Họ và tên: Phùng Thị Linh

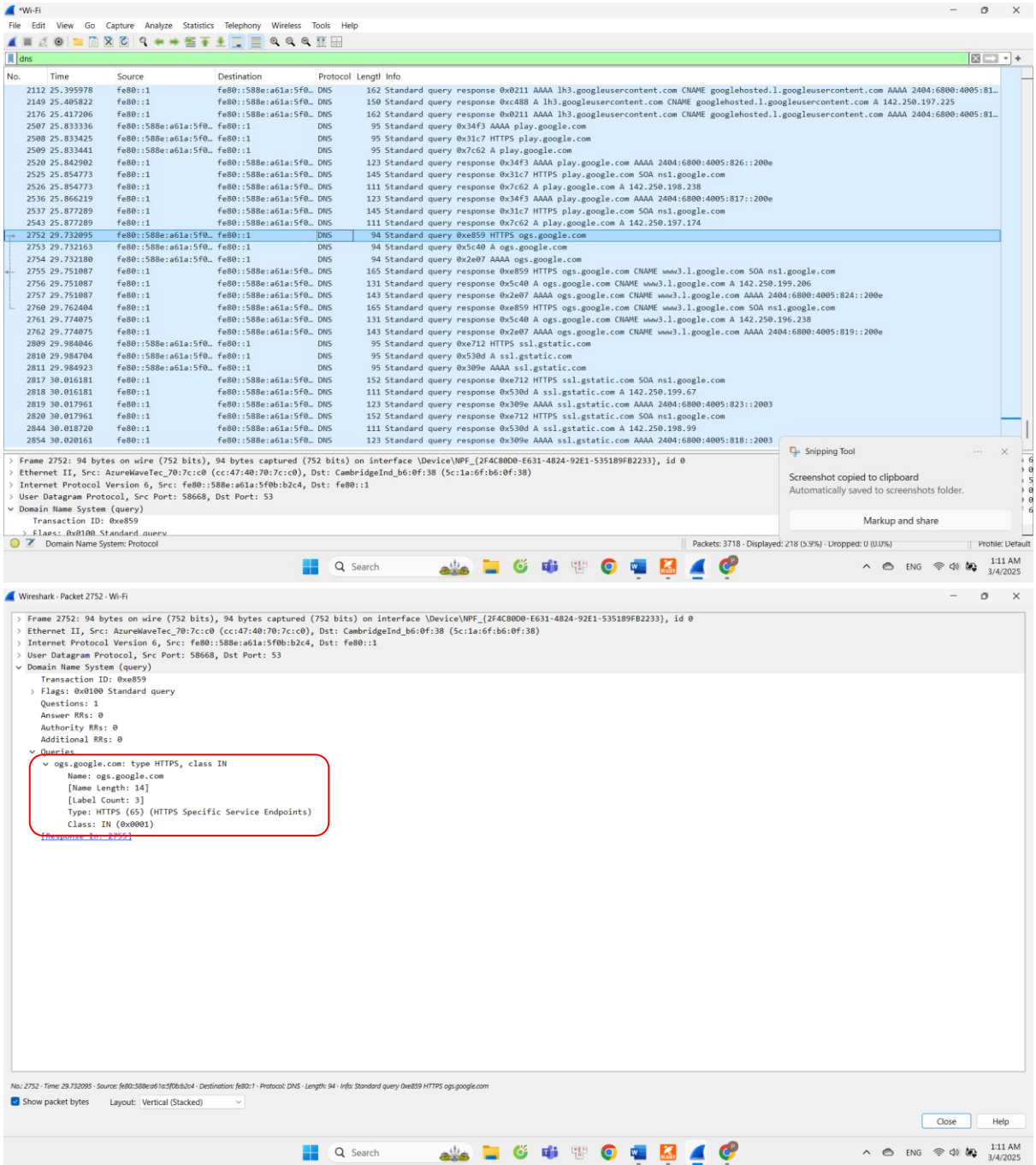
Mã sinh viên : 22174600001

Lớp : DHKL16A1HN

BÁO CÁO BÀI TẬP THỰC HÀNH CHƯƠNG 1

1. Bắt gói tin DNS bằng Wireshark

Kết quả:



2. Bắt và phân tích quá trình bắt tay 3 bước TCP bằng Wireshark

Bước 1: Mở Wireshark và bắt đầu thu thập gói tin

1. Mở Wireshark

2. Chọn card mạng đang sử dụng kết nối Internet (Wi-Fi hoặc Ethernet).

3. Nhập bộ lọc để chỉ hiển thị gói tin TCP liên quan đến quá trình bắt tay:

3.1. Nếu muốn lọc gói SYN hoặc ACK trong Capture Filter (ô nhập đ/k lọc), dùng cú pháp sau:

tcp[tcpflags] & (tcp-syn|tcp-ack) != 0

3.2. Nếu muốn lọc sau khi đã bắt gói tin (Display Filter), dùng cú pháp:

tcp.flags.syn == 1 || tcp.flags.ack == 1

Lưu ý: Display Filter chỉ hoạt động sau khi đã bắt gói tin xong.

Bước 2: Khởi tạo kết nối TCP

Cách 1: Truy cập một trang web bằng trình duyệt

- Mở trình duyệt và nhập một URL (ví dụ: http://www.example.com).
- Khi nhấn Enter, trình duyệt sẽ thực hiện kết nối TCP đến máy chủ web.

Cách 2: Sử dụng telnet để kết nối đến một máy chủ

- Mở Command Prompt (Windows) hoặc Terminal (Linux/macOS).
- Nhập lệnh sau để mở kết nối TCP đến cổng 80 (HTTP) của Google

  telnet www.google.com 80

Nếu telnet hiển thị Connected to www.google.com, nghĩa là kết nối TCP đã được thiết lập.

