxTech_4e:8b:40 (00:02:c9:4e:8b:40)
> Internet Protocol Version 4, Src: 10.0.231.26, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 57885, Dst Port: 80, Seq: 515, Ack: 1305, Len: 460
    Source Port: 57885
    Destination Port: 80
    [Stream index: 4]
    [Stream Packet Number: 8]
    > [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 460]
    Sequence Number: 515      (relative sequence number)
    Sequence Number (raw): 859755533
    ...

```

Hình 5: TCP Stream cho file pearson.png

- 8E_cover_small.jpg: `tcp.stream eq 6`



```
> Frame 176: Packet, 481 bytes on wire (3848 bits), 481 bytes captured (3848 bits) on interface \Device\NPF_{ACF4308D-C1C0-4727-B0B3-7493D58A1BFA}, id 0
> Ethernet II, Src: LiteonTechno_dc:74:25 (24:b2:b9:dc:74:25), Dst: MellanoxTech_4e:8b:40 (00:02:c9:4e:8b:40)
> Internet Protocol Version 4, Src: 10.0.231.26, Dst: 2.56.99.24
> Transmission Control Protocol, Src Port: 65162, Dst Port: 80, Seq: 1, Ack: 1, Len: 427
    Source Port: 65162
    Destination Port: 80
    [Stream index: 6]
    [Stream Packet Number: 4]
    > [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 427]
    Sequence Number: 1      (relative sequence number)
    Sequence Number (raw): 815437270
    [Next Sequence Number: 428  (relative sequence number)]

```

Hình 6: TCP Stream cho file 8E_cover_small.jpg

- Vậy có 2 kết nối TCP riêng biệt được thiết lập:

- `tcp.stream eq 4` (file HTML và file pearson.png)
- `tcp.stream eq 6` (file 8E_cover_small.jpg)

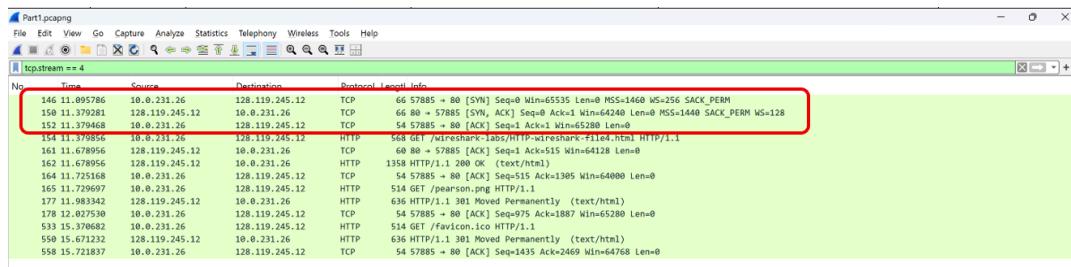
1.5 For the TCP connection that retrieved the initial HTML file, identify the three packets that form the TCP Three-Way Handshake.

List the TCP flags set in each of these three packets in order

- Lọc các gói TCP của kết nối HTML bằng Stream Index: 4.

- Trong mục Packet List, xác định 3 gói đầu tiên của kết nối TCP, đó chính là 3 gói tin tạo thành TCP Three-Way Handshake cần tìm:

STT	Dòng	Source	Destination	Source → Destination	TCP Flags
Gói 1	146	10.0.231.26	128.119.245.12	Client → Server	SYN
Gói 2	150	128.119.245.12	10.0.231.26	Server → Client	SYN, ACK
Gói 3	152	10.0.231.26	128.119.245.12	Client → Server	ACK



Hình 7: TCP Stream cho file 8E_cover_small.jpg

1.6 Select the largest data transfer packet (a packet with the PSH or ACK flag set and a large length) during the download of one of the image files. Examine the TCP Window Size Value in the packet details. What does this value represent, and why is it essential for the Transport Layer?

- Xét file ảnh 8E_cover_small.jpg với Stream index = 6.

