

Họ và tên: **Nguyễn Quang Thụy**

MSSV: **B1910306**

BÀI THỰC HÀNH SỐ 2

Bài tập 5

1) Quan sát mô hình mạng cần xây dựng và nhận diện các thiết bị, giao diện với các địa chỉ IP được gán trên các máy ảo.

2) Tạo thư mục BaiTap5 trong workspace của sinh viên. Thư mục sẽ này chứa các thư mục con và các file cấu hình (.startup, lab.conf) theo cấu trúc quy định của Kathará.

Trên máy thực, di chuyển đến thư mục BaiTap5 bằng lệnh

\$cd ~/BaiTapMMT/BaiTap5

```
→ ~ cd BaiTapMMT/BaiTH1/
→ BaiTH1 mkdir BaiTap5
→ BaiTH1 cd BaiTap5
→ BaiTap5 mkdir pc1 pc2 pc3 router1 router2
→ BaiTap5 touch lab.conf pc1.startup pc2.startup pc3.startup router1.startup router2.startup
→ BaiTap5 ls
lab.conf pc1 pc1.startup pc2 pc2.startup pc3 pc3.startup router1 router1.startup router2 router2.startup
→ BaiTap5
```

3) Trên file lab.conf, soạn thảo nội dung mô tả hình thái mạng theo thiết kế

pc1[0]=A

pc2[0]=B

pc3[0]=C

router1[0]=A

router1[1]=B

router2[0]=A

router2[1]=C

```
lab.conf ⚙
7 pc1[0]=A
6 pc2[0]=B
5 pc3[0]=C
4 router1[0]=A
3 router1[1]=B
2 router2[0]=A
1 router2[1]=C
8
```

4) Trên file pc1.startup chứa nội dung được miêu tả như sau

```
ifconfig eth0 10.0.0.101/24 up
route add -net 10.0.1.0/24 gw 10.0.0.1
route add -net 10.0.2.0/24 gw 10.0.0.2
```

```
pc1.startup ≡
3 ifconfig eth0 10.0.0.101/24 up
2 route add -net 10.0.1.0/24 gw 10.0.0.1
1 route add -net 10.0.2.0/24 gw 10.0.0.2
4 █
```

5) Thêm thông tin vạch đường đến nhánh LAN A, nhánh LAN C trên pc2.startup và thêm tin vạch đường đến nhánh LAN A, nhánh LAN B trên pc3.startup

```
pc2.startup ≡
3 ifconfig eth0 10.0.1.101/24 up
2 route add -net 10.0.0.0/24 gw 10.0.1.1
1 route add -net 10.0.2.0/24 gw 10.0.1.1
4 █
```

```
pc3.startup ≡
3 ifconfig eth0 10.0.2.101/24 up
2 route add -net 10.0.0.0/24 gw 10.0.2.1
1 route add -net 10.0.1.0/24 gw 10.0.2.1
4 █
```

6) Thêm thông tin vạch đường trên router1.startup và router2.startup bằng lệnh route add -net đã được hướng dẫn nhằm giúp cho router1 biết đường đi tới LAN C và router2 biết đường đi tới LAN B.

Nội dung file router1.startup có thể được trình bày như sau:

```
ifconfig eth0 10.0.0.1/24 up
ifconfig eth1 10.0.1.1/24 up
route add -net 10.0.2.0/24 gw 10.0.0.2
```

```
router1.startup ≡
3 ifconfig eth0 10.0.0.1/24 up
2 ifconfig eth1 10.0.1.1/24 up
1 route add -net 10.0.2.0/24 gw 10.0.0.2
4 █
```

```
router2.startup
3 ifconfig eth0 10.0.0.2/24 up
2 ifconfig eth1 10.0.2.1/24 up
1 route add -net 10.0.1.0/24 gw 10.0.0.1
4
```

7) Khởi động mạng ảo BaiTap5. Kiểm tra bảng vạch đường (bằng lệnh route) trên từng thiết bị mạng (máy ảo).

\$sudo kathara lstart

<pre>--- Startup Commands Log ++ ifconfig eth0 10.0.0.101/24 up ++ route add -net 10.0.1.0/24 gw 10.0.0.1 ++ route add -net 10.0.2.0/24 gw 10.0.0.2 --- End Startup Commands Log root@pc1:/#</pre>	<pre>--- Startup Commands Log ++ ifconfig eth0 10.0.1.101/24 up ++ route add -net 10.0.0.0/24 gw 10.0.1.1 ++ route add -net 10.0.2.0/24 gw 10.0.1.1 --- End Startup Commands Log root@pc2:/#</pre>	<pre>--- Startup Commands Log ++ ifconfig eth0 10.0.2.101/24 up ++ route add -net 10.0.0.0/24 gw 10.0.2.1 ++ route add -net 10.0.1.0/24 gw 10.0.2.1 --- End Startup Commands Log root@pc3:/#</pre>	<pre>--- Startup Commands Log ++ ifconfig eth0 10.0.0.1/24 up ++ ifconfig eth1 10.0.1.1/24 up ++ route add -net 10.0.2.0/24 gw 10.0.0.2 --- End Startup Commands Log root@router1:/#</pre>	<pre>--- Startup Commands Log ++ ifconfig eth0 10.0.0.2/24 up ++ ifconfig eth1 10.0.2.1/24 up ++ route add -net 10.0.1.0/24 gw 10.0.0.1 --- End Startup Commands Log root@router2:/#</pre>
--	--	--	--	--

8) Trường hợp bảng vạch đường của các thiết bị đều đúng, trên pc2 , router1 và router2 lần lượt thực hiện lệnh tcpdump với cú pháp như sau:

tcpdump -s 1536 -w /shared/BT5_pc2.pcap (trên máy ảo pc2)

```
root@pc2:/# tcpdump -s 1536 -w /shared/BT5_pc2.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 1536 bytes
```

tcpdump -s 1536 -w /shared/BT5_router1.pcap (trên máy ảo router1)

```
root@router1:/# tcpdump -s 1536 -w /shared/BT5_router1.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 1536 bytes
```

tcpdump -s 1536 -w /shared/BT5_router2.pcap (trên máy ảo router2)

```
root@router2:/# tcpdump -s 1536 -w /shared/BT5_router2.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 1536 bytes
```

9) Trên pc3 thực hiện gửi dữ liệu đến pc2 bằng lệnh:

ping 10.0.1.101

và chờ khoảng 10 giây, sau đó dừng lệnh ping lại.

```

root@pc3:/# ping 10.0.1.101
PING 10.0.1.101 (10.0.1.101) 56(84) bytes of data.
64 bytes from 10.0.1.101: icmp_seq=1 ttl=62 time=0.373 ms
64 bytes from 10.0.1.101: icmp_seq=2 ttl=62 time=0.170 ms
64 bytes from 10.0.1.101: icmp_seq=3 ttl=62 time=0.171 ms
64 bytes from 10.0.1.101: icmp_seq=4 ttl=62 time=0.171 ms
64 bytes from 10.0.1.101: icmp_seq=5 ttl=62 time=0.172 ms
64 bytes from 10.0.1.101: icmp_seq=6 ttl=62 time=0.180 ms
64 bytes from 10.0.1.101: icmp_seq=7 ttl=62 time=0.171 ms
64 bytes from 10.0.1.101: icmp_seq=8 ttl=62 time=0.169 ms
64 bytes from 10.0.1.101: icmp_seq=9 ttl=62 time=0.169 ms
64 bytes from 10.0.1.101: icmp_seq=10 ttl=62 time=0.169 ms
^C
--- 10.0.1.101 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9215ms
rtt min/avg/max/mdev = 0.169/0.191/0.373/0.062 ms
root@pc3:/# █

```

Dừng các lệnh tcpdump trên pc2 , router1 và router2 lại.

```

root@pc2:/# tcpdump -s 1536 -w /shared/BT5_pc2.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 1536 bytes
^C32 packets captured
32 packets received by filter
0 packets dropped by kernel
root@pc2:/# █

```

```

root@router1:/# tcpdump -s 1536 -w /shared/BT5_router1.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 1536 bytes
^C30 packets captured
30 packets received by filter
0 packets dropped by kernel
root@router1:/# █

```

```

root@router2:/# tcpdump -s 1536 -w /shared/BT5_router2.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 1536 bytes
^C28 packets captured
28 packets received by filter
0 packets dropped by kernel
root@router2:/# █

```

10) Trên máy thực, mở file BT5_pc2.pcap bằng Wireshark. Chọn khung vật lý (physical frame) số 3 và trả lời các câu hỏi sau đây:

- Toàn bộ khung số 3 có kích thước là bao nhiêu (Bytes)?

