Bài tập thực hành

Môn: TH Lập Trình Mạng

Chương 1: Cơ bản về mạng máy tính

1. MỤC TIÊU

Bài thực hành đưa ra với mục tiêu cho sinh viên ôn tập lại các kiến thức mạng máy tính cơ bản, đồng thời làm quen sử dụng các công cụ mạng cơ bản, sẽ trợ giúp cho các bài tập ở các chương sau.

2. YÊU CẦU

- Kiến thức cơ bản về mạng máy tính và các câu lệnh cơ bản Linux
- Máy tính cài đặt Hệ điều hành Linux (khuyến khích distro Ubuntu)

3. BÀI THỰC HÀNH

Bài 1. Cài đặt các máy tính kết nối với nhau trong mạng

Sinh viên tiến hành cài đặt 3 máy tính kết nối với nhau trong một mạng.

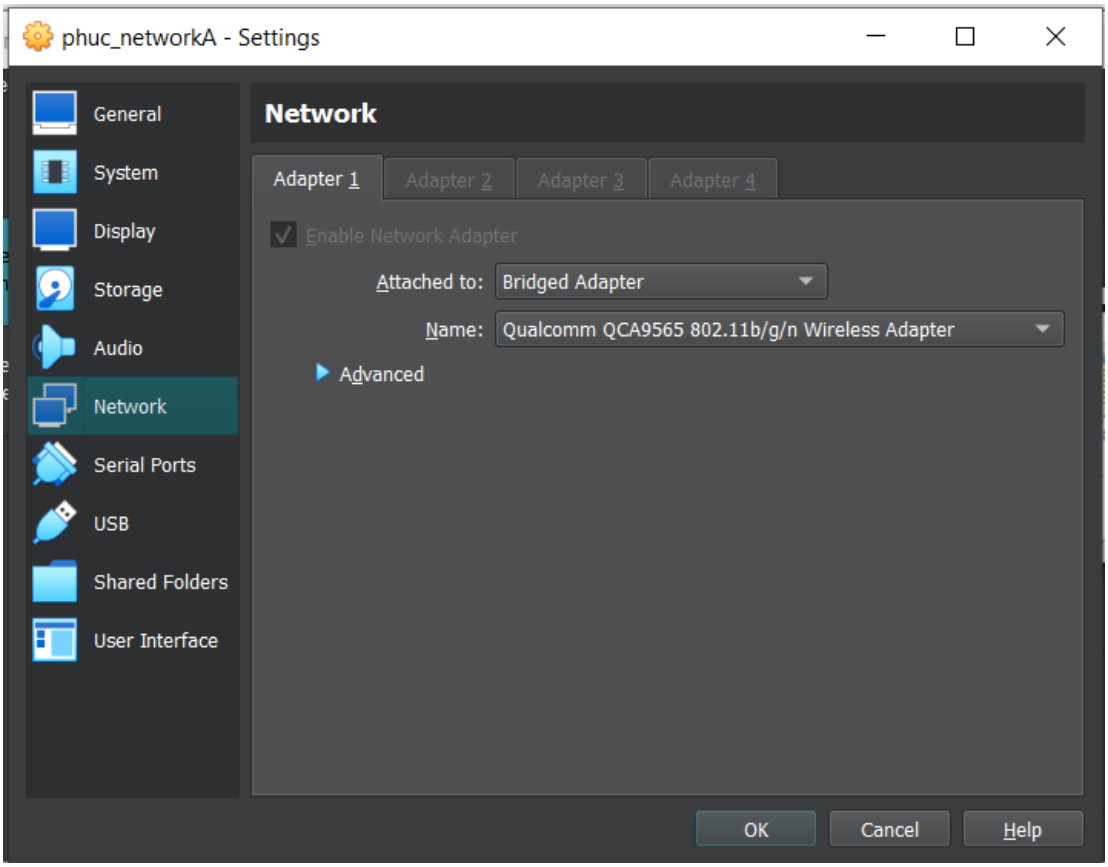
Sinh viên có thể sử dụng máy ảo (cài VirtualBox) hoặc máy thật. Hệ điều hành sử dụng là Ubuntu. Gọi 3 máy đó là A, B, và C.

Bài 2. Cấu hình mạng cho các máy

(gợi ý: sử dụng lệnh ifconfig)

Câu hỏi 1: Trình bày các bước (các lệnh) để thực hiện quá trình cấu hình mạng sao cho các máy nằm trong cùng một mạng đó. Em thực hiện lệnh nào để biết các máy đã được kết nối trong cùng một mạng?