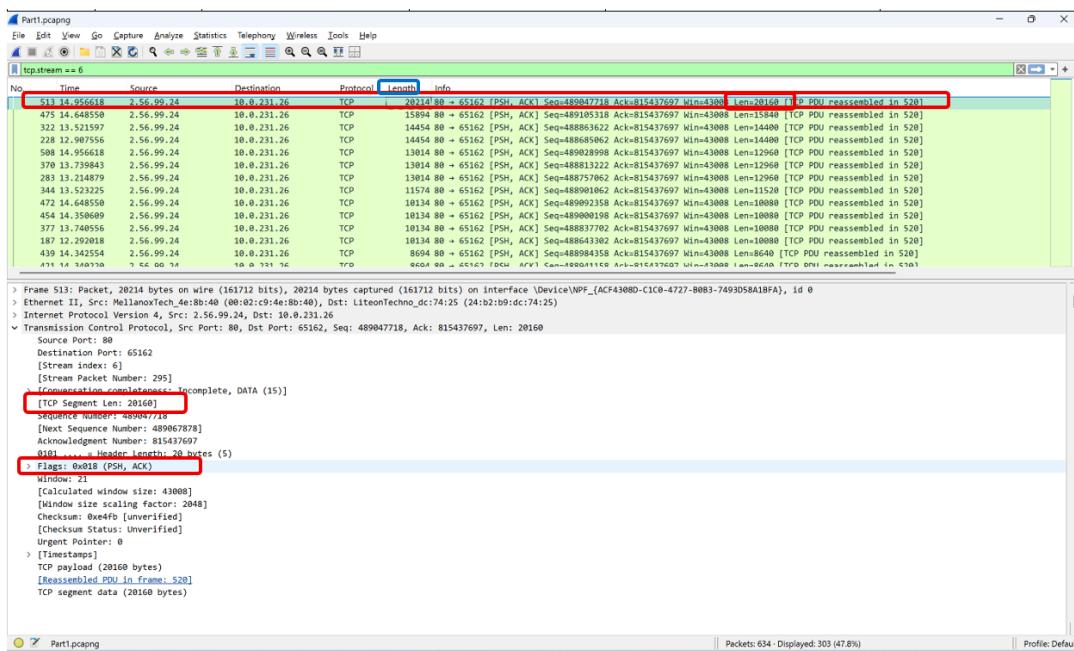
- Tại ô Display Filter của Wireshark, nhập:

```
1 tcp.stream == 6
```

- Nhập vào "Length" để sắp xếp theo thứ tự giảm dần, kiểm tra TCP Flags của gói có TCP Segment Len lớn nhất nếu thỏa đề thì đó là gói cần tìm.

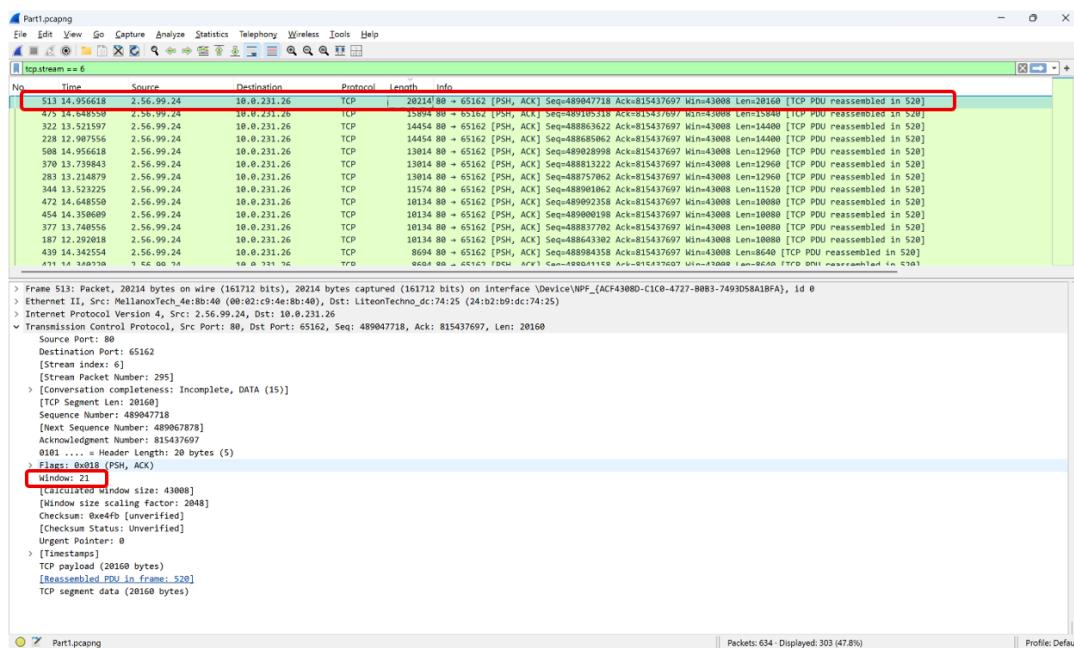
STT	Dòng	Source	Destination	TCP Segment Len	TCP Flags
1	513	2.56.99.24	10.0.231.26	20160	PHS, ACK

Network Packet Analysis with Wireshark



Hình 8: TCP Flags của gói tin có TCP Segment Len lớn nhất

- Click vào gói để mở mục Packet Details → Transmission Control Protocol
- Tìm mục Window để xác định TCP Window Size Value.



Hình 9: TCP Window Size Value

- TCP Window Size value: 21 bytes.

- TCP Window Size value đại diện cho dung lượng bộ đệm (buffer) sẵn có của bên nhận đã dành ra và sẵn sàng chấp nhận từ bên gửi tại thời điểm đó.