Trả lời: Toàn bộ khung số 3 có kích thước là 98 bytes.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fe80::b5:45ff:fe78:...	ff02::fb	MDNS	220	Standard query 0x0000 PTR _nfs._tcp.local
2	2.783815	fe80::b5:45ff:fe78:...	ff02::2	ICMPv6	70	Router Solicitation from 02:b5:45:78:db:1
3	21.215776	fe80::90b8:eeff:fe9...	ff02::2	ICMPv6	70	Router Solicitation from 02:b5:45:78:db:1
4	23.938432	fe80::90b8:eeff:fe9...	ff02::fb	MDNS	220	Standard query 0x0000 PTR _nfs._tcp.local
5	25.339613	8e:ea:43:97:cd:cc	Broadcast	ARP	42	Who has 10.0.1.101? Tell 10.0.1.1
6	25.339627	f2:11:2d:6d:b2:c0	8e:ea:43:97:cd:cc	ARP	42	10.0.1.101 is at f2:11:2d:6d:b2:c0
7	25.339640	10.0.2.101	10.0.1.101	ICMP	98	Echo (ping) request id=0x0035, seq=1/256
8	25.339670	10.0.1.101	10.0.2.101	ICMP	98	Echo (ping) reply id=0x0035, seq=1/256
9	26.367900	10.0.2.101	10.0.1.101	ICMP	98	Echo (ping) request id=0x0035, seq=2/512
10	26.367919	10.0.1.101	10.0.2.101	ICMP	98	Echo (ping) reply id=0x0035, seq=2/512
11	27.392115	10.0.2.101	10.0.1.101	ICMP	98	Echo (ping) request id=0x0035, seq=3/768
12	27.392135	10.0.1.101	10.0.2.101	ICMP	98	Echo (ping) reply id=0x0035, seq=3/768
13	28.416149	10.0.2.101	10.0.1.101	ICMP	98	Echo (ping) request id=0x0035, seq=4/1024
14	28.416170	10.0.1.101	10.0.2.101	ICMP	98	Echo (ping) reply id=0x0035, seq=4/1024
15	29.440069	10.0.2.101	10.0.1.101	ICMP	98	Echo (ping) request id=0x0035, seq=5/1280
16	29.440081	10.0.1.101	10.0.2.101	ICMP	98	Echo (ping) reply id=0x0035, seq=5/1280
17	30.432007	f2:11:2d:6d:b2:c0	8e:ea:43:97:cd:cc	ARP	42	Who has 10.0.1.1? Tell 10.0.1.101
18	30.432077	8e:ea:43:97:cd:cc	f2:11:2d:6d:b2:c0	ARP	42	10.0.1.1 is at 8e:ea:43:97:cd:cc
19	30.463956	10.0.2.101	10.0.1.101	ICMP	98	Echo (ping) request id=0x0035, seq=6/1536
20	30.463975	10.0.1.101	10.0.2.101	ICMP	98	Echo (ping) reply id=0x0035, seq=6/1536

Frame 7: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
Ethernet II, Src: 8e:ea:43:97:cd:cc (8e:ea:43:97:cd:cc), Dst: f2:11:2d:6d:b2:c0 (f2:11:2d:6d:b2:c0)
Internet Protocol Version 4, Src: 10.0.2.101, Dst: 10.0.1.101
Internet Control Message Protocol

0000	f2 11 2d 6d b2 c0 8e ea 43 97 cd cc 08 00 45 00	...	m	...	C	...	E
0010	00 54 da 17 40 00 3e 01 4a c8 0a 00 02 65 0a 00	.	T	.	@	.	>
0020	01 65 08 00 27 e6 00 35 00 01 af 74 5a 61 00 00	.	e	.	'	.	5
0030	00 00 06 3b 01 00 00 00 00 00 10 11 12 13 14 15
0040	16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25	!
0050	26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35	&	'	()	*	+
0060	36 37	.	.	/	0	1	2

- Chọn Header Internet Control Message Protocol trong khung và cho biết:

- + Gói tin này sử dụng giao thức gì? Giao thức này hoạt động trên tầng nào của mô hình OSI?

Trả lời: Gói tin này sử dụng giao thức ICMP. Giao thức này nằm ở tầng mạng (network) của mô hình OSI.

- + Nội dung thông điệp của giao thức này là gì? Thông điệp này có độ dài bao nhiêu (Bytes)?

Trả lời: Nội dung thông điệp của giao thức này là gửi yêu cầu (echo (ping) request). Thông điệp này có độ dài 48 bytes.

- Chọn Header Internet Protocol Version 4 và cho biết:

- + Địa chỉ IP của máy gửi dữ liệu là bao nhiêu? Địa chỉ IP này là của máy tính nào trong mạng?

Trả lời: Địa chỉ IP của máy gửi dữ liệu là 10.0.2.101. Địa chỉ IP này là của máy pc3.

- + Địa chỉ IP của máy nhận dữ liệu là bao nhiêu? Địa chỉ IP này là của máy tính nào trong mạng?

Trả lời: Địa chỉ IP của máy nhận dữ liệu là 10.0.1.101. Địa chỉ IP này của pc2.

- + Định danh (ID) của gói tin IP này là bao nhiêu (dạng hexadecimal).

Trả lời: Định danh của gói tin IP này là: (da 17).

- + Định danh của 1 gói tin có ý nghĩa gì trong thông điệp IP?

Trả lời: Định danh id giúp dễ nhận biết thông tin IP và để không bị mất trong quá trình truyền tải.

- + Độ dài phần Header của thông điệp IP là bao nhiêu? Phần Header bao gồm những trường (fields) nào? Mỗi trường có độ dài bao nhiêu?

Trả lời:

- Độ dài phần header là 20 bytes.
- Phần header gồm các trường có độ dài:
 - Version: 1 byte
 - Header Length: 1 byte
 - Difference Services Field: 1 byte
 - Total Length: 2 bytes
 - Identification: 2 bytes
 - Flags (3 bits) và Fragment offset (13 bits): 2 bytes
 - Time to live: 1 byte
 - Protocol: 1 byte
 - Header checksum: 2 bytes
 - Source: 4 bytes
 - Destination: 4 bytes

- + Trường Total Length có độ dài là bao nhiêu (Bytes). Hãy lý giải tại sao có độ dài như vậy?

Trả lời: Trường Total Length có độ dài là 84 bytes. Vì độ dài này bao gồm header và dữ liệu.

- Chọn Header Ethernet II và cho biết:

- + Địa chỉ MAC của máy gửi dữ liệu là bao nhiêu? Có phải là địa chỉ MAC của máy tính có địa chỉ IP (source) đã tìm được trong câu trên không? Nếu không, hãy lý giải và cho biết địa chỉ MAC này là của máy tính nào trong mạng?

