

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:

The image shows a Wireshark network traffic capture. The main window displays a list of packets, and the packet details pane shows the structure of a DNS query for ogs.google.com.

Packet List:

No.	Time	Source	Destination	Protocol	Length	Info
2112	25.305978	fe80::1	fe80::588e:a61a:5f0...	DNS	162	Standard query response 0x0211 AAAA lh3.googleusercontent.com CNAME googlehosted.l.googleusercontent.com AAAA 2404:6800:4005:81...
2149	25.405822	fe80::1	fe80::588e:a61a:5f0...	DNS	150	Standard query response 0xc488 A lh3.googleusercontent.com CNAME googlehosted.l.googleusercontent.com A 142.250.197.225
2176	25.417206	fe80::1	fe80::588e:a61a:5f0...	DNS	162	Standard query response 0x0211 AAAA lh3.googleusercontent.com CNAME googlehosted.l.googleusercontent.com AAAA 2404:6800:4005:81...
2507	25.833336	fe80::588e:a61a:5f0...	fe80::1	DNS	95	Standard query 0x34f3 AAAA play.google.com
2508	25.833425	fe80::588e:a61a:5f0...	fe80::1	DNS	95	Standard query 0x31c7 HTTPS play.google.com
2509	25.833441	fe80::588e:a61a:5f0...	fe80::1	DNS	95	Standard query 0x7c62 A play.google.com
2520	25.842902	fe80::1	fe80::588e:a61a:5f0...	DNS	123	Standard query response 0x34f3 AAAA play.google.com AAAA 2404:6800:4005:826::200e
2525	25.854773	fe80::1	fe80::588e:a61a:5f0...	DNS	145	Standard query response 0x31c7 HTTPS play.google.com SOA ns1.google.com
2526	25.854773	fe80::1	fe80::588e:a61a:5f0...	DNS	111	Standard query response 0x7c62 A play.google.com A 142.250.198.238
2536	25.866219	fe80::1	fe80::588e:a61a:5f0...	DNS	123	Standard query response 0x34f3 AAAA play.google.com AAAA 2404:6800:4005:817::200e
2537	25.877289	fe80::1	fe80::588e:a61a:5f0...	DNS	145	Standard query response 0x31c7 HTTPS play.google.com SOA ns1.google.com
2543	25.877289	fe80::1	fe80::588e:a61a:5f0...	DNS	111	Standard query response 0x7c62 A play.google.com A 142.250.197.174
2752	29.732095	fe80::588e:a61a:5f0...	fe80::1	DNS	94	Standard query 0xe859 HTTPS ogs.google.com
2753	29.732163	fe80::588e:a61a:5f0...	fe80::1	DNS	94	Standard query 0x5c40 A ogs.google.com
2754	29.732180	fe80::588e:a61a:5f0...	fe80::1	DNS	94	Standard query 0x2e07 AAAA ogs.google.com
2755	29.751087	fe80::1	fe80::588e:a61a:5f0...	DNS	165	Standard query response 0xe859 HTTPS ogs.google.com CNAME www3.l.google.com SOA ns1.google.com
2756	29.751087	fe80::1	fe80::588e:a61a:5f0...	DNS	131	Standard query response 0x5c40 A ogs.google.com CNAME www3.l.google.com A 142.250.199.206
2757	29.751087	fe80::1	fe80::588e:a61a:5f0...	DNS	143	Standard query response 0x2e07 AAAA ogs.google.com CNAME www3.l.google.com AAAA 2404:6800:4005:824::200e
2760	29.762404	fe80::1	fe80::588e:a61a:5f0...	DNS	165	Standard query response 0xe859 HTTPS ogs.google.com CNAME www3.l.google.com SOA ns1.google.com
2761	29.774075	fe80::1	fe80::588e:a61a:5f0...	DNS	131	Standard query response 0x5c40 A ogs.google.com CNAME www3.l.google.com A 142.250.196.238
2762	29.774075	fe80::1	fe80::588e:a61a:5f0...	DNS	143	Standard query response 0x2e07 AAAA ogs.google.com CNAME www3.l.google.com AAAA 2404:6800:4005:819::200e
2809	29.984046	fe80::588e:a61a:5f0...	fe80::1	DNS	95	Standard query 0xe712 HTTPS ssl.gstatic.com
2810	29.984704	fe80::588e:a61a:5f0...	fe80::1	DNS	95	Standard query 0x530d A ssl.gstatic.com
2811	29.984923	fe80::588e:a61a:5f0...	fe80::1	DNS	95	Standard query 0x309e AAAA ssl.gstatic.com
2817	30.016181	fe80::1	fe80::588e:a61a:5f0...	DNS	152	Standard query response 0xe712 HTTPS ssl.gstatic.com SOA ns1.google.com
2818	30.016181	fe80::1	fe80::588e:a61a:5f0...	DNS	111	Standard query response 0x530d A ssl.gstatic.com A 142.250.199.67
2819	30.017961	fe80::1	fe80::588e:a61a:5f0...	DNS	123	Standard query response 0x309e AAAA ssl.gstatic.com AAAA 2404:6800:4005:823::2003
2820	30.017961	fe80::1	fe80::588e:a61a:5f0...	DNS	152	Standard query response 0xe712 HTTPS ssl.gstatic.com SOA ns1.google.com
2844	30.018720	fe80::1	fe80::588e:a61a:5f0...	DNS	111	Standard query response 0x530d A ssl.gstatic.com A 142.250.198.99
2854	30.020161	fe80::1	fe80::588e:a61a:5f0...	DNS	123	Standard query response 0x309e AAAA ssl.gstatic.com AAAA 2404:6800:4005:818::2003