- Thông báo cho bên nhận rằng tại thời điểm này bên nhận có sẵn 21 bytes bộ đệm trống để chứa dữ liệu.
- Trong các mạng hiện đại, TCP Window Size value là giá trị cơ sở được sử dụng để tính toán kích thước cửa sổ thực tế. Nếu **Window Scaling** được kích hoạt, giá trị này sẽ được nhân với một hệ số mở rộng để tạo ra **Calculated window size**.

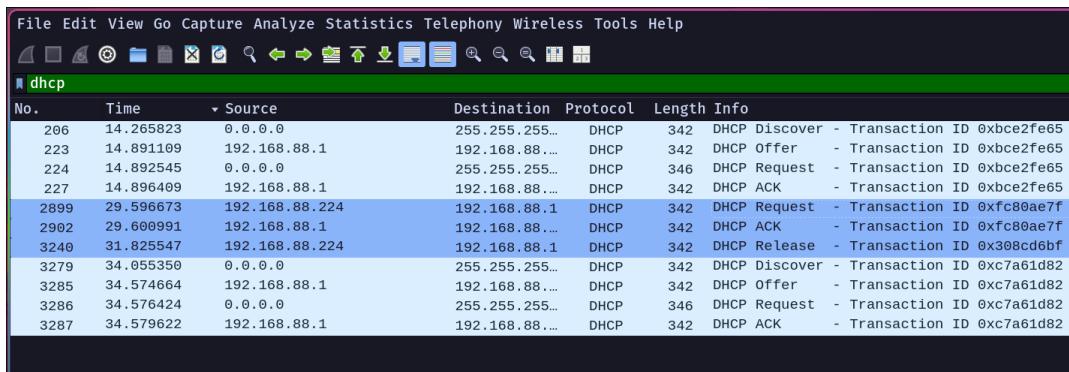
- Tầm quan trọng đối với Tầng Vận Chuyển:

- Ngăn chặn Tràn Bộ đệm: Giá trị này ngăn bên gửi truyền dữ liệu vượt quá khả năng xử lý của bên nhận. Nếu không có cơ chế này, bộ đệm của bên nhận sẽ bị tràn, dẫn đến mất gói tin và phải truyền lại, gây giảm hiệu suất mạng nghiêm trọng.
- Tính toán Cửa sổ Thực tế: Trong mạng hiện đại, giá trị này được nhân với một Hệ số Mở rộng (Window Scale Factor) để tạo ra **Calculated window size**.
