Lab
3

BÁO CÁO BÀI THỰC HÀNH SỐ [3]

[GIAO THỨC UDP & TCP]

Môn học: NHẬP MÔN MẠNG MÁY TÍNH

Sinh viên thực hiện	Nguyễn Dương Đại(23520217)
Thời gian thực hiện	15/10/2024 – 21/10/2024
Số câu đã hoàn thành	14/14

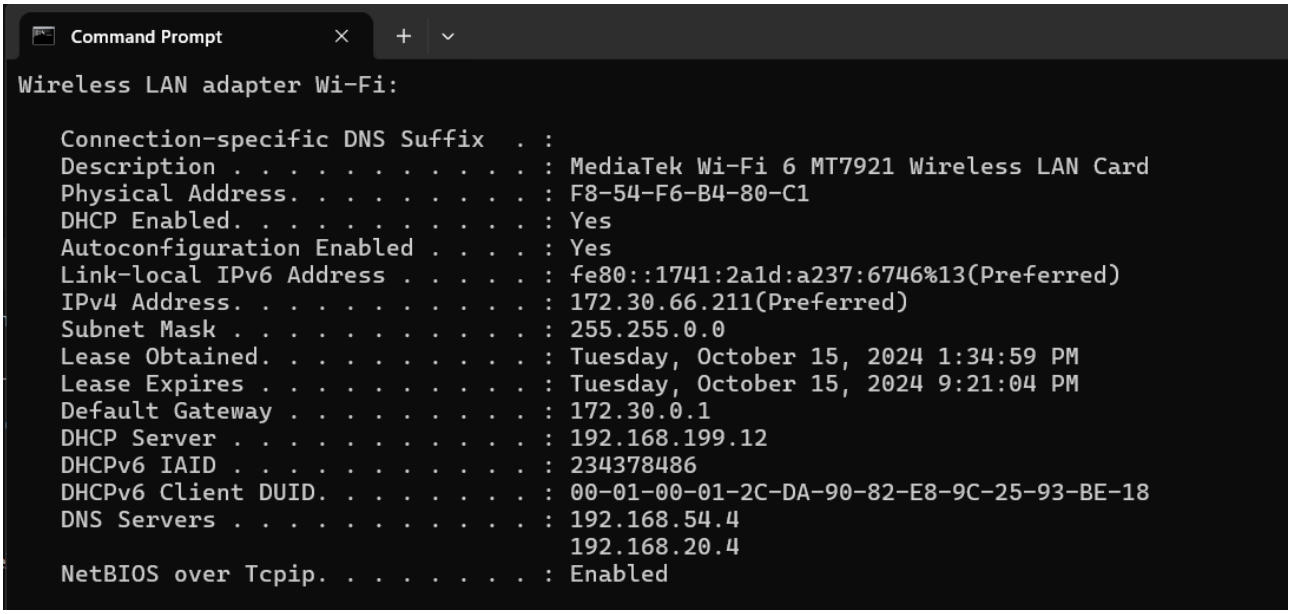
TRẢ LỜI CÁC CÂU HỎI

Câu 1: Điền thông tin vào bảng:

Trả lời:

IP address	172.30.66.211
MAC address	F8-54-F6-B4-80-C1
Default gateway IP address	172.30.0.1
DNS server IP address	192.168.54.4

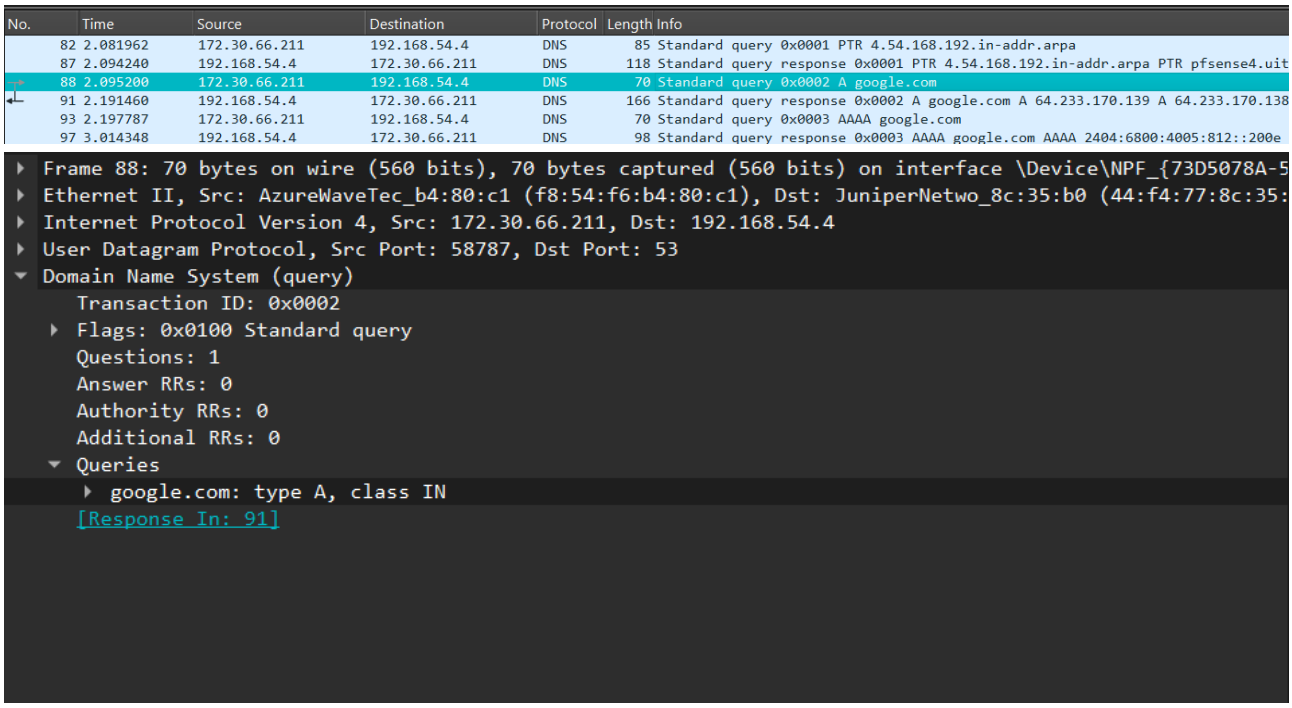
Minh chứng:



