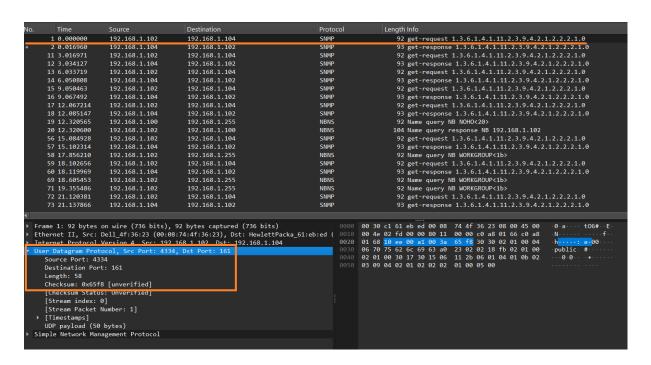
### Lab 3

Họ tên	Trần Nguyễn Thái Bình
MSSV	23520161
Lớp	IT005.P13.2

Từ câu 1 → 7 em sử dụng file <a href="http-ethereal-trace-5">http-ethereal-trace-5</a>

Từ câu 8 → 15 em sử dụng file tcp-ethereal-trace-1

### 1. Chọn một gói tin UDP, xác định các trường (field) trong UDP header?



Source Port: 4334

Destination Port: 161

• **Length**: 58

• Checksum: 0x65f8 [unverified]

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

Vì có trường Length: 58 (bytes) và UDP payload: 50 bytes, có 4 trường →
Mỗi trường có (58 - 50) / 4 = 2 bytes

#### 3. Giá trị của trường Length là độ dài của cái gì?

 Giá trị trường Length là độ dài của toàn packet (bao gồm 4 trường UDP và payload)

### 4. Số bytes lớn nhất mà payload của UDP có thể chứa?

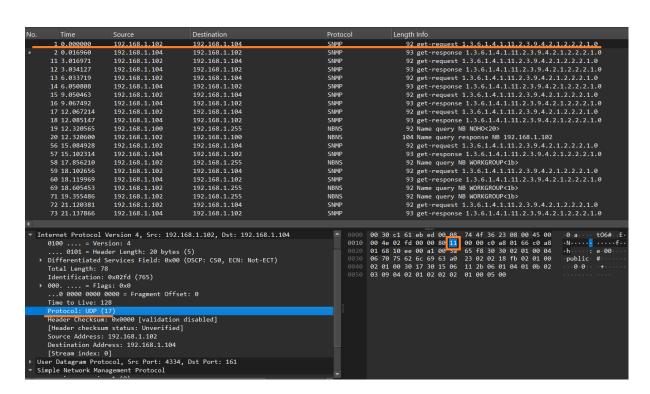
• IPv4: 65,507 bytes

• IPv6: 65,527 bytes

### 5. Giá trị lớn nhất có thể có của port nguồn?

Vì trường port có 2 bytes = 16 bits → Giá trị port tối đa 2^16 - 1 = 65535

6. Xác định protocol number của UDP (cả hệ 10 lẫn hệ 16)? Đế trả lời câu hỏi này, chúng ta cần phải xem trường Protocol của IP header.



Protocol: "17" (hệ 10) và "0x11" (hệ 16)

7. Kiểm tra một cặp gói tin gồm: gói tin do máy mình gửi và gói tin phản hồi của gói tin đó. Miêu tả mối quan hệ về port number

#### của 2 gói tin.

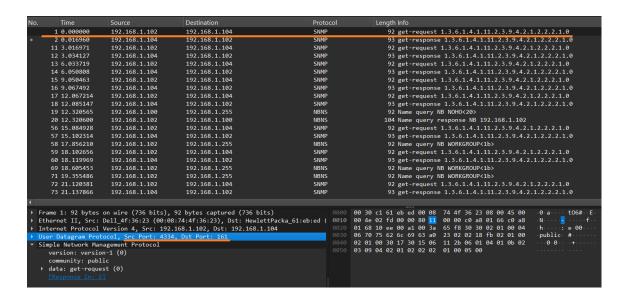
**Source Port** (Cổng nguồn): Là số cổng của thiết bị gửi gói tin trong một kết nối mạng. Nó được sử dụng để xác định ứng dụng hoặc dịch vụ cụ thể đang gửi dữ liệu trên máy tính nguồn.