- **Window size value** là giá trị duy nhất trong TCP Header cho phép bên nhận điều khiển trực tiếp tốc độ truyền dữ liệu của bên gửi, một trách nhiệm không thể thiếu của Tầng Vận chuyển.

2 Part 2: Analyzing DHCP Traffic (3.5 pt)

2.1 Were any ARP packets sent or received during the DHCP packet-exchange period? If so, explain the purpose of those ARP packets

- Xác định khoảng thời gian trao đổi DHCP, bằng cách lọc các gói tin DHCP trong Wireshark:



Hình 10: Lọc các gói tin DHCP trong Wireshark

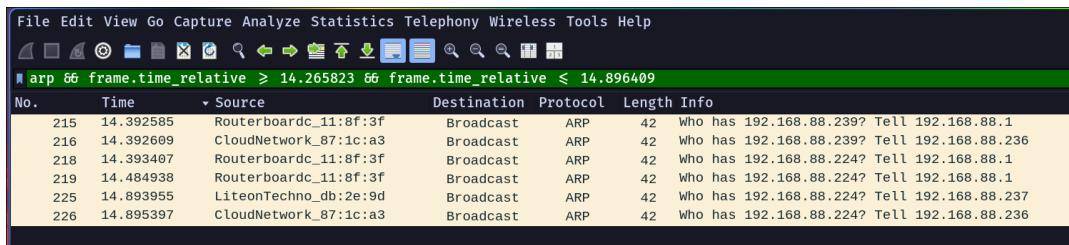
- Có thể thấy khoảng thời gian trao đổi DHCP (DHCP Discover → DHCP ACK) là:

1. **14.265823 → 14.896409**
2. **34.055350 → 34.579622**

- Lọc các gói ARP trong khoảng thời gian này bằng cách sử dụng bộ lọc:

- ARP trong DHCP Exchange 1:

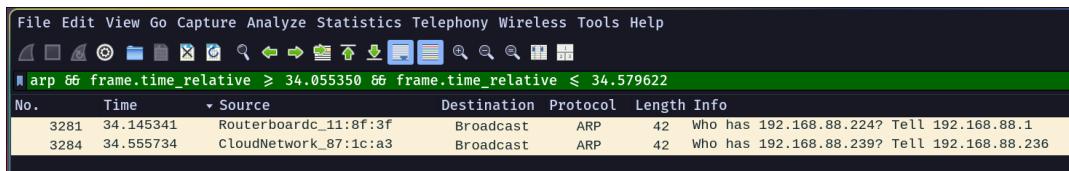
```
1 arp && frame.time_relative >= 14.265823 && frame.time_relative <=
    14.896409
```



Hình 11: Các gói tin ARP trong DHCP Exchange 1

- ARP trong DHCP Exchange 2:

```
1 arp && frame.time_relative >= 34.055350 && frame.time_relative <=
    34.579622
```



The screenshot shows a Wireshark capture window with the following details:

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

arp && frame.time_relative > 34.055350 && frame.time_relative < 34.579622

No.	Time	Source	Destination	Protocol	Length	Info
3281	34.145341	Routerboardc_11:8f:3f	Broadcast	ARP	42	Who has 192.168.88.224? Tell 192.168.88.1
3284	34.555734	CloudNetwork_87:1c:a3	Broadcast	ARP	42	Who has 192.168.88.239? Tell 192.168.88.236

Hình 12: Các gói tin ARP trong DHCP Exchange 2

- Nhiều gói ARP xuất hiện trong suốt quá trình chuyển đổi DHCP.

- Mục đích các gói ARP trên:

- ARP Request từ router/AP (**Routerboardc_11:8f:3f**): Được sử dụng để phân giải địa chỉ MAC của các host trong mạng cục bộ, nhằm duy trì và cập nhật bảng ARP của router/AP trong quá trình các thiết bị gia nhập hoặc rời mạng.
- ARP Request từ client (**CloudNetwork_87:1c:a3**): Được gửi sau khi client nhận DHCP ACK để kiểm tra xem địa chỉ IP vừa được cấp có đang được thiết bị khác sử dụng hay không (ARP Probe), đảm bảo tính duy nhất của địa chỉ IP trước khi client chính thức sử dụng.