- *Các bước (các lệnh) để thực hiện quá trình cấu hình mạng sao cho các máy nằm trong cùng một mạng:*
  + *Bước 1: Thiết lập network cho 2 máy ảo:*



+ *Bước 2: Xác định giao diện mạng:*

```
phucb@phucb-VirtualBox:~/Desktop$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::b53:8fab:2e4d:c88b  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:33:9f:9e  txqueuelen 1000  (Ethernet)
        RX packets 1290  bytes 1629950 (1.6 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 446  bytes 53409 (53.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 157  bytes 14222 (14.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 157  bytes 14222 (14.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

+ *Bước 3: Cấu hình địa chỉ IP tĩnh cho từng máy*

Máy A: 192.168.0.103

```
phuca@phuca-VirtualBox:~$ sudo ifconfig enp0s3 192.168.0.103 netmask 255.255.255.0 up
```

Máy B: 192.168.0.104

```
phucb@phucb-VirtualBox:~$ sudo ifconfig enp0s3 192.168.0.104 netmask 255.255.255.0 up
```

+ *Bước 4: Tắt tường lửa*

```
phucb@phucb-VirtualBox:~/Desktop$ sudo ufw status
[sudo] password for phucb:
Status: inactive
phucb@phucb-VirtualBox:~/Desktop$ sudo ufw disable
Firewall stopped and disabled on system startup
```

- *Lệnh để biết các máy đã được kết nối trong cùng một mạng*
  + *Máy A đến máy B:*

```
phuca@phuca-VirtualBox:~$ ping 192.168.0.104
PING 192.168.0.104 (192.168.0.104) 56(84) bytes of data.
64 bytes from 192.168.0.104: icmp_seq=1 ttl=64 time=0.353 ms
64 bytes from 192.168.0.104: icmp_seq=2 ttl=64 time=0.553 ms
64 bytes from 192.168.0.104: icmp_seq=3 ttl=64 time=0.431 ms
64 bytes from 192.168.0.104: icmp_seq=4 ttl=64 time=0.560 ms
^C
--- 192.168.0.104 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3064ms
rtt min/avg/max/mdev = 0.353/0.474/0.560/0.086 ms
```

  + *Máy B đến máy A:*

```
phucb@phucb-VirtualBox:~$ ping 192.168.0.103
PING 192.168.0.103 (192.168.0.103) 56(84) bytes of data.
64 bytes from 192.168.0.103: icmp_seq=1 ttl=64 time=0.263 ms
64 bytes from 192.168.0.103: icmp_seq=2 ttl=64 time=0.442 ms
64 bytes from 192.168.0.103: icmp_seq=3 ttl=64 time=0.559 ms
64 bytes from 192.168.0.103: icmp_seq=4 ttl=64 time=0.380 ms
64 bytes from 192.168.0.103: icmp_seq=5 ttl=64 time=0.774 ms
^C
--- 192.168.0.103 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4099ms
rtt min/avg/max/mdev = 0.263/0.483/0.774/0.173 ms
```

Câu hỏi 2: Em thực hiện lệnh nào để biết các máy đã được kết nối trong cùng một mạng?

Lệnh để biết các máy đã được kết nối trong cùng một mạng

+ *Máy A đến máy B:*

```
phuca@phuca-VirtualBox:~$ ping 192.168.0.104
PING 192.168.0.104 (192.168.0.104) 56(84) bytes of data.
64 bytes from 192.168.0.104: icmp_seq=1 ttl=64 time=0.353 ms
64 bytes from 192.168.0.104: icmp_seq=2 ttl=64 time=0.553 ms
64 bytes from 192.168.0.104: icmp_seq=3 ttl=64 time=0.431 ms
64 bytes from 192.168.0.104: icmp_seq=4 ttl=64 time=0.560 ms
^C
--- 192.168.0.104 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3064ms
rtt min/avg/max/mdev = 0.353/0.474/0.560/0.086 ms
```

+ *Máy B đến máy A:*

