LAB 1
CONSTRUCT A SIMPLE NETWORK

Name: **Đặng Nhật Tường**

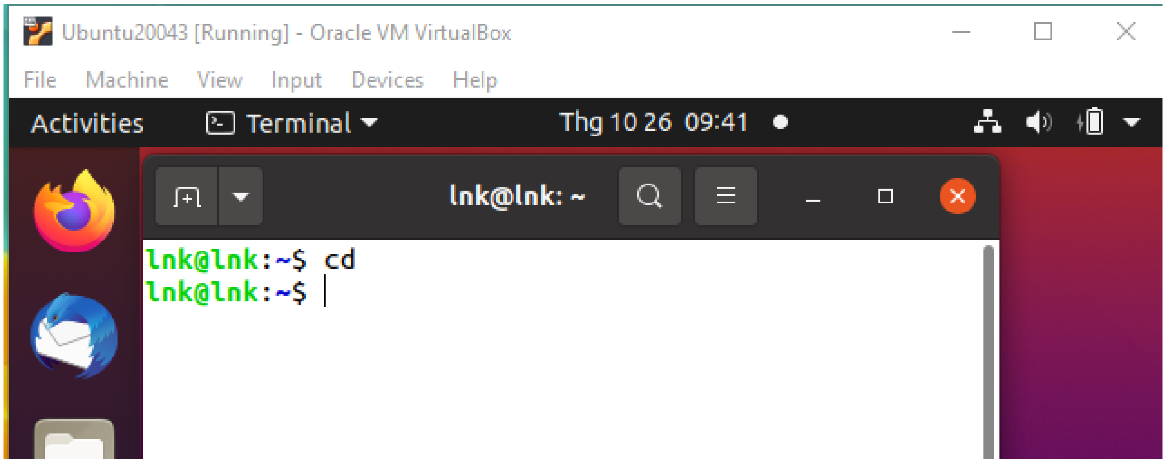
ID: **B2206021**

Group: **M02**

Submission: an ID_NAME_Lab01.pdf file describes clearly how did you solve the problem

Exercise 0: change the directory to your home directory

Answer: $cd



```
lnk@lnk:~$ cd
lnk@lnk:~$ 
```

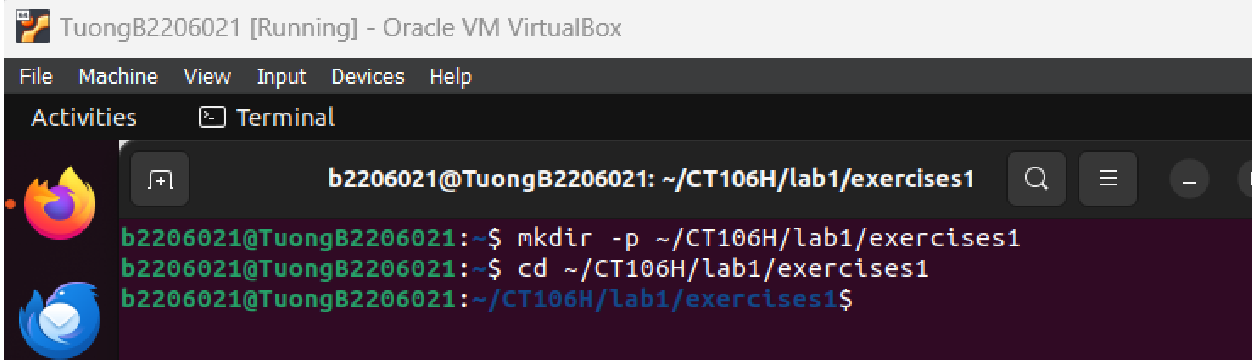
Exercise 1: Construct a simple network with two hosts connected to the same collision domain

Answer:

Create exercises1 directory

$ mkdir -p ~/CT106H/lab1/exercises1

$ cd ~/CT106H/lab1/exercises1

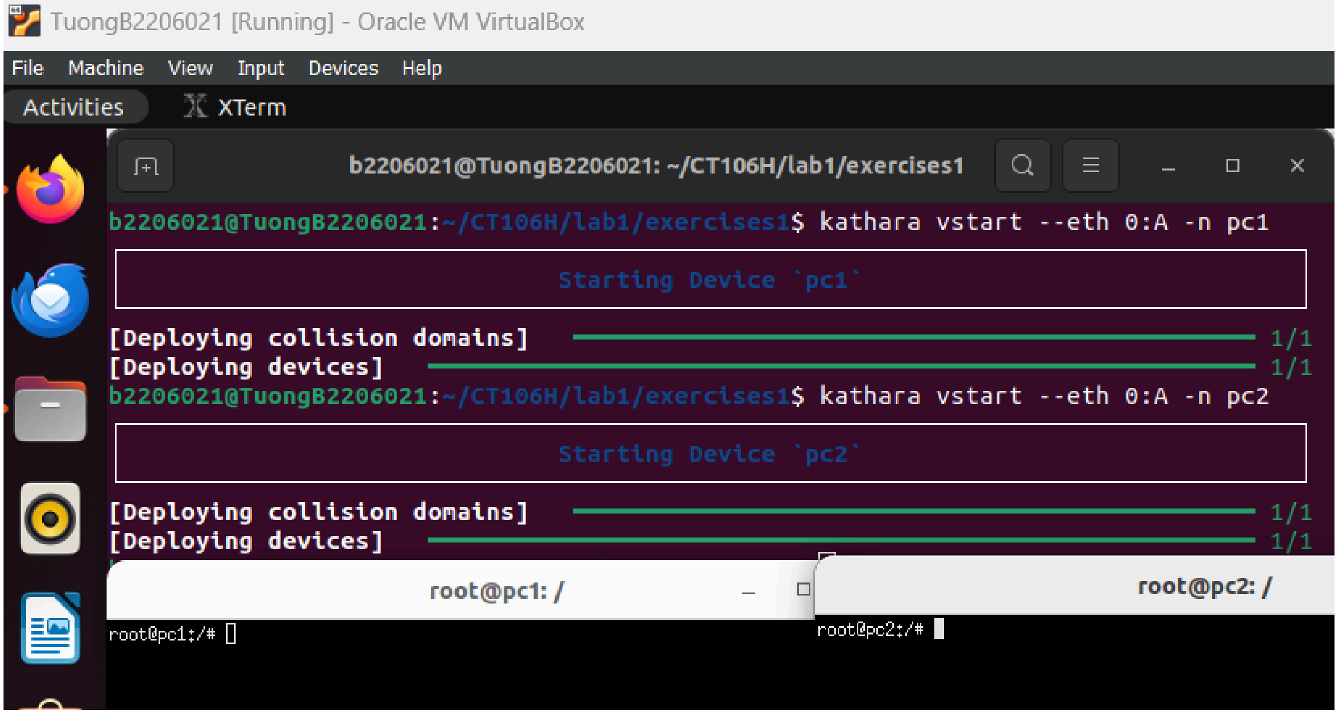

TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
b2206021@TuongB2206021: ~/CT106H/lab1/exercises1

```
b2206021@TuongB2206021:~$ mkdir -p ~/CT106H/lab1/exercises1
b2206021@TuongB2206021:~$ cd ~/CT106H/lab1/exercises1
b2206021@TuongB2206021:~/CT106H/lab1/exercises1$
```

Create two virtual devices, pc1 and pc2, and connect both to the same network A in the Kathara network simulation.

$kathara vstart --eth 0:A -n pc1

$kathara vstart --eth 0:A -n pc2


TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
XTerm
b2206021@TuongB2206021: ~/CT106H/lab1/exercises1
root@pc1: /
root@pc2: /

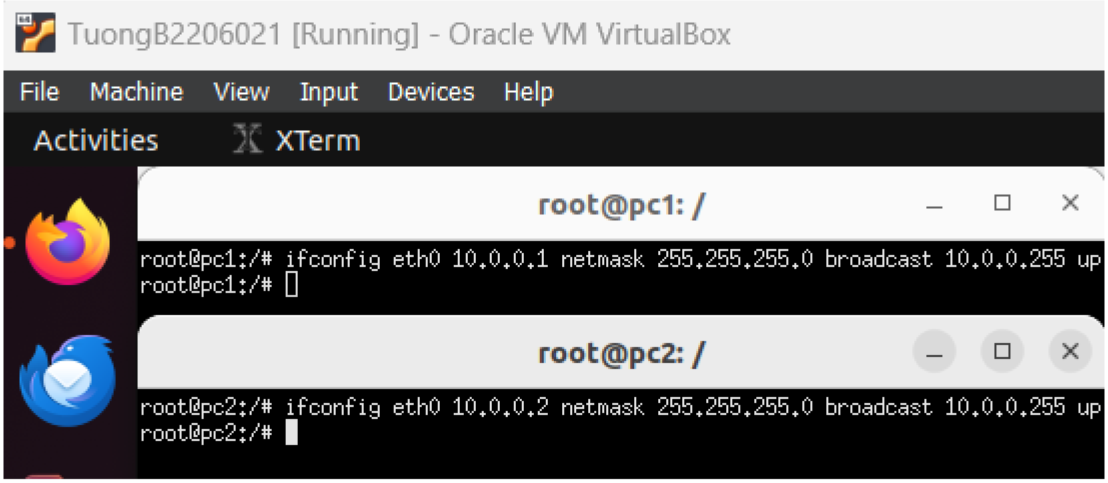
```
b2206021@TuongB2206021:~/CT106H/lab1/exercises1$ kathara vstart --eth 0:A -n pc1
                        Starting Device `pc1`
[Deploying collision domains]  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1/1
[Deploying devices]  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1/1
b2206021@TuongB2206021:~/CT106H/lab1/exercises1$ kathara vstart --eth 0:A -n pc2
                        Starting Device `pc2`
[Deploying collision domains]  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1/1
[Deploying devices]  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1/1
```

```
root@pc1:/#
```

```
root@pc2:/#
```

Configure network interfaces for pc1 & pc2

#ifconfig eth0 10.0.0.1 netmask 255.255.255.0 broadcast 10.0.0.255 up
#ifconfig eth0 10.0.0.2 netmask 255.255.255.0 broadcast 10.0.0.255 up

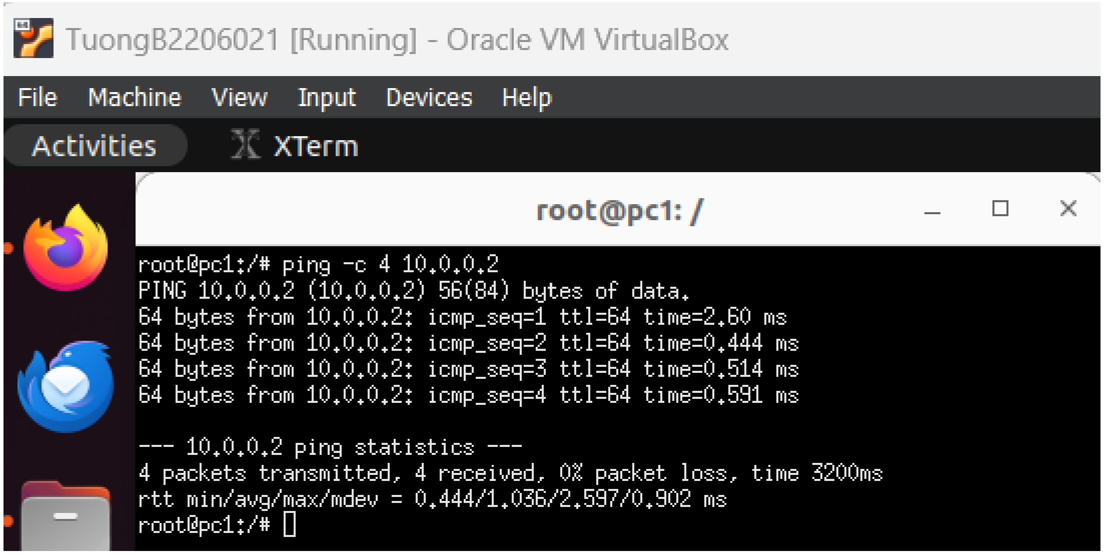

TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
XTerm
root@pc1: /
root@pc2: /

```
root@pc1:/# ifconfig eth0 10.0.0.1 netmask 255.255.255.0 broadcast 10.0.0.255 up
root@pc1:/# 
```

```
root@pc2:/# ifconfig eth0 10.0.0.2 netmask 255.255.255.0 broadcast 10.0.0.255 up
root@pc2:/# 
```