- ARP broadcast từ các thiết bị khác trong mạng: Xuất hiện do các thiết bị khác thực hiện phân giải địa chỉ MAC hoặc làm mới bảng ARP của chúng.

2.2 A host uses DHCP to obtain an IP address, among other configuration parameters. However, the host's IP address is not finalized until the completion of the four-message DHCP exchange. If the IP address is not assigned until the final message, what IP address values are used in the IP datagrams that carry these messages?

DHCP Message	Source IP	Destination IP	Giải thích
DHCP Discover	0.0.0.0	255.255.255.255	Client chưa có IP nên dùng địa chỉ không xác định (0.0.0.0) và gửi broadcast để tìm DHCP server.
DHCP Offer	IP của DHCP server	255.255.255.255	Server trả lời bằng broadcast vì client chưa thể nhận unicast.
DHCP Request	0.0.0.0	255.255.255.255	Client vẫn chưa được cấp IP chính thức, gửi broadcast để chấp nhận địa chỉ đề nghị.
DHCP ACK	IP của DHCP server	255.255.255.255 (hoặc unicast)	Server xác nhận cấp IP; client chỉ sử dụng IP sau khi nhận ACK.

Bảng 4: Bảng địa chỉ IP dùng trong 4 thông điệp DHCP

- Trong suốt quá trình DHCP, client chưa được phép sử dụng địa chỉ IP được đề nghị cho đến khi nhận DHCP ACK.
- Vì vậy, trong DHCP Discover và DHCP Request, client không thể dùng IP riêng và bắt buộc dùng source IP = 0.0.0.0
- Các gói DHCP được gửi dưới dạng broadcast (255.255.255.255) để đảm bảo DHCP server nhận được và client có thể nhận phản hồi dù chưa có IP hợp lệ
- Chỉ sau khi DHCP ACK được nhận, client mới gán IP được cấp vào interface và bắt đầu gửi các gói IP thông thường (ARP, ICMP, TCP, UDP...).

2.3 If, after receiving the ACK, the client sent an ARP probe for this new IP but received an ARP Reply from another device, what specific action is the client required to take according to the DHCP standard (RFC 2131)

- Client phải hủy IP vừa được cấp (decline) và gửi DHCPDECLINE cho server.

- Theo RFC 2131:

- nếu ARP probe trả về reply → IP bị xung đột
- client phải báo DHCPDECLINE
- sau đó quay lại trạng thái INIT và bắt đầu lại quá trình DHCP Discover.

2.4 Why does the DHCP Request message still use UDP Port 68 as the source port and UDP Port 67 as the destination port, even though a unique, high-numbered ephemeral port could technically be used? Explain how UDP's connectionless nature makes this fixed port usage simple and robust.

- UDP là connectionless → không cần thiết lập phiên: DHCP hoạt động ngay cả khi host chưa có IP, chưa có routing, chưa có state. Port cố định đảm bảo mọi thiết bị đều biết nơi gửi và nhận frame.

- Chuẩn hóa để tất cả DHCP server/relay có thể hoạt động thống nhất

- Client luôn gửi từ port 68
- Server luôn lắng nghe tại port 67

- Relay Agent (DHCP Relay) cần port cố định: Các router chuyển tiếp DHCP cần biết chính xác port để forward.

- Loại bỏ lỗi tương thích: Nếu dùng port ngẫu nhiên → server không biết nên gửi ACK về đâu.

2.5 UDP checksum

2.5.1 In the DHCP Discover packet, expand the User Datagram Protocol (UDP) layer and examine the Checksum field. Does Wireshark display the message "[UDP checksum is good]" or "[UDP checksum is incorrect]"?

- Trong Windows và Linux, Wireshark thường hiển thị: "[UDP checksum is good]" hoặc "Checksum validation disabled for UDP" (tùy OS checksum offloading) Dựa trên screenshot của bạn, gói không có lỗi → checksum GOOD.