Bước 3: Phân tích gói tin trong Wireshark

Sau khi thực hiện một trong các bước trên, quay lại Wireshark và dừng thu thập gói tin. nhấn Stop Capture (nút vuông đỏ) sau khi lệnh Telnet thực hiện xong.

Chúng ta sẽ thấy một loạt gói TCP.

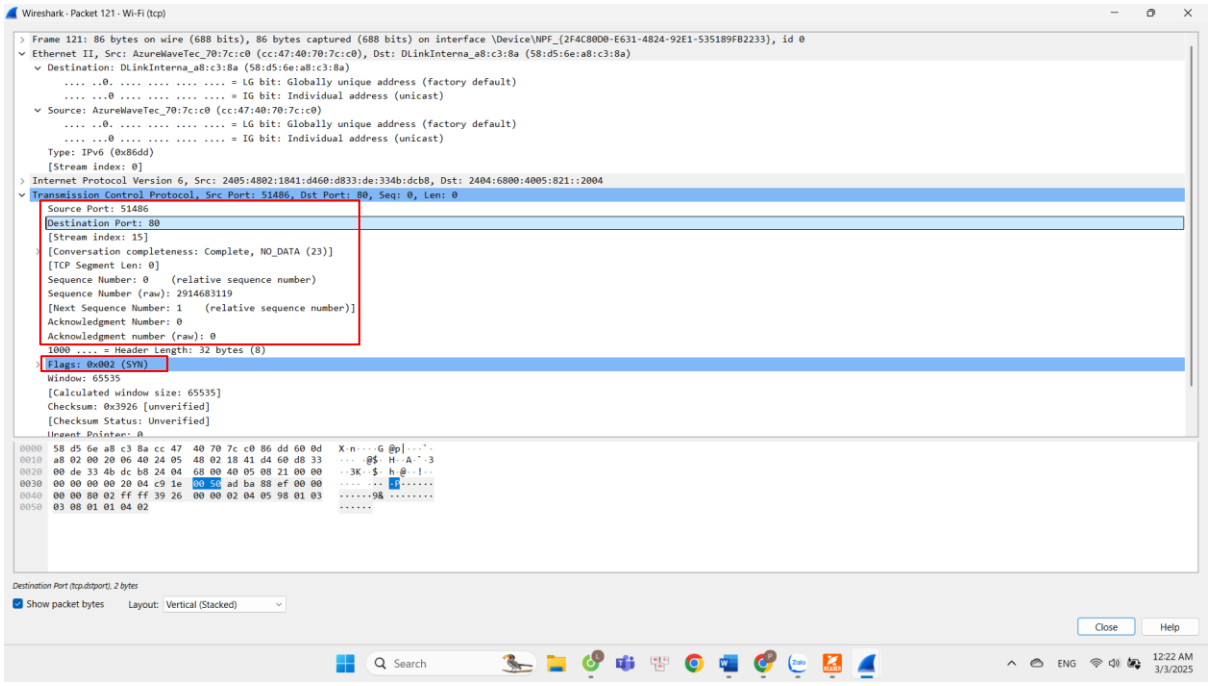
Trong ô Display Filter, nhập bộ lọc sau để chỉ hiển thị gói SYN:

tcp.flags.syn == 1 && tcp.flags.ack == 0

Nhấn Enter, bạn sẽ thấy gói SYN đầu tiên được gửi từ máy của mình đến www.google.com.

Bước 4: Phân tích gói SYN

❑ Nhấp vào gói SYN để xem chi tiết.



Bước 5: Tìm gói SYN-ACK và ACK để quan sát toàn bộ bắt tay 3 bước

❑ Tìm gói SYN-ACK từ Google:

Nhập bộ lọc:

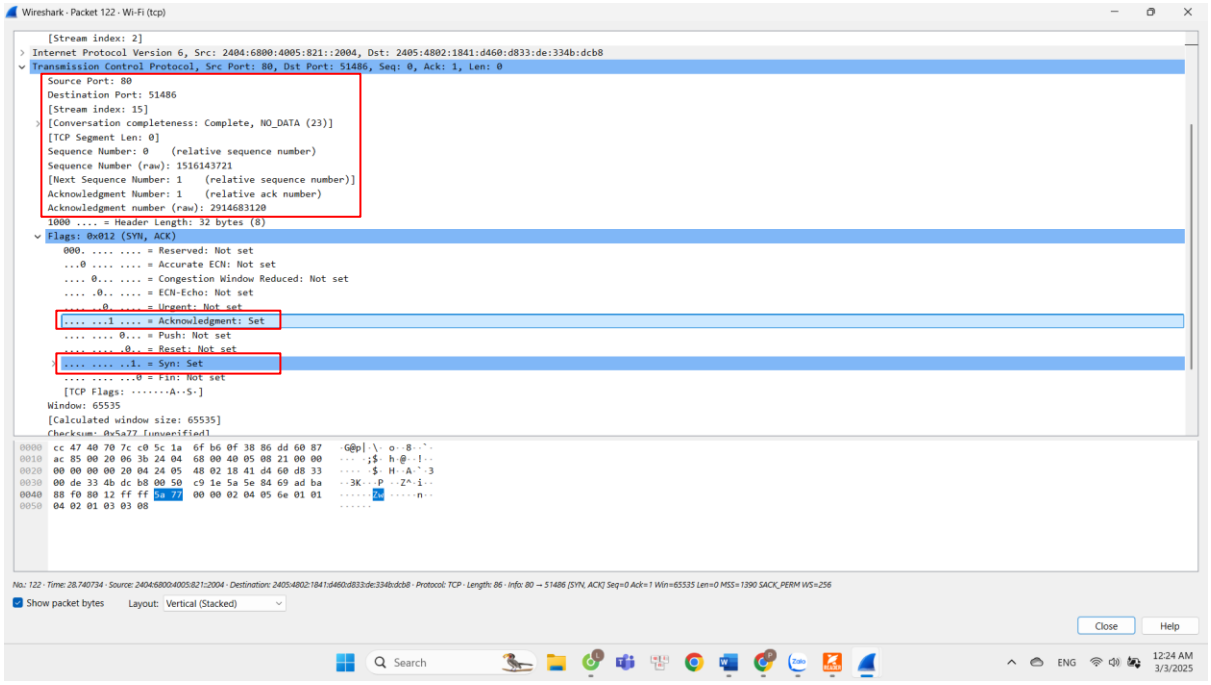
tcp.flags.syn == 1 && tcp.flags.ack == 1

Kiểm tra:

• Flags: SYN = 1, ACK = 1

• Acknowledgment Number: x + 1 (phản hồi từ Google).

