

Lab

3

BÁO CÁO BÀI THỰC HÀNH SỐ 3 Giao thức TCP & UDP

LAB 3 – UDP & TCP PROTOCOL

Môn học: Nhập môn Mạng máy tính

Sinh viên thực hiện	Mai Nguyễn Nam Phương (22521164)
Thời gian thực hiện	18/10/2023 – 24/10/2023
Tự chấm điểm	10/10

TRẢ LỜI CÁC CÂU HỎI

Câu 1. Điền thông tin vào bảng sau

Trả lời:

IP Address	172.30.220.151
MAC Address	2C-0D-A7-70-24-9D
Default gateway IP address	172.30.0.1
DNS sever IP address	8.8.8.8 8.8.4.4

```
Connection-specific DNS Suffix . :  
Description . . . . . : Intel(R) Wi-Fi 6E AX211 160MHz  
Physical Address. . . . . : 2C-0D-A7-70-24-9D  
DHCP Enabled. . . . . : Yes  
Autoconfiguration Enabled . . . . : Yes  
Link-local IPv6 Address . . . . . : fe80::f9d7:1a7e:5836:1736%17(Preferred)  
IPv4 Address. . . . . : 172.30.220.151(Preferred)  
Subnet Mask . . . . . : 255.255.0.0  
Lease Obtained. . . . . : Saturday, November 18, 2023 2:01:20 PM  
Lease Expires . . . . . : Saturday, November 18, 2023 9:00:45 PM  
Default Gateway . . . . . : 172.30.0.1  
DHCP Server . . . . . : 192.168.199.13  
DHCPv6 IAID . . . . . : 288099751  
DHCPv6 Client DUID. . . . . : 00-01-00-01-2C-91-64-E0-D4-93-90-1C-95-54  
DNS Servers . . . . . : 8.8.8.8  
                        8.8.4.4  
NetBIOS over Tcpip. . . . . : Enabled
```

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

Trả lời: Dựa vào hình ảnh ta thấy được gói tin truy vấn là gói tin thứ 194 và 201

Lab 1: Làm quen với Wireshark

190	3.353107	172.30.220.151	8.8.8.8	DNS	80 Standard query 0x0001 PTR 8.8.8.8.in-addr.arpa
193	3.411476	8.8.8.8	172.30.220.151	DNS	104 Standard query response 0x0001 PTR 8.8.8.8.in-addr.arpa PTR dns.google
194	3.413612	172.30.220.151	8.8.8.8	DNS	70 Standard query 0x0002 A google.com
200	3.475642	8.8.8.8	172.30.220.151	DNS	166 Standard query response 0x0002 A google.com A 74.125.68.113 A 74.125.68.138 A 74.125.68.102 A 74.125.68.100 A 74.125.68.139 A 74.125.68.101
201	3.478841	172.30.220.151	8.8.8.8	DNS	70 Standard query 0x0003 AAAA google.com
202	3.537529	8.8.8.8	172.30.220.151	DNS	182 Standard query response 0x0003 AAAA google.com AAAA 2404:6800:4003:c02::65 AAAA 2404:6800:4003:c02::71 AAAA 2404:6800:4003:c02::8a AAAA 2404:6800:4003:c02::8b

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

No.	Time	Source	Destination	Protocol	Length	Info
190	3.353107	172.30.220.151	8.8.8.8	DNS	80	Standard query 0x0001 PTR 8.8.8.8.in-addr.arpa
193	3.411476	8.8.8.8	172.30.220.151	DNS	104	Standard query response 0x0001 PTR 8.8.8.8.in-addr.arpa PTR dns.google
194	3.413612	172.30.220.151	8.8.8.8	DNS	70	Standard query 0x0002 A google.com
200	3.475642	8.8.8.8	172.30.220.151	DNS	166	Standard query response 0x0002 A google.com A 74.125.68.113 A 74.125.68.138 A 74.125.68.102 A 74.125.68.100 A 74.125.68.139 A 74.125.68.101
201	3.478841	172.30.220.151	8.8.8.8	DNS	70	Standard query 0x0003 AAAA google.com
202	3.537529	8.8.8.8	172.30.220.151	DNS	182	Standard query response 0x0003 AAAA google.com AAAA 2404:6800:4003:c02::65 AAAA 2404:6800:4003:c02::71 AAAA 2404:6800:4003:c02::8a AAAA 2404:6800:4003:c02::8b