-

2.5.2 If the checksum had been calculated as incorrect, what action would the recipient's Transport Layer (UDP handler) have immediately taken, and why would this lead the DHCP process to stall?

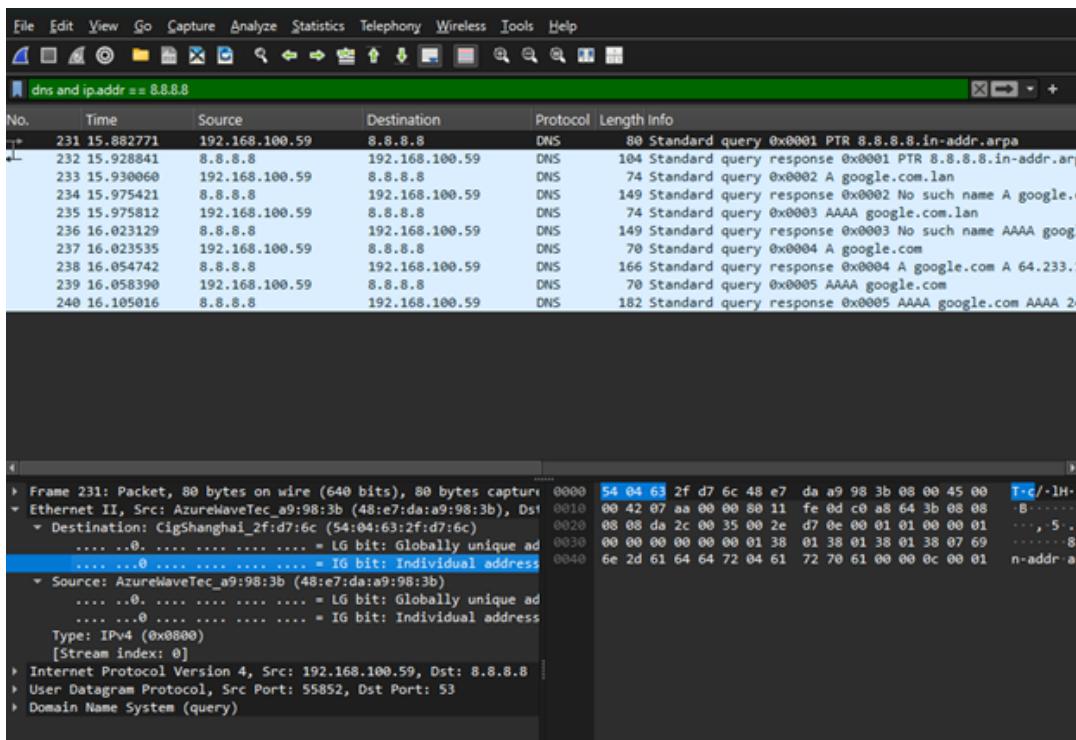
- UDP handler sẽ:
 - Loại bỏ gói (drop silently): UDP không tự sửa lỗi, không retransmission. Nó chỉ kiểm tra checksum → nếu sai → discard.
 - DHCP stall vì client không nhận Offer hoặc ACK: DHCP dựa vào chuỗi 4 gói BOOTP (Discover–Offer–Request–ACK). Nếu một gói bị drop:
 - Client không nhận Offer → phải timeout và gửi Discover lại.
 - Server không nhận Request → không cấp IP
 - Toàn bộ chu trình DHCP dừng hoặc timeout (30–60 giây)
 - DHCP không có cơ chế “retry nhanh” vì UDP không đảm bảo delivery.