Kết quả :



❑Tìm gói ACK từ máy người dùng:

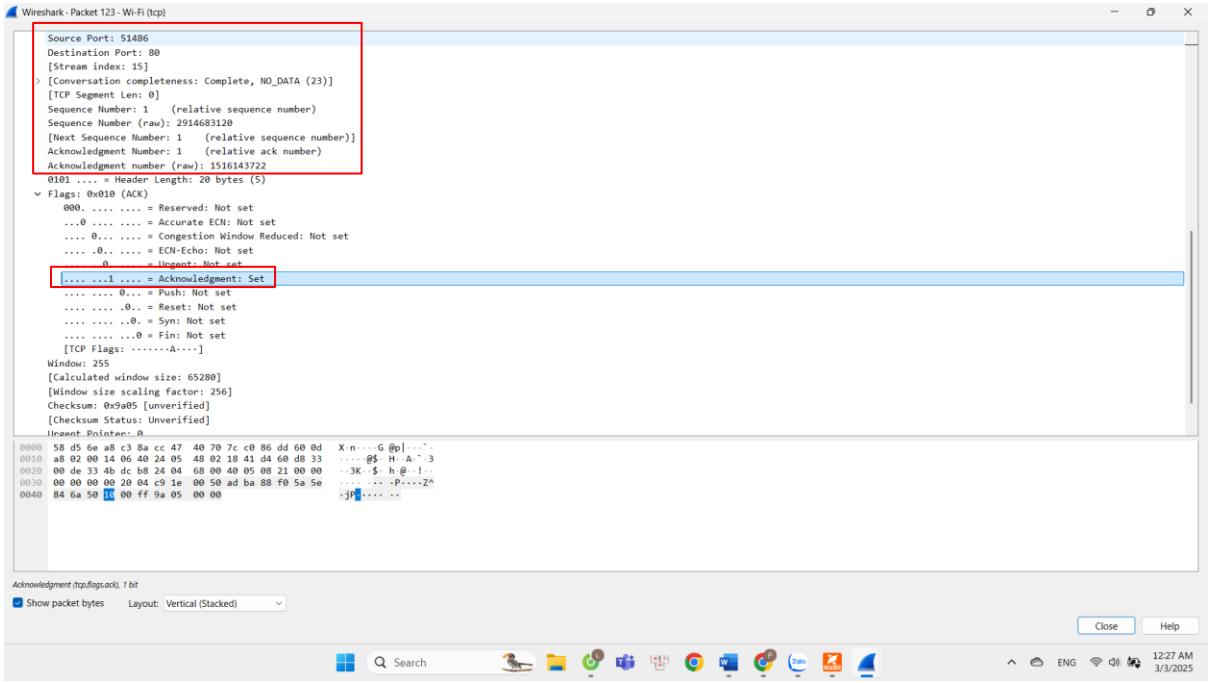
tcp.flags.ack == 1 && tcp.flags.syn == 0

Kiểm tra

Flags: ACK = 1

Acknowledgment Number: y + 1 (phản hồi từ máy người dùng).

Kết quả:



Tổng kết

Chạy Wireshark trước khi thực hiện lệnh telnet www.google.com 80 để ghi lại gói SYN.

Dùng tcp.flags.syn == 1 && tcp.flags.ack == 0 để lọc gói SYN.

Dùng tcp.flags.syn == 1 && tcp.flags.ack == 1 để tìm SYN-ACK.

Dùng tcp.flags.ack == 1 && tcp.flags.syn == 0 để tìm ACK.

Ta sẽ thấy toàn bộ quá trình bắt tay 3 bước TCP!

3. Bắt gói tin Ethernet bằng Wireshark để phân tích các trường dữ liệu.

Bước 1: Mở Wireshark và chọn giao diện mạng

1.Mở Wireshark.

2.Chọn giao diện mạng đang sử dụng (Ethernet hoặc Wi-Fi).

3.Nhấn Start để bắt gói tin.