Test connectivity from pc1 to pc2, "-c 4" send 4 ping requests

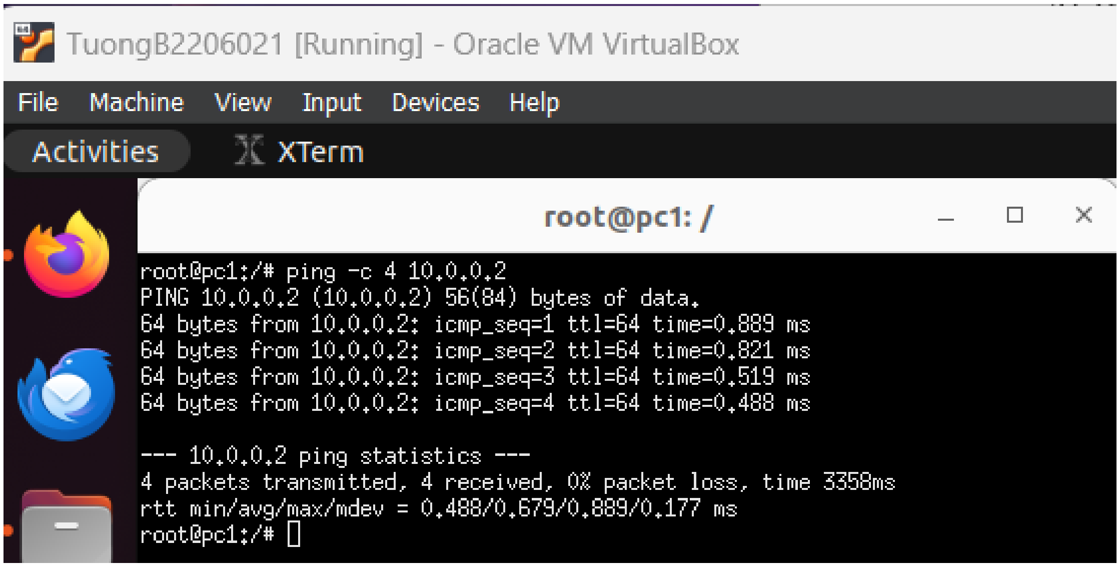
pc1:/#ping -c 4 10.0.0.2


TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
XTerm
root@pc1: /

```
root@pc1:/# ping -c 4 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=2.60 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.444 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.514 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.591 ms

--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3200ms
rtt min/avg/max/mdev = 0.444/1.036/2.597/0.902 ms
root@pc1:/# 
```

We can sniff the packets using "tcpdump" command

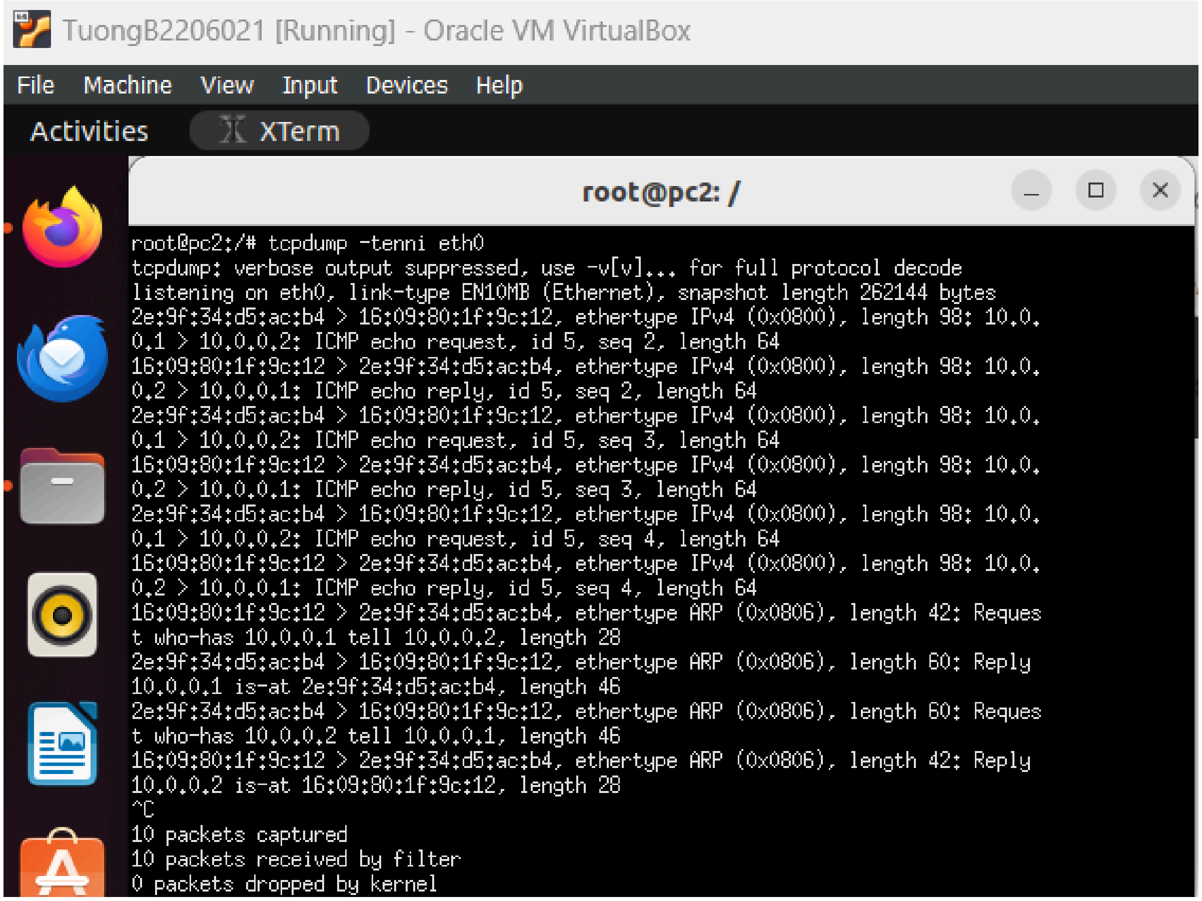
# ping -c 4 10.0.0.2


TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
XTerm
root@pc1: /

```
root@pc1:/# ping -c 4 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.889 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.821 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.519 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.488 ms

--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3358ms
rtt min/avg/max/mdev = 0.488/0.679/0.889/0.177 ms
root@pc1:/# 
```

As the same time, type in the following command on pc2 and press Ctrl + C to stop

# tcpdump -tenni eth0


TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
XTerm
root@pc2: /

```
root@pc2:/# tcpdump -tenni eth0
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
2e:9f:34:d5:ac:b4 > 16:09:80:1f:9c:12, ethertype IPv4 (0x0800), length 98: 10.0.
0.1 > 10.0.0.2: ICMP echo request, id 5, seq 2, length 64
16:09:80:1f:9c:12 > 2e:9f:34:d5:ac:b4, ethertype IPv4 (0x0800), length 98: 10.0.
0.2 > 10.0.0.1: ICMP echo reply, id 5, seq 2, length 64
2e:9f:34:d5:ac:b4 > 16:09:80:1f:9c:12, ethertype IPv4 (0x0800), length 98: 10.0.
0.1 > 10.0.0.2: ICMP echo request, id 5, seq 3, length 64
16:09:80:1f:9c:12 > 2e:9f:34:d5:ac:b4, ethertype IPv4 (0x0800), length 98: 10.0.
0.2 > 10.0.0.1: ICMP echo reply, id 5, seq 3, length 64
2e:9f:34:d5:ac:b4 > 16:09:80:1f:9c:12, ethertype IPv4 (0x0800), length 98: 10.0.
0.1 > 10.0.0.2: ICMP echo request, id 5, seq 4, length 64
16:09:80:1f:9c:12 > 2e:9f:34:d5:ac:b4, ethertype IPv4 (0x0800), length 98: 10.0.
0.2 > 10.0.0.1: ICMP echo reply, id 5, seq 4, length 64
16:09:80:1f:9c:12 > 2e:9f:34:d5:ac:b4, ethertype ARP (0x0806), length 42: Reques
t who-has 10.0.0.1 tell 10.0.0.2, length 28
2e:9f:34:d5:ac:b4 > 16:09:80:1f:9c:12, ethertype ARP (0x0806), length 60: Reply
10.0.0.1 is-at 2e:9f:34:d5:ac:b4, length 46
2e:9f:34:d5:ac:b4 > 16:09:80:1f:9c:12, ethertype ARP (0x0806), length 60: Reques
t who-has 10.0.0.2 tell 10.0.0.1, length 46
16:09:80:1f:9c:12 > 2e:9f:34:d5:ac:b4, ethertype ARP (0x0806), length 42: Reply
10.0.0.2 is-at 16:09:80:1f:9c:12, length 28
^C
10 packets captured
10 packets received by filter
0 packets dropped by kernel
```

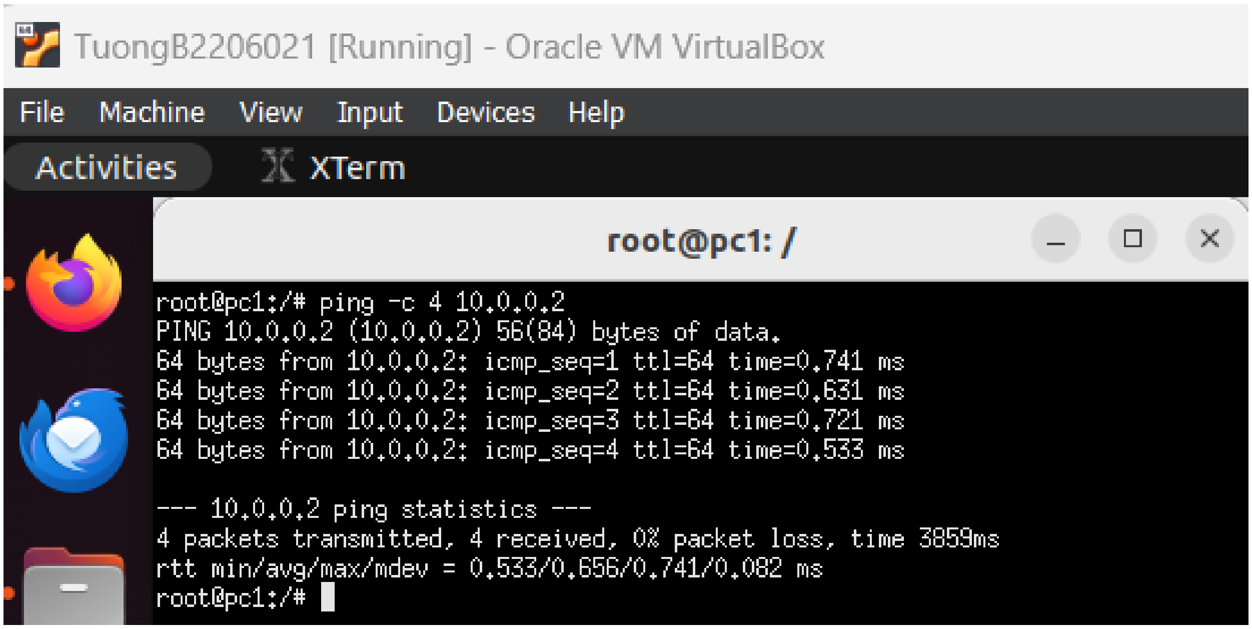
The command captures and displays network traffic on the eth0 interface with the following options:

- *t: Omits the timestamp in the output.*
- *e: Shows the link-layer (Ethernet) header.*
- *n: Disables DNS resolution, showing IP addresses instead of hostnames.*
- *nn: Shows numeric ports as well (without resolving service names).*
- *i eth0: Specifies the eth0 interface to capture traffic from.*