Trả lời: Ta sẽ thấy được gói tin phản hồi là gói 200 và 202

Từ đó có IP của domain google.com là:

74.125.68.113 A 74.125.68.138 A 74.125.68.102 A 74.125.68.100 A 74.125.68.139 A 74.125.68.101 (IPv4)

2404:6800:4003:c02::65 AAAA 2404:6800:4003:c02::71 AAAA 2404:6800:4003:c02::8a AAAA 2404:6800:4003:c02::8b (IPv6)

Câu 4. Chọn 1 gói tin DNS, xác định các trường (field) có trong UDP header và giải thích ý nghĩa của mỗi trường đó

Trả lời: Bao gồm 4 field

Source Port: Port gốc

Destination Port: Port đích

Length: Độ dài gói tin

Checksum: Giá trị kiểm tra

- Giải thích:

+ Source Port: Trường này xác định cổng của người gửi thông tin và có ý nghĩa nếu muốn nhận thông tin phản hồi từ người nhận.

+ Destination Port: Trường xác định cổng nhận thông tin

+ Length: Trường có độ dài 16-bit xác định chiều dài của toàn bộ datagram: phần header và dữ liệu. Chiều dài tối thiểu là 8-byte khi gói tin không có dữ liệu, chỉ có header.

+ Checksum: Trường checksum 16-bit dùng cho việc kiểm tra lỗi của phần header và dữ liệu. Phương pháp tính checksum được định nghĩa trong RFC 768.

Lab 1: Làm quen với Wireshark

```
> Frame 194: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface \Device\NPF_{DE018CBF-DAC5-4360-BF93-EF11FDB427A}
> Ethernet II, Src: Intel_70:24:9d (2c:0d:a7:70:24:9d), Dst: JuniperNetwo_8c:35:b0 (44:f4:77:8c:35:b0)
> Internet Protocol Version 4, Src: 172.30.220.151, Dst: 8.8.8.8
- User Datagram Protocol, Src Port: 57593, Dst Port: 53
  Source Port: 57593
  Destination Port: 53
  Length: 36
  Checksum: 0x98fb [unverified]
  [Checksum Status: Unverified]
  [Stream index: 100]
  > [Timestamps]
    UDP payload (28 bytes)
  > Domain Name System (query)
```

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

Trả lời:

UDP gồm 4 trường, kích thước mỗi trường:

- Source Port: 2 bytes
- Destination port: 2 bytes
- Length: 2 bytes
- Checksum: 2 bytes

Câu 6. Giá trị của trường Length trong UDP header là độ dài của gì? Chứng minh nhận định này

Trả lời: Giá trị của trường Length là trường chỉ định độ dài tổng cộng của gói UDP, bao gồm cả header và dữ liệu. Trường này đo lường bằng byte và bao gồm độ dài của cả phần header và phần dữ liệu

Ở đây độ dài của gói tin bao gồm 8 bytes header + độ dài data

Length (36) = 8 + data (28)

```
Source Port: 57593
Destination Port: 53
Length: 36
Checksum: 0x98fb [unverified]
[Checksum Status: Unverified]
[Stream index: 100]
> [Timestamps]
  UDP payload (28 bytes)
```

Câu 7. Quan sát 2 gói tin tìm được ở câu 2 và câu 3, mô tả mối quan hệ giữa các địa chỉ IP và port number của 2 gói tin này