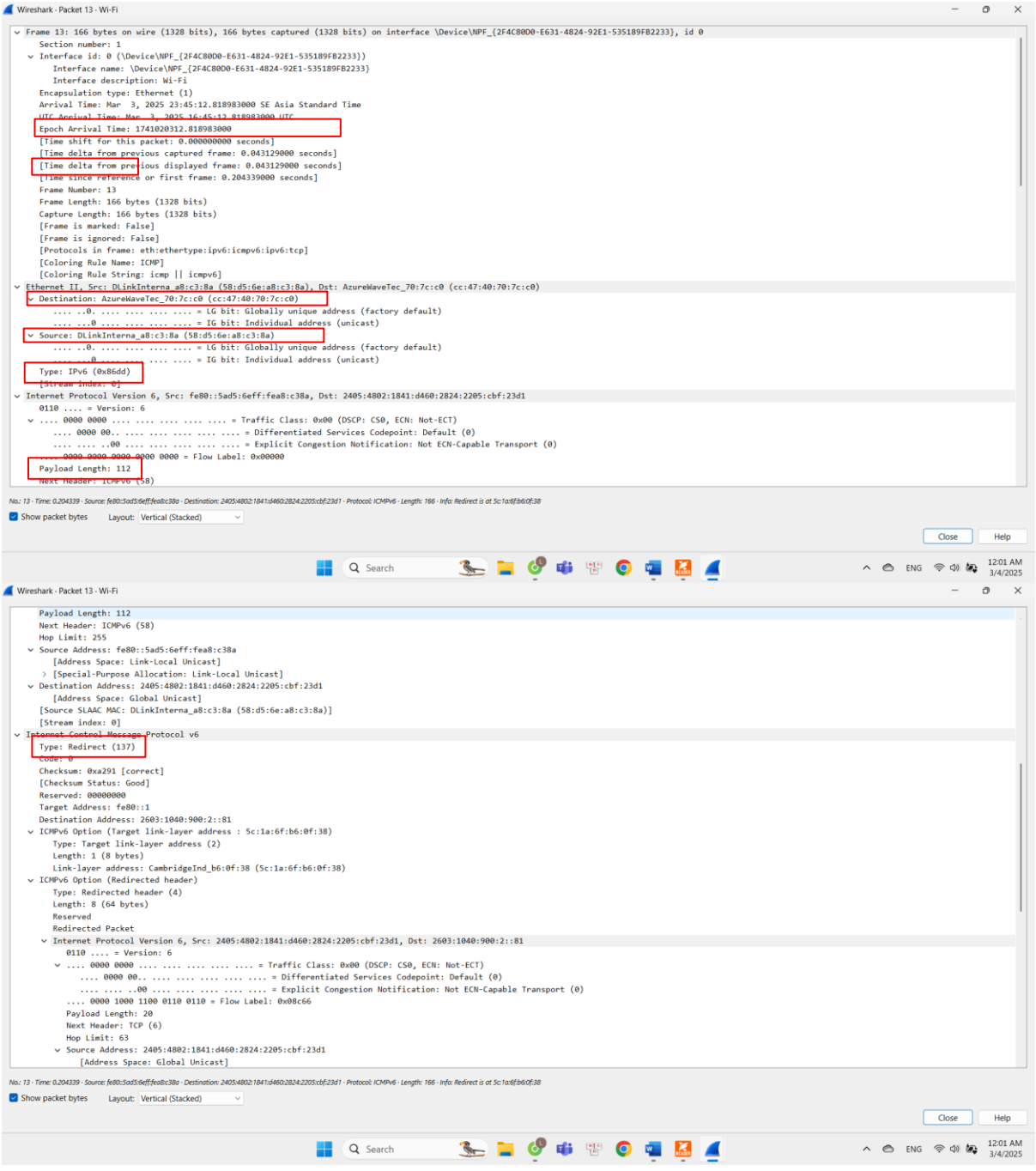
Bước 2: Bắt gói tin Ethernet

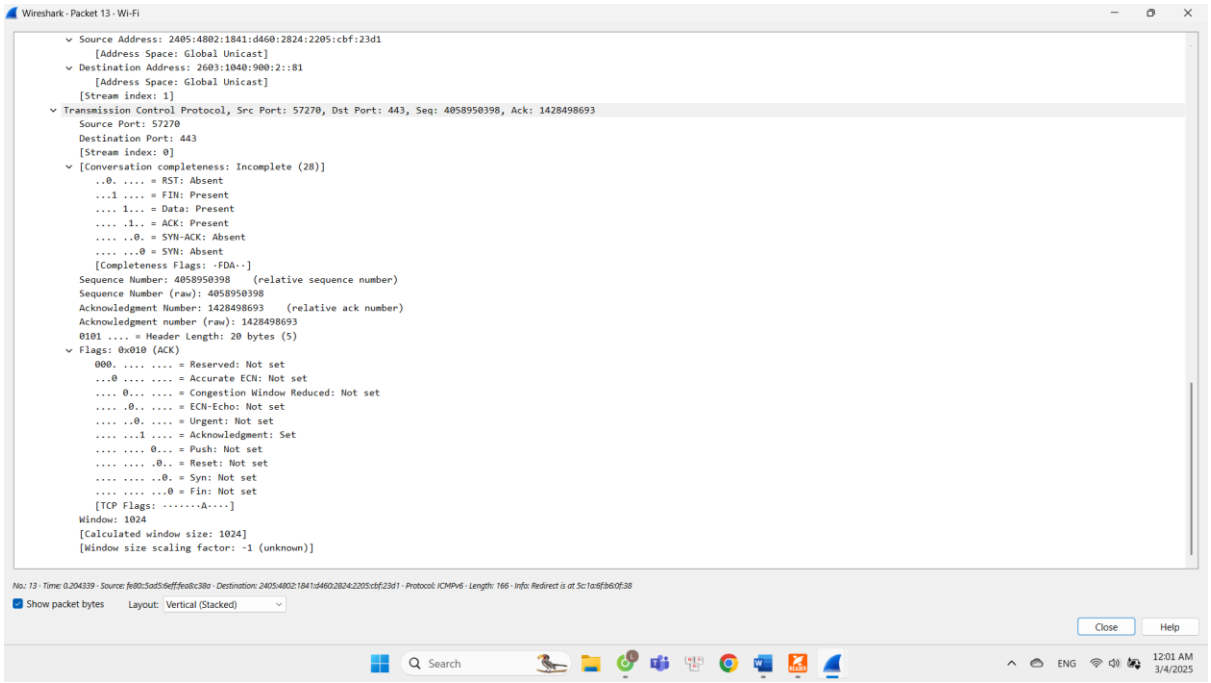
Để lọc chỉ các gói tin Ethernet, nhập vào thanh Filter:

- ethernet
- eth.dst == xx:xx:xx:xx:xx:xx để lọc theo địa chỉ MAC đích cụ thể.

Quan sát danh sách gói tin được bắt.