3 Part 3: Network and Link Layer Analysis (3 pt)

3.1 Source/Destination IP: Locate the DNS Query packet sent from your host to the Google DNS server

- Để tìm vị trí truy vấn gói tin DNS được gửi từ máy chủ đến máy chủ DNS của server, ta lọc các ip của máy chủ DNS của Google (8.8.8.8), tại ô Display filter của Wireshark, nhập:

1 dns and ip.addr == 8.8.8.8



Hình 13: Lọc gói tin DNS với ip của Google DNS server

- Sau khi lọc ta có thể dễ dàng nhìn thấy tương tác giữa máy chủ với máy chủ DNS của Google. Qua hình trên ta biết:

3.1.1 What is the Source IP Address?

- Địa chỉ IP nguồn (Host): 192.168.100.59

3.1.2 What is the Destination IP Address?

- Địa chỉ IP đích (Google DNS) : 8.8.8.8

3.1.3 Explain why these Source and Destination IP addresses will remain unchanged as the packet travels across the internet to the Google DNS server.

- Các router mạng sử dụng địa chỉ IP đích để đưa ra quyết định định tuyến. Nếu một router thay đổi địa chỉ IP nguồn hoặc đích thì nó sẽ phá vỡ tính toàn vẹn (End-to-end Integrity). Nghĩa là, Khi địa chỉ IP đích bị thay đổi trong lúc truyền, gói tin sẽ không bao giờ đến được

dịch thực sự của nó. Và ngược lại, khi địa chỉ nguồn bị thay đổi, máy chủ sẽ gửi trả lời đến một địa chỉ sai.

Source	Destination
192.168.100.59	8.8.8.8
8.8.8.8	192.168.100.59
192.168.100.59	8.8.8.8
8.8.8.8	192.168.100.59
192.168.100.59	8.8.8.8
8.8.8.8	192.168.100.59
192.168.100.59	8.8.8.8
8.8.8.8	192.168.100.59
192.168.100.59	8.8.8.8
8.8.8.8	192.168.100.59

Hình 14: Tính toán vẹn end-to-end của địa chỉ IP

3.2 Find the Time to Live (TTL) field in the IP header of the DNS Query. What does the initial value of the TTL represent, and how does your local router/gateway modify this value when forwarding the packet?

3.2.1 Find the Time to Live (TTL) field in the IP header of the DNS Query.

```

▼ Internet Protocol Version 4, Src: 192.168.100.59, Dst: 8.8.8.8
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECN)
    Total Length: 56
    Identification: 0x07ae (1966)
  ▶ 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: UDP (17)
    Header Checksum: 0xfe13 [validation disabled]
    [Header checksum status: Unverified]
  
```

Hình 15: TTL Field in IP Header of DNS Query

- Time to Live (TTL): 128

3.2.2 Giá trị TTL ban đầu đại diện cho điều gì?

- Giá trị TTL ban đầu (initial TTL) đại diện cho số lượng bước nhảy (hops) tối đa mà gói tin được phép đi qua các router trên mạng trước khi nó bị hủy. Nó ngăn các gói tin vào vòng lặp (packet loop) vô tận tại tầng mạng. Thông thường, TTL mặc định cho Windows là 128, đối với Linux và macOS là 64.

3.2.3 Router điều chỉnh TTL như nào

- Nếu có packet loop trong mạng, khi nó chạm vào cổng định tuyến (routing interface) giá trị của TTL giảm đi 1 và cuối cùng vòng lặp này bị huỷ khi TTL giảm về còn 0 và trả về thông báo ICMP loại 11 và mã 0 (quá thời gian).

3.3 Find the MAC Address of your local router/gateway (either by using arp -a in the command prompt or by finding the MAC address associated with the Gateway IP in your capture)

- MAC Address : 54-04-63-2f-d7-6c
- Địa chỉ MAC hiện thị qua Wireshark:



Hình 16: MAC Address trong Ethernet Header của DNS Query

- Tìm địa chỉ MAC hiện thị qua command prompt: (địa chỉ IP đầu tiên trong bảng)

Internet Address	Physical Address	Type
192.168.100.1	54-04-63-2f-d7-6c	dynamic
192.168.100.255	ff-ff-ff-ff-ff-ff	static
224.0.0.2	01-00-5e-00-00-02	static
224.0.0.22	01-00-5e-00-00-16	static
224.0.0.251	01-00-5e-00-00-fb	static
224.0.0.252	01-00-5e-00-00-fc	static
239.255.255.250	01-00-5e-7f-ff-fa	static
255.255.255.255	ff-ff-ff-ff-ff-ff	static

Hình 17: MAC Address trong Command Prompt

3.4 What are the Source and Destination addresses in the Link Layer header (Ethernet or 802.11)? How are these addresses fundamentally different from the IP addresses you found?