Câu 2: Tại danh sách các gói tin bắt được, định vị gói truy vấn domain google.com.

*Gợi ý: chứa "**standard query**" và "**A www.google.com**".*

Trả lời: Gói domain google.com là gói thứ 88

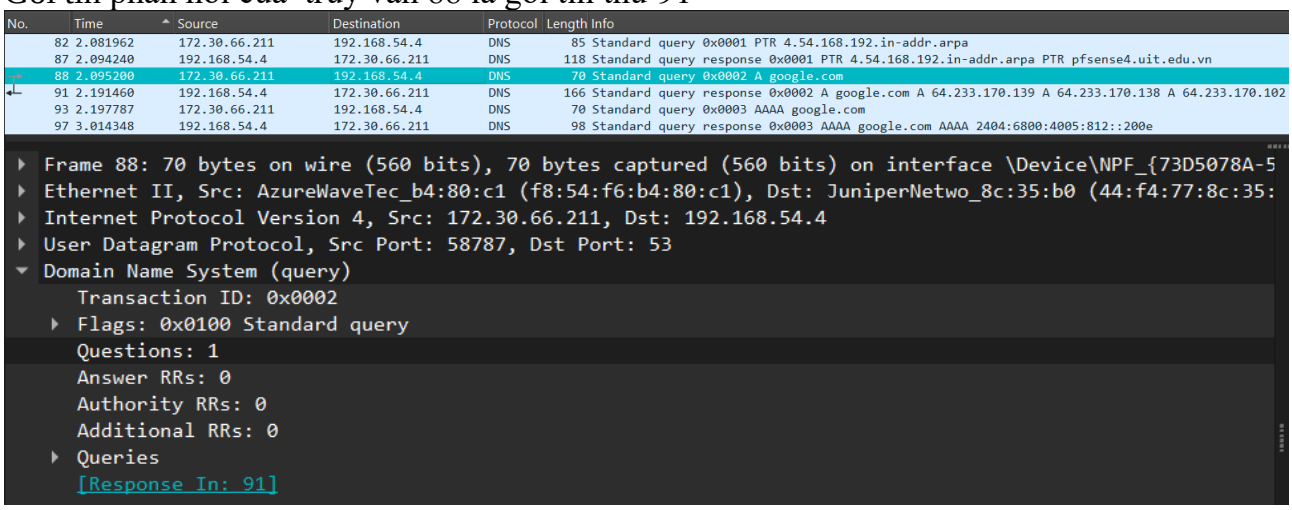


Ta thấy tại cột info dòng **A google.com** và trong gói tin tại phần "**Domain Name System**" ở phần **Queries** có chứa google.com: type A

Câu 3: Đinh vị gói tin phản hồi của truy vấn trên? Từ thông điệp trả lời, ghi lại địa chỉ IP của domain google.com.

Trả lời:

Gói tin phản hồi của truy vấn 88 là gói tin thứ 91



Ta có thể thấy rõ tại gói tin thứ 88 tại phần "**Domain Name System**" có chứa "**Response In : 91**" .

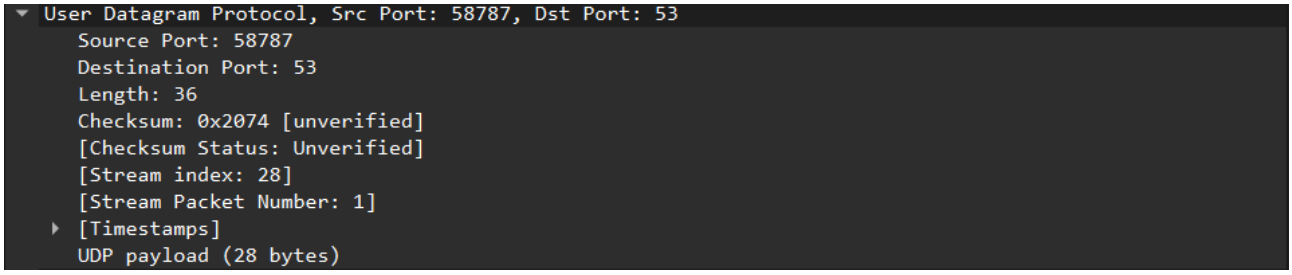
Câu 4: Chọn một gói tin DNS, xác định các trường (field) có trong UDP header và giải thích ý nghĩa của mỗi trường đó?