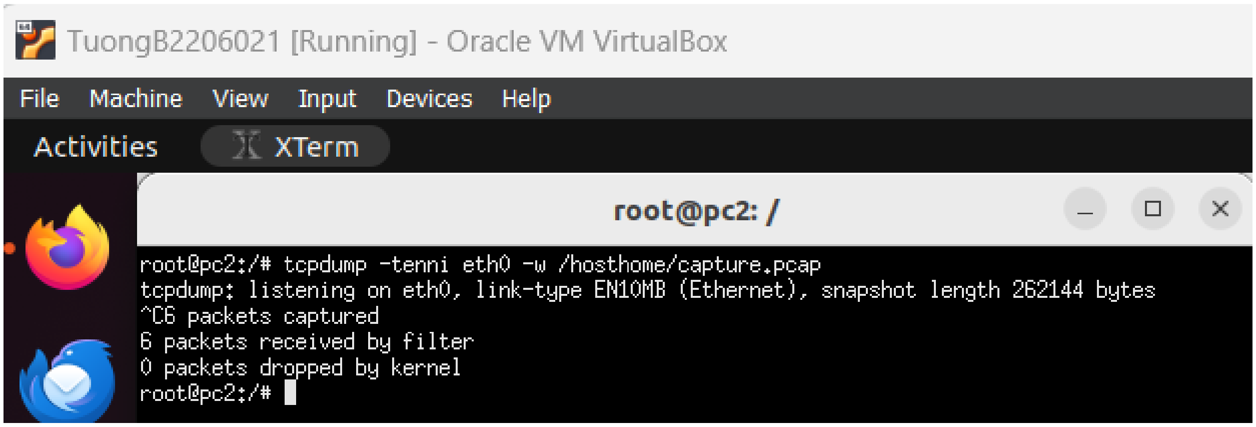
Same as above, but now store sniffed packets into file capture.pcap

# ping -c 4 10.0.0.2

# tcpdump -tenni eth0 -w /hosthome/capture.pcap *(The "-w" option store captured packets to specified file)*



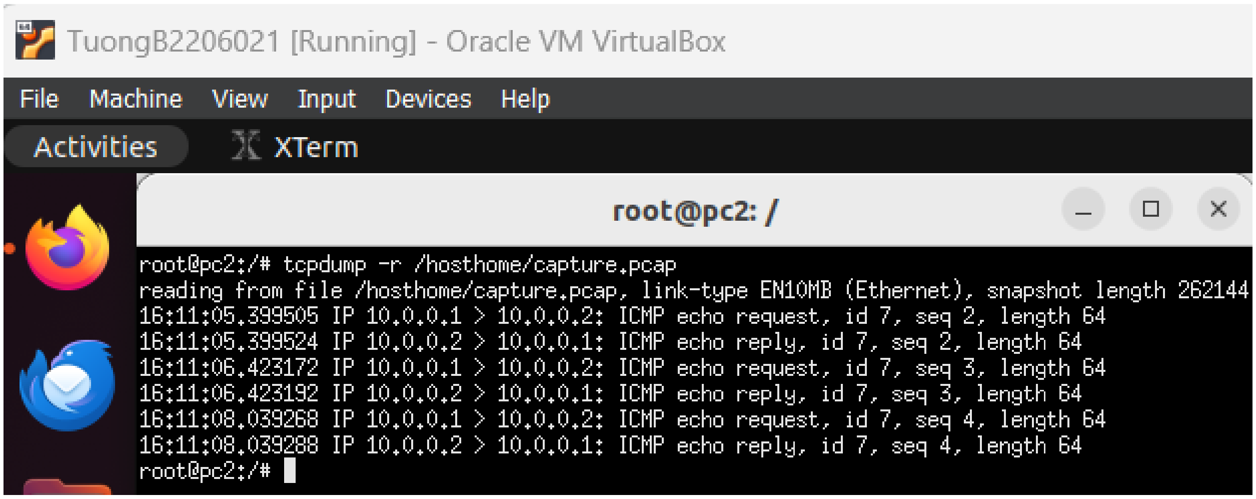
```
root@pc1:/# ping -c 4 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.741 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.631 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.721 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.533 ms

--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3859ms
rtt min/avg/max/mdev = 0.533/0.656/0.741/0.082 ms
root@pc1:/# 
```



```
root@pc2:/# tcpdump -tenni eth0 -w /hosthome/capture.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C6 packets captured
6 packets received by filter
0 packets dropped by kernel
root@pc2:/# 
```

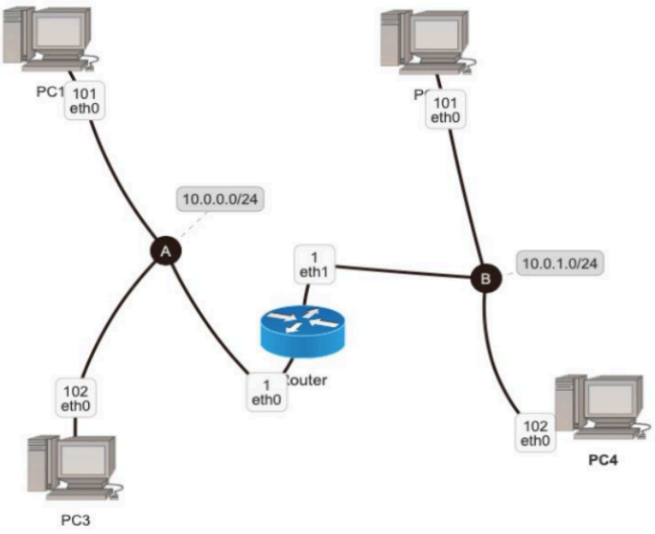
And to look for that packet, we can use the option "-r" in tcpdump command to open the file

tcpdump -r /hosthome/capture.pcap



```
root@pc2:/# tcpdump -r /hosthome/capture.pcap
reading from file /hosthome/capture.pcap, link-type EN10MB (Ethernet), snapshot length 262144
16:11:05.399505 IP 10.0.0.1 > 10.0.0.2: ICMP echo request, id 7, seq 2, length 64
16:11:05.399524 IP 10.0.0.2 > 10.0.0.1: ICMP echo reply, id 7, seq 2, length 64
16:11:06.423172 IP 10.0.0.1 > 10.0.0.2: ICMP echo request, id 7, seq 3, length 64
16:11:06.423192 IP 10.0.0.2 > 10.0.0.1: ICMP echo reply, id 7, seq 3, length 64
16:11:08.039268 IP 10.0.0.1 > 10.0.0.2: ICMP echo request, id 7, seq 4, length 64
16:11:08.039288 IP 10.0.0.2 > 10.0.0.1: ICMP echo reply, id 7, seq 4, length 64
root@pc2:/# 
```

Exercise 2: Construct the following network

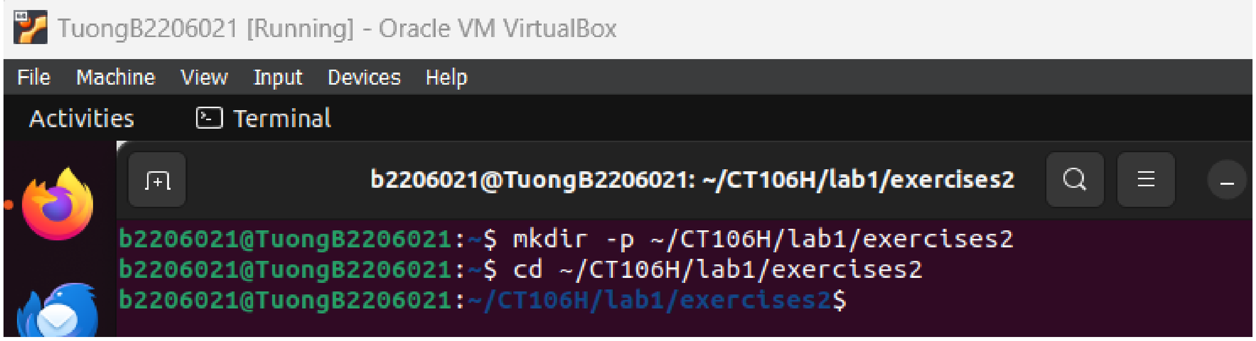


Answer:

Create exercise 2 directory

$ mkdir -p ~/CT106H/lab1/exercises2

$ cd ~/CT106H/lab1/exercises2


TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
b2206021@TuongB2206021: ~/CT106H/lab1/exercises2

```
b2206021@TuongB2206021:~$ mkdir -p ~/CT106H/lab1/exercises2
b2206021@TuongB2206021:~$ cd ~/CT106H/lab1/exercises2
b2206021@TuongB2206021:~/CT106H/lab1/exercises2$
```