```
phucb@phucb-VirtualBox:~$ ping 192.168.0.103
PING 192.168.0.103 (192.168.0.103) 56(84) bytes of data.
64 bytes from 192.168.0.103: icmp_seq=1 ttl=64 time=0.263 ms
64 bytes from 192.168.0.103: icmp_seq=2 ttl=64 time=0.442 ms
64 bytes from 192.168.0.103: icmp_seq=3 ttl=64 time=0.559 ms
64 bytes from 192.168.0.103: icmp_seq=4 ttl=64 time=0.380 ms
64 bytes from 192.168.0.103: icmp_seq=5 ttl=64 time=0.774 ms
^C
--- 192.168.0.103 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4099ms
rtt min/avg/max/mdev = 0.263/0.483/0.774/0.173 ms
```

Bài 3. Cài đặt wireshark cho máy A. Thử kết nối giữa các máy. Quan sát màn hìnhwireshark của máy A khi được B và C thực hiện lệnh *ping*.

Cài đặt wireshark cho máy A

```
phuca@phuca-VirtualBox:~/Desktop$ sudo apt update
phuca@phuca-VirtualBox:~/Desktop$ sudo apt install wireshark
```

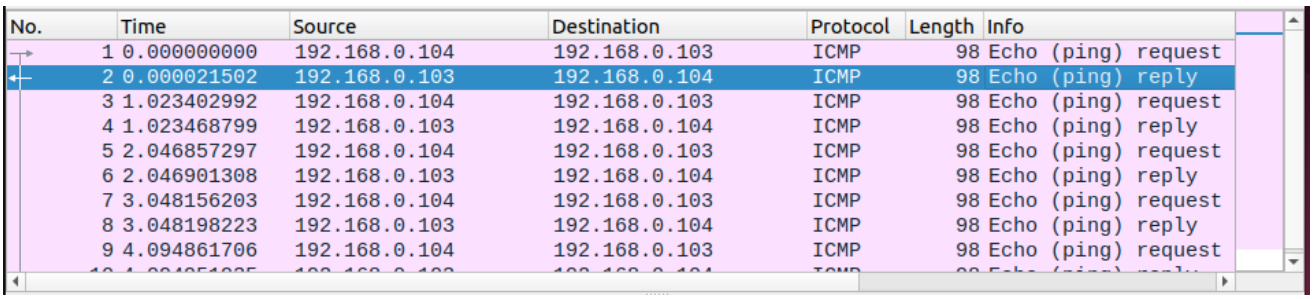
Thực hiện bắt các gói tin với wireshark:

- Ấn vào nút Capture
- Lựa chọn giao diện mạng (interface) phù hợp (chú ý: phải chọn đúng giao diện đang có kết nối giữa các máy)

Câu hỏi 3: Thực hiện lệnh ping giữa các máy. Những dòng thông tin nào trên cửa sổ wireshark cho thấy thông tin của lệnh *ping* đó?

Trên cửa sổ wireshark:

- *Những trường thông tin hiển thị thông tin của lệnh **ping** đã thực hiện là: Source, Destination, Protocol, Info.*
- *Trong trường hợp này, cứ 2 dòng sẽ hiển thị thông tin của 1 lệnh **ping**.*

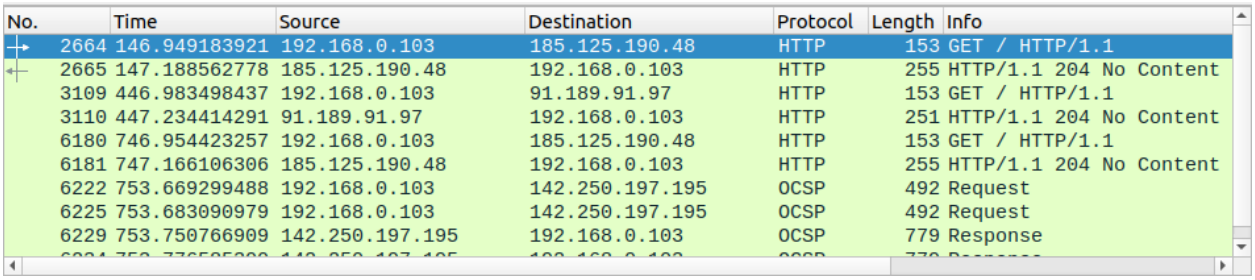