Trả lời:
Một gói tin DNS khi được truyền đi sẽ được đóng gói trong một datagram UDP. UDP header gồm các trường sau:

- **Source Port:**
  Ý nghĩa: Chỉ định cổng nguồn của máy gửi gói tin. Trong trường hợp của DNS, thường là một cổng ngẫu nhiên hoặc một cổng cố định được cấu hình trước.
- **Destination Port:**
  Ý nghĩa: Chỉ định cổng đích của máy nhận gói tin. Đối với DNS, cổng đích thường là 53.
- **Length:**
  Ý nghĩa: Xác định tổng độ dài của datagram UDP, bao gồm cả header và dữ liệu.
- **Checksum:**
  Ý nghĩa: Một giá trị kiểm tra lỗi được tính toán dựa trên nội dung của datagram. Người nhận sẽ tính toán lại checksum và so sánh với giá trị nhận được để đảm bảo tính toàn vẹn của dữ liệu.

Giải thích:

- **Source Port:** Giúp xác định máy tính nào đã gửi gói tin. Nó cho phép máy chủ DNS phân biệt các yêu cầu từ các máy khách khác nhau.
- **Destination Port:** Luôn là 53 vì đây là cổng chuẩn dành cho dịch vụ DNS. Khi một máy khách muốn thực hiện một truy vấn DNS, nó sẽ gửi một gói tin UDP đến cổng 53 của máy chủ DNS.
- **Length:** Cho biết tổng kích thước của gói tin, bao gồm cả phần header UDP và phần dữ liệu chứa yêu cầu DNS.
- **Checksum:** Được sử dụng để phát hiện các lỗi xảy ra trong quá trình truyền dữ liệu. Nếu checksum tính toán lại không khớp với checksum nhận được, gói tin sẽ bị loại bỏ.

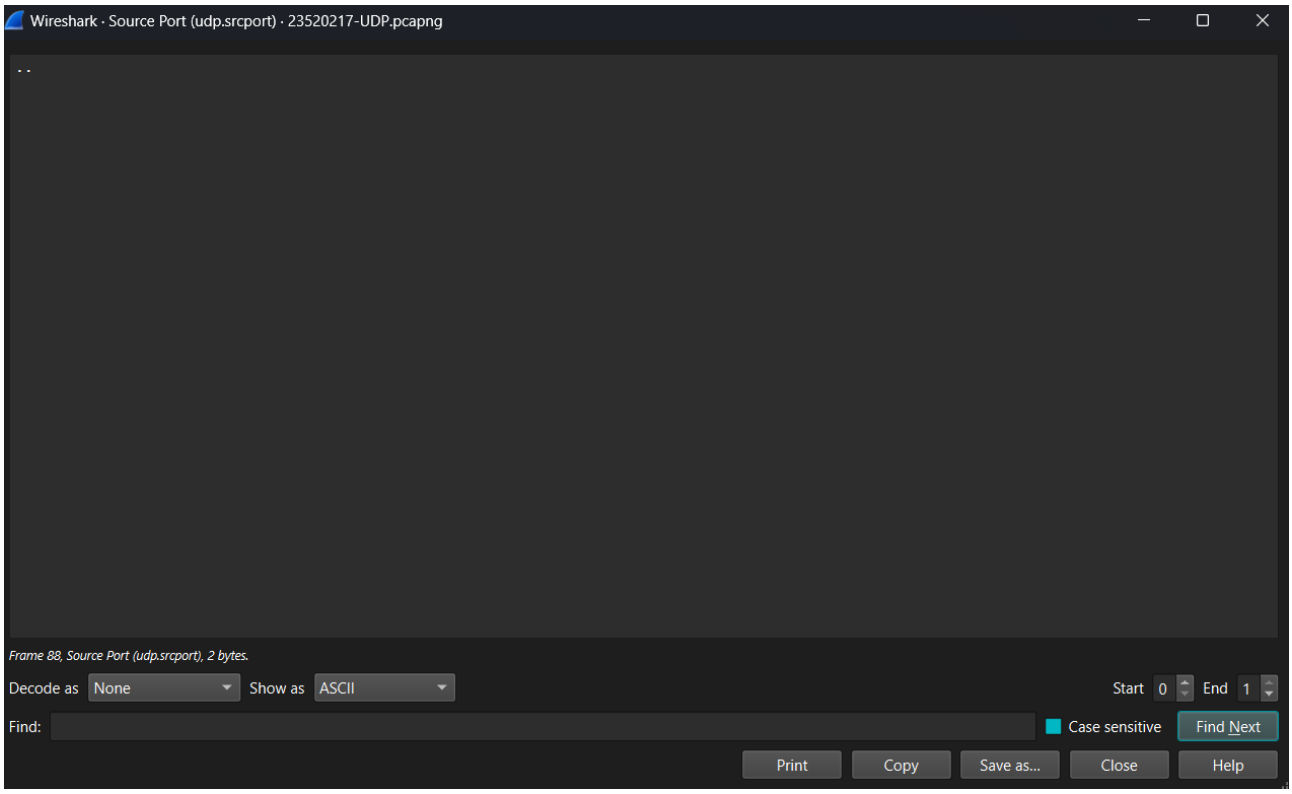


*Câu 5: Qua thông tin hiển thị của Wireshark, xác định độ dài (**tính theo byte**) của mỗi trường trong UDP header?*

Trả lời:
Độ dài các trường UDP header

- Source Port: 16 bit (2 byte)
- Destination Port: 16 bit (2 byte)
- Length: 16 bit (2 byte)
- Checksum: 16 bit (2 byte)

Minh chứng: Ta có thể thấy tại mỗi trường ta nhấn chuột phải và chọn Show Packet Bytes.Ta nhìn thấy giá trị của mỗi trường.