Bước 3: Phân tích gói tin Ethernet





1. Địa chỉ MAC nguồn và đích:

- MAC nguồn: DLinkInterna_a8:c3:8a (58:d5:6e:a8:c3:8a)
- MAC đích: AzureWaveTec_70:7c:c0 (cc:47:40:70:7c:c0)

2. Gói tin này sử dụng giao thức IPv6. Giao thức cụ thể là ICMPv6. Loại ICMPv6: 137 (Redirect).

3. Trường Padding

- Không có thông tin về Padding.
- Padding thường dùng để đảm bảo độ dài tối thiểu của khung Ethernet (thường là 64 bytes). Nếu gói tin đã đủ dài, sẽ không cần Padding.

4. Kiểm tra CRC (Cyclic Redundancy Check)

- Không có cảnh báo về lỗi CRC, nên có thể gói tin có CRC hợp lệ.
- Hoặc Wireshark không hiển thị thông tin này do card mạng đang tắt kiểm tra CRC.
- Một số giao diện mạng (như Wi-Fi) thường không hỗ trợ kiểm tra CRC do cách chúng thu thập dữ liệu.

Bước 4: Kiểm tra loại gói tin Ethernet

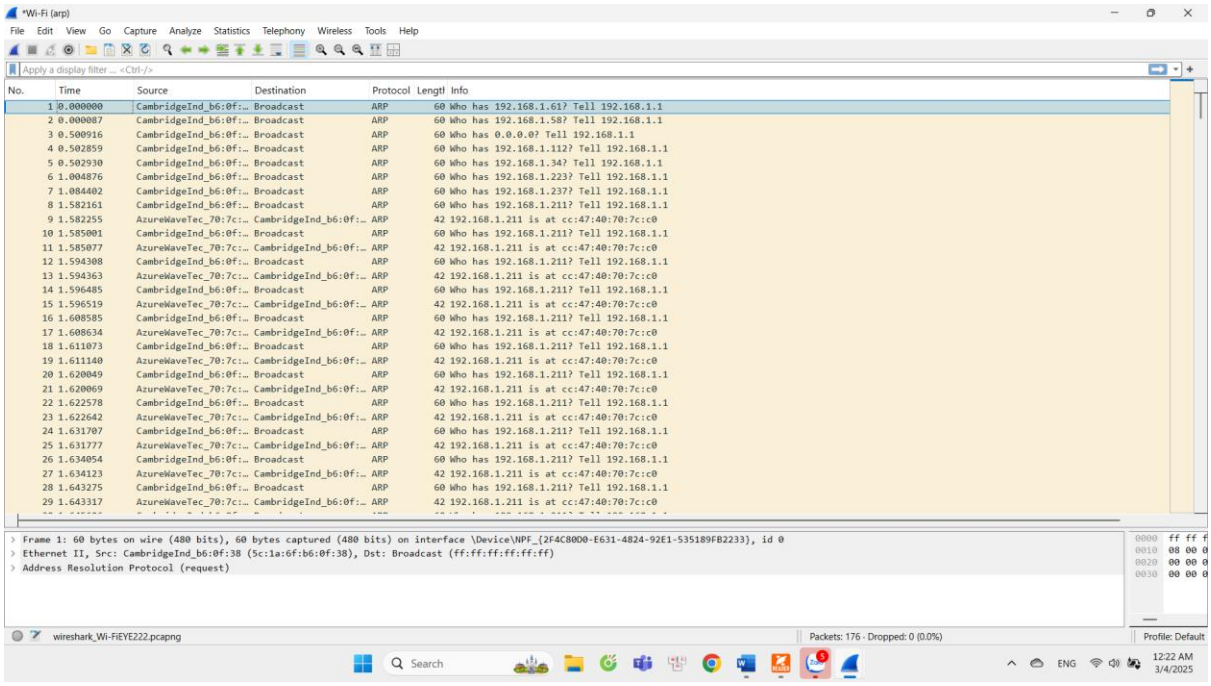
Trong ảnh gói tin có EtherType = 0x86DD, tức là IPv6.

Dựa trên quy tắc phân loại:

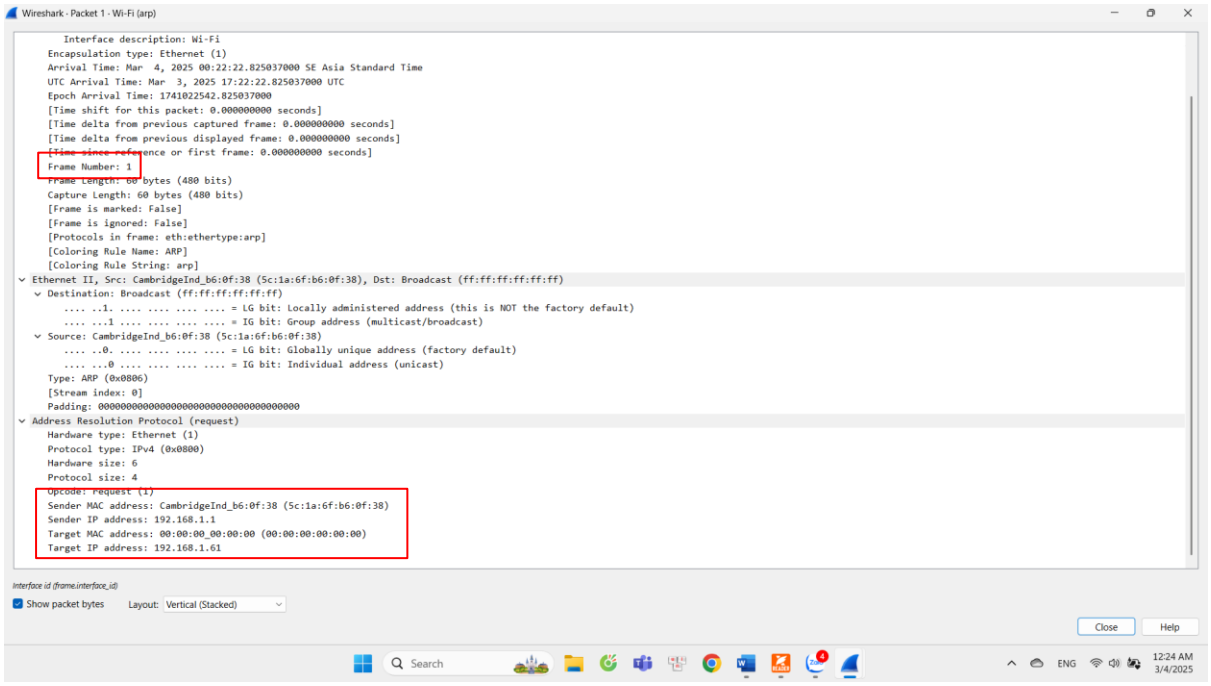
- EtherType ≥ 1536 (0x0600) → Ethernet II.
- EtherType < 1536 → IEEE 802.3 (sử dụng trường Length thay vì Type).