Packet Details (Packet 2752):

- Frame 2752: 94 bytes captured on interface \Device\NPF_{2F4C8000-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 6, Src: fe80::588e:a61a:5f0b:b2c4, Dst: fe80::1
- User Datagram Protocol, Src Port: 58668, Dst Port: 53
- Domain Name System (query)
 - Transaction ID: 0xe859
 - Flags: 0x0100 Standard query
 - Questions: 1
 - Answer RRs: 0
 - Authority RRs: 0
 - Additional RRs: 0
 - Queries
 - ogs.google.com: type HTTPS, class IN
 - Name: ogs.google.com
 - [Name Length: 14]
 - [Label Count: 3]
 - Type: HTTPS (65) (HTTPS Specific Service Endpoints)
 - Class: IN (0x0001)

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.

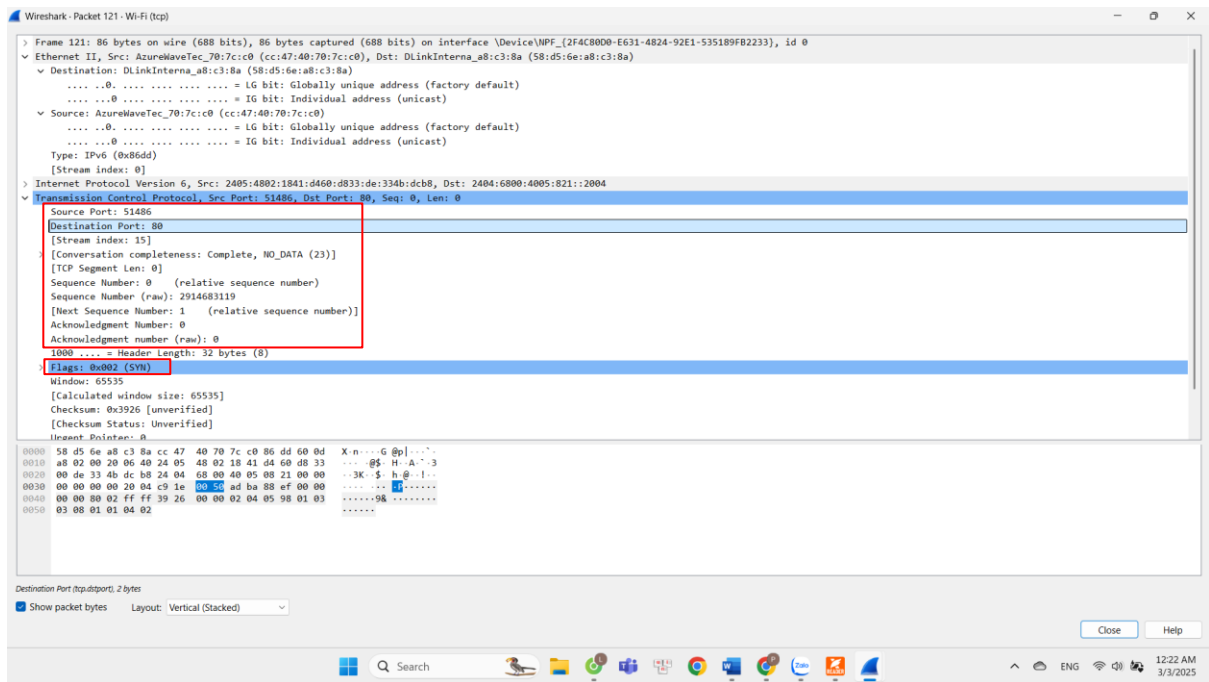
Trong ô Display Filter, nhập bộ lọc sau để chỉ hiển thị gói SYN:

`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:

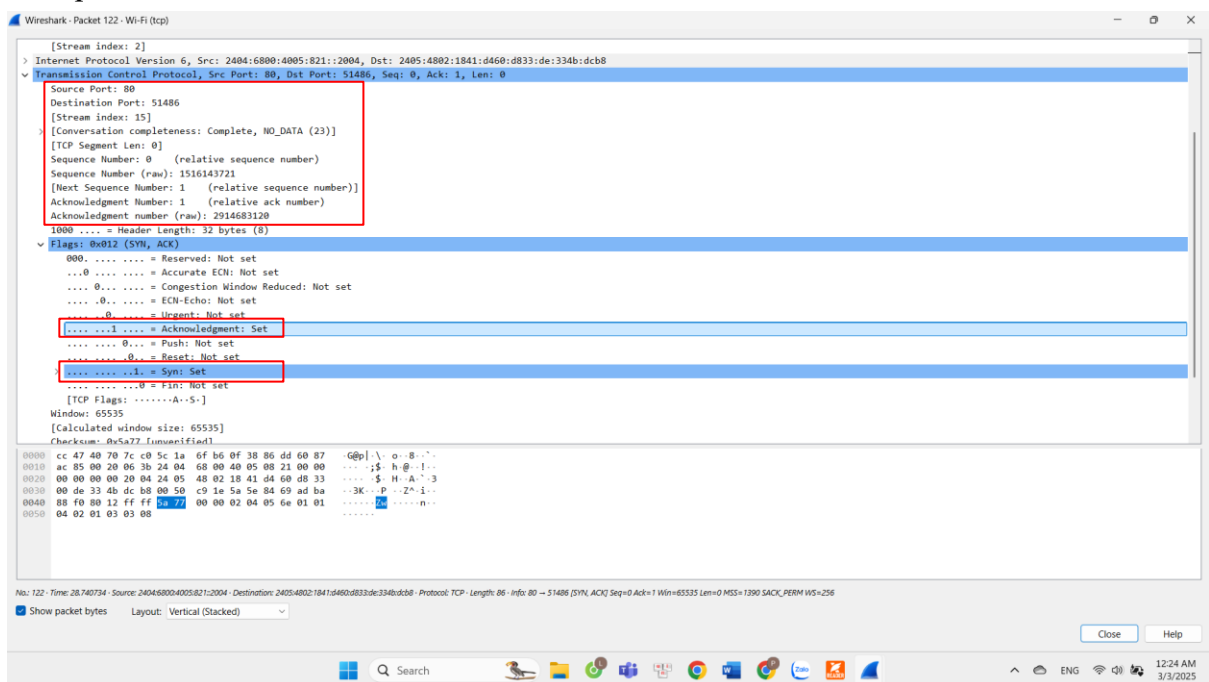
Nhập bộ lọc:

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:

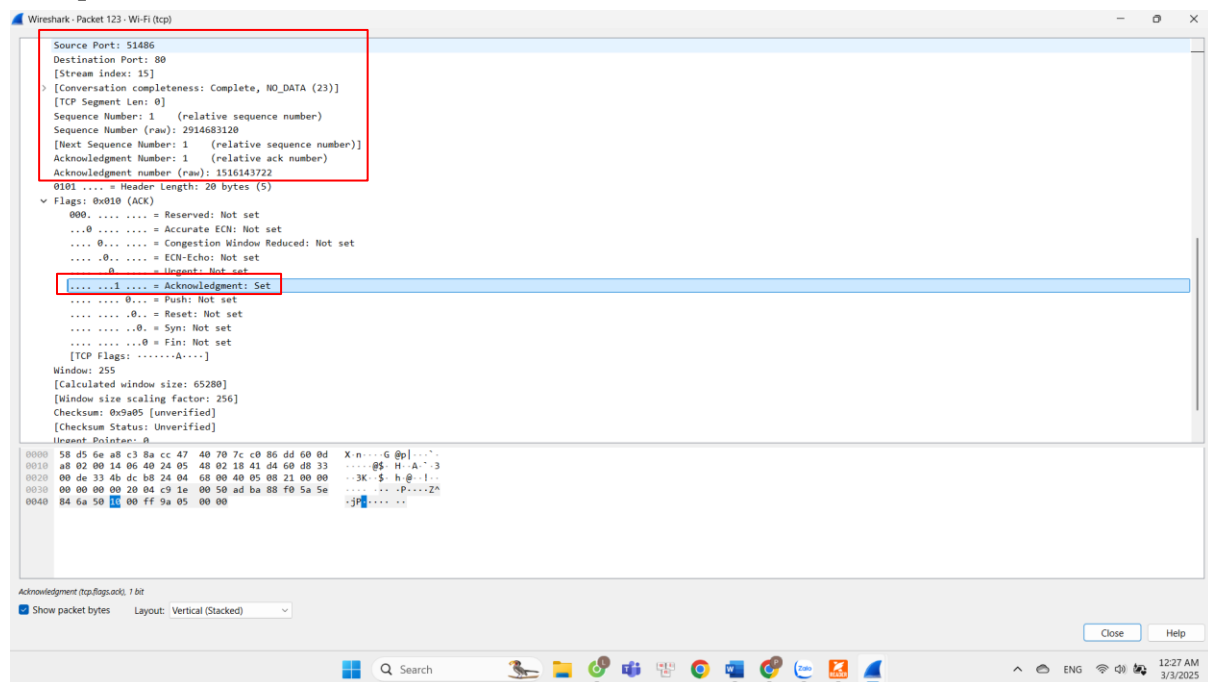
`tcp.flags.ack == 1 && tcp.flags.syn == 0`