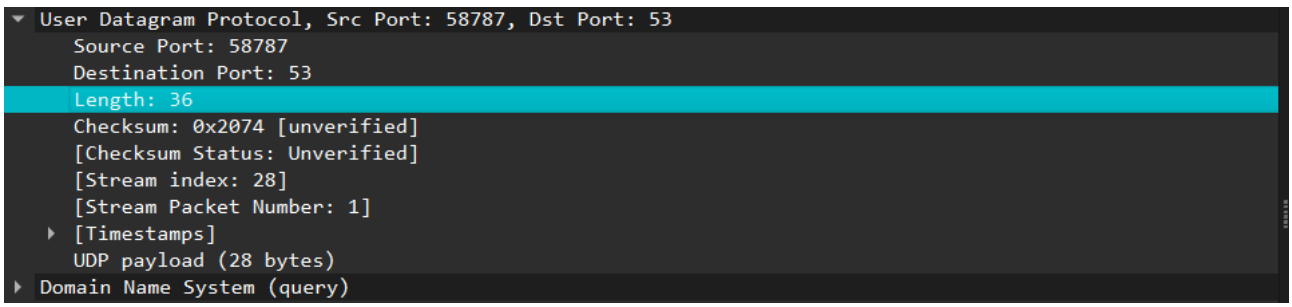


*Câu 6: Giá trị của trường **Length** trong UDP header là độ dài của gì? Chứng minh nhận định này bằng thông tin hiển thị của Wireshark?*

Trả lời:

Giá trị của trường Length trong UDP header là độ dài của toàn bộ datagram bao gồm header và dữ liệu. Giá trị Length trong trường hợp này là 36.

Minh chứng:

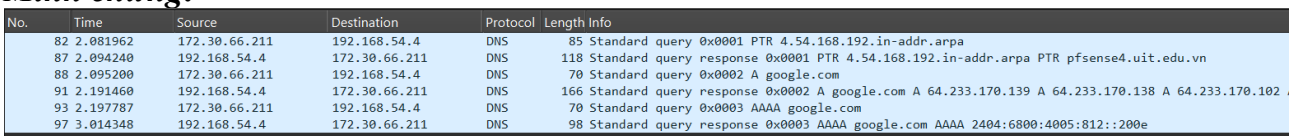


Câu 7: Quan sát 2 gói tin tìm được ở Câu 1 và 2, mô tả mối quan hệ giữa các địa chỉ IP và các port number của 2 gói tin này.

Trả lời:

Mối quan hệ giữa các địa chỉ IP và các port number: *Source và Destination của 2 gói tin trái ngược nhau.*

Minh chứng:



No.	Time	Source	Destination	Protocol	Length	Info
82	2.081962	172.30.66.211	192.168.54.4	DNS	85	Standard query 0x0001 PTR 4.54.168.192.in-addr.arpa
87	2.094240	192.168.54.4	172.30.66.211	DNS	118	Standard query response 0x0001 PTR 4.54.168.192.in-addr.arpa PTR pfsense4.uit.edu.vn
88	2.095200	172.30.66.211	192.168.54.4	DNS	70	Standard query 0x0002 A google.com
91	2.191460	192.168.54.4	172.30.66.211	DNS	166	Standard query response 0x0002 A google.com A 64.233.170.139 A 64.233.170.138 A 64.233.170.102
93	2.197787	172.30.66.211	192.168.54.4	DNS	70	Standard query 0x0003 AAAA google.com
97	3.014348	192.168.54.4	172.30.66.211	DNS	98	Standard query response 0x0003 AAAA google.com AAAA 2404:6800:4005:812::200e

Câu 8: Xác định IP và TCP port của client sử dụng để chuyển tệp sang gaia.cs.umass.edu là gì?

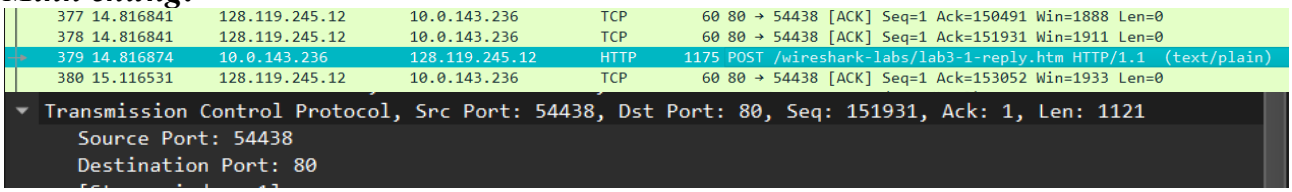
Trả lời:

Địa chỉ IP client sử dụng để chuyển tệp sang gaia.cs.umass.edu là: **10.0.143.236.**

TCP port client sử dụng để chuyển tệp sang gaia.cs.umass.edu là:

- Source Port: 54438.
- Destination Port: 80.

Minh chứng:

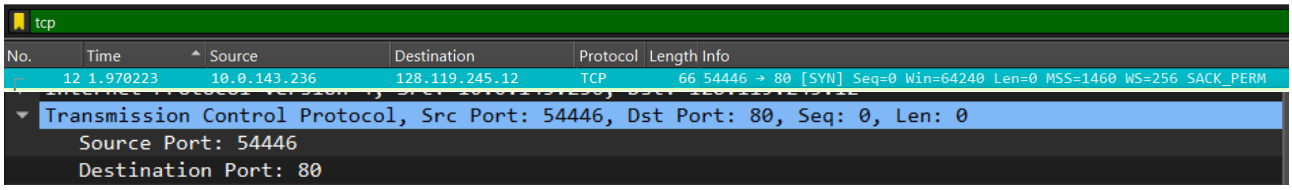


Câu 9: Địa chỉ IP của gaia.cs.umass.edu là gì? Trên số cổng nào nó gửi và nhận các segment TCP cho kết nối này?