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.0.104	192.168.0.103	ICMP	98	Echo (ping) request
2	0.000021502	192.168.0.103	192.168.0.104	ICMP	98	Echo (ping) reply
3	1.023402992	192.168.0.104	192.168.0.103	ICMP	98	Echo (ping) request
4	1.023468799	192.168.0.103	192.168.0.104	ICMP	98	Echo (ping) reply
5	2.046857297	192.168.0.104	192.168.0.103	ICMP	98	Echo (ping) request
6	2.046901308	192.168.0.103	192.168.0.104	ICMP	98	Echo (ping) reply
7	3.048156203	192.168.0.104	192.168.0.103	ICMP	98	Echo (ping) request
8	3.048198223	192.168.0.103	192.168.0.104	ICMP	98	Echo (ping) reply
9	4.094861706	192.168.0.104	192.168.0.103	ICMP	98	Echo (ping) request

Câu hỏi 4: Dùng trình duyệt của máy đang chạy wireshark truy cập vào các trang web khác nhau. Những dòng thông tin nào trên cửa sổ wireshark cho thấy thông tin của quá trình duyệt web đó (các gói tin liên quan HTTP/HTTPS traffic).

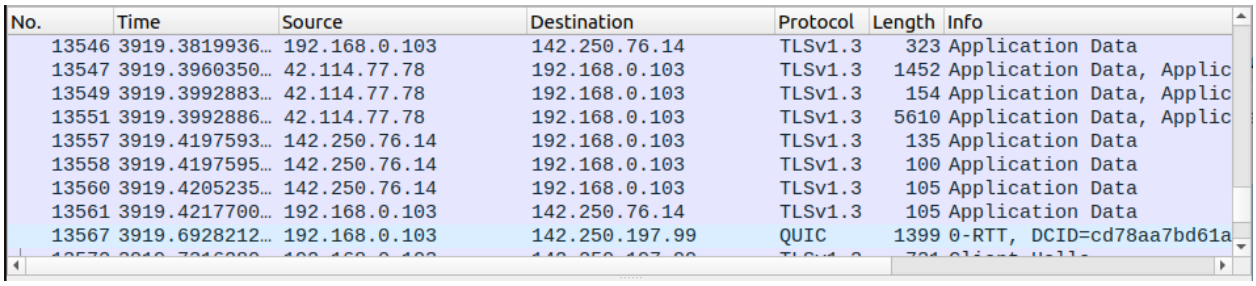
Những dòng thông tin trên cửa sổ wireshark cho thấy thông tin của quá trình duyệt web đó là:

- *Dòng có giao thức HTTP để yêu cầu (GET, POST), phản hồi (HTTP/1.1 204 No Content)*



No.	Time	Source	Destination	Protocol	Length	Info
2664	146.949183921	192.168.0.103	185.125.190.48	HTTP	153	GET / HTTP/1.1
2665	147.188562778	185.125.190.48	192.168.0.103	HTTP	255	HTTP/1.1 204 No Content
3109	446.983498437	192.168.0.103	91.189.91.97	HTTP	153	GET / HTTP/1.1