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

Đề 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

The image shows two screenshots of the Wireshark network protocol analyzer. The top screenshot displays the details of a captured packet (Packet 13) on a Wi-Fi interface. The packet is an Ethernet II frame with a destination MAC address of DLinkInterna_a8:c3:8a and a source MAC address of DLinkInterna_a8:c3:8a. The frame is marked as unicast and has a length of 166 bytes. The payload is an ICMPv6 packet. The bottom screenshot shows the expanded details of the ICMPv6 packet, which is a Redirect message (Type 137). The message includes a checksum, a target address, and a link-layer address. The packet is captured on the same interface as the top screenshot.

Wireshark - Packet 13 - Wi-Fi

Frame 13: 166 bytes on wire (1328 bits), 166 bytes captured (1328 bits) on interface \Device\NPF_{2F4C8000-E631-4824-92E1-535189FB2233}, id 0

Section number: 1

Interface id: 0 (\Device\NPF_{2F4C8000-E631-4824-92E1-535189FB2233})

Interface name: \Device\NPF_{2F4C8000-E631-4824-92E1-535189FB2233}

Interface description: Wi-Fi

Encapsulation type: Ethernet (1)

Arrival Time: Mar 3, 2025 23:45:12.818983000 SE Asia Standard Time

UTC Arrival Time: Mar 3, 2025 16:45:12.818983000 UTC

Epoch Arrival Time: 1741620312.818983000

[Time shift for this packet: 0.000000000 seconds]

[Time delta from previous captured frame: 0.043129000 seconds]

[Time delta from previous displayed frame: 0.043129000 seconds]

[Time since reference or first frame: 0.204339000 seconds]

Frame Number: 13

Frame Length: 166 bytes (1328 bits)

Capture Length: 166 bytes (1328 bits)

[Frame is marked: False]

[Frame is ignored: False]

[Protocols in frame: ethII:ethertype:ipv6:icmpv6:ipv6:tcp]

[Coloring Rule Name: ICMP]

[Coloring Rule String: icmp || icmpv6]

Ethernet II, Src: DLinkInterna_a8:c3:8a (58:d5:6e:a8:c3:8a), Dst: AzureWaveTec_70:7c:c0 (cc:47:40:70:7c:c0)

Destination: AzureWaveTec_70:7c:c0 (cc:47:40:70:7c:c0)

Source: DLinkInterna_a8:c3:8a (58:d5:6e:a8:c3:8a)

Type: IPv6 (0x86dd)

Stream index: 0

Internet Protocol Version 6, Src: fe80::5ad5:6eff:fea8:c38a, Dst: 2405:4802:1841:d460:2824:2205:cbf:23d1

0110 = Version: 6

.... 0000 0000 = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)

.... 0000 00.. = Differentiated Services Codepoint: Default (0)

.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)

.... 0000 0000 0000 0000 = Flow Label: 0x000000

Payload Length: 112

Next Header: ICMPv6 (58)

No. 13 - Time: 0.204339 - Source: fe80::5ad5:6eff:fea8:c38a - Destination: 2405:4802:1841:d460:2824:2205:cbf:23d1 - Protocol: ICMPv6 - Length: 166 - Info: Redirect is at Sc:1a:6f:b6:0f:38

Show packet bytes Layout: Vertical (Stacked)

Close Help

Wireshark - Packet 13 - Wi-Fi

Payload Length: 112

Next Header: ICMPv6 (58)

Hop Limit: 255

Source Address: fe80::5ad5:6eff:fea8:c38a

[Address Space: Link-Local Unicast]

[Special-Purpose Allocation: Link-Local Unicast]

Destination Address: 2405:4802:1841:d460:2824:2205:cbf:23d1

[Address Space: Global Unicast]

[Source SLAAC MAC: DLinkInterna_a8:c3:8a (58:d5:6e:a8:c3:8a)]

[Stream index: 0]