Trả lời:

Địa chỉ IP của gaia.cs.umass.edu là: **128.119.245.12**. Trên số cổng **80** nó gửi và nhận các segment TCP cho kết nối này.

Minh chứng:

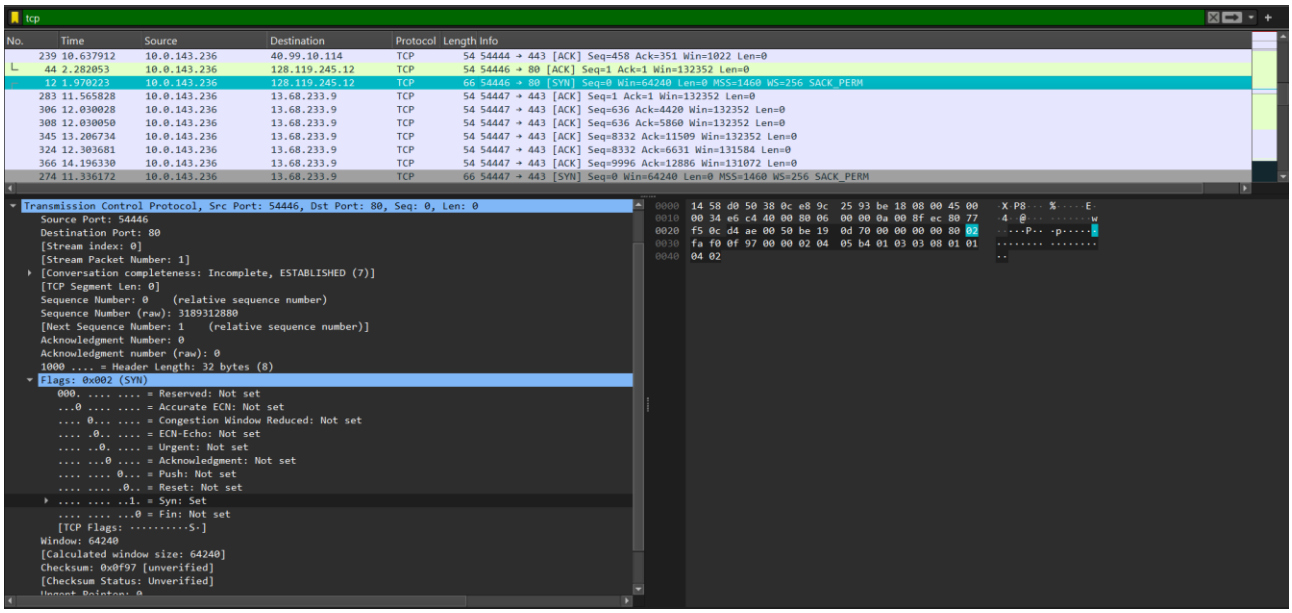


*Câu 10: TCP SYN segment (gói tin TCP có cờ SYN) sử dụng **sequence number** nào để khởi tạo kết nối TCP giữa client và server? Thành phần nào trong segment cho ta biết segment đó là TCP SYN segment?*

Trả lời: TCP SYN segment sử dụng sequence number là : 0.

Ta chọn gói tin vào trường **Flags** ta tìm dòng "**Syn**" nếu là giá trị "**Set**" và có cờ Syn là 1 vậy đây là TCP SYN segment.

Minh chứng:

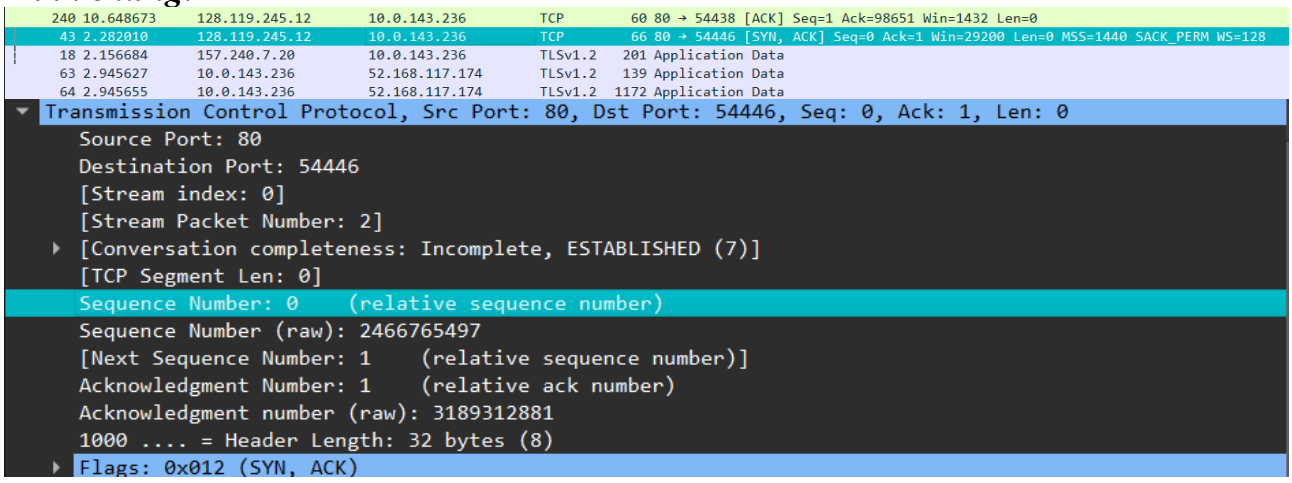


*Câu 11: Tìm **sequence number** của gói tin **SYN/ACK segment** được tạo bởi Server đến client để trả lời cho SYN segment?*

Trả lời:

Sequence number của gói tin SYN/ACK segment là: 0.