Trả lời: Trong quá trình gửi yêu cầu, IP nguồn gửi Request sẽ trở thành điểm đích và Source Port sẽ trở thành Destination Port. IP người gửi Response sẽ trở thành IP nguồn hay có thể hiểu là các giá trị IP và Port của nguồn và đích sẽ đảo lại với nhau

Lab 1: Làm quen với Wireshark

No.	Time	Source	Destination	Protocol	Length	Info
190	3.353107	172.30.220.151	8.8.8.8	DNS	80	Standard query 0x0001 PTR 8.8.8.8.in-addr.arpa
193	3.411476	8.8.8.8	172.30.220.151	DNS	104	Standard query response 0x0001 PTR 8.8.8.8.in-addr.arpa
194	3.413612	172.30.220.151	8.8.8.8	DNS	70	Standard query 0x0002 A google.com
200	3.475642	8.8.8.8	172.30.220.151	DNS	166	Standard query response 0x0002 A google.com A 74.125.68.1
201	3.478841	172.30.220.151	8.8.8.8	DNS	70	Standard query 0x0003 AAAA google.com
202	3.537529	8.8.8.8	172.30.220.151	DNS	182	Standard query response 0x0003 AAAA google.com AAAA 2404:

<

> Frame 194: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface \Device\NPF_{DE018CBF-DAC5-4360-BF93-EF11FDB427AC},
 > Ethernet II, Src: Intel_70:24:9d (2c:0d:a7:70:24:9d), Dst: JuniperNetwo_8c:35:b0 (44:f4:77:8c:35:b0)
 > Internet Protocol Version 4, Src: 172.30.220.151, Dst: 8.8.8.8
 ✓ User Datagram Protocol, Src Port: 57593, Dst Port: 53
 Source Port: 57593
 Destination Port: 53
 Length: 36
 Checksum: 0x98fb [unverified]
 [Checksum Status: Unverified]
 [Stream index: 100]
 > [Timestamps]
 UDP payload (28 bytes)
 > Domain Name System (query)

No.	Time	Source	Destination	Protocol	Length	Info
190	3.353107	172.30.220.151	8.8.8.8	DNS	80	Standard query 0x0001 PTR 8.8.8.8.in-addr.arpa
193	3.411476	8.8.8.8	172.30.220.151	DNS	104	Standard query response 0x0001 PTR 8.8.8.8.in-addr.arpa
194	3.413612	172.30.220.151	8.8.8.8	DNS	70	Standard query 0x0002 A google.com
200	3.475642	8.8.8.8	172.30.220.151	DNS	166	Standard query response 0x0002 A google.com A 74.125.68.1
201	3.478841	172.30.220.151	8.8.8.8	DNS	70	Standard query 0x0003 AAAA google.com
202	3.537529	8.8.8.8	172.30.220.151	DNS	182	Standard query response 0x0003 AAAA google.com AAAA 2404:

<

> Frame 200: 166 bytes on wire (1328 bits), 166 bytes captured (1328 bits) on interface \Device\NPF_{DE018CBF-DAC5-4360-BF93-EF11FDB427AC},
 > Ethernet II, Src: JuniperNetwo_8c:35:b0 (44:f4:77:8c:35:b0), Dst: Intel_70:24:9d (2c:0d:a7:70:24:9d)
 > Internet Protocol Version 4, Src: 8.8.8.8, Dst: 172.30.220.151
 ✓ User Datagram Protocol, Src Port: 53, Dst Port: 57593
 Source Port: 53
 Destination Port: 57593
 Length: 132
 Checksum: 0x111d [unverified]
 [Checksum Status: Unverified]
 [Stream index: 100]
 > [Timestamps]
 UDP payload (124 bytes)
 > Domain Name System (response)

Lab 1: Làm quen với Wireshark

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

Trả lời: Dựa trên gợi ý ta chọn vào gói tin thông điệp HTTP là 218 và biết được IP của client là 172.30.220.151 và sử dụng Port 54074