Internet Control Message Protocol v6

Type: Redirect (137)

Checksum: 0xa291 [correct]

[Checksum Status: Good]

Reserved: 00000000

Target Address: fe80::1

Destination Address: 2603:1040:900:2:81

ICMPv6 Option (Target link-layer address : 5c:1a:6f:b6:0f:38)

Type: Target link-layer address (2)

Length: 1 (8 bytes)

Link-layer address: CambridgeInd_b6:0f:38 (5c:1a:6f:b6:0f:38)

ICMPv6 Option (Redirected header)

Type: Redirected header (4)

Length: 8 (64 bytes)

Reserved

Redirected Packet

Internet Protocol Version 6, Src: 2405:4802:1841:d460:2824:2205:cbf:23d1, Dst: 2603:1040:900:2:81

0110 = Version: 6

.... 0000 0000 = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)

.... 0000 00.. = Differentiated Services Codepoint: Default (0)

.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)

.... 0000 1000 1100 0110 0110 = Flow Label: 0x08c66

Payload Length: 20

Next Header: TCP (6)

Hop Limit: 63

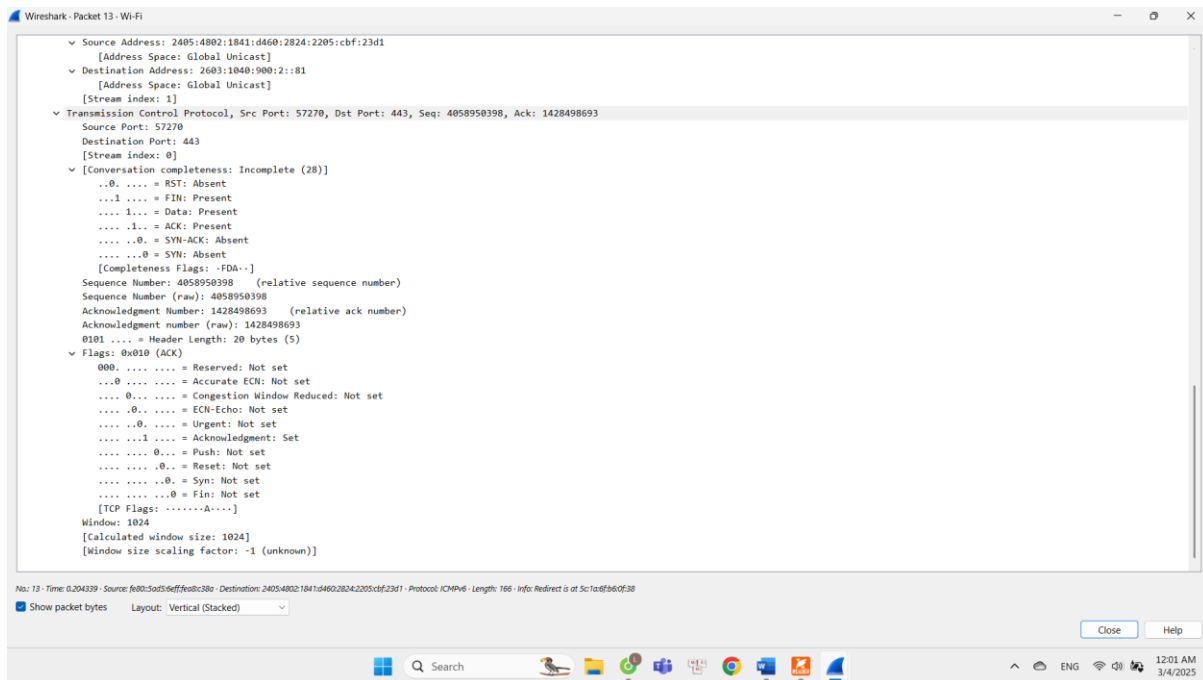
Source Address: 2405:4802:1841:d460:2824:2205:cbf:23d1

[Address Space: Global Unicast]

No. 13 - Time: 0.204339 - Source: fe80::5ad5:6eff:fea8:c38a - Destination: 2405:4802:1841:d460:2824:2205:cbf:23d1 - Protocol: ICMPv6 - Length: 166 - Info: Redirect is at Sc:1a:6f:b6:0f:38

Show packet bytes Layout: Vertical (Stacked)