Minh chứng:



*Câu 12. Tìm giá trị của **Acknowledgement** trong SYN/ACK segment? Làm sao sever có thể xác định giá trị đó? Thành phần nào trong segment cho ta biết segment đó là SYN/ACK segment?*

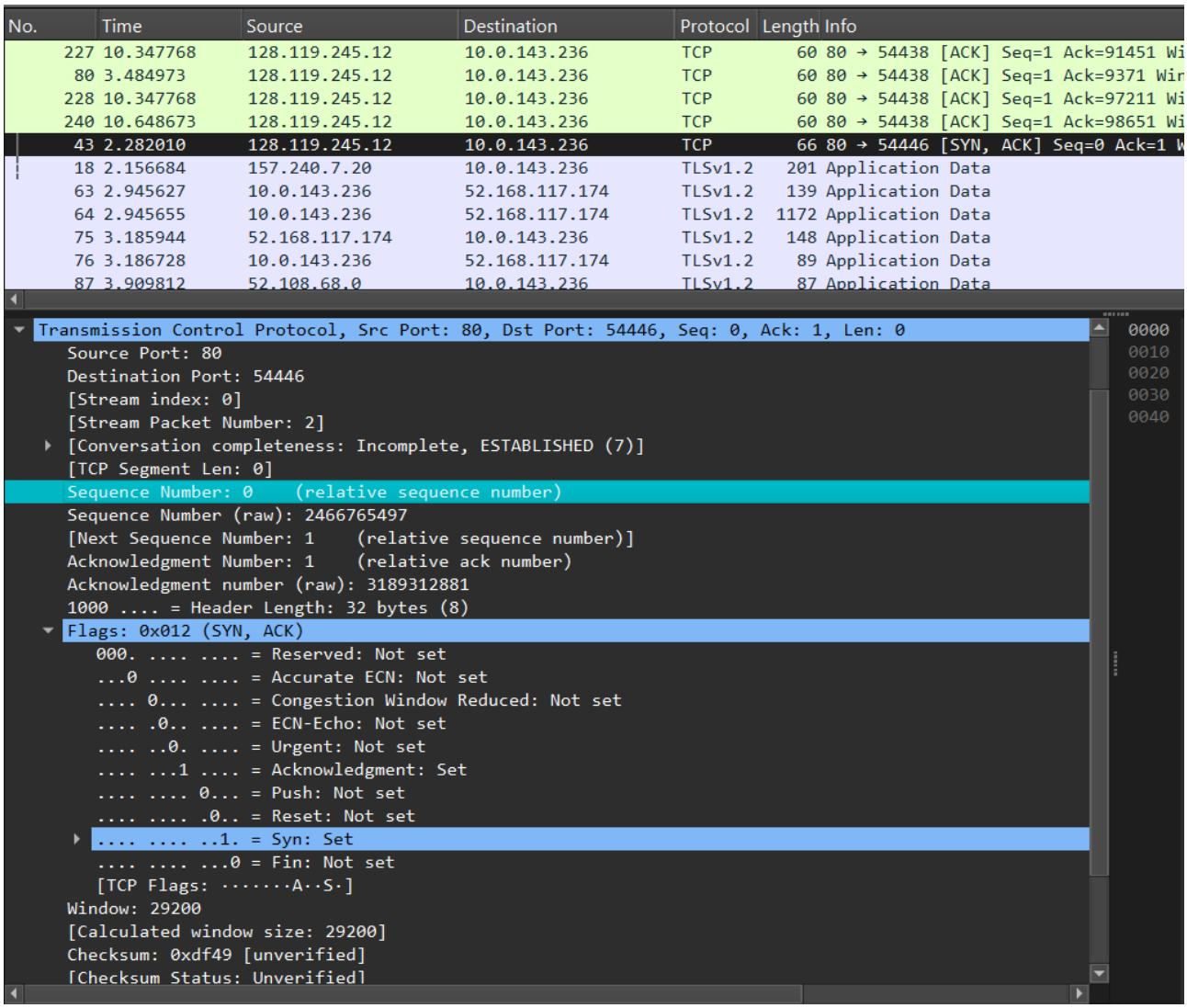
Trả lời:

Giá trị của Acknowledgement trong SYN/ACK segment là: 1.

Sever xác định giá trị của Acknowledgement = X+1 với X là giá trị mà sequence number ở gói tin SYN mà client đã gửi trước đó.

Thành phần cờ Acknowledgement và cờ Syn cho ta biết segment đó là SYN/ACK segment.

Minh chứng:



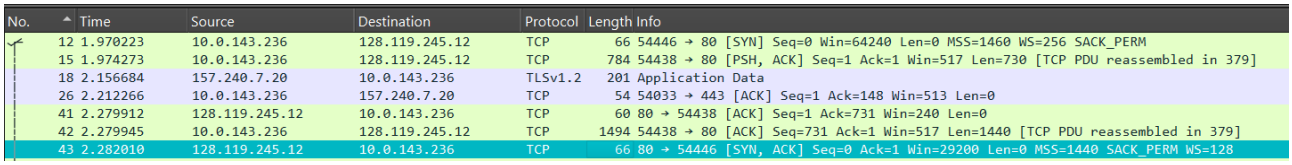
Câu 13. Tìm độ dài của từng segment trong bộ 6 segments đầu tiên trên? Tìm lượng buffer còn trống nhỏ nhất mà bên nhận thông báo cho bên gửi trong suốt quá trình truyền tin

Trả lời:

Độ dài của từng segment trong 6 bộ segment đầu tiên:

- Gói thứ 1 (Số 12): TCP segment có chiều dài là **66** bytes.
- Gói thứ 2 (Số 18): TCP segment có chiều dài là **784** bytes.
- Gói thứ 3 (Số 26): TLSv1.2 segment với Application Data có chiều dài **201** bytes (TLSv1.2, không phải TCP thông thường).
- Gói thứ 4 (Số 41): TCP segment có chiều dài là **60** bytes.
- Gói thứ 5 (Số 42): TCP segment có chiều dài là **1494** bytes.
- Gói thứ 6 (Số 43): TCP segment có chiều dài là **66** bytes.