3110	447.234414291	91.189.91.97	192.168.0.103	HTTP	251	HTTP/1.1 204 No Content
6180	746.954423257	192.168.0.103	185.125.190.48	HTTP	153	GET / HTTP/1.1
6181	747.166106306	185.125.190.48	192.168.0.103	HTTP	255	HTTP/1.1 204 No Content
6222	753.669299488	192.168.0.103	142.250.197.195	OCSP	492	Request
6225	753.683090979	192.168.0.103	142.250.197.195	OCSP	492	Request
6229	753.750766909	142.250.197.195	192.168.0.103	OCSP	779	Response

- *Dòng có giao thức TLSv1.2 hoặc TLSv1.3 (quá trình bắt tay)*



No.	Time	Source	Destination	Protocol	Length	Info
13546	3919.3819936…	192.168.0.103	142.250.76.14	TLSv1.3	323	Application Data
13547	3919.3960350…	42.114.77.78	192.168.0.103	TLSv1.3	1452	Application Data, Applic
13549	3919.3992883…	42.114.77.78	192.168.0.103	TLSv1.3	154	Application Data, Applic
13551	3919.3992886…	42.114.77.78	192.168.0.103	TLSv1.3	5610	Application Data, Applic
13557	3919.4197593…	142.250.76.14	192.168.0.103	TLSv1.3	135	Application Data
13558	3919.4197595…	142.250.76.14	192.168.0.103	TLSv1.3	100	Application Data
13560	3919.4205235…	142.250.76.14	192.168.0.103	TLSv1.3	105	Application Data
13561	3919.4217700…	192.168.0.103	142.250.76.14	TLSv1.3	105	Application Data
13567	3919.6928212…	192.168.0.103	142.250.197.99	QUIC	1399	0-RTT, DCID=cd78aa7bd61a

Thực hiện phân tích các luồng dữ liệu TCP và UDP với wireshark:

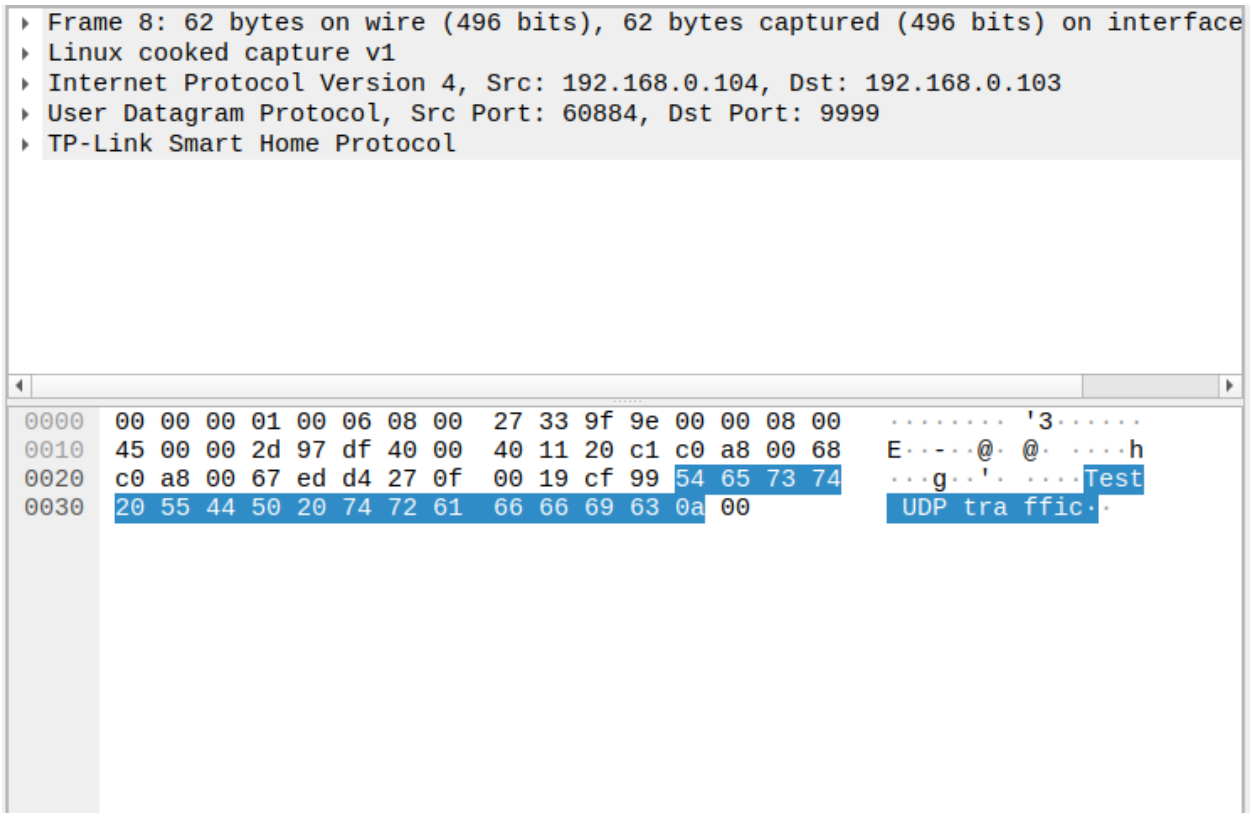
- Đối với UDP:
  - Ở máy server, sử dụng công cụ netcat để chạy lệnh: `nc -u -l 9999`