Close Help



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

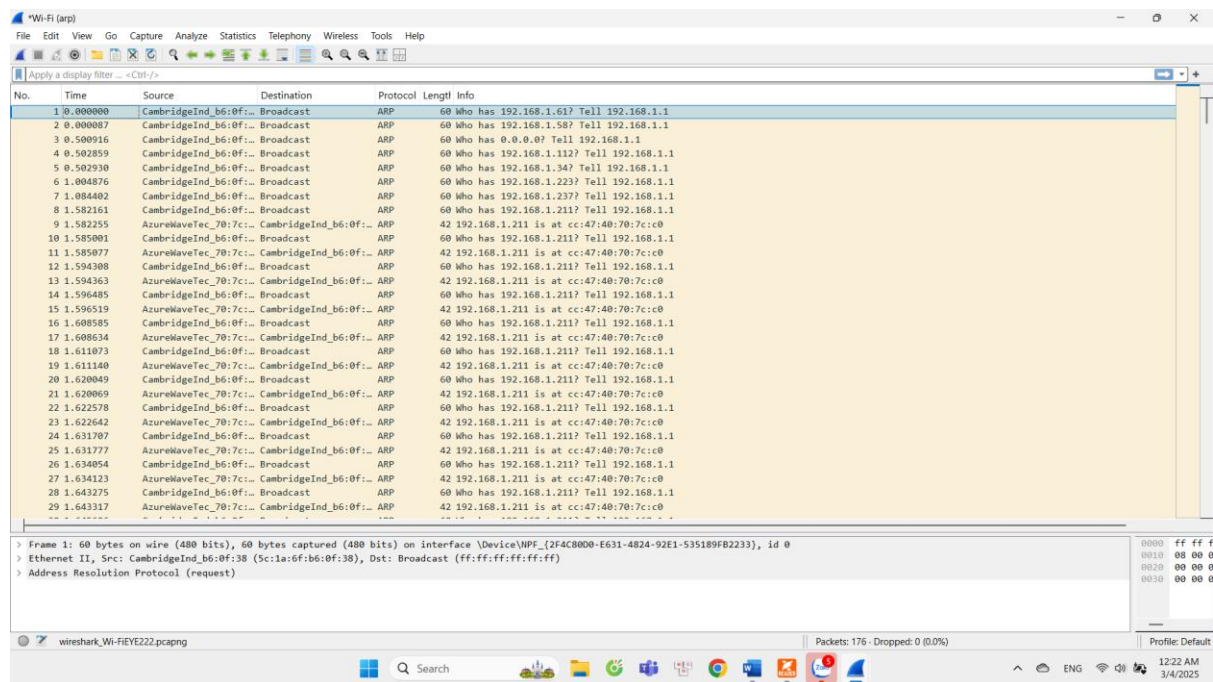
Trong ảnh gói tin có EtherType = 0x86DD, tức là IPv6.

Dựa trên quy tắc phân loại:

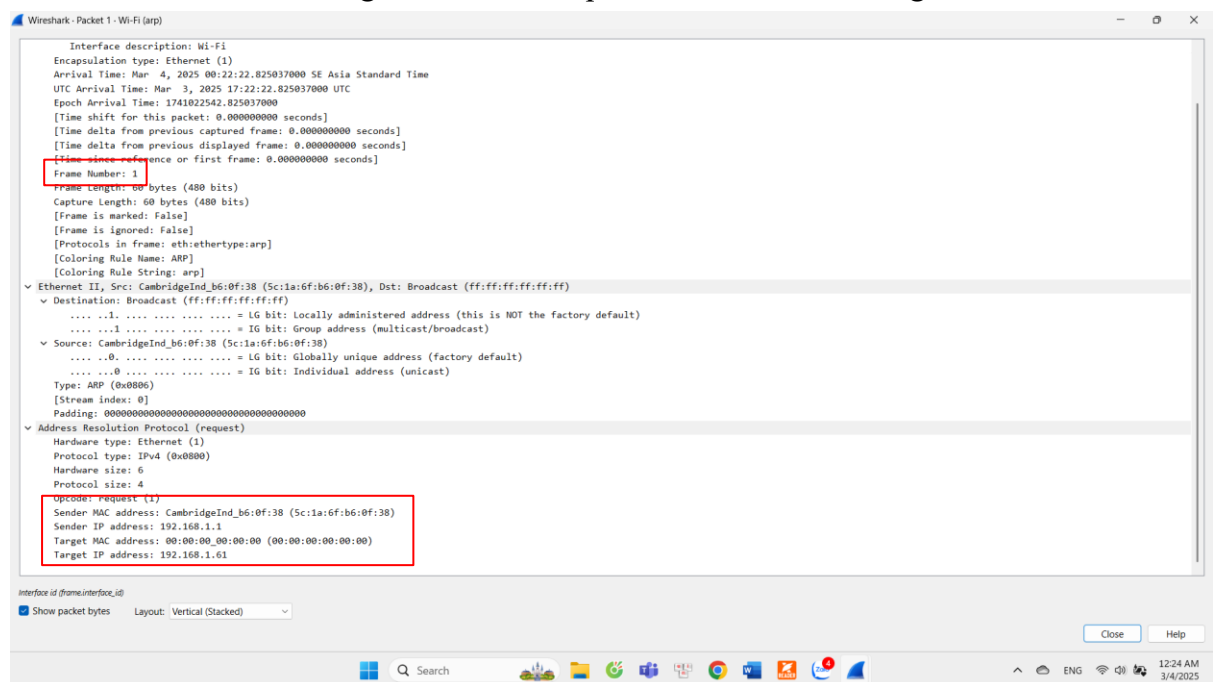
- EtherType ≥ 1536 (0x0600) \rightarrow Ethernet II.
- EtherType $< 1536 \rightarrow$ IEEE 802.3 (sử dụng trường Length thay vì Type).