Prepare the lab

$ mkdir pc1 pc2 pc3 pc4 router1 shared

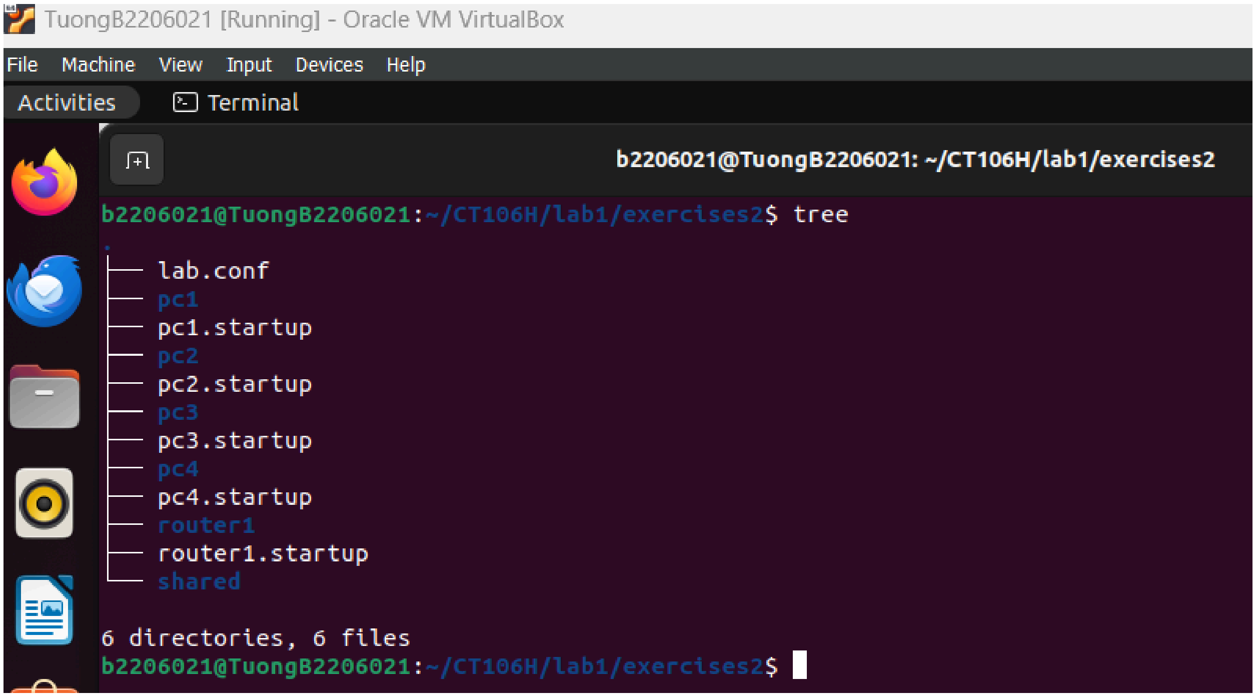
$ gedit pc1.startup

$ gedit pc2.startup

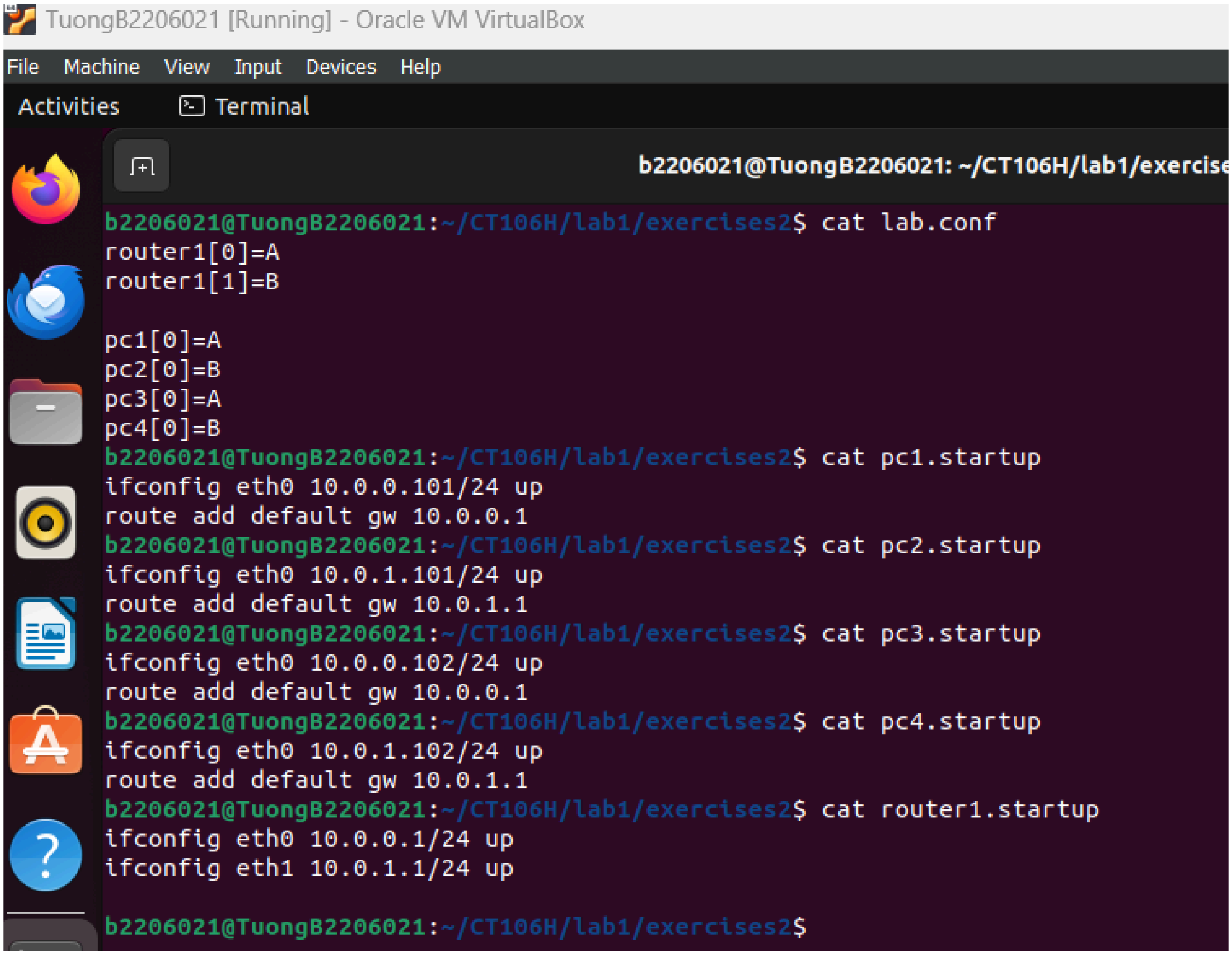
$ gedit pc3.startup

$ gedit pc4.startup

$ gedit router1.startup


TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
b2206021@TuongB2206021: ~/CT106H/lab1/exercises2

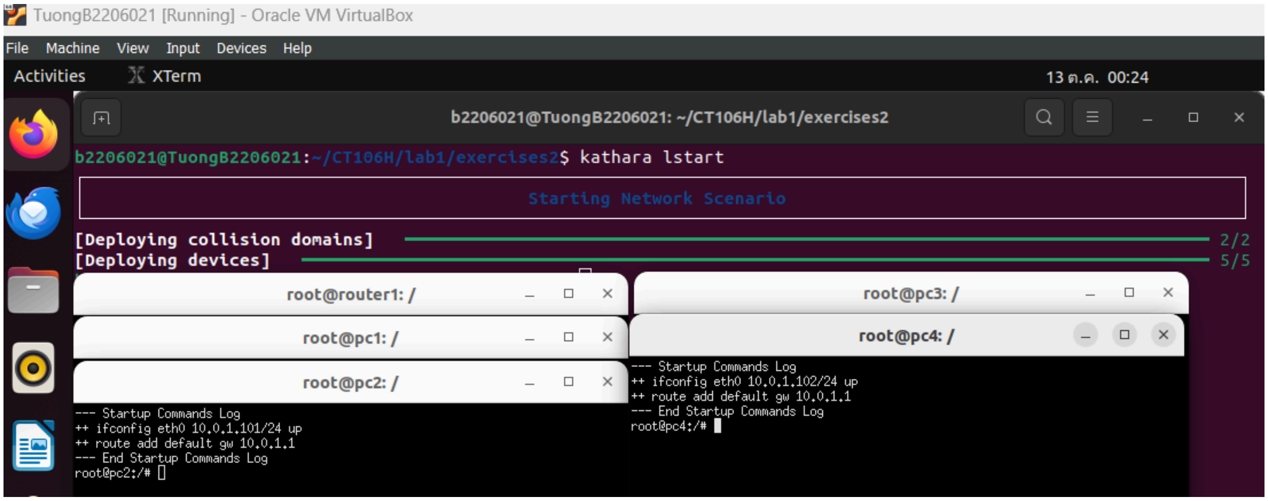
```
b2206021@TuongB2206021:~/CT106H/lab1/exercises2$ tree
.
├── lab.conf
├── pc1
├── pc1.startup
├── pc2
├── pc2.startup
├── pc3
├── pc3.startup
├── pc4
├── pc4.startup
├── router1
├── router1.startup
└── shared

6 directories, 6 files
b2206021@TuongB2206021:~/CT106H/lab1/exercises2$
```


TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
b2206021@TuongB2206021: ~/CT106H/lab1/exercise

```
b2206021@TuongB2206021:~/CT106H/lab1/exercises2$ cat lab.conf
router1[0]=A
router1[1]=B

pc1[0]=A
pc2[0]=B
pc3[0]=A
pc4[0]=B
b2206021@TuongB2206021:~/CT106H/lab1/exercises2$ cat pc1.startup
ifconfig eth0 10.0.0.101/24 up
route add default gw 10.0.0.1
b2206021@TuongB2206021:~/CT106H/lab1/exercises2$ cat pc2.startup
ifconfig eth0 10.0.1.101/24 up
route add default gw 10.0.1.1
b2206021@TuongB2206021:~/CT106H/lab1/exercises2$ cat pc3.startup
ifconfig eth0 10.0.0.102/24 up
route add default gw 10.0.0.1
b2206021@TuongB2206021:~/CT106H/lab1/exercises2$ cat pc4.startup
ifconfig eth0 10.0.1.102/24 up
route add default gw 10.0.1.1
b2206021@TuongB2206021:~/CT106H/lab1/exercises2$ cat router1.startup
ifconfig eth0 10.0.0.1/24 up
ifconfig eth1 10.0.1.1/24 up

b2206021@TuongB2206021:~/CT106H/lab1/exercises2$
```

Start the lab

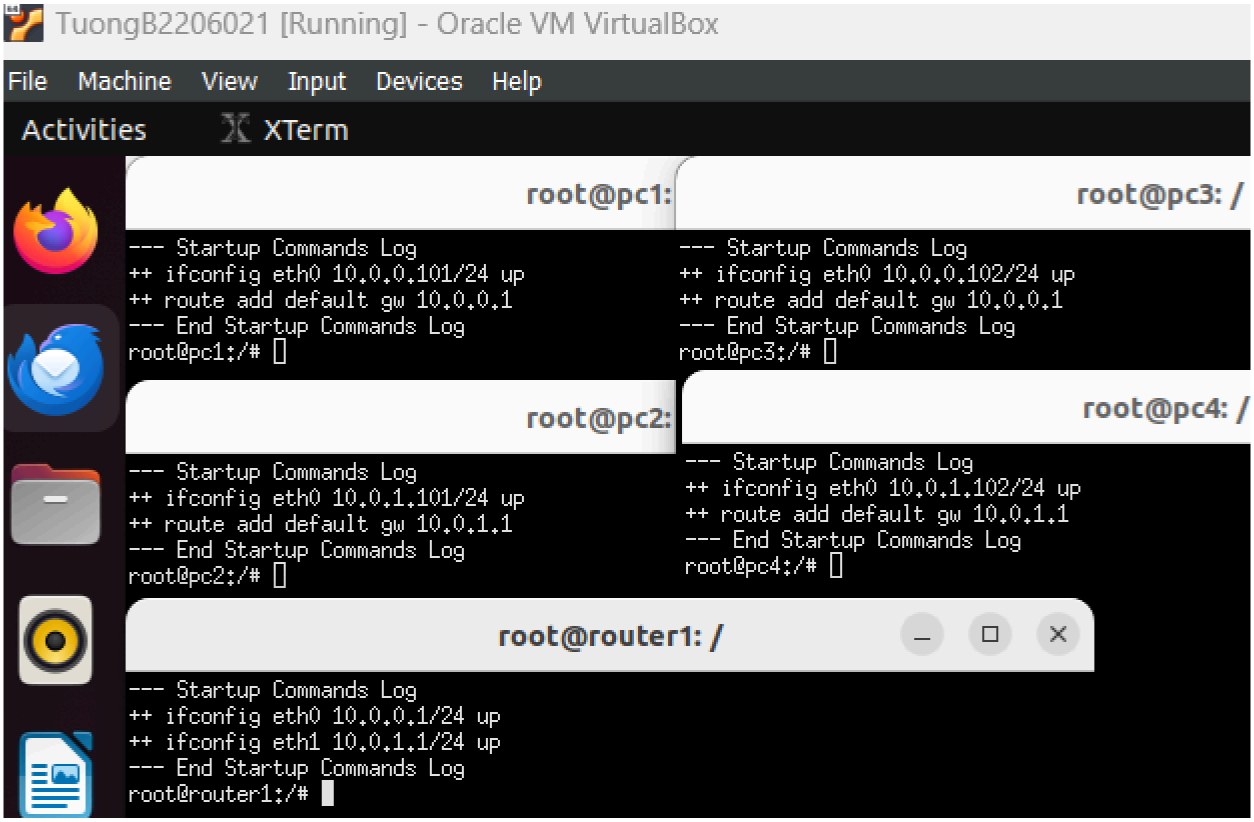
$ kathara lstart


TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
XTerm
13 ต.ค. 00:24
b2206021@TuongB2206021: ~/CT106H/lab1/exercises2
root@router1: /
root@pc1: /
root@pc2: /
root@pc3: /
root@pc4: /

```
b2206021@TuongB2206021:~/CT106H/lab1/exercises2$ kathara lstart
                          Starting Network Scenario
[Deploying collision domains] ──────────────────────────────── 2/2
[Deploying devices] ──────────────────────────────────────── 5/5
```

```
--- Startup Commands Log
++ ifconfig eth0 10.0.1.101/24 up
++ route add default gw 10.0.1.1
--- End Startup Commands Log
root@pc2:/# 
```

```
--- Startup Commands Log
++ ifconfig eth0 10.0.1.102/24 up
++ route add default gw 10.0.1.1
--- End Startup Commands Log
root@pc4:/# 
```


TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
XTerm

root@pc1:

```
--- Startup Commands Log
++ ifconfig eth0 10.0.0.101/24 up
++ route add default gw 10.0.0.1
--- End Startup Commands Log
root@pc1:/#
```

root@pc3: /

```
--- Startup Commands Log
++ ifconfig eth0 10.0.0.102/24 up
++ route add default gw 10.0.0.1
--- End Startup Commands Log
root@pc3:/#
```

root@pc2:

```
--- Startup Commands Log
++ ifconfig eth0 10.0.1.101/24 up
++ route add default gw 10.0.1.1
--- End Startup Commands Log
root@pc2:/#
```

root@pc4: /

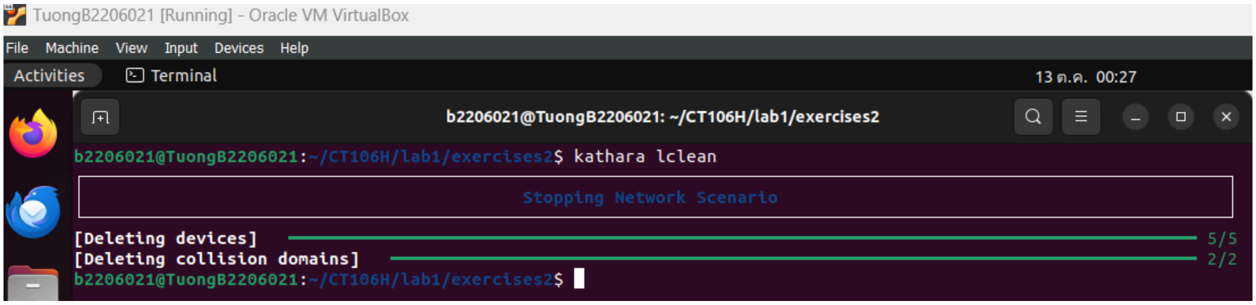
```
--- Startup Commands Log
++ ifconfig eth0 10.0.1.102/24 up
++ route add default gw 10.0.1.1
--- End Startup Commands Log
root@pc4:/#
```

root@router1: /

```
--- Startup Commands Log
++ ifconfig eth0 10.0.0.1/24 up
++ ifconfig eth1 10.0.1.1/24 up
--- End Startup Commands Log
root@router1:/#
```

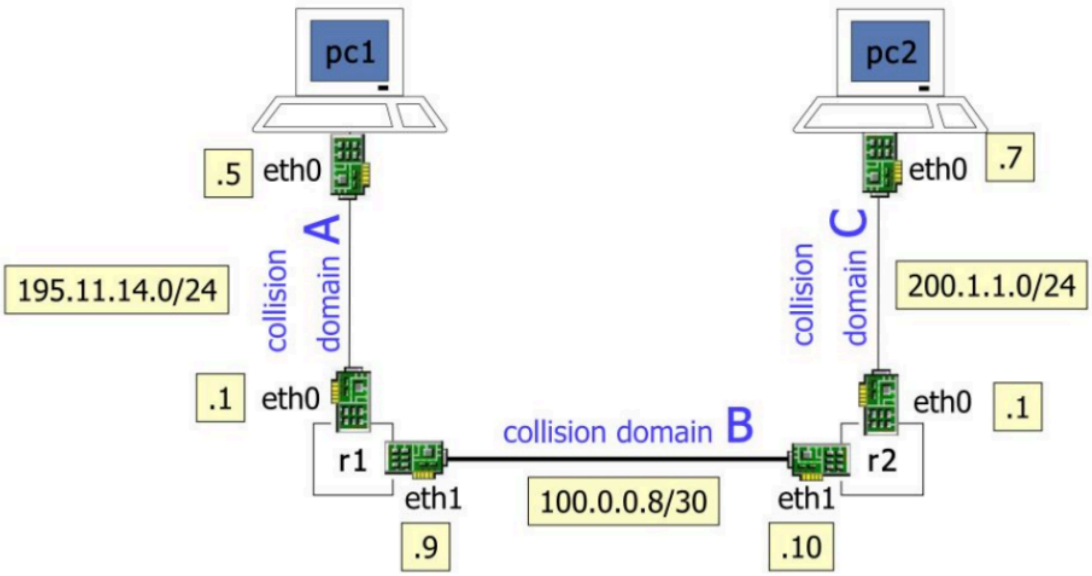
Run "kathara lclean" to ensure there are no lingering processes, nodes, or configurations from previous simulations, preventing conflicts or issues when starting a new lab setup.

$ kathara lclean


TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
13 ต.ค. 00:27
b2206021@TuongB2206021: ~/CT106H/lab1/exercises2

```
b2206021@TuongB2206021:~/CT106H/lab1/exercises2$ kathara lclean
                          Stopping Network Scenario
[Deleting devices]          ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5/5
[Deleting collision domains] ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2/2
b2206021@TuongB2206021:~/CT106H/lab1/exercises2$
```

Exercise 3: Construct the following network

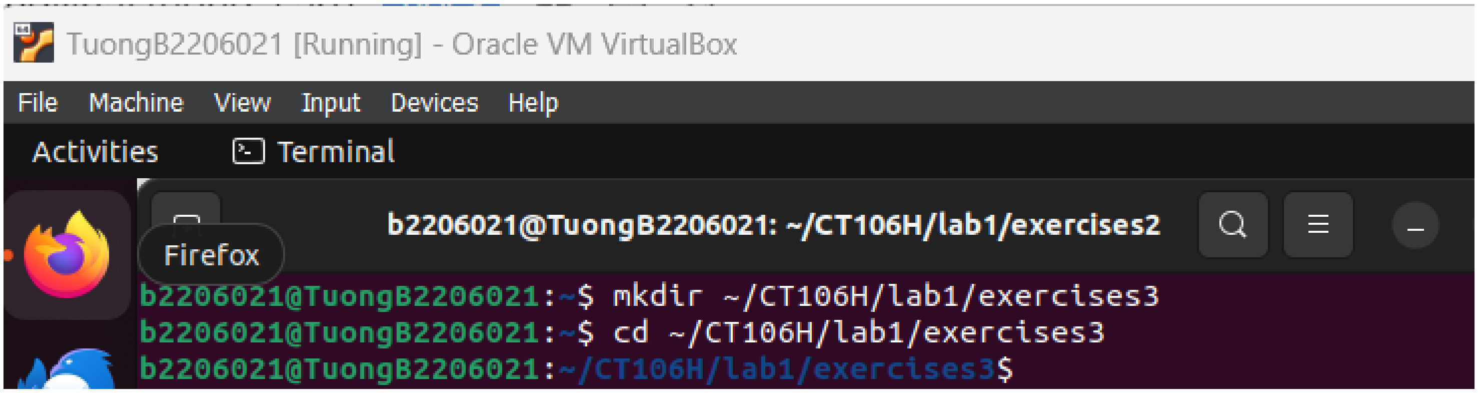


Answer:

Create exercise 3 directory

$ mkdir -p ~/CT106H/lab1/exercises3

$ cd ~/CT106H/lab1/exercises3


TuongB2206021 [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
Activities
Terminal
Firefox
b2206021@TuongB2206021: ~/CT106H/lab1/exercises2

```
b2206021@TuongB2206021:~$ mkdir ~/CT106H/lab1/exercises3
b2206021@TuongB2206021:~$ cd ~/CT106H/lab1/exercises3
b2206021@TuongB2206021:~/CT106H/lab1/exercises3$
```

Prepare the lab

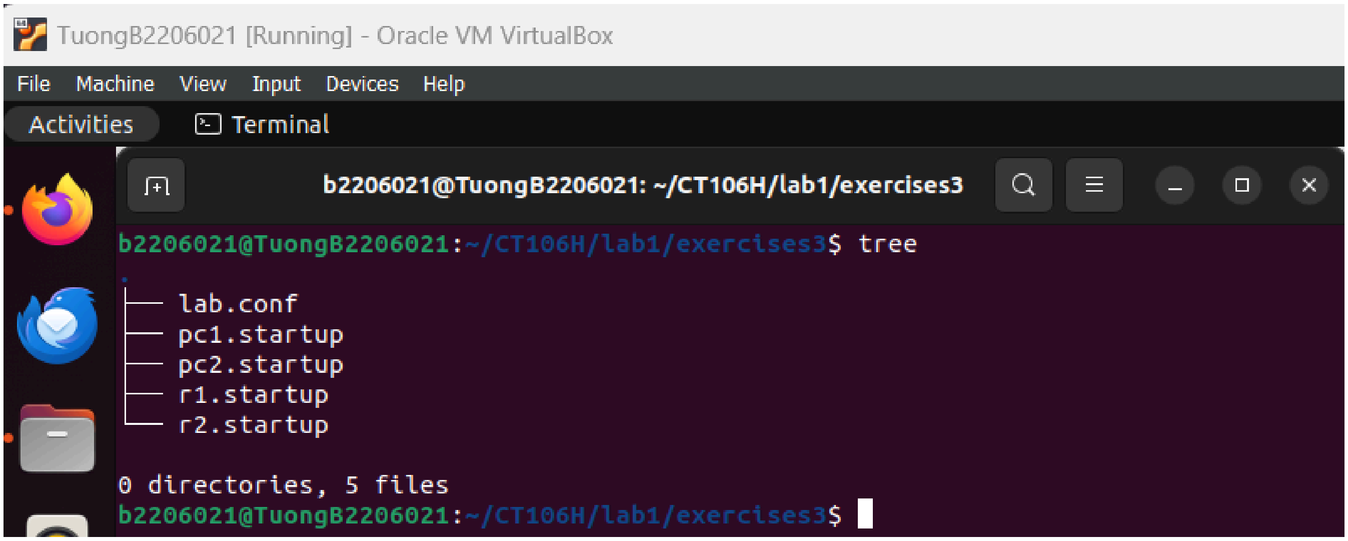
$ gedit lab.conf

$ gedit pc1.startup

$ gedit pc2.startup

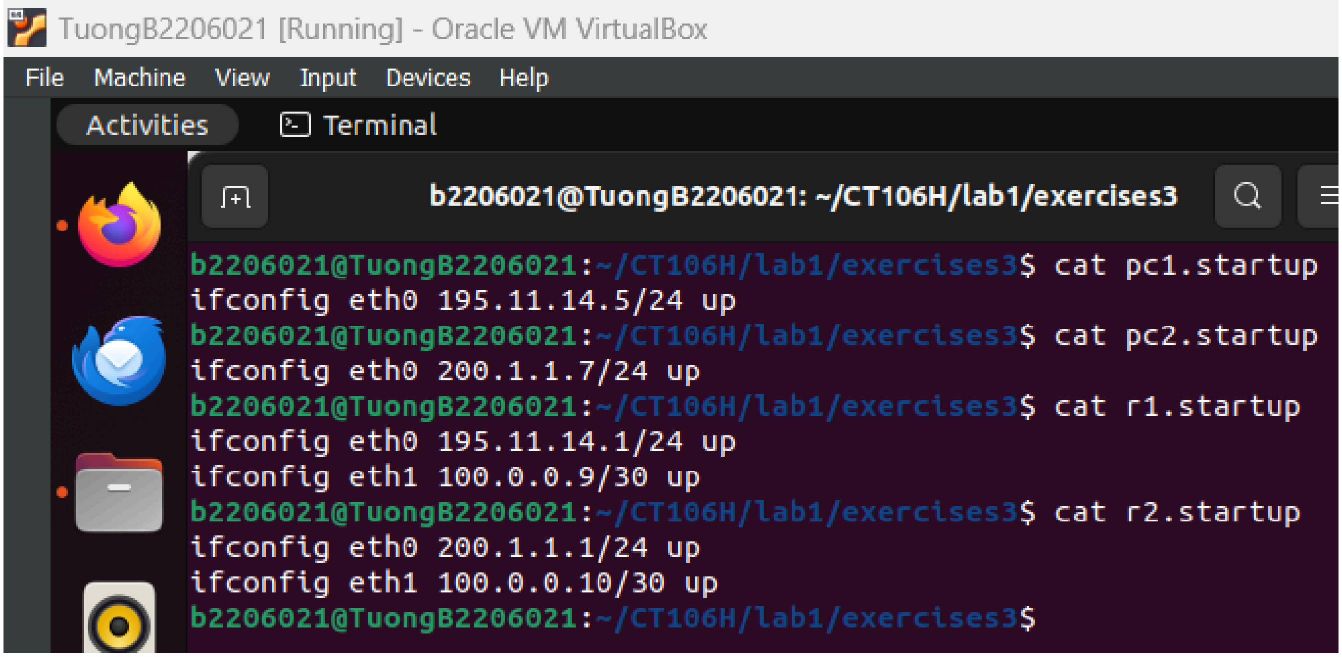
$ gedit r1.startup

$ gedit r2.startup



TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
b2206021@TuongB2206021: ~/CT106H/lab1/exercises3

```
b2206021@TuongB2206021:~/CT106H/lab1/exercises3$ tree
.
├── lab.conf
├── pc1.startup
├── pc2.startup
├── r1.startup
└── r2.startup

0 directories, 5 files
b2206021@TuongB2206021:~/CT106H/lab1/exercises3$ 
```