3.4.1 What are the Source and Destination addresses in the Link Layer header (Ethernet or 802.11)?

- Source address :48:e7:da:a9:98:3b
- Destination address: 54:04:63:2f:d7:6c

▼ Ethernet II, Src: AzureWaveTec_a9:98:3b (48:e7:da:a9:98:3b), Dst: CigShanghai_2f:d7:6c (54:04:63:2f:d7:6c)
▼ Destination: CigShanghai_2f:d7:6c (54:04:63:2f:d7:6c)
.... ..0. = LG bit: Globally unique address (factory default)
.... ..0. = IG bit: Individual address (unicast)
▼ Source: AzureWaveTec_a9:98:3b (48:e7:da:a9:98:3b)
.... ..0. = LG bit: Globally unique address (factory default)
.... ..0. = IG bit: Individual address (unicast)
Type: IPv4 (0x0800)
[Stream index: 0]

Hình 18: Source and Destination MAC Address in Link Layer Header

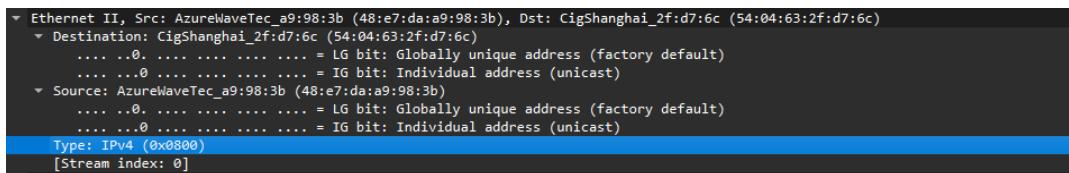
3.4.2 How are these addresses fundamentally different from the IP addresses you found?

Đặc điểm	Địa chỉ MAC	Địa chỉ IP
Tầng mạng	Tầng 2 - Data Link (TCP/IP)	Tầng 3 - Network (OSI)
Tính chất	Được mã hoá cứng vào card mạng	Được gán động hoặc tinh thông qua cấu hình phần mềm.
Phạm vi sử dụng	Dùng để truyền dữ liệu giữa các thiết bị kết nối trên cùng một mạng LAN (hop-by-hop)	Dùng để định tuyến gói tin qua Internet từ nguồn đến đích (end-to-end)
Tính ổn định	MAC nguồn và đích bị thay thế tại mỗi router	IP nguồn và IP đích giữ nguyên từ thiết bị gửi đến thiết bị nhận
Dịnh dạng	48 bit (6 octet)	IPv4 32 bit hoặc IPv6 128 bit

Bảng 5: So sánh địa chỉ MAC và địa chỉ IP

3.5 Examine the Link Layer header (Ethernet II or similar). What is the Type field's value, and what does it tell the receiving device (your router/gateway) about the protocol that follows?

3.6 Giá trị của trường Type là gì?



Hình 19: Trường Type trong Link Layer Header

- Giá trị của trường Type là: 0x0800

3.7 Ý nghĩa của Trường Type với thiết bị nhận (gateway/router)

- Quyết định giao thức cho tầng trên : Trường Type cho Router biết loại giao thức nào được đóng gói bên trong phần Dữ liệu (Payload) của khung Ethernet.
- Là điểm kết nối giữa **Layer 2** và **Layer 3** trong mô hình TCP/IP: Sau khi một thiết bị (như router) xử lý xong tiêu đề Tầng 2 (Địa chỉ MAC), nó cần biết giao thức Tầng 3 nào được đóng gói bên trong. Trường type cung cấp thông tin cho việc này.

- Cho router/gateway biết cách xử lý gói tin: Giá trị 16-bit trong trường Type cho thiết bị nhận biết giao thức tầng trên đang được sử dụng là gì.