Trả lời: Địa chỉ MAC của máy gửi dữ liệu là: 8e:ea:43:97:cd:cc. Đây không phải là địa chỉ MAC đã tìm. Mà là địa chỉ MAC của router đầu tiên mà trên đường từ máy nguồn pc3 đến máy đích cuối cùng là pc2. Địa chỉ MAC này là của router 2 trong mạng.

- + Địa chỉ MAC của máy nhận dữ liệu là bao nhiêu? Có phải là địa chỉ MAC của máy tính có địa chỉ IP (destination) đã tìm được trong câu trên không? Nếu không, hãy lý giải và cho biết địa chỉ MAC này là của máy tính nào trong mạng?

Trả lời: Địa chỉ MAC của máy nhận dữ liệu là: f2:11:2d:6d:b2:c0. Đây không phải là địa chỉ MAC đã tìm. Mà là địa chỉ MAC của router đầu tiên mà trên đường từ máy nguồn pc3 đến máy đích cuối cùng là pc2. Địa chỉ MAC này là của router 1 trong mạng.

- + Trường Type mang giá trị (hexadecimal) bằng bao nhiêu? Thông tin thể hiện là gì?

Trả lời: Trường Type mang giá trị (hexadecimal) bằng 0x0800. Trường "Type" trong các khung Ethernet II cho HĐH biết loại dữ liệu mà khung mang theo - 0x0800 có nghĩa là khung có gói tin IPv4.

- + Hãy chỉ ra trường Payload của khung Ethernet II? Trường Payload này có độ dài bằng bao nhiêu (Bytes)?

Trả lời: Trường Payload của khung Ethernet II có độ dài 14 bytes.

11) Hủy mạng ảo bằng lệnh wipe sau khi đã thực hiện xong Bài tập 5

\$sudo kathara wipe

```
→ BaiTap5 sudo kathara wipe
[sudo] password for thuyyy:
Are you sure to wipe Kathara? (y/n) y
→ BaiTap5
```

Bài tập 6

1) Quan sát mô hình mạng cần xây dựng và nhận diện các thiết bị, giao diện với các địa chỉ IP được gán trên các máy ảo.

2) Tạo thư mục BaiTap6 trong workspace của sinh viên. Thư mục sẽ này chứa các thư mục con và các file cấu hình (.startup, lab.conf) theo cấu trúc quy định của Kathara.

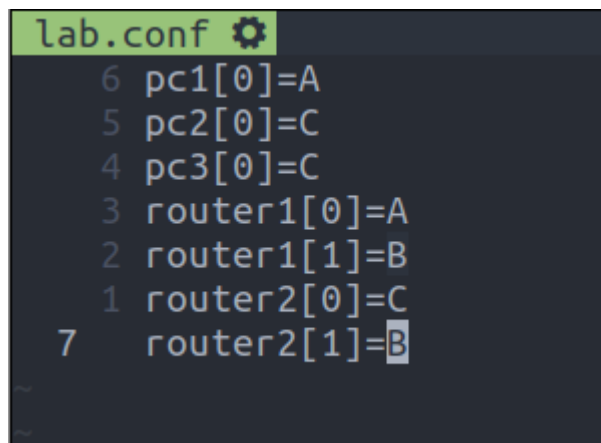
Trên máy thực, di chuyển đến thư mục BaiTap6 bằng lệnh:

\$cd ~/BaiTapMMT/BaiTap6

```
→ BaiTapMMT mkdir BaiTap6
→ BaiTapMMT cd BaiTap6
→ BaiTap6 mkdir pc1 pc2 pc3 router1 router2
→ BaiTap6 touch lab.conf pc1.startup pc2.startup pc3.startup router1.startup router2.startup
→ BaiTap6 ls
lab.conf      pc2          pc3.startup  router2
pc1           pc2.startup  router1      router2.startup
pc1.startup  pc3          router1.startup
→ BaiTap6
```

3) Trên file lab.conf, soạn thảo nội dung mô tả hình thái mạng theo thiết kế

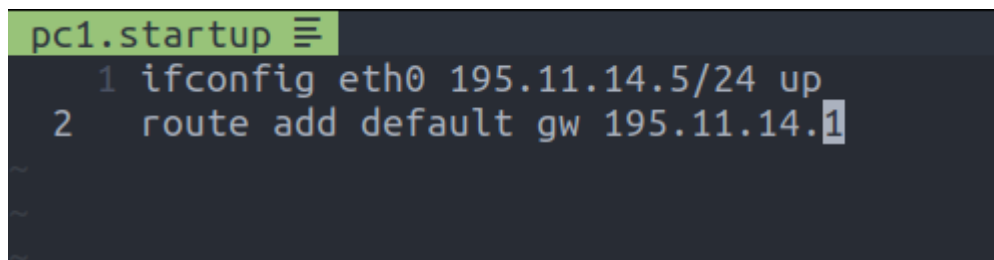
```
pc1[0]=A
pc2[0]=C
pc3[0]=C
router1[0]=A
router1[1]=B
router2[0]=C
router2[1]=B
```



```
lab.conf
6 pc1[0]=A
5 pc2[0]=C
4 pc3[0]=C
3 router1[0]=A
2 router1[1]=B
1 router2[0]=C
7 router2[1]=B
```

4) Trên file pc1.startup để vạch đường mặc nhiên thì sẽ chứa nội dung được miêu tả như sau

```
ifconfig eth0 195.11.14.5/24 up
route add default gw 195.11.14.1
```



```
pc1.startup
1 ifconfig eth0 195.11.14.5/24 up
2 route add default gw 195.11.14.1
```

Trên file pc2.startup để vạch đường mặc nhiên sẽ chứa nội dung được miêu tả như sau

```
ifconfig eth0 200.1.1.7/24 up
route add default gw 200.1.1.1
```



```
pc2.startup ≡  
1 ifconfig eth0 200.1.1.7/24 up  
2 route add default gw 200.1.1.1
```

Thực hiện tương tự trên pc3.startup

```
pc3.startup ≡  
1 ifconfig eth0 200.1.1.3/24 up  
2 route add default gw 200.1.1.1
```

5) Trên file router1.startup và router2.startup cũng thực hiện thêm thông tin vạch đường tĩnh sao cho router1 biết hướng đi tới LAN C và router2 biết hướng đi tới LAN A.

Nội dung file router1.startup có thể được trình bày như sau:

```
ifconfig eth0 195.11.14.1/24 up  
ifconfig eth1 100.0.0.9/30 up  
route add -net 200.1.1.0/24 gw 100.0.0.10
```

```
router1.startup ≡  
2 ifconfig eth0 195.11.14.1/24 up  
1 ifconfig eth1 100.0.0.9/30 up  
3 route add -net 200.1.1.0/24 gw 100.0.0.10
```

Nội dung file router2.startup có thể được trình bày như sau:

```
router2.startup ≡  
2 ifconfig eth0 200.1.1.1/24 up  
1 ifconfig eth1 100.0.0.10/30 up  
3 route add -net 195.11.14.0/24 gw 100.0.0.9
```

2.3.2.1. Giao thức ARP giữa 2 thiết bị trong cùng nhánh mạng LAN C

6) Khởi động mạng ảo BaiTap6.