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 :



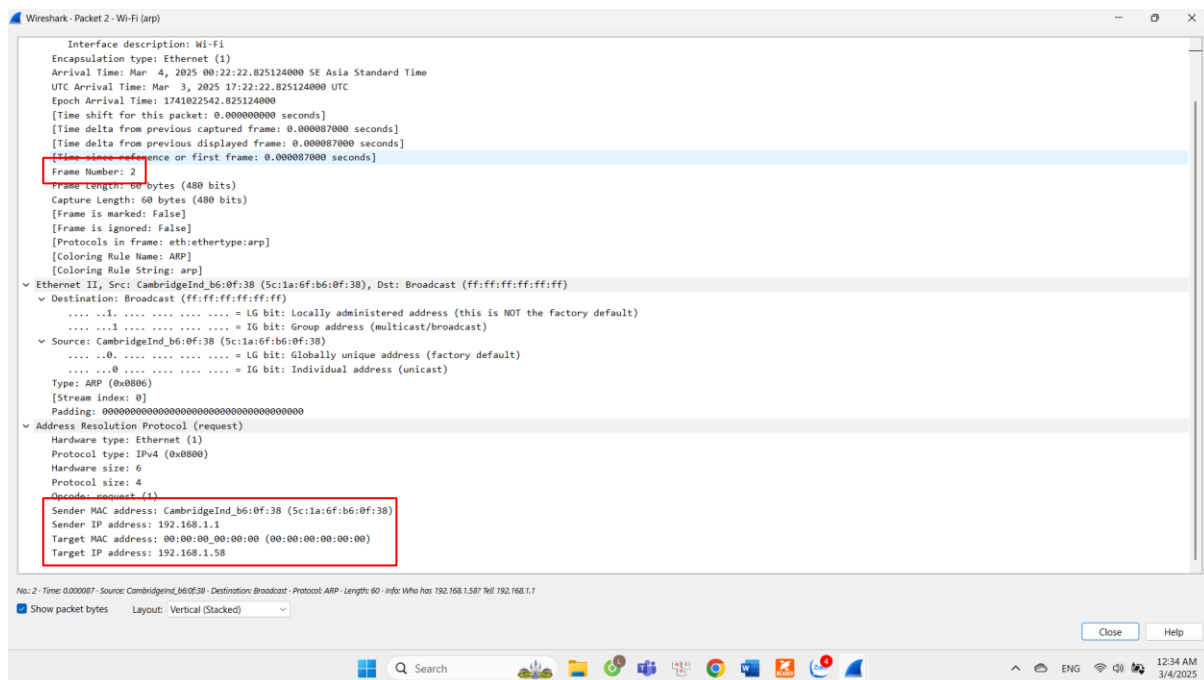
Địa chỉ MAC nguồn: 5c:1a:6f:b6:0f:38 (CambridgeInd_b6:0f:38).