 Destination Port (Cổng đích): Là số cổng của thiết bị nhận gói tin. Nó giúp xác định ứng dụng hoặc dịch vụ cụ thể trên thiết bị đích mà gói tin sẽ được chuyển đến.

#### Với:

- IP client: 192.168.1.102- IP server: 192.168.1.104

Gửi từ máy đến server



Source port: 4434

Destination port: 161

Gửi từ server về máy

		-	- · ·		
No.	Time	Source	Destination	Protocol	Length Info
•	1 0.000000	192.168.1.102	192.168.1.104	SNMP	92 get-request 1.3.6.1.4.1.11.2.3.9.4.2.1.2.2.2.1.0
	2 0.016960	192.168.1.104	192.168.1.102	SNMP	93 get-response 1.3.6.1.4.1.11.2.3.9.4.2.1.2.2.2.1.0
	11 3.016971	192.168.1.102	192.168.1.104	SNMP	92 get-request 1.3.6.1.4.1.11.2.3.9.4.2.1.2.2.2.1.0
	12 3.034127	192.168.1.104	192.168.1.102	SNMP	93 get-response 1.3.6.1.4.1.11.2.3.9.4.2.1.2.2.2.1.0
	13 6.033719	192.168.1.102	192.168.1.104	SNMP	92 get-request 1.3.6.1.4.1.11.2.3.9.4.2.1.2.2.2.1.0
	14 6.050808	192.168.1.104	192.168.1.102	SNMP	93 get-response 1.3.6.1.4.1.11.2.3.9.4.2.1.2.2.2.1.0
	15 9.050463	192.168.1.102	192.168.1.104	SNMP	92 get-request 1.3.6.1.4.1.11.2.3.9.4.2.1.2.2.2.1.0
	16 9.067492	192.168.1.104	192.168.1.102	SNMP	93 get-response 1.3.6.1.4.1.11.2.3.9.4.2.1.2.2.2.1.0
	17 12.067214	192.168.1.102	192.168.1.104	SNMP	92 get-request 1.3.6.1.4.1.11.2.3.9.4.2.1.2.2.2.1.0
	18 12.085147	192.168.1.104	192.168.1.102	SNMP	93 get-response 1.3.6.1.4.1.11.2.3.9.4.2.1.2.2.2.1.0
	19 12.320565	192.168.1.100	192.168.1.255	NBNS	92 Name query NB NOHO<20>
	20 12.320600	192.168.1.102	192.168.1.100	NBNS	104 Name query response NB 192.168.1.102
	56 15.084928	192.168.1.102	192.168.1.104	SNMP	92 get-request 1.3.6.1.4.1.11.2.3.9.4.2.1.2.2.2.1.0
	57 15.102314	192.168.1.104	192.168.1.102	SNMP	93 get-response 1.3.6.1.4.1.11.2.3.9.4.2.1.2.2.2.1.0
	58 17.856210	192.168.1.102	192.168.1.255	NBNS	92 Name query NB WORKGROUP<1b>
	59 18.102656	192.168.1.102	192.168.1.104	SNMP	92 get-request 1.3.6.1.4.1.11.2.3.9.4.2.1.2.2.2.1.0
	60 18.119969	192.168.1.104	192.168.1.102	SNMP	93 get-response 1.3.6.1.4.1.11.2.3.9.4.2.1.2.2.2.1.0
	69 18.605453	192.168.1.102	192.168.1.255	NBNS	92 Name query NB WORKGROUP<1b>
	71 19.355486	192.168.1.102	192.168.1.255	NBNS	92 Name query NB WORKGROUP<1b>
	72 21.120381	192.168.1.102	192.168.1.104	SNMP	92 get-request 1.3.6.1.4.1.11.2.3.9.4.2.1.2.2.2.1.0
	73 21.137866	192.168.1.104	192.168.1.102	SNMP	93 get-response 1.3.6.1.4.1.11.2.3.9.4.2.1.2.2.2.1.0
4					
b Er	ame 2. 93 hytes	on wire (744 hits)	93 bytes captured (744	hits) 000	00 00 08 74 4f 36 23 00 30 c1 61 eb ed 08 00 45 00 · t06# 0 a · · · E
			ed (00:30:c1:61:eb:ed),		
			.168.1.104, Dst: 192.168		00 01 66 00 a1 10 ee 00 3b 53 f2 30 31 02 01 00 04 f; S.01
		ocol, Src Port: 161,		003	06 70 75 62 6c 69 63 a2 24 02 02 18 fb 02 01 00 public \$
		agement Protocol		004	00 02 01 00 30 18 30 16 06 11 2b 06 01 04 01 0b 02 ···0·0····+·····
	version: version-1 (0)				00 03 09 04 02 01 02 02 02 01 00 04 01 10
	community: public				
	b data: get-response (2)				
Response To: 1					
	[Time: 0.016960000 seconds]				
	[		·		