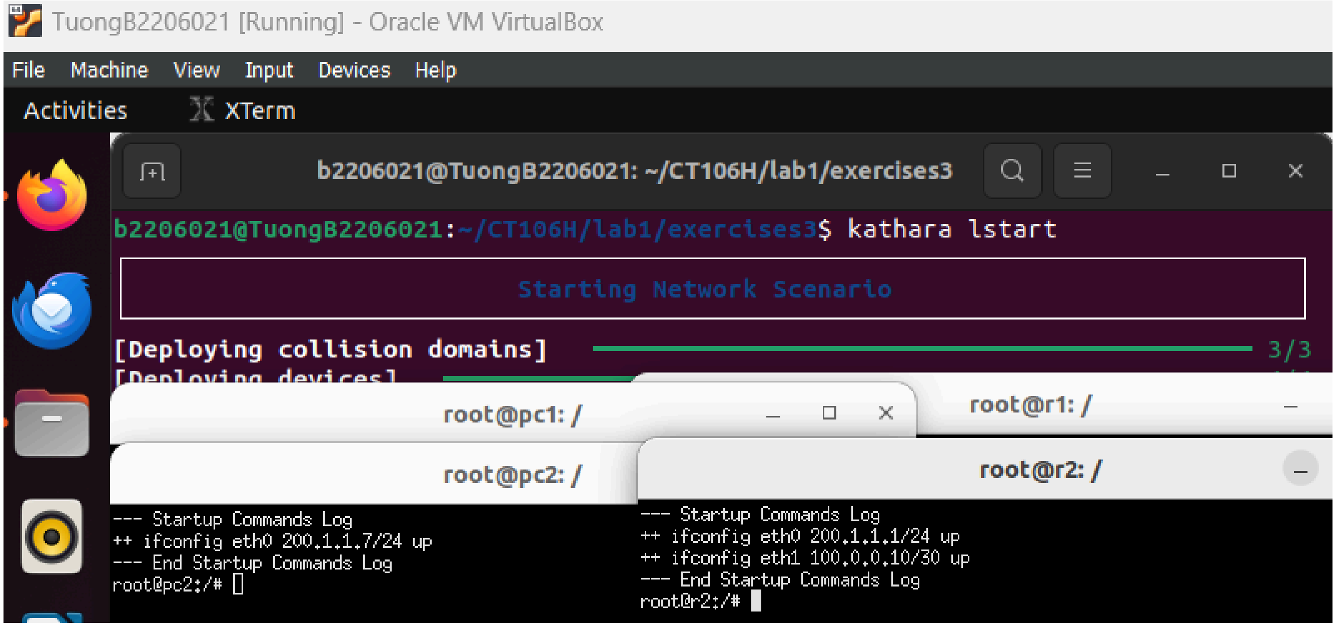


TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
b2206021@TuongB2206021: ~/CT106H/lab1/exercises3

```
b2206021@TuongB2206021:~/CT106H/lab1/exercises3$ cat pc1.startup
ifconfig eth0 195.11.14.5/24 up
b2206021@TuongB2206021:~/CT106H/lab1/exercises3$ cat pc2.startup
ifconfig eth0 200.1.1.7/24 up
b2206021@TuongB2206021:~/CT106H/lab1/exercises3$ cat r1.startup
ifconfig eth0 195.11.14.1/24 up
ifconfig eth1 100.0.0.9/30 up
b2206021@TuongB2206021:~/CT106H/lab1/exercises3$ cat r2.startup
ifconfig eth0 200.1.1.1/24 up
ifconfig eth1 100.0.0.10/30 up
b2206021@TuongB2206021:~/CT106H/lab1/exercises3$
```

Start the lab

$ kathara lstart


TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities XTerm
b2206021@TuongB2206021: ~/CT106H/lab1/exercises3
root@pc1: /
root@r1: /
root@pc2: /
root@r2: /

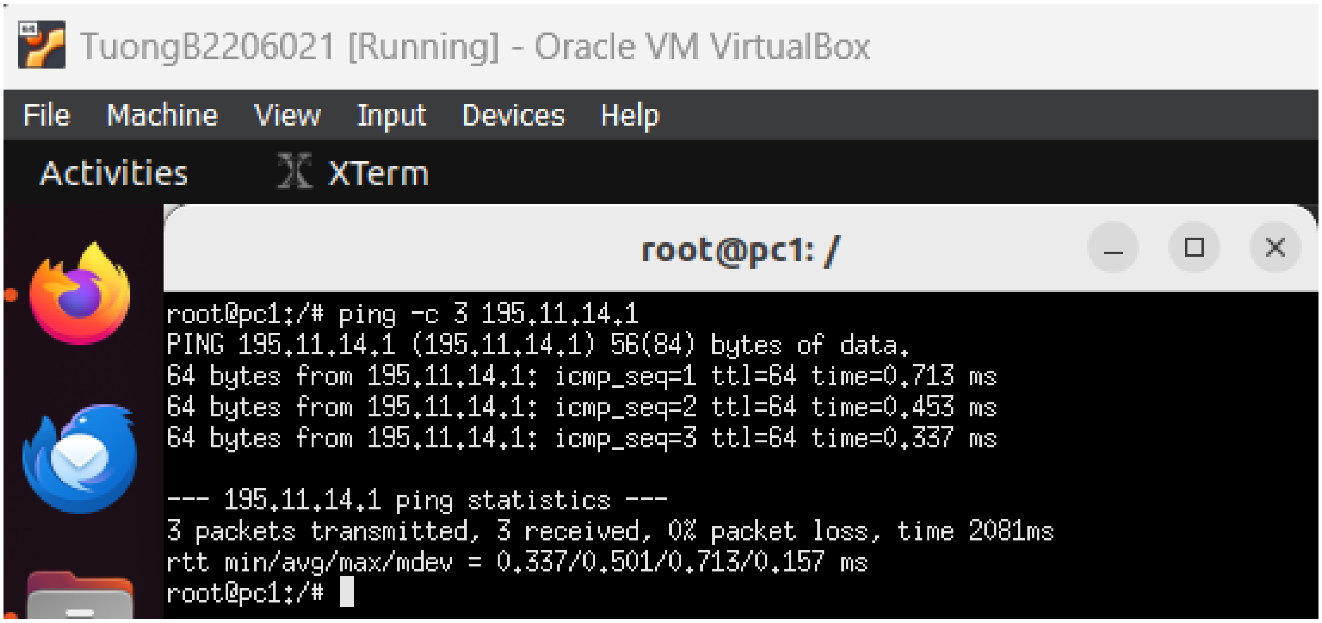
```
b2206021@TuongB2206021:~/CT106H/lab1/exercises3$ kathara lstart
                  Starting Network Scenario
[Deploying collision domains]                                    3/3
[Deploying devices]
```

```
--- Startup Commands Log
++ ifconfig eth0 200.1.1.7/24 up
--- End Startup Commands Log
root@pc2:/# 
```

```
--- Startup Commands Log
++ ifconfig eth0 200.1.1.1/24 up
++ ifconfig eth1 100.0.0.10/30 up
--- End Startup Commands Log
root@r2:/# 
```

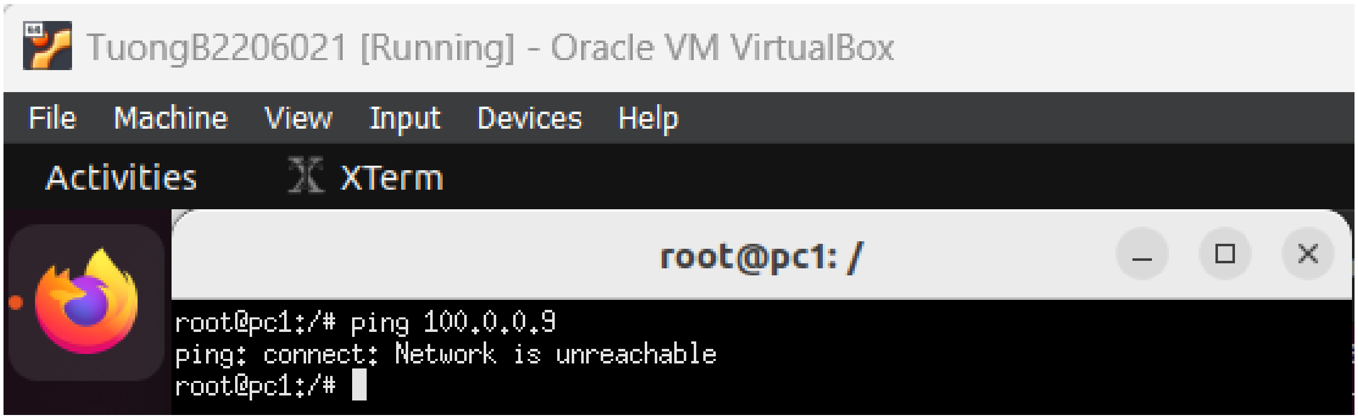
Testing connectivity

# ping -c 3 195.11.14.1


TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities XTerm
root@pc1: /

```
root@pc1:/# ping -c 3 195.11.14.1
PING 195.11.14.1 (195.11.14.1) 56(84) bytes of data.
64 bytes from 195.11.14.1: icmp_seq=1 ttl=64 time=0.713 ms
64 bytes from 195.11.14.1: icmp_seq=2 ttl=64 time=0.453 ms
64 bytes from 195.11.14.1: icmp_seq=3 ttl=64 time=0.337 ms

--- 195.11.14.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2081ms
rtt min/avg/max/mdev = 0.337/0.501/0.713/0.157 ms
root@pc1:/# 
```

# ping 100.0.0.9

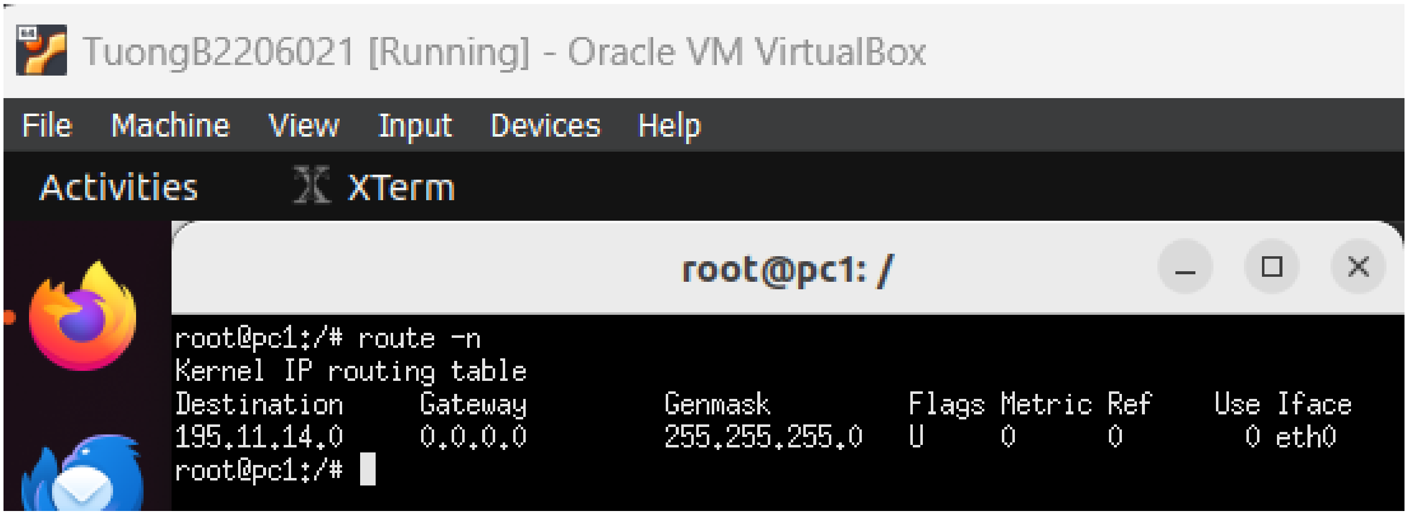

TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities XTerm
root@pc1: /

```
root@pc1:/# ping 100.0.0.9
ping: connect: Network is unreachable
root@pc1:/#
```