\$sudo kathara lstart

Kiểm tra bảng vạch đường (lệnh route) và địa chỉ IP của các giao diện mạng (lệnh ifconfig) trên từng máy ảo để đảm bảo tính đúng đắn của mô hình mạng Bài Tập 6.

```
root@pc1:/# route
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
default          195.11.14.1     0.0.0.0         UG      0      0      0 eth0
195.11.14.0      0.0.0.0         255.255.255.0   U        0      0      0 eth0
root@pc1:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 195.11.14.5  netmask 255.255.255.0  broadcast 195.11.14.255
    ether 0a:f6:c2:5a:44:b5  txqueuelen 1000  (Ethernet)
    RX packets 25  bytes 3742 (3.6 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 0  bytes 0 (0.0 B)
    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
    loop txqueuelen 1000  (Local Loopback)
    RX packets 4  bytes 336 (336.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 4  bytes 336 (336.0 B)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@pc1:/#
```

```
root@pc2:/# route
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
default          200.1.1.1       0.0.0.0         UG      0      0      0 eth0
200.1.1.0        0.0.0.0         255.255.255.0   U        0      0      0 eth0
root@pc2:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 200.1.1.7  netmask 255.255.255.0  broadcast 200.1.1.255
    ether fa:a4:c8:0c:0e:03  txqueuelen 1000  (Ethernet)
    RX packets 30  bytes 4542 (4.4 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 0  bytes 0 (0.0 B)
    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
    loop txqueuelen 1000  (Local Loopback)
    RX packets 4  bytes 328 (328.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 4  bytes 328 (328.0 B)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@pc2:/#
```

```

root@pc3:/# route
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
default          200.1.1.1       0.0.0.0          UG      0      0      0 eth0
200.1.1.0        0.0.0.0         255.255.255.0    U       0      0      0 eth0
root@pc3:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 200.1.1.3 netmask 255.255.255.0 broadcast 200.1.1.255
    ether 9e:28:5a:f2:80:12 txqueuelen 1000 (Ethernet)
    RX packets 32 bytes 4832 (4.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    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
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4 bytes 328 (328.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 328 (328.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@pc3:/# █

```

```

root@router1:/# route
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
100.0.0.8        0.0.0.0         255.255.255.252  U       0      0      0 eth1
195.11.14.0      0.0.0.0         255.255.255.0    U       0      0      0 eth0
200.1.1.0        100.0.0.10      255.255.255.0    UG      0      0      0 eth1
root@router1:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 195.11.14.1 netmask 255.255.255.0 broadcast 195.11.14.255
    ether 4a:25:b2:a1:9f:d3 txqueuelen 1000 (Ethernet)
    RX packets 34 bytes 5122 (5.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 100.0.0.9 netmask 255.255.255.252 broadcast 100.0.0.11
    ether 0e:d8:83:d5:44:0c txqueuelen 1000 (Ethernet)
    RX packets 34 bytes 5122 (5.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    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

```

```

root@router2:/# route
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
100.0.0.8        0.0.0.0          255.255.255.252 U        0      0      0 eth1
195.11.14.0      100.0.0.9        255.255.255.0   UG       0      0      0 eth1
200.1.1.0        0.0.0.0          255.255.255.0   U        0      0      0 eth0
root@router2:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 200.1.1.1 netmask 255.255.255.0 broadcast 200.1.1.255
    ether e6:b9:29:f4:2d:60 txqueuelen 1000 (Ethernet)
    RX packets 34 bytes 5122 (5.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 100.0.0.10 netmask 255.255.255.252 broadcast 100.0.0.11
    ether 7a:17:95:de:e3:3d txqueuelen 1000 (Ethernet)
    RX packets 34 bytes 5122 (5.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    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

```

7) Trên máy ảo pc3 , pc2 và router2 , lần lượt dùng lệnh arp. Nhận xét kết quả

Trả lời: Không hiển thị gì cả.

8) Lần lượt thực hiện lệnh tcpdump với cú pháp như sau:

tcpdump -s 1536 -w /shared/BT6_pc2_A.pcap (trên máy ảo pc2)

```

root@pc2:/# tcpdump -s 1536 -w /shared/BT6_pc2_A.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 1536 bytes

```

tcpdump -s 1536 -w /shared/BT6_router1_A.pcap (trên máy ảo router1)

```

root@router1:/# tcpdump -s 1536 -w /shared/BT6_router1_A.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 1536 bytes

```

tcpdump -s 1536 -w /shared/BT6_router2_A.pcap (trên máy ảo router2)

```

root@router2:/# tcpdump -s 1536 -w /shared/BT6_router2_A.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 1536 bytes

```

9) Trên pc3 thực hiện gửi dữ liệu đến pc2 bằng lệnh:

ping 200.1.1.7

và chờ khoảng 10 giây, sau đó dừng lệnh ping trên pc3 lại.

```
root@pc3:/# ping 200.1.1.7
PING 200.1.1.7 (200.1.1.7) 56(84) bytes of data.
64 bytes from 200.1.1.7: icmp_seq=1 ttl=64 time=0.211 ms
64 bytes from 200.1.1.7: icmp_seq=2 ttl=64 time=0.116 ms
64 bytes from 200.1.1.7: icmp_seq=3 ttl=64 time=0.111 ms
64 bytes from 200.1.1.7: icmp_seq=4 ttl=64 time=0.111 ms
64 bytes from 200.1.1.7: icmp_seq=5 ttl=64 time=0.111 ms
64 bytes from 200.1.1.7: icmp_seq=6 ttl=64 time=0.110 ms
64 bytes from 200.1.1.7: icmp_seq=7 ttl=64 time=0.110 ms
64 bytes from 200.1.1.7: icmp_seq=8 ttl=64 time=0.046 ms
64 bytes from 200.1.1.7: icmp_seq=9 ttl=64 time=0.102 ms
64 bytes from 200.1.1.7: icmp_seq=10 ttl=64 time=0.125 ms
^C
--- 200.1.1.7 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9193ms
rtt min/avg/max/mdev = 0.046/0.115/0.211/0.038 ms
root@pc3:/#
```

Dừng các lệnh tcpdump trên pc2 , router1 và router2 lại.

```
root@pc2:/# tcpdump -s 1536 -w /shared/BT6_pc2_A.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 1536 bytes
^C32 packets captured
33 packets received by filter
0 packets dropped by kernel
root@pc2:/#
```

```
root@router1:/# tcpdump -s 1536 -w /shared/BT6_router1_A.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 1536 bytes
^C6 packets captured
6 packets received by filter
0 packets dropped by kernel
root@router1:/#
```

```
root@router2:/# tcpdump -s 1536 -w /shared/BT6_router2_A.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 1536 bytes
^C29 packets captured
29 packets received by filter
0 packets dropped by kernel
root@router2:/#
```

10) Trên pc3 thực hiện lại lệnh arp và nhận xét kết quả hiển thị.

Lưu ý sự thay đổi so với kết quả ở 7). Lý giải cho sự thay đổi này.

Ghi nhận kết quả hiển thị để so sánh với 10) ở mục 2.3.2.2.

```

root@pc3:/# arp
Address          Hwtype  Hwaddress      Flags Mask      Iface
200.1.1.7        ether    1a:e7:df:52:12:a9  C              eth0
root@pc3:/#

```

Trả lời:

- + Khi thực hiện ping từ pc3 đến pc2 thì sử dụng lệnh arp trên pc3 kết quả hiển thị một địa chỉ IP của pc2 thành địa chỉ MAC.
- + Lý do thay đổi: Lệnh arp dùng phân giải địa chỉ IP thành địa chỉ MAC của nó. Ban đầu khi thực hiện arp do chưa nhận được IP nào từ các máy khác nên nó không hiển thị gì, đến khi nhận được địa chỉ IP pc3 khi ping đến pc2 thì nó bắt đầu phân giải IP của pc2.

11) Trên pc2, thực hiện lại lệnh arp và nhận xét kết quả hiển thị.

Lưu ý sự thay đổi so với kết quả ở bước số 7). Lý giải cho sự thay đổi này.

```

root@pc2:/# arp
Address          Hwtype  Hwaddress      Flags Mask      Iface
200.1.1.3        ether    ca:fe:8f:69:76:0a  C              eth0
root@pc2:/#

```

Trả lời: Giống như pc3, pc2 cũng nhận được địa chỉ IP của pc3 khi thực hiện ping đến pc2, do 2 máy được liên thông với nhau nên arp trên pc2 cũng phân giải IP pc3 nhận được và thực hiện phân giải địa chỉ MAC của nó.