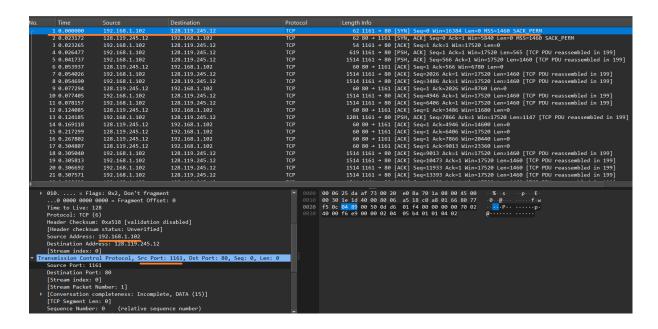
Source port: 161

o Destination port: 4334

 Vậy gửi từ port nào từ client đến port nào của server thì khi nhận cũng sẽ nhận tương ứng

• Client (4434) - Server (80)

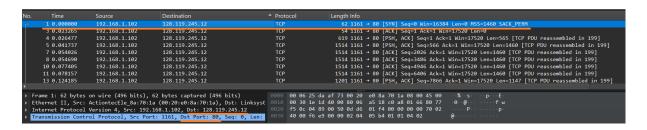
# 8. Tìm địa chỉ IP và TCP port của máy khách gửi file cho gaia.cs.umass.edu?



• IP: 192.168.1.102

• TCP port: 1161

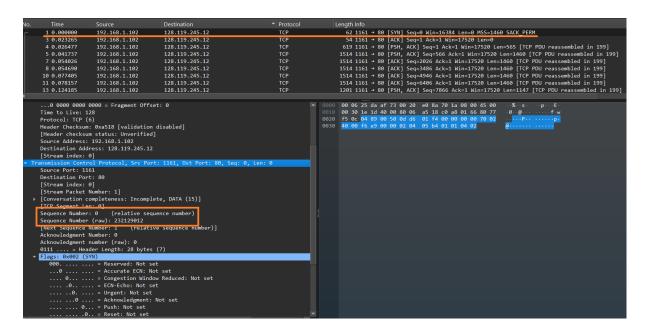
# 9. Tìm địa chỉ IP của <u>gaia.cs.umass.edu</u>? Kết nối TCP dùng để gửi và nhận các segments sử dụng port nào?



• IP: 128.119.245.12

Port: 80

# 10. TCP SYN segment sử dụng sequence number nào để khởi tạo kết nối TCP giữa máy khách và gaia.cs.umass.edu? Thành phần nào trong segment cho ta biết segment đó là TCP SYN segment?