The connectivity test fails because pc1 and pc2 are in different IP subnets. pc1 is in the 195.11.14.0/24 subnet, and the target is in the 100.0.0.0/24 subnet. pc1 doesn't know how to reach this subnet

Inspecting routing tables

# route -n

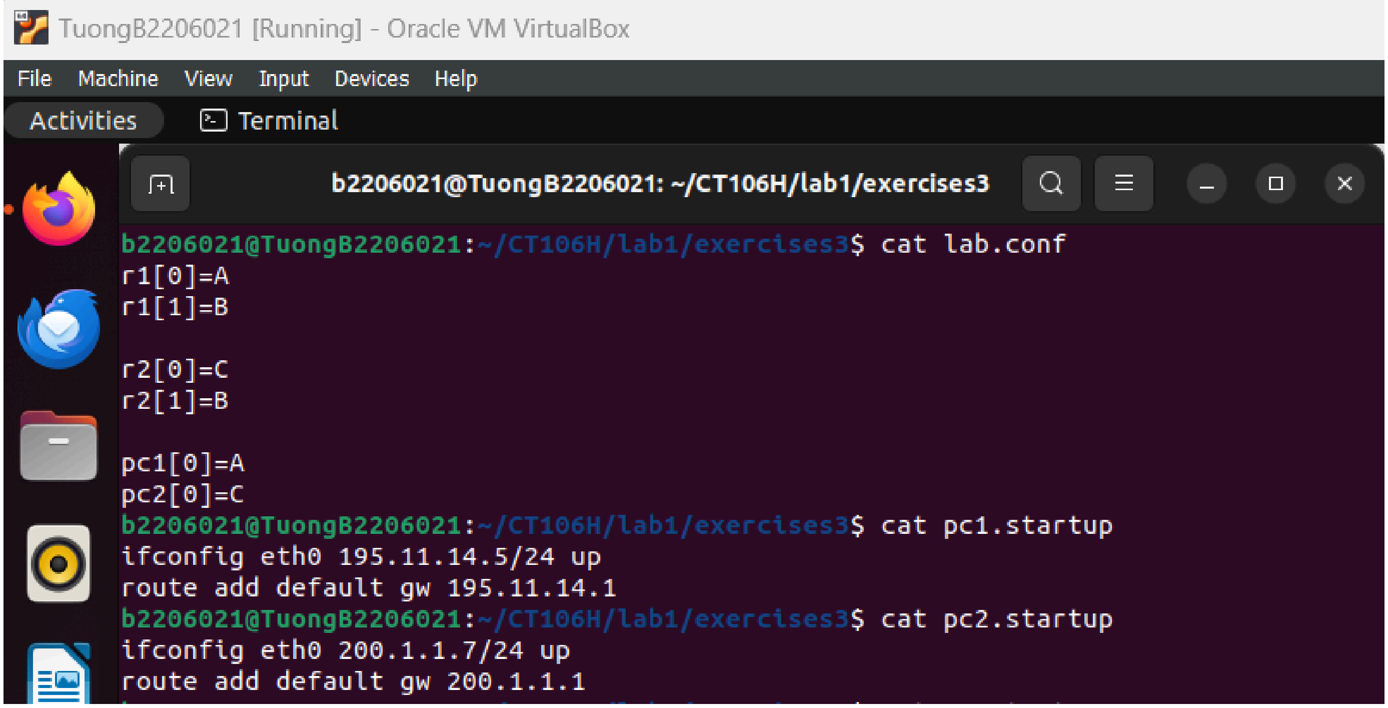

TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities XTerm
root@pc1: /

```
root@pc1:/# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
195.11.14.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
root@pc1:/#
```

To fix the problem we could specify the default route on the pcs: "through this gateway (IP number) you can reach all the other networks":

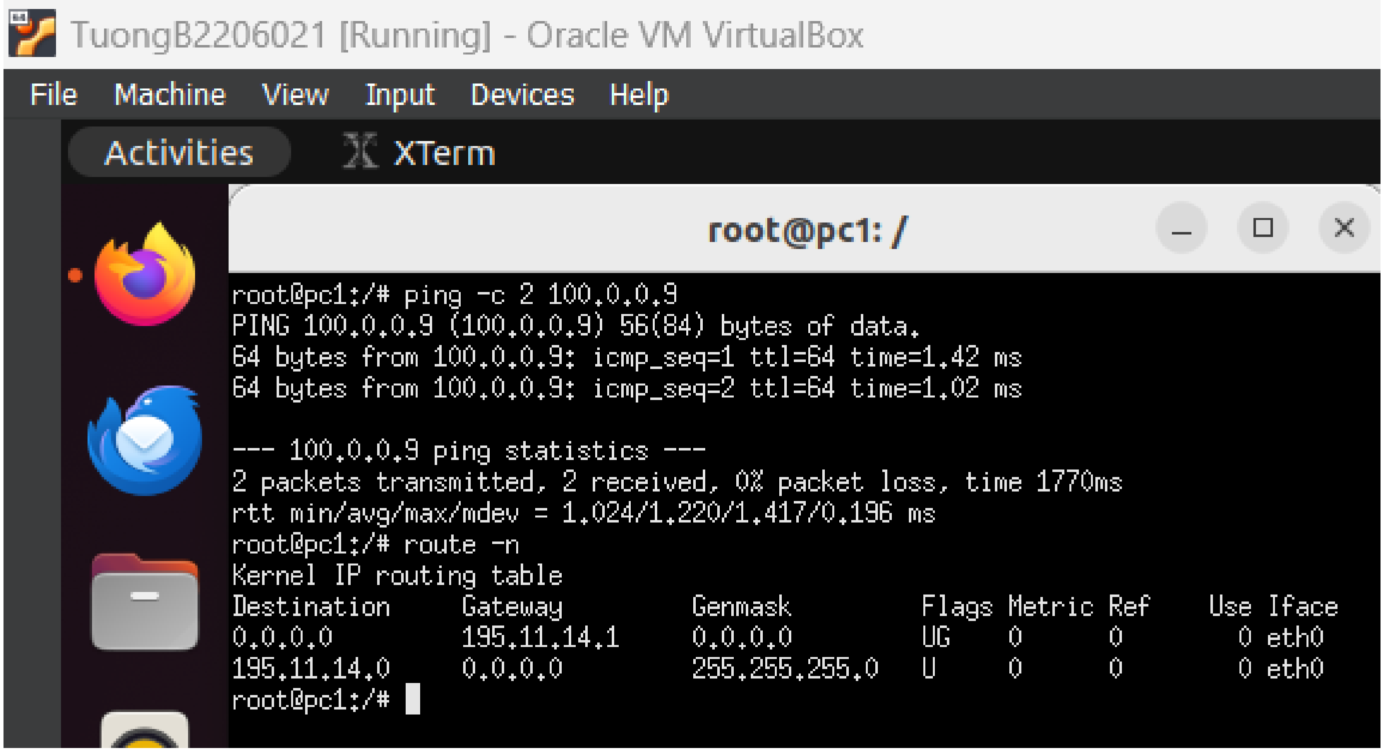
Add **"route add default gw 195.11.14.1"** to pc1.startup file
& **"route add default gw 200.1.1.1"** to pc2.startup file

This will set the default gateway for pc1 & pc2, allowing it to route traffic to networks outside its local subnet.


TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
b2206021@TuongB2206021: ~/CT106H/lab1/exercises3

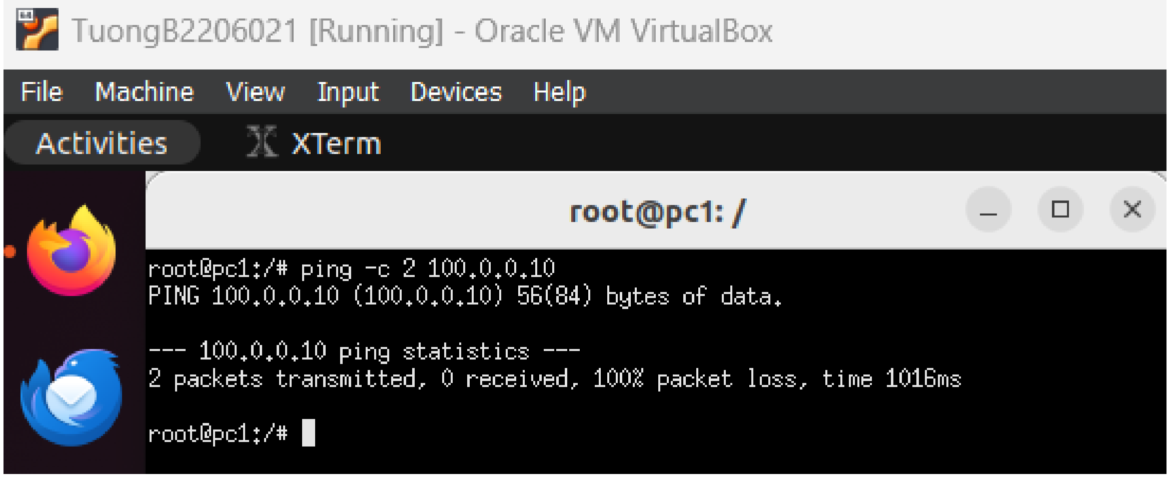
```
b2206021@TuongB2206021:~/CT106H/lab1/exercises3$ cat lab.conf
r1[0]=A
r1[1]=B

r2[0]=C
r2[1]=B

pc1[0]=A
pc2[0]=C
b2206021@TuongB2206021:~/CT106H/lab1/exercises3$ cat pc1.startup
ifconfig eth0 195.11.14.5/24 up
route add default gw 195.11.14.1
b2206021@TuongB2206021:~/CT106H/lab1/exercises3$ cat pc2.startup
ifconfig eth0 200.1.1.7/24 up
route add default gw 200.1.1.1
```