Vì 0x86DD > 1536, nên gói tin là Ethernet II.

4. Phân tích ARP



Chọn frame đầu tiên của giao thức ARP, quan sát chi tiết nội dung frame :



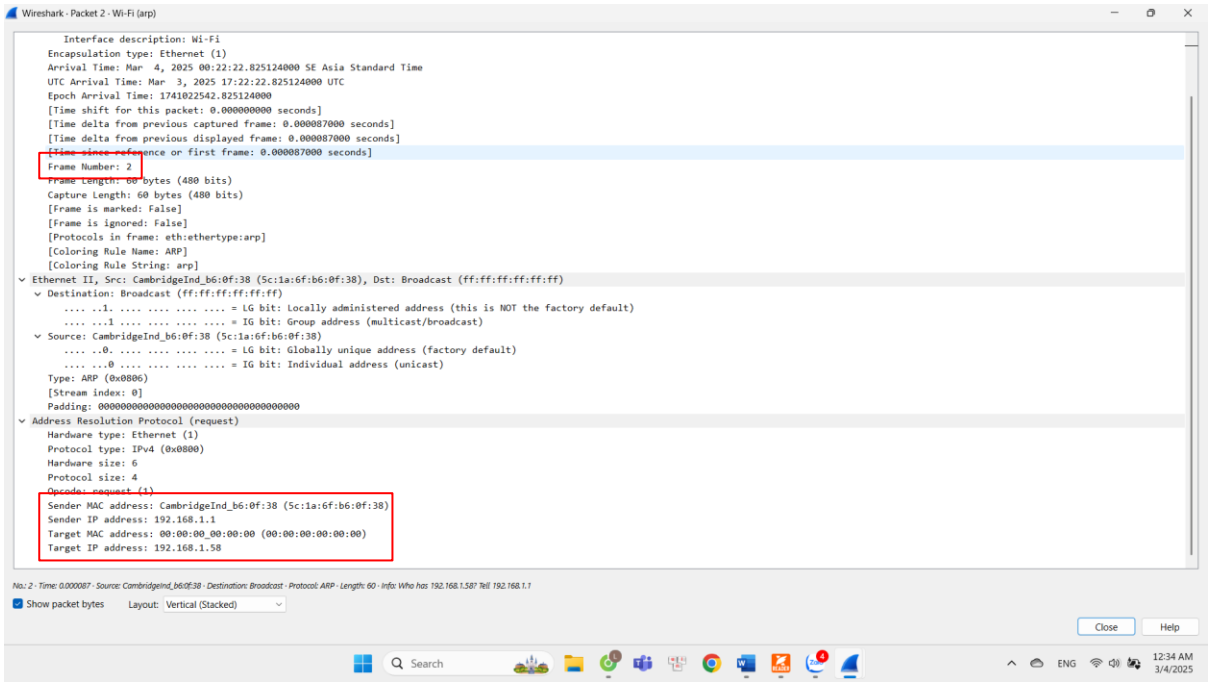
Địa chỉ MAC nguồn: 5c:1a:6f:b6:0f:38 (CambridgeInd_b6:0f:38).

Địa chỉ MAC đích: 00:00:00_00:00:00 (00:00:00:00:00:00)

Địa chỉ IP nguồn: 192.168.1.1.

Địa chỉ IP đích: 192.168.1.61.

Chọn frame thứ hai của giao thức ARP hiển thị ở cửa sổ Packet List



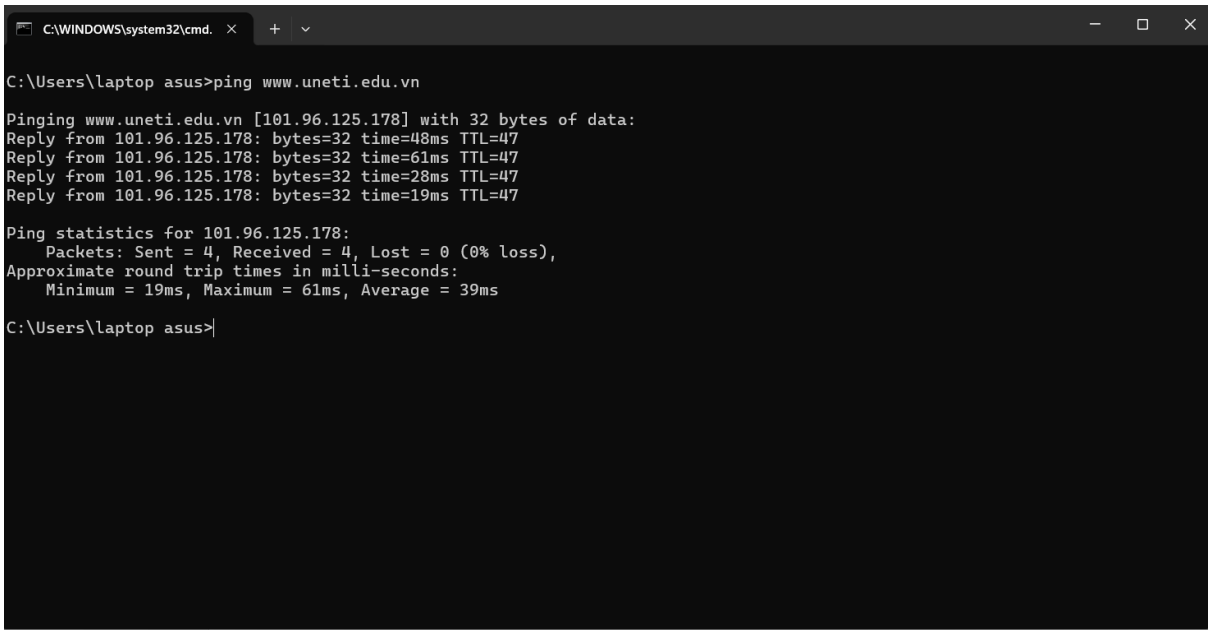
Địa chỉ MAC nguồn: 5c:1a:6f:b6:0f:38 (CambridgeInd_b6:0f:38).