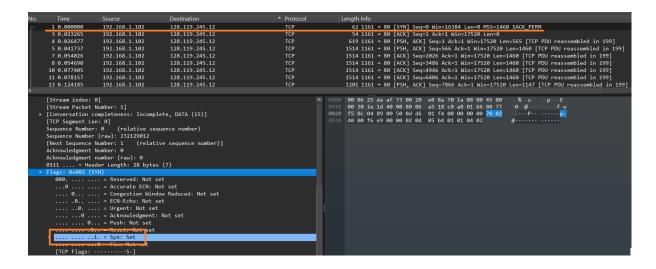


TCP SYN segment sử dụng:

Sequence number: 0 (relative sequence number)

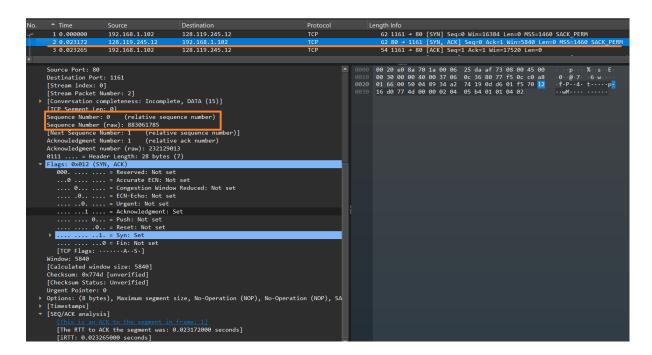
Sequence number (raw): 232129012

Lab 3 5



Thành phần TCP Flags cho ta thấy flag Syn được set là 1

# 11. Tìm sequence number của SYNACK segment được gửi bởi gaia.cs.umass.edu đến máy khách để trả lời cho SYN segment? Tìm giá trị của Acknowledgement trong SYNACK segment?

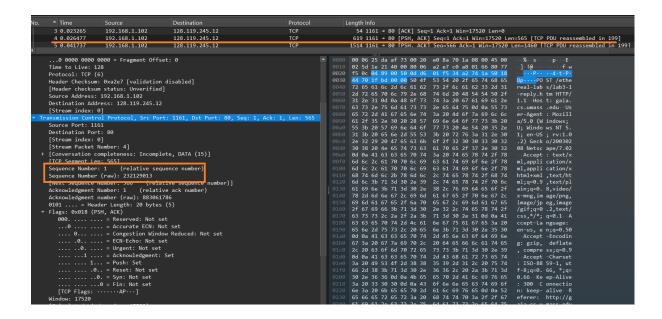


Sequence number: 0 (relative sequence number)

Sequence number (raw): 88301785

- Acknowledgement number: 1 (relative ack number)
- Acknowledgement number (raw): 232129013

## 12. Tìm sequence number của TCP segment có chứa lệnh HTTP POST?



- Sequence number: 1 (relative sequence number)
- Sequence number (raw): 232129013

Lab 3 7

13. Giả thiết rằng TCP segment chứa lệnh HTTP POST là segment đầu tiên của kết nối TCP. Tìm sequence number của 6 segments đầu tiên (tính cả segment có chứa HTTP POST)? Thời gian mà mỗi segment được gửi? Thời gian ACK cho mỗi segment được nhận? Đưa ra sự khác nhau giữa thời gian mà mỗi segment được gửi và thời gian ACK cho mỗi segment được nhận, tính RTT cho 6 segments? Tính EstimatedRTT sau khi nhận mỗi ACK? Giả sử EstimatedRTT bằng với RTT cho segment đầu tiên, sau đó tính EstimatedRTT với công thức trong giáo trình trang 239 cho các segment tiếp theo.

 6 sequences của HTTP POST (các giá trị tính bằng s kể từ khi client kết nối đến server)

Em sử dụng thêm filter <u>ip.src\_host == "192.168.1.102"</u> để lọc ra request gửi từ máy client

<b>∏</b> tcp	tcp && ip.src_host == "192.168.1.102"				
No.		Source	Destination	Protocol	Length Info
	4 0.026477	192.168.1.102	128.119.245.12	TCP	619 1161 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=565 [TCP PDU reassembled in 199]
	5 0.041737	192.168.1.102	128.119.245.12		1514 1161 → 80 [PSH, ACK] Seq=566 Ack=1 Win=17520 Len=1460 [TCP PDU reassembled in 199]
	7 0.054026	192.168.1.102	128.119.245.12		1514 1161 → 80 [ACK] Seq=2026 Ack=1 Win=17520 Len=1460 [TCP PDU reassembled in 199]
	8 0.054690	192.168.1.102	128.119.245.12		1514 1161 → 80 [ACK] Seq=3486 Ack=1 Win=17520 Len=1460 [TCP PDU reassembled in 199]
	10 0.077405	192.168.1.102	128.119.245.12		1514 1161 → 80 [ACK] Seq=4946 Ack=1 Win=17520 Len=1460 [TCP PDU reassembled in 199]
	11 0.078157	192.168.1.102	128.119.245.12		1514 1161 → 80 [ACK] Seq=6406 Ack=1 Win=17520 Len=1460 [TCP PDU reassembled in 199]