No.	Time	Source	Destination	Protocol	Length	Info
212	3.741778	172.30.220.151	128.119.245.12	TCP	14574	54074 → 80 [PSH, ACK] Seq=102226 Ack=1 Win=132096 Len=14520 [TCP segment of a reassembled PDU]
213	3.741852	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=58666 Win=146560 Len=0
214	3.741863	172.30.220.151	128.119.245.12	TCP	14574	54074 → 80 [ACK] Seq=116746 Ack=1 Win=132096 Len=14520 [TCP segment of a reassembled PDU]
215	3.742732	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=65926 Win=161152 Len=0
216	3.742739	172.30.220.151	128.119.245.12	TCP	14574	54074 → 80 [PSH, ACK] Seq=131266 Ack=1 Win=132096 Len=14520 [TCP segment of a reassembled PDU]
217	3.744742	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=73186 Win=175616 Len=0
218	3.744764	172.30.220.151	128.119.245.12	HTTP	7175	POST /wireshark-labs/lab3-1-reply.htm HTTP/1.1 (text/plain)
219	3.745214	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=80446 Win=179584 Len=0
220	3.745797	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=87706 Win=179584 Len=0
221	3.745979	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=94966 Win=179584 Len=0
222	3.746954	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=96418 Win=178560 Len=0
225	3.990931	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=102226 Win=180608 Len=0
234	4.018656	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=109486 Win=197760 Len=0
235	4.018880	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=116746 Win=212224 Len=0
236	4.019464	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=124006 Win=226816 Len=0
237	4.019553	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=131266 Win=241280 Len=0
238	4.020305	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=138526 Win=255872 Len=0
239	4.020878	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=145786 Win=270336 Len=0
240	4.021688	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=152907 Win=284544 Len=0
241	4.023961	128.119.245.12	172.30.220.151	HTTP	831	HTTP/1.1 200 OK (text/html)
242	4.073573	172.30.220.151	128.119.245.12	TCP	54	54074 → 80 [ACK] Seq=152907 Ack=778 Win=131328 Len=0
259	4.602224	172.30.220.151	52.64.218.251	TCP	55	53981 → 443 [ACK] Seq=1 Ack=1 Win=4139 Len=1 [TCP segment of a reassembled PDU]
266	4.756927	52.64.218.251	172.30.220.151	TCP	66	443 → 53981 [ACK] Seq=1 Ack=2 Win=425 Len=0 SLE=1 SRE=2

> Frame 218: 7175 bytes on wire (57400 bits), 7175 bytes captured (57400 bits) on interface \Device\NPF_{DE018CBF-DAC5-4360-BF93-EF1...}

> Ethernet II, Src: Intel_70:24:9d (2c:0d:a7:70:24:9d), Dst: JuniperNetwo_8c:35:b0 (44:f4:77:8c:35:b0)

> Internet Protocol Version 4, Src: 172.30.220.151, Dst: 128.119.245.12

> Transmission Control Protocol, Src Port: 54074, Dst Port: 80, Seq: 145786, Ack: 1, Len: 7121

> [15 Reassembled TCP Segments (152906 bytes): #140(585), #141(13068), #167(1452), #169(14520), #171(11616), #184(2904), #186(14520)]

> Hypertext Transfer Protocol

> MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "----WebKitFormBoundaryARAZy0wPvq4h0ZXd"

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

Trả lời: Tương tự như câu trên thì địa chỉ IP của `gaia.cs.umass.edu` là 128.119.245.12 và nó gửi và nhận các segment TCP cho kết nối bằng số cổng 80