```
phuca@phuca-VirtualBox:~/Desktop$ nc -u -l 9999
Test UDP traffic
```

  - Ở máy client: `echo "Test UDP traffic" | nc -u 127.0.0.1 9999`

```
phucb@phucb-VirtualBox:~$ echo "Test UDP traffic" | nc -u 192.168.0.103 9999
```

- Sau khi chạy được một lúc, dừng wireshark lại và nhập “udp” vào thanhfilter để lọc ra các thông điệp UDP



Ở bảng giữa sẽ có các thông tin chi tiết về cấu trúc gói:

- Frame: Thông tin chung về gói (kích thước, thời gian, v.v.).
- Ethernet: Địa chỉ MAC nguồn và đích.
- IP: Địa chỉ IP nguồn và đích.
- UDP: Cổng nguồn và đích, độ dài và checksum.

Câu hỏi 5: Quan sát UDP packet trên wireshark, phân tích về tính đơn giản của UDP. Gợi ý: không có kết nối, do đó không có cờ (flags) để thiết lập hoặc hủy kết nối.

- *UDP là giao thức không có kết nối, nghĩa là khi gửi 1 gói tin, nó không cần thiết lập kết nối trước 2 bên như TCP (bắt tay 3 bước).*
- *UDP không có flags, vì thế không kiểm tra kết nối giữa 2 bên.*
- *UDP không có cơ chế xác nhận gói tin đã được nhận nên không thể kiểm tra và gửi lại khi mất gói tin. Bên cạnh đấy, các gói tin có thể bị tràn hoặc bỏ qua khi mạng quá tải.*

Do đó, UDP sẽ được ứng dụng khi cần truyền dữ liệu thời gian thực (như: trò chơi trực tuyến, video streaming,...)

- Đối với TCP:
    - Ở máy server, sử dụng công cụ netcat để chạy lệnh: `nc -l 8888`

```
phuca@phuca-VirtualBox:~/Desktop$ nc -l 8888
Test TCP traffic
```

    - Ở máy client: `echo "Test TCP traffic" | nc 127.0.0.1 8888`

```
phucb@phucb-VirtualBox:~$ echo "Test TCP traffic" | nc 192.168.0.103 8888
```

    - Sau khi chạy được một lúc, dừng wireshark lại và nhập "tcp" vào thanh filter để lọc ra các thông điệp TCP

```
▸ Frame 15: 85 bytes on wire (680 bits), 85 bytes captured (680 bits) on interface any, id 0
▸ Linux cooked capture v1
▸ Internet Protocol Version 4, Src: 192.168.0.104, Dst: 192.168.0.103
▸ Transmission Control Protocol, Src Port: 43894, Dst Port: 8888, Seq: 1, Ack: 1, Len: 17
▸ Data (17 bytes)
```

```
0000  00 00 00 01 00 06 08 00  27 33 9f 9e 00 00 08 00   ........ '3......
0010  45 00 00 45 2a d2 40 00  40 06 8d c1 c0 a8 00 68   E..E*.@. @......h
0020  c0 a8 00 67 ab 76 22 b8  5b 1c eb e2 17 c4 08 4f   ...g.v". [......O
0030  80 18 01 f6 9e a7 00 00  01 01 08 0a 0d 96 60 a6   ........ ......`.
0040  f7 3e 21 0d 54 65 73 74  20 54 43 50 20 74 72 61   .>!.Test  TCP tra
0050  66 66 69 63 0a                                    ffic.
```

Quan sát thông tin 1 TCP packet, chúng ta sẽ thấy các thông tin sau:

Frame: Chi tiết chung về gói tin.

Ethernet: Địa chỉ MAC.

IP: Thông tin địa chỉ IP.

Câu hỏi 6: Ấn vào trường thông tin TCP, quan sát sẽ thấy nhiều trường hơn so với UDP. Đó là những trường nào? Ý nghĩa của từng trường là gì?

Ý nghĩa các trường thông tin của TCP (nhiều hơn so với UDP):

- *Source Port: Là cổng của ứng dụng trên máy gửi, giúp xác định ứng dụng nào đang gửi dữ liệu.*

- *Destination Port: Là cổng của ứng dụng trên máy nhận, giúp chuyển dữ liệu đến đúng ứng dụng trên máy nhận.*
- *Sequence Number: Chứa số thứ tự của byte đầu trong dữ liệu được gửi, giúp theo dõi và đảm bảo dữ liệu được nhận theo đúng thứ tự.*
- *Acknowledgment Number: Chứa số thứ tự byte tiếp theo mà bên gửi mong muốn nhận được, giúp xác nhận đã nhận được dữ liệu.*
- *Data Offset: Xác định độ dài tiêu đề TCP, giúp xác định dữ liệu bắt đầu từ byte nào trong gói tin.*
- *Flags: Hiển thị trạng thái kết nối*
  + *SYN: Thiết lập kết nối trong quá trình bắt tay 3 bước.*
  + *ACK: Xác nhận đã nhận được dữ liệu.*
  + *FIN: Kết thúc kết nối.*
  + *RST: Đặt lại kết nối nếu có lỗi.*
  + *PSH: Yêu cầu gửi dự liệu ngay lập tức.*
  + *URG: Dữ liệu khẩn cấp cần được xử lý.*
- *Window Size: Cho biết lượng dữ liệu bên nhận có thể tiếp tục nhận tiếp mà không cần xác nhận lại.*
- *Checksum: Chứa giá trị kiểm tra lỗi của gói tin.*

Liên quan đến Bắt tay ba bước:

Gói SYN từ máy khách.

Gói SYN-ACK từ máy chủ.

Gói ACK từ máy khách để thiết lập kết nối.

Câu hỏi 7: Giải thích ý nghĩa của quy trình thiết lập kết nối bắt tay 3 bước này đối với TCP.

Ý nghĩa của quy trình thiết lập kết nối bắt tay 3 bước đối với TCP là: Đảm bảo kết nối được thiết lập đáng tin cậy, phía nguồn chỉ gửi dữ liệu khi nào phía đích đã sẵn sàng, giúp tránh mất mát và nhầm lẫn trong quá trình trao đổi dữ liệu.

Bài 4. Cài đặt Webserver apache2

Giao thức http: Cài đặt webserver apache2 cho máy A. Thử truy cập vào trang web của A từ 2 máy B và C (cổng 80). Quan sát màn hình wireshark của máy A.

Cài đặt webserver apache2:

```
phuca@phuca-VirtualBox:~/Desktop$ sudo apt update
phuca@phuca-VirtualBox:~/Desktop$ sudo apt install apache2
```

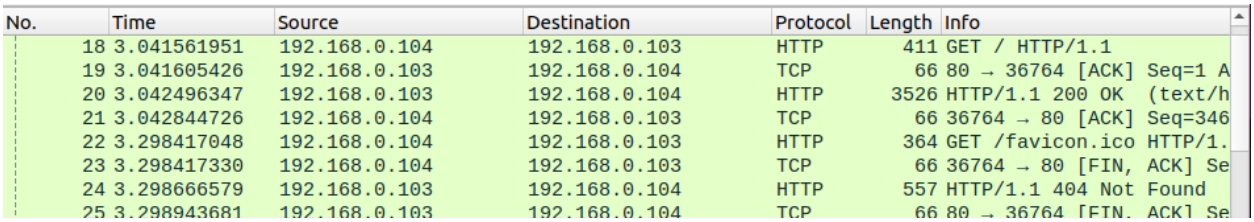
Truy cập trang web của A từ máy B:

Câu hỏi 8: Những dòng thông tin nào trên cửa sổ wireshark cho thấy thông tin của việc truy cập web đó?

Những dòng thông tin trên cửa sổ wireshark cho thấy thông tin việc truy cập web là:

- *Dòng có giao thức HTTP: Quá trình yêu cầu (GET) từ máy B và phản hồi (200 OK) từ máy A.*

- *Dòng có giao thức TCP: Quá trình bắt tay 3 bước giữa 2 máy A và B.*



No.	Time	Source	Destination	Protocol	Length	Info
18	3.041561951	192.168.0.104	192.168.0.103	HTTP	411	GET / HTTP/1.1
19	3.041605426	192.168.0.103	192.168.0.104	TCP	66	80 → 36764 [ACK] Seq=1 A
20	3.042496347	192.168.0.103	192.168.0.104	HTTP	3526	HTTP/1.1 200 OK (text/h
21	3.042844726	192.168.0.104	192.168.0.103	TCP	66	36764 → 80 [ACK] Seq=346
22	3.298417048	192.168.0.104	192.168.0.103	HTTP	364	GET /favicon.ico HTTP/1.
23	3.298417330	192.168.0.104	192.168.0.103	TCP	66	36764 → 80 [FIN, ACK] Se
24	3.298666579	192.168.0.103	192.168.0.104	HTTP	557	HTTP/1.1 404 Not Found
25	3.298943681	192.168.0.103	192.168.0.104	TCP	66	80 → 36764 [FIN, ACK] Se

(gợi ý: có thể tham khảo đường link sau để cài đặt apache2: https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-16- 04)