12) Trên router2, thực hiện lại lệnh arp và nhận xét kết quả hiển thị. Ghi nhận kết quả hiển thị để so sánh với bước 12) ở mục 2.3.2.2.

```

root@router2:/# arp
root@router2:/#

```

Trả lời: Do router 2 không nhận được địa chỉ nào liên thông đến nên kết quả không hiển thị gì khi thực hiện lệnh arp.

13) Trên máy thực, dùng Wireshark mở file BT6_router2_A.pcap, chọn khung vật lý số thứ tự 1. Trả lời các câu hỏi sau đây:

- Toàn bộ khung số thứ tự 1 có kích thước là bao nhiêu (Bytes)?

Trả lời: Toàn bộ khung 1 có kích thước là 124 bytes.

- Chọn Header Address Resolution Protocol và cho biết:
 - Trường Opcode có giá trị (hexadecimal) là bao nhiêu? Giá trị của trường này thể hiện thông tin gì? Trường Opcode này còn có thể có giá trị (hexadecimal) là bao nhiêu nữa và thể hiện thông tin gì?

Trả lời: Trường Opcode có giá trị là 00 01. Giá trị của trường này thể hiện thông tin yêu cầu (request) các gói tin từ máy nguồn. Trường Opcode còn có thể có giá trị là 00 02, thể hiện thông tin phản hồi (reply) từ máy nguồn.

- Địa chỉ IP và địa chỉ MAC của máy gửi dữ liệu? Đây là địa chỉ IP và MAC của máy tính nào trong mạng?

Trả lời:

- Địa chỉ IP và địa chỉ MAC của máy gửi dữ liệu:

Sender MAC address: ca:fe:8f:69:76:0a (ca:fe:8f:69:76:0a)
Sender IP address: 200.1.1.3

- Đây là địa chỉ IP và MAC của máy pc3 trong mạng.

- Địa chỉ IP và địa chỉ MAC của máy nhận dữ liệu? Đây là địa chỉ IP và MAC của máy tính nào trong mạng? Nhận xét về cặp địa chỉ IP và MAC của máy nhận dữ liệu.

Trả lời:

- Địa chỉ IP và địa chỉ MAC của máy nhận dữ liệu:

Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
Target IP address: 200.1.1.7

- Đây là địa chỉ IP và MAC bị hỏng.

➤ Chọn Header Ethernet II và cho biết:

- Địa chỉ MAC của máy gửi dữ liệu là bao nhiêu? Địa chỉ MAC này là của máy tính nào trong mạng?

Trả lời:

- Địa chỉ MAC của máy gửi dữ liệu:

▶ Source: ca:fe:8f:69:76:0a (ca:fe:8f:69:76:0a)

- Địa chỉ MAC này là của máy tính pc2 trong mạng.

- Địa chỉ MAC của máy nhận dữ liệu là bao nhiêu? Địa chỉ MAC này là của máy tính nào trong mạng? Nhận xét về địa chỉ MAC này và địa chỉ MAC của máy nhận dữ liệu đã quan sát được ở phần Header Address Resolution Protocol

Trả lời:

- Địa chỉ MAC của máy nhận dữ liệu:

▶ Destination: Broadcast (ff:ff:ff:ff:ff:ff)

- Đây là địa chỉ MAC máy tính đang dùng (máy thực)

- Địa chỉ MAC này và địa chỉ MAC của máy nhận dữ liệu khác nhau.
- Trường Type mang giá trị (hexadecimal) bằng bao nhiêu? Thông tin thể hiện là gì?

Trả lời: Trường Type mang giá trị 0x0806. Thể hiện loại giao thức ARP.

14) Hủy mạng ảo bằng lệnh wipe sau khi đã thực hiện xong phần 2.3.2.1

\$sudo kathara wipe

```
→ BaiTap6 sudo kathara wipe
[sudo] password for thuyyy:
Are you sure to wipe Kathara? (y/n) y
→ BaiTap6
```

2.3.2.2. Giao thức ARP giữa 2 thiết bị khác nhánh mạng LAN

7) Mở lại mạng ảo bằng lệnh lstart.

\$sudo kathara lstart

Trên máy ảo pc1 và router1, lần lượt dùng lệnh arp, nhận xét kết quả hiển thị.

<pre>root@router1:/# arp root@router1:/#</pre>	<pre>root@pc1:/# arp root@pc1:/#</pre>
--	--

Trả lời: Không hiển thị gì cả.

8) Lần lượt thực hiện lệnh tcpdump với cú pháp như sau:

tcpdump -s 1536 -w /shared/BT6_pc2_B.pcap (trên máy ảo pc2)

```
root@pc1:/# tcpdump -s 1536 -w /shared/BT6_pc1_B.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 1536 bytes
```

tcpdump -s 1536 -w /shared/BT6_router1_B.pcap (trên máy ảo router1)

```
root@router1:/# tcpdump -s 1536 -w /shared/BT6_router1_B.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 1536 bytes
```

tcpdump -s 1536 -w /shared/BT6_router2_B.pcap (trên máy ảo router2)


```
root@router2:/# tcpdump -s 1536 -w /shared/BT6_router2_B.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 1536 bytes
```

9) Trên pc3 thực hiện gửi dữ liệu đến pc1 bằng lệnh:

ping 195.11.14.5

và chờ khoảng 10 giây, sau đó dừng lệnh ping trên pc3 lại.

```
root@pc3:/# ping 195.11.14.5
PING 195.11.14.5 (195.11.14.5) 56(84) bytes of data.
64 bytes from 195.11.14.5: icmp_seq=1 ttl=62 time=0.433 ms
64 bytes from 195.11.14.5: icmp_seq=2 ttl=62 time=0.195 ms
64 bytes from 195.11.14.5: icmp_seq=3 ttl=62 time=0.170 ms
64 bytes from 195.11.14.5: icmp_seq=4 ttl=62 time=0.076 ms
64 bytes from 195.11.14.5: icmp_seq=5 ttl=62 time=0.174 ms
64 bytes from 195.11.14.5: icmp_seq=6 ttl=62 time=0.173 ms
64 bytes from 195.11.14.5: icmp_seq=7 ttl=62 time=0.170 ms
64 bytes from 195.11.14.5: icmp_seq=8 ttl=62 time=0.173 ms
64 bytes from 195.11.14.5: icmp_seq=9 ttl=62 time=0.170 ms
64 bytes from 195.11.14.5: icmp_seq=10 ttl=62 time=0.177 ms
^C
--- 195.11.14.5 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9197ms
rtt min/avg/max/mdev = 0.076/0.191/0.433/0.086 ms
root@pc3:/#
```

Dừng các lệnh tcpdump trên pc1 , router1 và router2 .

```
root@pc1:/# tcpdump -s 1536 -w /shared/BT6_pc1_B.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 1536 bytes
^C30 packets captured
30 packets received by filter
0 packets dropped by kernel
root@pc1:/#
```

```
root@router1:/# tcpdump -s 1536 -w /shared/BT6_router1_B.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 1536 bytes
^C30 packets captured
30 packets received by filter
0 packets dropped by kernel
root@router1:/#
```

```
root@router2:/# tcpdump -s 1536 -w /shared/BT6_router2_B.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 1536 bytes
^C28 packets captured
28 packets received by filter
0 packets dropped by kernel
root@router2:/#
```

10) Trên pc3 thực hiện lại lệnh arp và nhận xét kết quả hiển thị. Lưu ý: so sánh với 10) ở phần 2.3.2.1.

```
root@pc3:/# arp
Address          HWtype  HWaddress      Flags Mask    Iface
200.1.1.1        ether    86:bc:cd:6a:b1:e5  C             eth0
root@pc3:/#
```

Trả lời: Nhận thấy khi ping trong cùng nhánh mạng thì có thể lấy địa IP chuyển thành MAC thông qua arp. Trong trường hợp này ping khác nhánh mạng (pc3 -> pc1) nên địa chỉ MAC chính là địa chỉ MAC router đầu tiên đi đến từ máy nguồn đến máy đích.