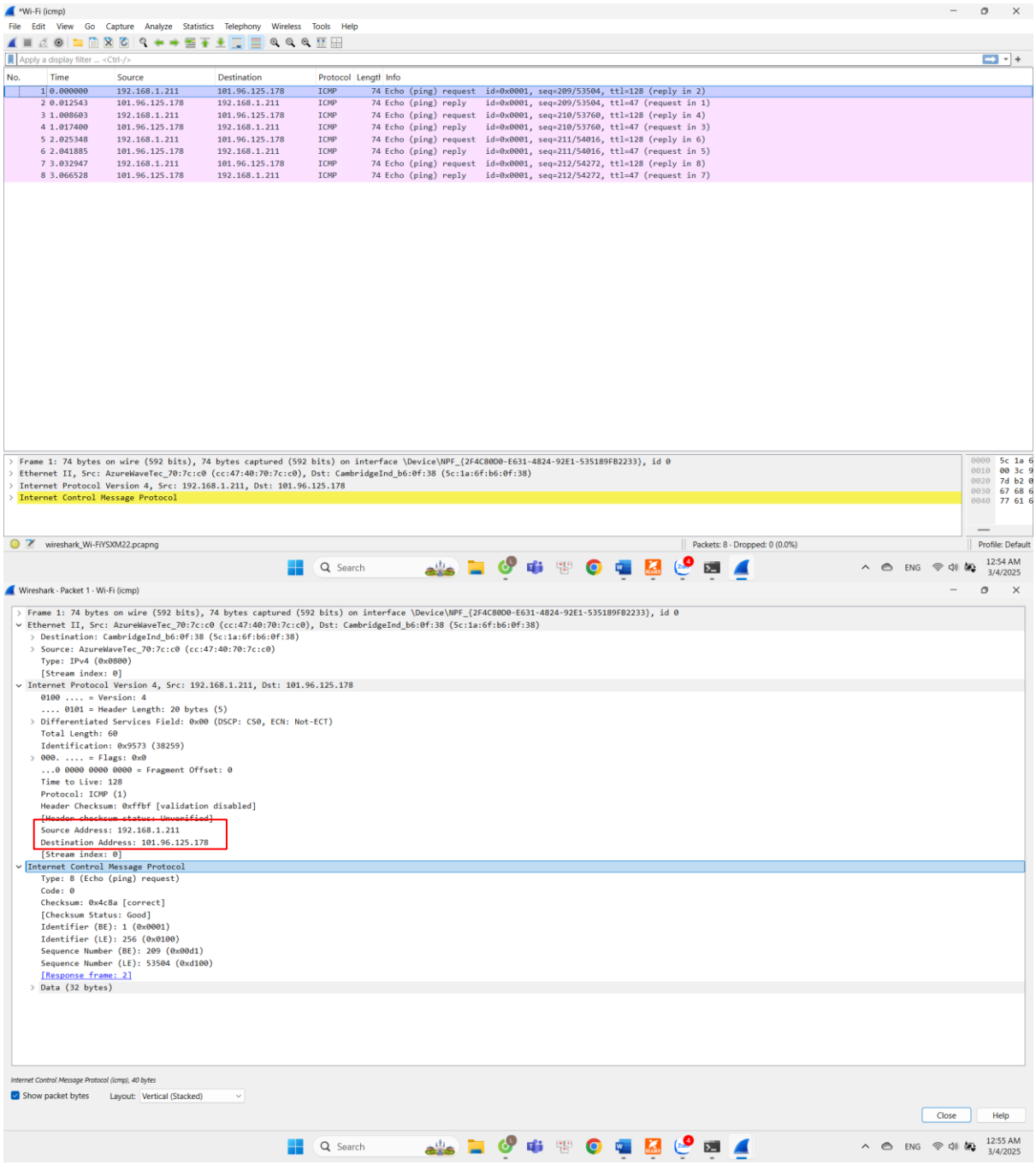
Địa chỉ MAC đích: 00:00:00_00:00:00 (00:00:00:00:00:00)

Địa chỉ IP nguồn: 192.168.1.1.