• Thời gian 6 ACK mà server gửi về tương ứng

Em sử dụng thêm filter <u>ip.src\_host</u> == "128.119.245.12" để lọc ra request gửi từ server



Bảng giá trị thời gian

STT	Client → Server (s)	Server → Client (s)	RTT (s)
1	0.026477000	0.053937000	0.02746
2	0.041737000	0.077294000	0.035557
3	0.054026000	0.124085000	0.070059
4	0.054690000	0.169118000	0.114428

STT	Client → Server (s)	Server → Client (s)	RTT (s)
5	0.077405000	0.217299000	0.139894
6	0.078157000	0.267802000	0.189645

#### · Công thức

EstimatedRTT<sub>new</sub> = 
$$(1 - \alpha) \times \text{EstimatedRTT}_{\text{old}} + \alpha \times \text{sampleRTT}$$

• Chọn

D

#### Tính toán

EstimatedRTT $_1 = 0.027460$ 

EstimatedRTT<sub>2</sub> = 
$$(1 - 0.125) \times 0.027460 + 0.125 \times 0.035557 = 0.02404875 + 0.004444625 = 0.031205375$$

EstimatedRTT<sub>4</sub> = 
$$(1 - 0.125) \times 0.036562075625 + 0.125 \times 0.114428 = 0.03174156609375 + 0.0143035 = 0.04604506609375$$

$$\begin{aligned} \text{EstimatedRTT}_5 &= (1-0.125) \times 0.04604506609375 + 0.125 \times \\ 0.139894 &= 0.04028943453320313 + 0.01748675 = \\ 0.05777618453320313 \end{aligned}$$

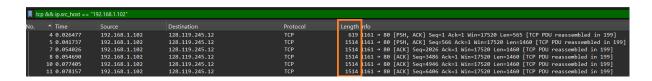
$$\begin{array}{l} \text{EstimatedRTT}_6 = (1-0.125) \times 0.05777618453320313 + 0.125 \times \\ 0.189645 = 0.05055316121415273 + 0.023705625 = \\ 0.07425878621415273 \end{array}$$

#### Vậy giá trị RTT của mỗi packet

STT	Estimated RTT (s)
1	0.027460
2	0.031205375
3	0.036562075625
4	0.04604506609375
5	0.05777618453320313

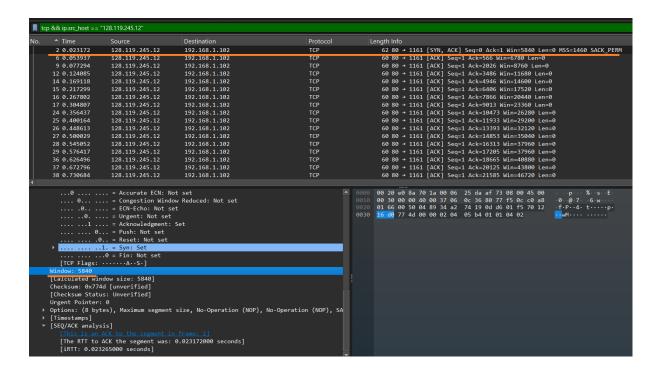
STT	Estimated RTT (s)
6	0.07425878621415273

### 14. Tìm độ dài của 6 segment đầu tiên?



STT	Length (byte)
1	619
2	1514
3	1514
4	1514
5	1514
6	1514

# 15. Tìm lượng buffer còn trống nhỏ nhất mà bên nhận thông báo cho bên gửi trong suốt file trace?



 Lượng buffer nhỏ nhất còn trống (trường window) mà server gửi cho client khi bắt tay có giá trị là 5840

Lab 3 11