11) Trên router2, thực hiện lại lệnh arp và nhận xét kết quả hiển thị. Lưu ý: so sánh với 12) ở phần 2.3.2.1.

```
root@router2:/# arp
Address          HWtype  HWaddress      Flags Mask    Iface
100.0.0.9        ether    fa:5e:dd:38:5a:73  C             eth1
200.1.1.3        ether    c2:2e:df:8c:36:f0  C             eth0
root@router2:/#
```

Trả lời: Do ở câu 12 phần 2.3.2.1 thực hiện trong cùng nhánh mạng nên không cần qua router 2 nên không hiển thị gì. Còn ở đây ping từ LAN C đến LAN A cần thông qua router 2 nên router 2 sẽ nhận được địa chỉ MAC của máy nguồn đến máy đích.

12) Trên router1, thực hiện lại lệnh arp và nhận xét kết quả hiển thị.

```
root@router1:/# arp
Address          HWtype  HWaddress      Flags Mask    Iface
195.11.14.5      ether    26:56:94:e2:e8:af  C             eth0
100.0.0.10       ether    fe:dd:61:09:7c:32  C             eth1
root@router1:/#
```

Trả lời: Khi ping giữa 2 máy pc 3 và pc1 sẽ thông qua 2 router 1 và router 2. Vì vậy khi pc1 nhận được yêu cầu từ pc3 thì pc1 sẽ gửi phản hồi lại thông qua địa chỉ MAC router 1.

13) Trên pc1, thực hiện lại lệnh arp và nhận xét kết quả hiển thị. Lưu ý: so sánh với thông tin hiển thị của máy pc1 tại 7) ở phần 2.3.2.2.

```
root@pc1:/# arp
Address          HWtype  HWaddress      Flags Mask    Iface
195.11.14.1      ether    e2:16:95:3d:d9:c8  C             eth0
root@pc1:/#
```

Trả lời: Ở câu 7 khi pc3 chưa ping đến pc1 nên không nhận được phản hồi từ pc1 nên không hiển thị gì. Vì vậy khi pc1 nhận được yêu cầu từ pc3 thì pc1 sẽ gửi phản hồi lại thông qua địa chỉ MAC router 1 và pc1 cũng nhận được địa chỉ MAC từ router 1.

14) Trên máy thực, dùng Wireshark mở file BT6_Router1_B.pcap , chọn khung vật lý số thứ tự 1.

- Toàn bộ khung số thứ tự 1 có kích thước là bao nhiêu (Bytes)?

Trả lời: Toàn bộ khung số 1 có kích thước là 42 bytes.

- Chọn Header Address Resolution Protocol và cho biết:

- Trường Opcode có giá trị (hexadecimal) là bao nhiêu? Giá trị của trường này thể hiện thông tin gì? Trường Opcode này còn có thể có giá trị (hexadecimal) là bao nhiêu nữa và thể hiện thông tin gì?

Trả lời: Trường Opcode có giá trị là 00 01. Thể hiện thông tin yêu cầu (request) các gói tin từ máy nguồn. Trường Opcode này còn có thể có giá trị là 00 02. Thể hiện thông tin phản hồi (reply) từ máy nguồn.

- Địa chỉ IP và địa chỉ MAC của máy gửi dữ liệu? Đây là địa chỉ IP và MAC của máy tính nào trong mạng?

Trả lời:

- Địa chỉ MAC của máy gửi dữ liệu:

```
Sender MAC address: e2:16:95:3d:d9:c8 (e2:16:95:3d:d9:c8)  
Sender IP address: 195.11.14.1
```

- Địa chỉ MAC này là của máy tính pc1 trong mạng.

- Địa chỉ IP và địa chỉ MAC của máy nhận dữ liệu? Đây là địa chỉ IP và MAC của máy tính nào trong mạng? Nhận xét về cặp địa chỉ IP và MAC của máy nhận dữ liệu.

Trả lời:

- Địa chỉ IP và địa chỉ MAC của máy nhận dữ liệu:

```
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)  
Target IP address: 195.11.14.5
```

- Đây là địa chỉ IP và MAC của máy router1 trong mạng.

- Chọn Header Ethernet II và cho biết:

- Địa chỉ MAC của máy gửi dữ liệu là bao nhiêu? Địa chỉ MAC này là của máy tính nào trong mạng?

Trả lời:

- Địa chỉ MAC của máy gửi dữ liệu:

```
Source: e2:16:95:3d:d9:c8 (e2:16:95:3d:d9:c8)
```

- Địa chỉ MAC này là của máy tính pc3 trong mạng.

- Địa chỉ MAC của máy nhận dữ liệu là bao nhiêu? Địa chỉ MAC này là của máy tính nào trong mạng? Nhận xét về địa chỉ MAC này và địa chỉ

MAC của máy nhận dữ liệu đã quan sát được ở phần Header Address Resolution Protocol

Trả lời:

- Địa chỉ MAC của máy nhận dữ liệu:

► Destination: Broadcast (ff:ff:ff:ff:ff:ff)

- Đây là địa chỉ MAC máy tính đang dùng (máy thực)
- Địa chỉ MAC này và địa chỉ MAC của máy nhận dữ liệu khác nhau.
- Trường Type mang giá trị (hexadecimal) bằng bao nhiêu? Thông tin thể hiện là gì?

Trả lời: Trường Type mang giá trị 0x0806. Thể hiện loại giao thức ARP.

Vẽ sơ đồ tuần tự (sequence diagram) thể hiện vai trò của giao thức ARP trong việc truyền tải dữ liệu từ pc3 đến pc1 bằng lệnh ping

15) Hủy mạng ảo bằng lệnh `lwiipe` sau khi đã thực hiện xong 2.3.2.2

```
→ BaiTap6 sudo kathara wipe
Are you sure to wipe Kathara? (y/n) y
→ BaiTap6
```

2.3.2.3. Giao thức ARP với địa chỉ không nằm trong bảng vạch đường

- Trên pc1 , gửi dữ liệu đến Google DNS (địa chỉ ngoài mạng ảo) bằng lệnh `ping 8.8.8.8`

```
root@pc1:/# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
From 195.11.14.1 icmp_seq=1 Destination Net Unreachable
From 195.11.14.1 icmp_seq=2 Destination Net Unreachable
From 195.11.14.1 icmp_seq=3 Destination Net Unreachable
From 195.11.14.1 icmp_seq=4 Destination Net Unreachable
From 195.11.14.1 icmp_seq=5 Destination Net Unreachable
From 195.11.14.1 icmp_seq=6 Destination Net Unreachable
From 195.11.14.1 icmp_seq=7 Destination Net Unreachable
From 195.11.14.1 icmp_seq=8 Destination Net Unreachable
^C
--- 8.8.8.8 ping statistics ---
8 packets transmitted, 0 received, +8 errors, 100% packet loss, time 7144ms
root@pc1:/#
```

- Ghi nhận kết quả hiển thị khi dùng lệnh arp trên pc1 và router1. Nhận xét kết quả này.

```

root@pc1:/# arp
Address          Hwtype  Hwaddress      Flags Mask      Iface
195.11.14.1      ether   62:f2:00:f4:7d:78  C          eth0
root@pc1:/# █

```

```

root@router1:/# arp
Address          Hwtype  Hwaddress      Flags Mask      Iface
195.11.14.5      ether   fe:9c:9f:af:b3:1b  C          eth0
root@router1:/# █

```

Trả lời: Do khi ping đến địa chỉ không có trong mạng nên router sẽ không tìm thấy route nên không ping đến được. Do đó địa chỉ MAC sẽ không thể hoàn thiện.