No.	Time	Source	Destination	Protocol	Length	Info
212	3.741778	172.30.220.151	128.119.245.12	TCP	14574	54074 → 80 [PSH, ACK] Seq=102226 Ack=1 Win=132096 Len=14520 [TCP segment of a reassembled PDU]
213	3.741852	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=58666 Win=146560 Len=0
214	3.741863	172.30.220.151	128.119.245.12	TCP	14574	54074 → 80 [ACK] Seq=116746 Ack=1 Win=132096 Len=14520 [TCP segment of a reassembled PDU]
215	3.742732	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=65926 Win=161152 Len=0
216	3.742739	172.30.220.151	128.119.245.12	TCP	14574	54074 → 80 [PSH, ACK] Seq=131266 Ack=1 Win=132096 Len=14520 [TCP segment of a reassembled PDU]
217	3.744742	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=73186 Win=175616 Len=0
218	3.744764	172.30.220.151	128.119.245.12	HTTP	7175	POST /wireshark-labs/lab3-1-reply.htm HTTP/1.1 (text/plain)
219	3.745214	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=80446 Win=179584 Len=0
220	3.745797	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=87706 Win=179584 Len=0
221	3.745979	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=94966 Win=179584 Len=0
222	3.746954	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=96418 Win=178560 Len=0
225	3.990931	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=102226 Win=180608 Len=0
234	4.018656	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=109486 Win=197760 Len=0
235	4.018880	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=116746 Win=212224 Len=0
236	4.019464	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=124006 Win=226816 Len=0
237	4.019553	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=131266 Win=241280 Len=0
238	4.020305	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=138526 Win=255872 Len=0
239	4.020878	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=145786 Win=270336 Len=0
240	4.021688	128.119.245.12	172.30.220.151	TCP	56	80 → 54074 [ACK] Seq=1 Ack=152907 Win=284544 Len=0
241	4.023961	128.119.245.12	172.30.220.151	HTTP	831	HTTP/1.1 200 OK (text/html)
242	4.073573	172.30.220.151	128.119.245.12	TCP	54	54074 → 80 [ACK] Seq=152907 Ack=778 Win=131328 Len=0
259	4.602224	172.30.220.151	52.64.218.251	TCP	55	53981 → 443 [ACK] Seq=1 Ack=1 Win=4139 Len=1 [TCP segment of a reassembled PDU]
266	4.756927	52.64.218.251	172.30.220.151	TCP	66	443 → 53981 [ACK] Seq=1 Ack=2 Win=425 Len=0 SLE=1 SRE=2

> Frame 218: 7175 bytes on wire (57400 bits), 7175 bytes captured (57400 bits) on interface \Device\NPF_{DE018CBF-DAC5-4360-BF93-EF1...}

> Ethernet II, Src: Intel_70:24:9d (2c:0d:a7:70:24:9d), Dst: JuniperNetwo_8c:35:b0 (44:f4:77:8c:35:b0)

> Internet Protocol Version 4, Src: 172.30.220.151, Dst: 128.119.245.12

> Transmission Control Protocol, Src Port: 54074, Dst Port: 80, Seq: 145786, Ack: 1, Len: 7121

> [15 Reassembled TCP Segments (152906 bytes): #140(585), #141(13068), #167(1452), #169(14520), #171(11616), #184(2904), #186(14520)]

> Hypertext Transfer Protocol

> MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "----WebKitFormBoundaryARAZy0wPvq4h0ZXd"

Lab 1: Làm quen với Wireshark

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 bằng 0 để tạo kết nối TCP giữa client và server

No.	Time	Source	Destination	Protocol	Length	Info
69	1.350127	157.240.7.50	172.30.220.151	TLSv1.2	124	Application Data
70	1.393622	172.30.220.151	157.240.7.50	TCP	54	52658 → 443 [ACK] Seq=1 Ack=71 Win=512 Len=0
112	2.588612	172.30.220.151	128.119.245.12	TCP	54	54065 → 443 [FIN, ACK] Seq=1 Ack=1 Win=514 Len=0
113	2.588686	172.30.220.151	128.119.245.12	TCP	54	54065 → 443 [RST, ACK] Seq=2 Ack=1 Win=0 Len=0
124	2.661909	172.30.220.151	128.119.245.12	TCP	66	54074 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
134	2.840677	172.30.220.151	128.119.245.12	TCP	66	54075 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
135	2.850117	128.119.245.12	172.30.220.151	TCP	56	443 → 54065 [RST] Seq=1 Win=0 Len=0
138	2.924971	128.119.245.12	172.30.220.151	TCP	66	80 → 54074 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1452 SACK_P
139	2.925022	172.30.220.151	128.119.245.12	TCP	54	54074 → 80 [ACK] Seq=1 Ack=1 Win=132096 Len=0