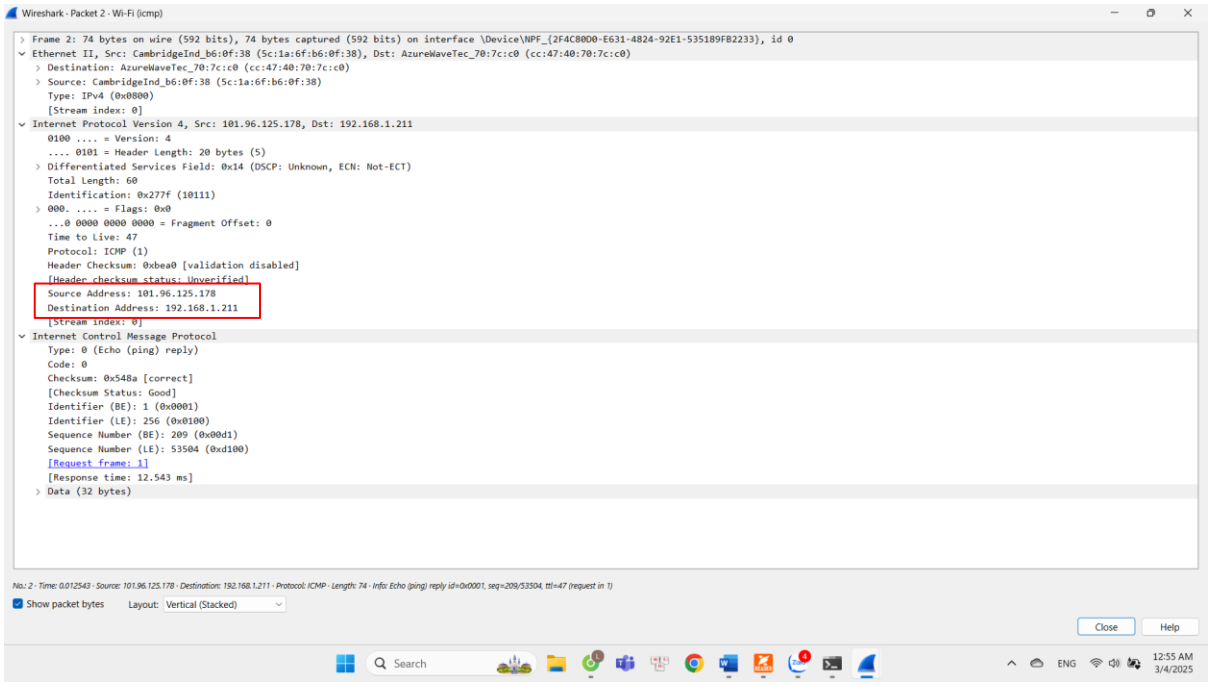
Địa chỉ IP đích: 192.168.1.58.

5. Ping một server ngoài Internet




*Wi-Fi (icmp)
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Lengtl Info
1 0.000000 192.168.1.211 101.96.125.178 ICMP 74 Echo (ping) request id=0x0001, seq=209/53504, ttl=128 (reply in 2)
2 0.012543 101.96.125.178 192.168.1.211 ICMP 74 Echo (ping) reply id=0x0001, seq=209/53504, ttl=47 (request in 1)
3 1.008603 192.168.1.211 101.96.125.178 ICMP 74 Echo (ping) request id=0x0001, seq=210/53760, ttl=128 (reply in 4)
4 1.017400 101.96.125.178 192.168.1.211 ICMP 74 Echo (ping) reply id=0x0001, seq=210/53760, ttl=47 (request in 3)
5 2.025348 192.168.1.211 101.96.125.178 ICMP 74 Echo (ping) request id=0x0001, seq=211/54016, ttl=128 (reply in 6)
6 2.041885 101.96.125.178 192.168.1.211 ICMP 74 Echo (ping) reply id=0x0001, seq=211/54016, ttl=47 (request in 5)
7 3.032947 192.168.1.211 101.96.125.178 ICMP 74 Echo (ping) request id=0x0001, seq=212/54272, ttl=128 (reply in 8)
8 3.066528 101.96.125.178 192.168.1.211 ICMP 74 Echo (ping) reply id=0x0001, seq=212/54272, ttl=47 (request in 7)
Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{2F4C80D0-E631-4824-92E1-535189FB2233}, id 0
Ethernet II, Src: AzureWaveTec_70:7c:c0 (cc:47:40:70:7c:c0), Dst: CambridgeInd_b6:0f:38 (5c:1a:6f:b6:0f:38)
Internet Protocol Version 4, Src: 192.168.1.211, Dst: 101.96.125.178
Internet Control Message Protocol
wireshark_Wi-FiYSXM22.pcapng
Packets: 8 · Dropped: 0 (0.0%)
Profile: Default
Search
ENG
12:54 AM
3/4/2025

Wireshark · Packet 1 · Wi-Fi (icmp)
Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{2F4C80D0-E631-4824-92E1-535189FB2233}, id 0
Ethernet II, Src: AzureWaveTec_70:7c:c0 (cc:47:40:70:7c:c0), Dst: CambridgeInd_b6:0f:38 (5c:1a:6f:b6:0f:38)
Destination: CambridgeInd_b6:0f:38 (5c:1a:6f:b6:0f:38)
Source: AzureWaveTec_70:7c:c0 (cc:47:40:70:7c:c0)
Type: IPv4 (0x0800)
[Stream index: 0]
Internet Protocol Version 4, Src: 192.168.1.211, Dst: 101.96.125.178
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 60
Identification: 0x9573 (38259)
000. .... = Flags: 0x0
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 128
Protocol: ICMP (1)
Header Checksum: 0xffbf [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.1.211
Destination Address: 101.96.125.178
[Stream index: 0]
Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0x4c8a [correct]
[Checksum Status: Good]
Identifier (BE): 1 (0x0001)
Identifier (LE): 256 (0x0100)
Sequence Number (BE): 209 (0x00d1)
Sequence Number (LE): 53504 (0xd100)
[Response frame: 2]
Data (32 bytes)
Internet Control Message Protocol (icmp), 40 bytes
Show packet bytes
Layout: Vertical (Stacked)
Close
Help
Search
ENG
12:55 AM
3/4/2025



Không biết được địa chỉ MAC của server **www.uneti.edu.vn** hoặc bất kỳ server nào ngoài Internet khi thực hiện lệnh ping. Lý do chính là do cách thức hoạt động của giao thức mạng.