2.3.2.4. Giao thức ARP với địa chỉ thuộc nhánh mạng LAN không sử dụng

- Trên pc1 , gửi dữ liệu đến 195.11.14.200 (địa chỉ không tồn tại có thể được cấp phát trong nhánh LAN A) bằng lệnh

ping 195.11.14.200

```

root@pc1:/# ping 195.11.14.200
PING 195.11.14.200 (195.11.14.200) 56(84) bytes of data.
From 195.11.14.5 icmp_seq=1 Destination Host Unreachable
From 195.11.14.5 icmp_seq=2 Destination Host Unreachable
From 195.11.14.5 icmp_seq=3 Destination Host Unreachable
From 195.11.14.5 icmp_seq=4 Destination Host Unreachable
From 195.11.14.5 icmp_seq=5 Destination Host Unreachable
From 195.11.14.5 icmp_seq=6 Destination Host Unreachable
^C
--- 195.11.14.200 ping statistics ---
7 packets transmitted, 0 received, +6 errors, 100% packet loss, time 6151ms
pipe 4
root@pc1:/# █

```

- Ghi nhận kết quả hiển thị khi dùng lệnh arp trên pc1 và router1 . Nhận xét kết quả này.

```

root@pc1:/# arp
Address          Hwtype  Hwaddress      Flags Mask      Iface
195.11.14.200    ether   (incomplete)   C          eth0
195.11.14.1      ether   62:f2:00:f4:7d:78  C          eth0
root@pc1:/# █

```

```

root@router1:/# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
195.11.14.5              ether    fe:9c:9f:af:b3:1b   C                     eth0
root@router1:/# █

```

Trả lời: Do khi ping đến địa chỉ không có trong mạng nên router sẽ không tìm thấy route nên không ping đến được. Do đó địa chỉ MAC sẽ không thể hoàn thiện.

16) Hủy mạng ảo bằng lệnh Iwipe sau khi đã thực hiện xong Bài tập 6.

\$sudo kathara wipe

```

→ BaiTap6 sudo kathara wipe
Are you sure to wipe Kathara? (y/n) y
→ BaiTap6 █

```

Bài tập 7

1) Quan sát mô hình mạng cần xây dựng. Nhận diện các thiết bị (PC, Router...), giao diện (eth0, eth1...) với các địa chỉ IP được gán.

2) Tạo thư mục BaiTap7 trong workspace của sinh viên.

```

→ BaiTap7 mkdir pc1 pc2 pc3 router1 router2 router3
→ BaiTap7 touch lab.conf pc1.startup pc2.startup pc3.startup router1.startup router2.startup router3.startup
→ BaiTap7 ls
lab.conf      pc2      pc3.startup  router2      router3.startup
pc1          pc2.startup router1      router2.startup
pc1.startup  pc3      router1.startup router3
→ BaiTap7 █

```

3) Soạn thảo nội dung mô tả hình thái mạng theo thiết kế trên file lab.conf

```

lab.conf
14 pc1[0]=A
13 pc2[0]=C
12 pc3[0]=F
11
10 router1[0]=A
9 router1[1]=B
8 router1[2]=D
7
6 router2[0]=B
5 router2[1]=C
4 router2[2]=E
3
2 router3[0]=D
1 router3[1]=E
15 router3[2]=F

```

4) Đối với các file pc1.startup , pc2.startup và pc3.startup : thực hiện vạch đường mặc nhiên thông qua các Router tương ứng trong nhánh mạng.

pc1.startup ≡

```
1 ifconfig eth0 192.168.0.40/27 up
2 route add default gw 192.168.0.33
```

pc2.startup ≡

```
1 ifconfig eth0 192.168.0.100/27 up
2 route add default gw 192.168.0.97
```

pc3.startup ≡

```
1 ifconfig eth0 192.168.0.200/27 up
2 route add default gw 192.168.0.193
```

5) Đối với các file router1.startup , router2.startup và router3.startup: thực hiện vạch đường tĩnh và vạch đường mặc nhiên (nếu cần).

router1.startup ≡

```
6 ifconfig eth0 192.168.0.33/27 up
5 ifconfig eth1 192.168.0.65/27 up
4 ifconfig eth2 192.168.0.129/27 up
3 route add -net 192.168.0.96/27 gw 192.168.0.66
2 route add -net 192.168.0.160/27 gw 192.168.0.66
1 route add -net 192.168.0.192/27 gw 192.168.0.130
7 route add -net 192.168.0.160/27 gw 192.168.0.130
```

router2.startup ≡

```
6 ifconfig eth0 192.168.0.66/27 up
5 ifconfig eth1 192.168.0.97/27 up
4 ifconfig eth2 192.168.0.161/27 up
3 route add -net 192.168.0.32/27 gw 192.168.0.65
2 route add -net 192.168.0.128/27 gw 192.168.0.65
1 route add -net 192.168.0.128/27 gw 192.168.0.162
7 route add -net 192.168.0.192/27 gw 192.168.0.162
```



```

router3.startup
6 ifconfig eth0 192.168.0.130/27 up
5 ifconfig eth1 192.168.0.162/27 up
4 ifconfig eth2 192.168.0.193/27 up
3 route add -net 192.168.0.32/27 gw 192.168.0.129
2 route add -net 192.168.0.64/27 gw 192.168.0.129
1 route add -net 192.168.0.96/27 gw 192.168.0.161
7 route add -net 192.168.0.64/27 gw 192.168.0.161

```

6) Khởi động mạng ảo BaiTap7. Kiểm tra bảng vạch đường (bảng route) và địa chỉ IP của các giao diện mạng (bảng ifconfig) trên từng máy ảo để đảm bảo tính đúng đắn của mô hình mạng Bài Tập 7.

\$sudo kathara lstart

```

root@pc1:/# route
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
default          192.168.0.33   0.0.0.0         UG    0      0      0 eth0
192.168.0.32     0.0.0.0        255.255.255.224 U     0      0      0 eth0
root@pc1:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.0.40 netmask 255.255.255.224 broadcast 192.168.0.63
    ether 32:db:bc:2d:ef:10 txqueuelen 1000 (Ethernet)
    RX packets 25  bytes 3742 (3.6 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 0  bytes 0 (0.0 B)
    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
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4  bytes 340 (340.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 4  bytes 340 (340.0 B)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@pc1:/#

```



```

root@pc2:/# route
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
default          192.168.0.97    0.0.0.0          UG      0      0      0 eth0
192.168.0.96     0.0.0.0         255.255.255.224 U        0      0      0 eth0
root@pc2:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.100 netmask 255.255.255.224 broadcast 192.168.0.127
    ether be:19:ff:6a:e3:5a txqueuelen 1000 (Ethernet)
    RX packets 28 bytes 4252 (4.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    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
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4 bytes 340 (340.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 340 (340.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@pc2:/# █

```

```

root@pc3:/# route
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
default          192.168.0.193   0.0.0.0          UG      0      0      0 eth0
192.168.0.192    0.0.0.0         255.255.255.224 U        0      0      0 eth0
root@pc3:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.200 netmask 255.255.255.224 broadcast 192.168.0.223
    ether 0e:c8:9c:ba:18:4d txqueuelen 1000 (Ethernet)
    RX packets 30 bytes 4392 (4.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    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
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4 bytes 344 (344.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 344 (344.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@pc3:/# █

```

```

root@router1:/# route
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
192.168.0.32     0.0.0.0         255.255.255.224 U        0      0      0 eth0
192.168.0.64     0.0.0.0         255.255.255.224 U        0      0      0 eth1
192.168.0.96     192.168.0.66    255.255.255.224 UG       0      0      0 eth1
192.168.0.128    0.0.0.0         255.255.255.224 U        0      0      0 eth2
192.168.0.160    192.168.0.130   255.255.255.224 UG       0      0      0 eth2
192.168.0.160    192.168.0.66    255.255.255.224 UG       0      0      0 eth1
192.168.0.192    192.168.0.130   255.255.255.224 UG       0      0      0 eth2