Ta kiểm tra trong trường Flags của gói, nếu cờ SYN được set = 1 thì đó là TCP SYN segment

```
Wireshark · Packet 124 · 22521164-TCP.pcapng

[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 0
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
  .... .0.. = ECN-Echo: Not set
  .... ..0. = Urgent: Not set
  .... ...0 = Acknowledgment: Not set
  .... ....0... = Push: Not set
  .... .... .0.. = Reset: Not set
  > .... .... ..1. = Syn: Set
  .... .... ...0 = Fin: Not set
  [TCP Flags: .....S.]
Window: 64240
[Calculated window size: 64240]
```

Câu 11. Tìm sequence number của gói tin SYN/ACK segment được gửi bởi server đến client để trả lời cho SYN segment

Trả lời: Ta có thể thấy gói tin SYN/ACK segment được gửi bởi server đến client để trả lời cho SYN segment có rất nhiều, ở đây ta lấy ví dụ là gói tin 138 => có thể thấy sequence number của nó là 0

Lab 1: Làm quen với Wireshark

124	2.661909	172.30.220.151	128.119.245.12	TCP	66 54074 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
134	2.840677	172.30.220.151	128.119.245.12	TCP	66 54075 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
135	2.850117	128.119.245.12	172.30.220.151	TCP	56 443 → 54065 [RST] Seq=1 Win=0 Len=0
138	2.924971	128.119.245.12	172.30.220.151	TCP	66 80 → 54074 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1452 SACK_PERM WS=128
139	2.925022	172.30.220.151	128.119.245.12	TCP	54 54074 → 80 [ACK] Seq=1 Ack=1 Win=132096 Len=0
140	2.925496	172.30.220.151	128.119.245.12	TCP	639 54074 → 80 [PSH, ACK] Seq=1 Ack=1 Win=132096 Len=585 [TCP segment of a reassemb]
141	2.925655	172.30.220.151	128.119.245.12	TCP	13122 54074 → 80 [ACK] Seq=586 Ack=1 Win=132096 Len=13068 [TCP segment of a reassemb]

Câu 12. Tìm giá trị của Acknowledgement trong SYN/ACK segment? Làm sao server 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, server có thể xác định nó nhờ vào thông tin của gói tin và có thể nhận biết nó qua trường Flags trong SYN/ACK segment

```
▼ Transmission Control Protocol, Src Port: 80, Dst Port: 54074, Seq: 0, Ack: 1, Len: 0
  Source Port: 80
  Destination Port: 54074
  [Stream index: 2]
  > [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 0]
  Sequence Number: 0 (relative sequence number)
  Sequence Number (raw): 3724965091
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 1475741997
  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
    > ....0... = Syn: Set
    ....0... = Fin: Not set
  [TCP Flags: .....A..S.]
  ... ..
```

Ta có thể biết segment đó là SYN/ACK segment nhờ vào trường Info của Wireshark

135	2.850117	128.119.245.12	172.30.220.151	TCP	56 443 → 54065 [RST] Seq=1 Win=0 Len=0
138	2.924971	128.119.245.12	172.30.220.151	TCP	66 80 → 54074 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1452 SACK
139	2.925022	172.30.220.151	128.119.245.12	TCP	54 54074 → 80 [ACK] Seq=1 Ack=1 Win=132096 Len=0
140	2.925496	172.30.220.151	128.119.245.12	TCP	639 54074 → 80 [PSH, ACK] Seq=1 Ack=1 Win=132096 Len=585 [TCP segme