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



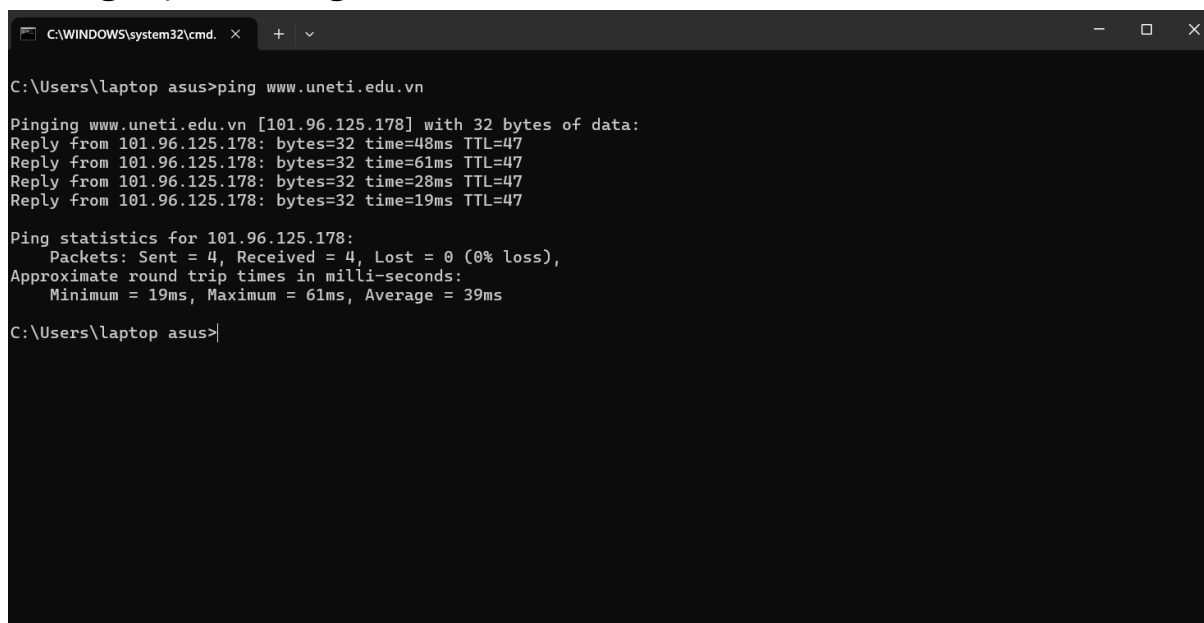
Địa chỉ MAC nguồn: 5c:1a:6f:b6:0f:38 (CambridgeInd_b6:0f:38).

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-F>

No.	Time	Source	Destination	Protocol	Length	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	1.000693	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_{2F4C8000-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-Fi (icmp)

Packets: 8 - Dropped: 0 (0.0%)

Profile: Default

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_{2F4C8000-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 (0x0000)

[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: 0xffff [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]

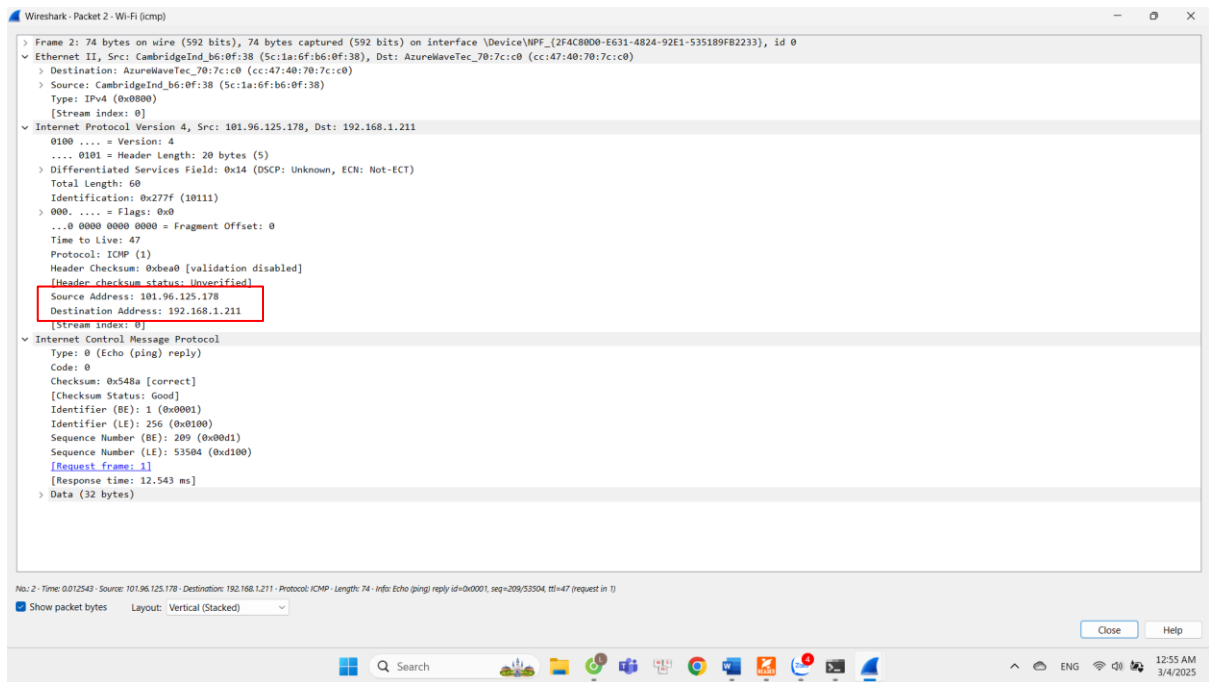
> Data (32 bytes)

Internet Control Message Protocol (icmp), 40 bytes

Show packet bytes Layout: Vertical (Stacked)

Close Help

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.