```

```

root@router1:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.33 netmask 255.255.255.224 broadcast 192.168.0.63
    ether 8e:ac:72:3d:f6:78 txqueuelen 1000 (Ethernet)
    RX packets 31  bytes 4612 (4.5 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 0  bytes 0 (0.0 B)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.65 netmask 255.255.255.224 broadcast 192.168.0.95
    ether d6:06:17:8a:08:04 txqueuelen 1000 (Ethernet)
    RX packets 32  bytes 4742 (4.6 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 0  bytes 0 (0.0 B)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.129 netmask 255.255.255.224 broadcast 192.168.0.159
    ether 8a:7c:b3:01:c7:82 txqueuelen 1000 (Ethernet)
    RX packets 31  bytes 4612 (4.5 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 0  bytes 0 (0.0 B)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

```

```

root@router2:/# route
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
192.168.0.32     192.168.0.65    255.255.255.224 UG       0      0      0 eth0
192.168.0.64     0.0.0.0         255.255.255.224 U        0      0      0 eth0
192.168.0.96     0.0.0.0         255.255.255.224 U        0      0      0 eth1
192.168.0.128    192.168.0.162   255.255.255.224 UG       0      0      0 eth2
192.168.0.128    192.168.0.65    255.255.255.224 UG       0      0      0 eth0
192.168.0.160    0.0.0.0         255.255.255.224 U        0      0      0 eth2
192.168.0.192    192.168.0.162   255.255.255.224 UG       0      0      0 eth2

```

```

root@router2:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.66 netmask 255.255.255.224 broadcast 192.168.0.95
    ether e2:fa:cf:c0:08:8e txqueuelen 1000 (Ethernet)
    RX packets 33 bytes 4872 (4.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.97 netmask 255.255.255.224 broadcast 192.168.0.127
    ether d6:a1:8c:26:e7:4c txqueuelen 1000 (Ethernet)
    RX packets 30 bytes 4482 (4.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.161 netmask 255.255.255.224 broadcast 192.168.0.191
    ether ea:32:2b:1f:fc:09 txqueuelen 1000 (Ethernet)
    RX packets 32 bytes 4682 (4.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

```

root@router3:/# route
Kernel IP routing table

```

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.0.32	192.168.0.129	255.255.255.224	UG	0	0	0	eth0
192.168.0.64	192.168.0.161	255.255.255.224	UG	0	0	0	eth1
192.168.0.64	192.168.0.129	255.255.255.224	UG	0	0	0	eth0
192.168.0.96	192.168.0.161	255.255.255.224	UG	0	0	0	eth1
192.168.0.128	0.0.0.0	255.255.255.224	U	0	0	0	eth0
192.168.0.160	0.0.0.0	255.255.255.224	U	0	0	0	eth1
192.168.0.192	0.0.0.0	255.255.255.224	U	0	0	0	eth2

```

root@router3:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.130 netmask 255.255.255.224 broadcast 192.168.0.159
    ether de:4f:ed:c7:2c:a6 txqueuelen 1000 (Ethernet)
    RX packets 35 bytes 5192 (5.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.162 netmask 255.255.255.224 broadcast 192.168.0.191
    ether 26:3c:8e:81:ea:04 txqueuelen 1000 (Ethernet)
    RX packets 34 bytes 5122 (5.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.193 netmask 255.255.255.224 broadcast 192.168.0.223
    ether 76:48:e2:ad:f7:0a txqueuelen 1000 (Ethernet)
    RX packets 32 bytes 4862 (4.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Kiểm tra tính liên thông giữa pc1 , pc2 và pc3 trong mạng (bằng ping).

- Ping pc1 đến pc2:

```
root@pc1:/# ping 192.168.0.100
PING 192.168.0.100 (192.168.0.100) 56(84) bytes of data.
64 bytes from 192.168.0.100: icmp_seq=1 ttl=62 time=0.326 ms
64 bytes from 192.168.0.100: icmp_seq=2 ttl=62 time=0.150 ms
64 bytes from 192.168.0.100: icmp_seq=3 ttl=62 time=0.151 ms
^C
--- 192.168.0.100 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2050ms
rtt min/avg/max/mdev = 0.150/0.209/0.326/0.082 ms
```

- Ping pc1 đến pc3:

```
root@pc1:/# ping 192.168.0.200
PING 192.168.0.200 (192.168.0.200) 56(84) bytes of data.
64 bytes from 192.168.0.200: icmp_seq=1 ttl=62 time=0.292 ms
64 bytes from 192.168.0.200: icmp_seq=2 ttl=62 time=0.175 ms
64 bytes from 192.168.0.200: icmp_seq=3 ttl=62 time=0.147 ms
^C
--- 192.168.0.200 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2053ms
rtt min/avg/max/mdev = 0.147/0.204/0.292/0.064 ms
root@pc1:/# █
```

- Ping pc2 đến pc1:

```
root@pc2:/# ping 192.168.0.40
PING 192.168.0.40 (192.168.0.40) 56(84) bytes of data.
64 bytes from 192.168.0.40: icmp_seq=1 ttl=62 time=0.194 ms
64 bytes from 192.168.0.40: icmp_seq=2 ttl=62 time=0.151 ms
64 bytes from 192.168.0.40: icmp_seq=3 ttl=62 time=0.152 ms
^C
--- 192.168.0.40 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2036ms
rtt min/avg/max/mdev = 0.151/0.165/0.194/0.024 ms
root@pc2:/# █
```

- Ping pc2 đến pc3:


```
root@pc2:/# ping 192.168.0.200
PING 192.168.0.200 (192.168.0.200) 56(84) bytes of data.
64 bytes from 192.168.0.200: icmp_seq=1 ttl=62 time=0.287 ms
64 bytes from 192.168.0.200: icmp_seq=2 ttl=62 time=0.154 ms
64 bytes from 192.168.0.200: icmp_seq=3 ttl=62 time=0.150 ms
^C
--- 192.168.0.200 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2032ms
rtt min/avg/max/mdev = 0.150/0.197/0.287/0.063 ms
root@pc2:/# █
```

- Ping pc3 đến pc1:

```
root@pc3:/# ping 192.168.0.40
PING 192.168.0.40 (192.168.0.40) 56(84) bytes of data.
64 bytes from 192.168.0.40: icmp_seq=1 ttl=62 time=0.252 ms
64 bytes from 192.168.0.40: icmp_seq=2 ttl=62 time=0.147 ms
64 bytes from 192.168.0.40: icmp_seq=3 ttl=62 time=0.152 ms
^C
--- 192.168.0.40 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2056ms
rtt min/avg/max/mdev = 0.147/0.183/0.252/0.050 ms
```

- Ping pc3 đến pc2:

```
root@pc3:/# ping 192.168.0.100
PING 192.168.0.100 (192.168.0.100) 56(84) bytes of data.
64 bytes from 192.168.0.100: icmp_seq=1 ttl=62 time=0.251 ms
64 bytes from 192.168.0.100: icmp_seq=2 ttl=62 time=0.155 ms
64 bytes from 192.168.0.100: icmp_seq=3 ttl=62 time=0.153 ms
^C
--- 192.168.0.100 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2056ms
rtt min/avg/max/mdev = 0.153/0.186/0.251/0.047 ms
root@pc3:/# █
```

7) Hủy mạng ảo bằng lệnh wipe sau khi đã thực hiện xong Bài tập 7

\$sudo kathara wipe

```
→ BaiTap7 sudo kathara wipe
Are you sure to wipe Kathara? (y/n) y
→ BaiTap7 █
```