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

Trả lời: 6 segment đầu tiên là 140,141,167,169,171 và 184 với độ dài lần lượt là 585 bytes, 13068 bytes, 1452 bytes, 14520 bytes, 11616 bytes, 2904 bytes

Lượng buffer còn trống nhỏ nhất là 29200

Lab 1: Làm quen với Wireshark

```
> Frame 218: 7175 bytes on wire (57400 bits), 7175 bytes captured (57400 bits) on interface \Device\NPF_{DE018CBF-DAC5-4360-BF93-EF11F0
> Ethernet II, Src: Intel_70:24:9d (2c:0d:a7:70:24:9d), Dst: JuniperNetwo_8c:35:b0 (44:f4:77:8c:35:b0)
> Internet Protocol Version 4, Src: 172.30.220.151, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 54074, Dst Port: 80, Seq: 145786, Ack: 1, Len: 7121
▼ [15 Reassembled TCP Segments (152906 bytes): #140(585), #141(13068), #167(1452), #169(14520), #171(11616), #184(2904), #186(14520), #
  [Frame: 140, payload: 0-584 (585 bytes)]
  [Frame: 141, payload: 585-13652 (13068 bytes)]
  [Frame: 167, payload: 13653-15104 (1452 bytes)]
  [Frame: 169, payload: 15105-29624 (14520 bytes)]
  [Frame: 171, payload: 29625-41240 (11616 bytes)]
  [Frame: 184, payload: 41241-44144 (2904 bytes)]
  [Frame: 186, payload: 44145-58664 (14520 bytes)]
  [Frame: 188, payload: 58665-73184 (14520 bytes)]
  [Frame: 190, payload: 73185-87704 (14520 bytes)]
  [Frame: 192, payload: 87705-96416 (8712 bytes)]
  [Frame: 210, payload: 96417-102224 (5808 bytes)]
  [Frame: 212, payload: 102225-116744 (14520 bytes)]
  [Frame: 214, payload: 116745-131264 (14520 bytes)]
  [Frame: 216, payload: 131265-145784 (14520 bytes)]
  [Frame: 218, payload: 145785-152905 (7121 bytes)]
  [Segment count: 15]
  [Reassembled TCP length: 152906]
  [Reassembled TCP Data [truncated]: 504f5354202f77697265736861726b2d6c6162732f6c6162332d312d7265706c792e68746d20485454502f312e310d0
> Hypertext Transfer Protocol
> MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "----WebKitFormBoundaryARAZy0wPvq4h0ZXd"

> Frame 138: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{DE018CBF-DAC5-4360-BF93-EF11FDB427AC},
> Ethernet II, Src: JuniperNetwo_8c:35:b0 (44:f4:77:8c:35:b0), Dst: Intel_70:24:9d (2c:0d:a7:70:24:9d)
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 172.30.220.151
▼ Transmission Control Protocol, Src Port: 80, Dst Port: 54074, Seq: 0, Ack: 1, Len: 0
  Source Port: 80
  Destination Port: 54074
  [Stream index: 2]
> [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 0]
  Sequence Number: 0 (relative sequence number)
  Sequence Number (raw): 3724965091
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 1475741997
  1000 .... = Header Length: 32 bytes (8)
> Flags: 0x012 (SYN, ACK)
  Window: 29200
  [Calculated window size: 29200]
  Checksum: 0x7726 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
> Options: (12 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted, No-Operation (NOP), Window scal
> [Timestamps]
> [SEQ/ACK analysis]
```

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

- Không có segment nào được gửi lại
- Dựa vào biểu đồ phân tích truyền gói tin có thể thấy các gói tin có cùng sequence number ở các thời điểm khác nhau không được tìm thấy

Lab 1: Làm quen với Wireshark