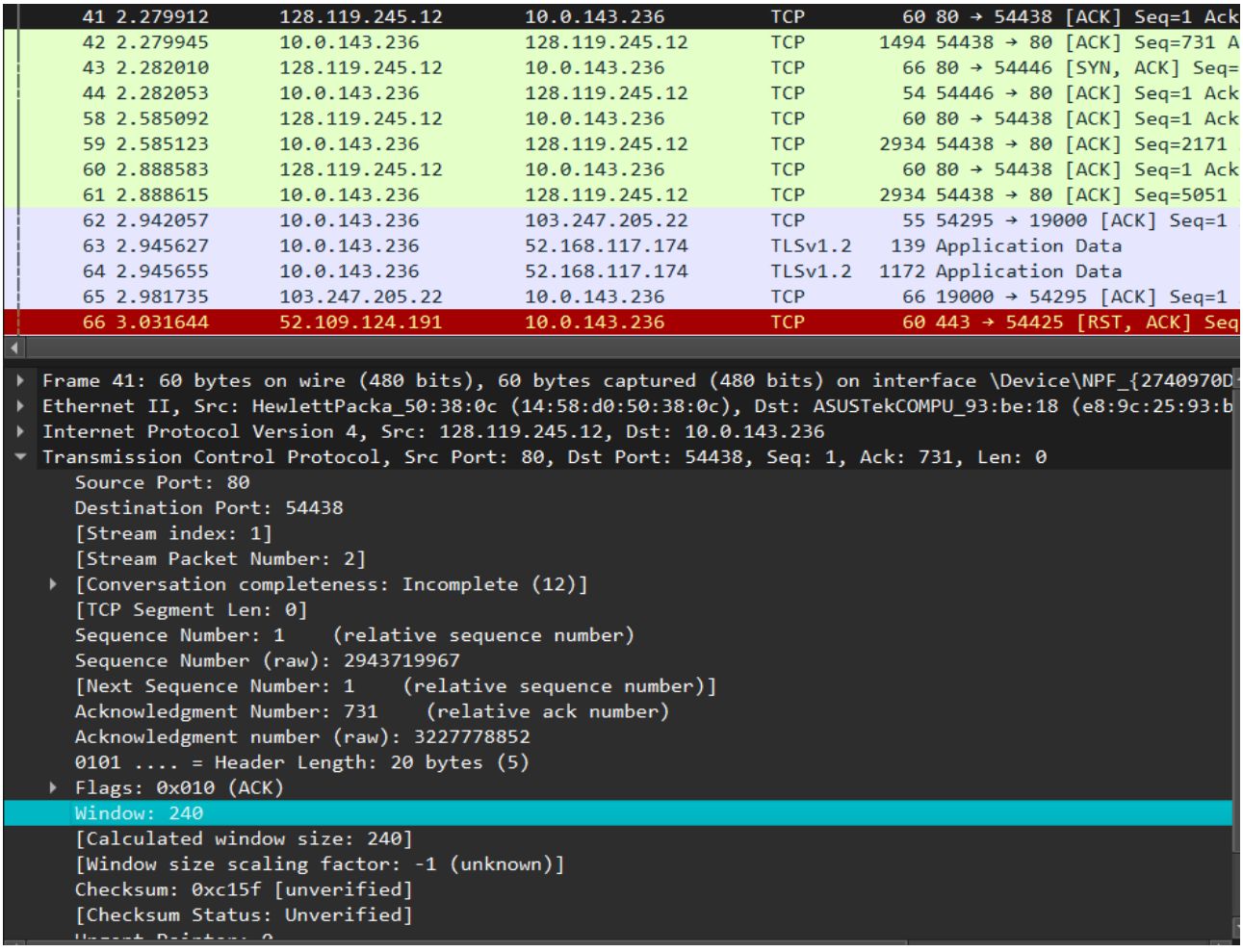
Minh chứng



No.	Time	Source	Destination	Protocol	Length	Info
12	1.970223	10.0.143.236	128.119.245.12	TCP	66	54446 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
15	1.974273	10.0.143.236	128.119.245.12	TCP	784	54438 → 80 [PSH, ACK] Seq=1 Ack=1 Win=517 Len=730 [TCP PDU reassembled in 379]
18	2.156684	157.240.7.20	10.0.143.236	TLSv1.2	201	Application Data
26	2.212266	10.0.143.236	157.240.7.20	TCP	54	54033 → 443 [ACK] Seq=1 Ack=148 Win=513 Len=0
41	2.279912	128.119.245.12	10.0.143.236	TCP	60	80 → 54438 [ACK] Seq=1 Ack=731 Win=240 Len=0
42	2.279945	10.0.143.236	128.119.245.12	TCP	1494	54438 → 80 [ACK] Seq=731 Ack=1 Win=517 Len=1440 [TCP PDU reassembled in 379]
43	2.282010	128.119.245.12	10.0.143.236	TCP	66	80 → 54446 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1440 SACK_PERM WS=128

Lượng buffer còn trống nhỏ nhất mà bên nhận thông báo cho bên gửi trong suốt quá trình truyền tin:

Vào **Request** -> chọn **Transmission Control Protocol** -> ở phần Window(cửa sổ nhận) hiển thị lượng buffer còn trống. Từ đó ta sẽ tìm được lượng buffer còn trống nhỏ nhất là **240** tại gói 41.



```
    41 2.279912     128.119.245.12     10.0.143.236       TCP      60 80 → 54438 [ACK] Seq=1 Ack
    42 2.279945     10.0.143.236       128.119.245.12     TCP    1494 54438 → 80 [ACK] Seq=731 A
    43 2.282010     128.119.245.12     10.0.143.236       TCP      66 80 → 54446 [SYN, ACK] Seq=
    44 2.282053     10.0.143.236       128.119.245.12     TCP      54 54446 → 80 [ACK] Seq=1 Ack
    58 2.585092     128.119.245.12     10.0.143.236       TCP      60 80 → 54438 [ACK] Seq=1 Ack
    59 2.585123     10.0.143.236       128.119.245.12     TCP    2934 54438 → 80 [ACK] Seq=2171
    60 2.888583     128.119.245.12     10.0.143.236       TCP      60 80 → 54438 [ACK] Seq=1 Ack
    61 2.888615     10.0.143.236       128.119.245.12     TCP    2934 54438 → 80 [ACK] Seq=5051
    62 2.942057     10.0.143.236       103.247.205.22     TCP      55 54295 → 19000 [ACK] Seq=1
    63 2.945627     10.0.143.236       52.168.117.174     TLSv1.2   139 Application Data
    64 2.945655     10.0.143.236       52.168.117.174     TLSv1.2  1172 Application Data
    65 2.981735     103.247.205.22     10.0.143.236       TCP      66 19000 → 54295 [ACK] Seq=1
    66 3.031644     52.109.124.191     10.0.143.236       TCP      60 443 → 54425 [RST, ACK] Seq

▸ Frame 41: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{2740970D
▸ Ethernet II, Src: HewlettPacka_50:38:0c (14:58:d0:50:38:0c), Dst: ASUSTekCOMPU_93:be:18 (e8:9c:25:93:b
▸ Internet Protocol Version 4, Src: 128.119.245.12, Dst: 10.0.143.236
▾ Transmission Control Protocol, Src Port: 80, Dst Port: 54438, Seq: 1, Ack: 731, Len: 0
    Source Port: 80
    Destination Port: 54438
    [Stream index: 1]
    [Stream Packet Number: 2]
  ▸ [Conversation completeness: Incomplete (12)]
    [TCP Segment Len: 0]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 2943719967
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 731    (relative ack number)
    Acknowledgment number (raw): 3227778852
    0101 .... = Header Length: 20 bytes (5)
  ▸ Flags: 0x010 (ACK)
    Window: 240
    [Calculated window size: 240]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0xc15f [unverified]
    [Checksum Status: Unverified]
```

Câu 14: Có segment nào được gửi lại hay không? Thông tin nào trong quá trình truyền tin cho chúng ta biết điều đó?

Trả lời:

Có segment được gửi lại. Thông tin trong quá trình truyền tin cho chúng ta biết điều đó là:

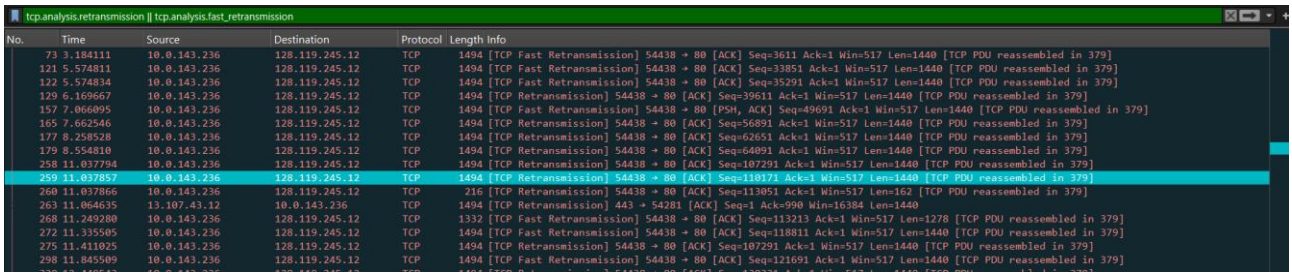
Để xác định liệu có **segment nào được gửi lại** trong quá trình truyền tin, ta có thể sử dụng:

Cách 1: Nhập vào filter: **tcp.analysis.retransmission || tcp.analysis.fast_retransmission**

- **tcp.analysis.retransmission**: Dùng để tìm các gói tin được gửi lại sau khi không nhận được ACK đúng hạn.
- **tcp.analysis.fast retransmission**: Dùng để tìm gói tin được gửi lại nhanh (sau khi nhận nhiều duplicate ACKs).

Nhấn Enter.

Nếu có bất kỳ segment nào được gửi lại, Wireshark sẽ hiển thị chúng.



Cách 2: Sử dụng biểu đồ Time-Sequence Graph trong Wireshark

Vào **Statistics → TCP Stream Graph → Time-Sequence Graph (Stevens)**.

Trong biểu đồ:

Nếu có **điểm trùng sequence number** trên trục Y tại **nhiều thời điểm khác nhau** trên trục X, điều này cho thấy một gói đã được gửi lại.