Testing connectivity


TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
XTerm
root@pc1: /

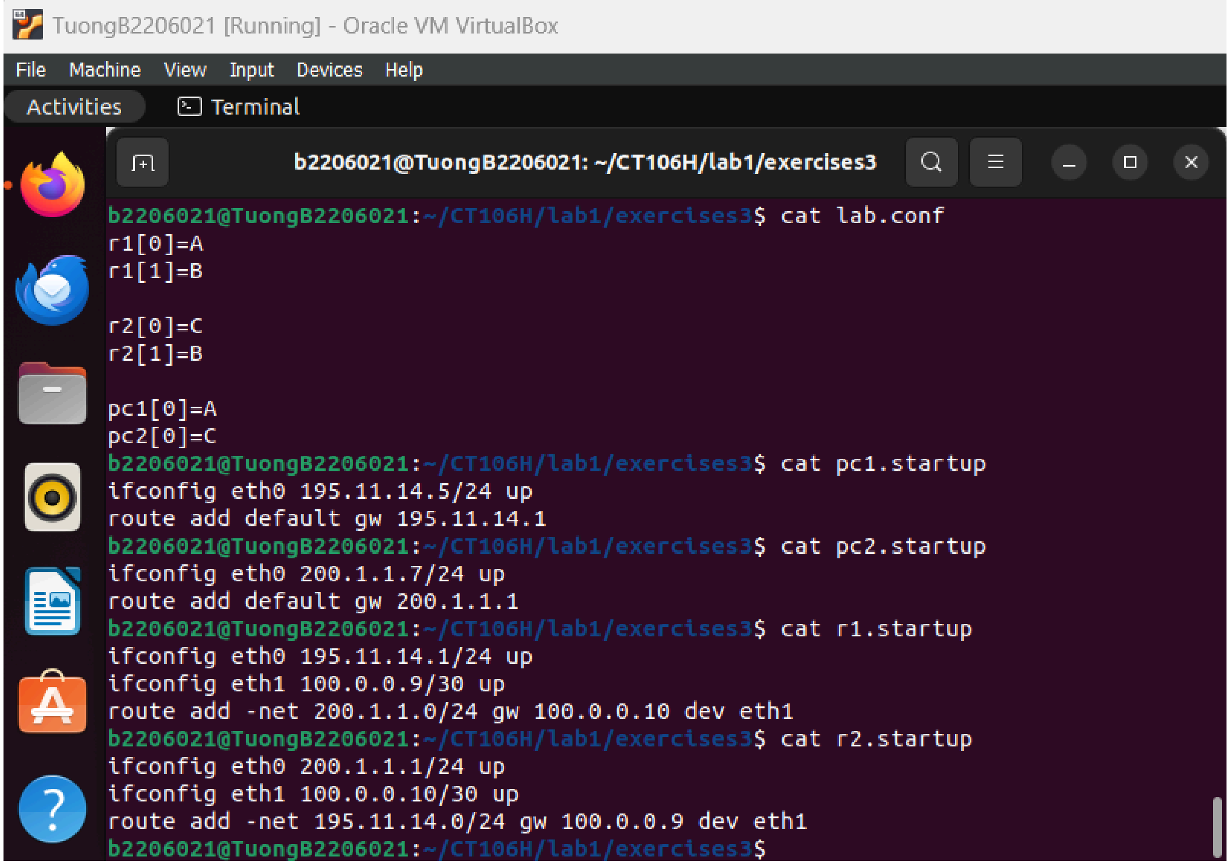
```
root@pc1:/# ping -c 2 100.0.0.9
PING 100.0.0.9 (100.0.0.9) 56(84) bytes of data.
64 bytes from 100.0.0.9: icmp_seq=1 ttl=64 time=1.42 ms
64 bytes from 100.0.0.9: icmp_seq=2 ttl=64 time=1.02 ms

--- 100.0.0.9 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1770ms
rtt min/avg/max/mdev = 1.024/1.220/1.417/0.196 ms
root@pc1:/# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         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:/#
```

After that, interfaces on r2 seem unreachable because r2 does not know how to reach such an address


TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
XTerm
root@pc1: /

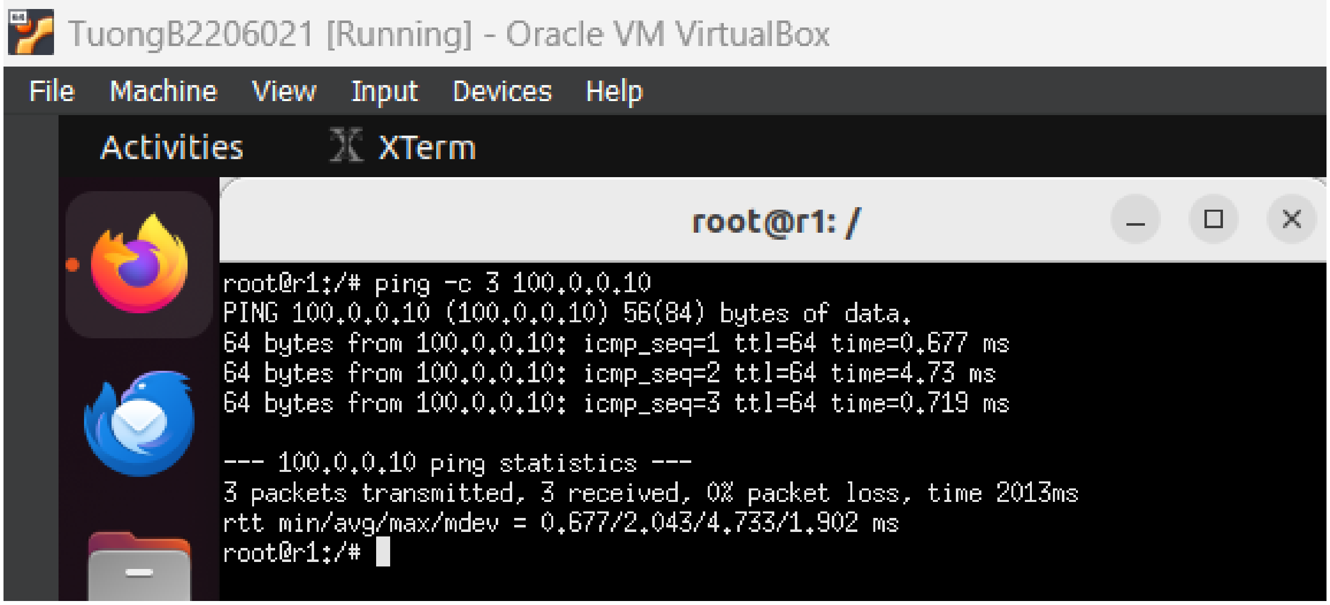
```
root@pc1:/# ping -c 2 100.0.0.10
PING 100.0.0.10 (100.0.0.10) 56(84) bytes of data.

--- 100.0.0.10 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1016ms

root@pc1:/#
```

To fix the issue, add a default route into the routing table of ***r2.startup*** file
route add -net 195.11.14.0/24 gw 100.0.0.9 dev eth1 for r2.startup file
route add -net 200.1.1.0/24 gw 100.0.0.10 dev eth1 for r1.startup file (to make sure r1 can reach the necessary subnets too)


TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
b2206021@TuongB2206021: ~/CT106H/lab1/exercises3

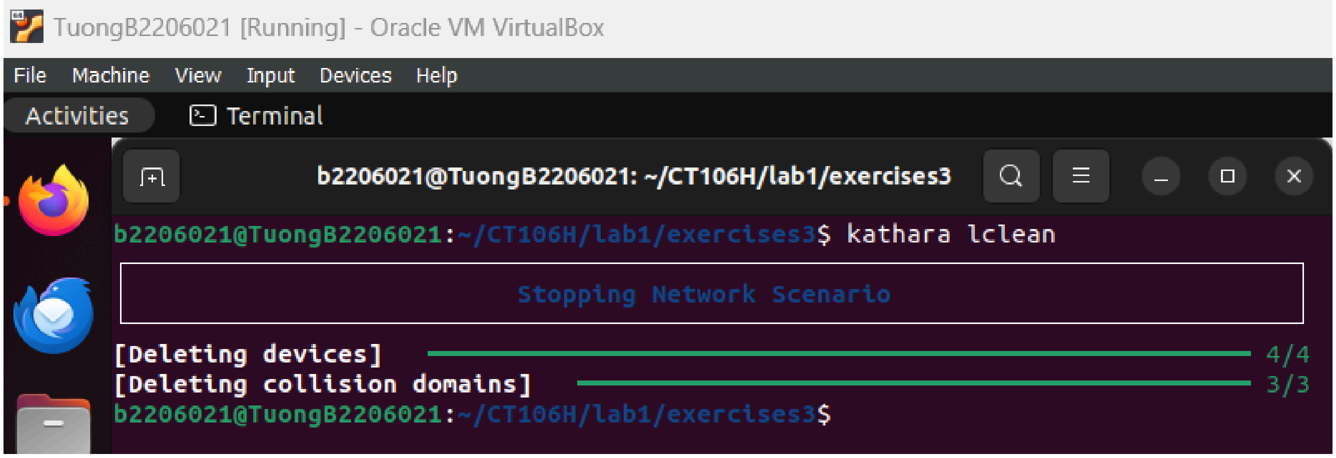
```
b2206021@TuongB2206021:~/CT106H/lab1/exercises3$ cat lab.conf
r1[0]=A
r1[1]=B

r2[0]=C
r2[1]=B

pc1[0]=A
pc2[0]=C
b2206021@TuongB2206021:~/CT106H/lab1/exercises3$ cat pc1.startup
ifconfig eth0 195.11.14.5/24 up
route add default gw 195.11.14.1
b2206021@TuongB2206021:~/CT106H/lab1/exercises3$ cat pc2.startup
ifconfig eth0 200.1.1.7/24 up
route add default gw 200.1.1.1
b2206021@TuongB2206021:~/CT106H/lab1/exercises3$ cat r1.startup
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 dev eth1
b2206021@TuongB2206021:~/CT106H/lab1/exercises3$ cat r2.startup
ifconfig eth0 200.1.1.1/24 up
ifconfig eth1 100.0.0.10/30 up
route add -net 195.11.14.0/24 gw 100.0.0.9 dev eth1
b2206021@TuongB2206021:~/CT106H/lab1/exercises3$
```

Testing connectivity


TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
XTerm
root@r1: /

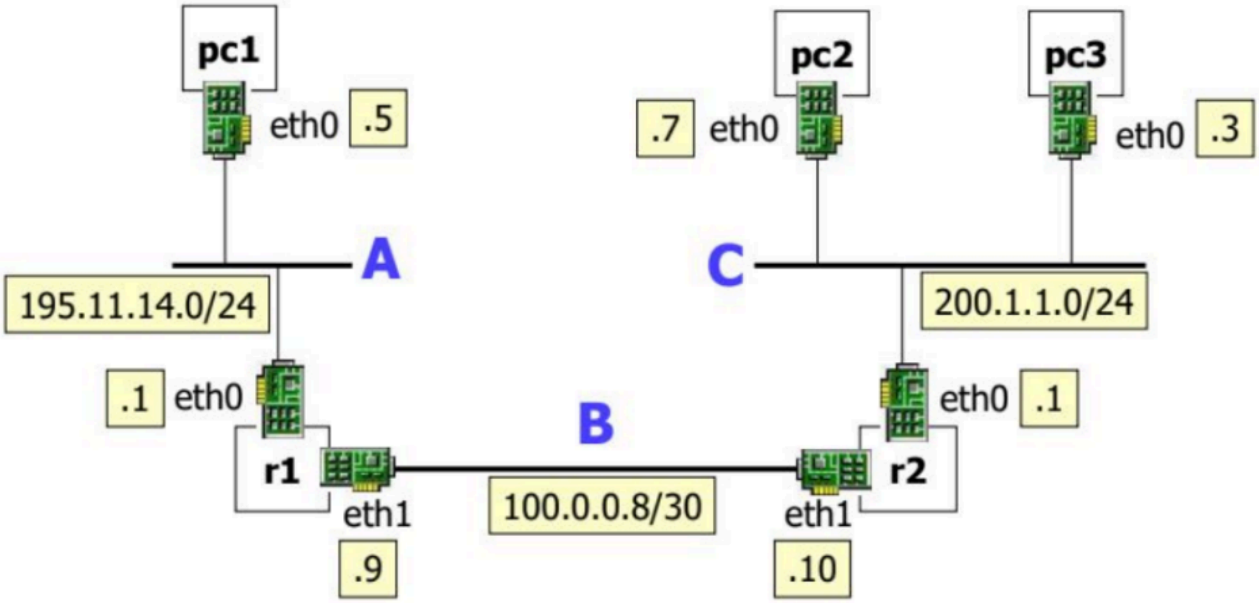
```
root@r1:/# ping -c 3 100.0.0.10
PING 100.0.0.10 (100.0.0.10) 56(84) bytes of data.
64 bytes from 100.0.0.10: icmp_seq=1 ttl=64 time=0.677 ms
64 bytes from 100.0.0.10: icmp_seq=2 ttl=64 time=4.73 ms
64 bytes from 100.0.0.10: icmp_seq=3 ttl=64 time=0.719 ms

--- 100.0.0.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2013ms
rtt min/avg/max/mdev = 0.677/2.043/4.733/1.902 ms
root@r1:/#
```

Stop the environment


TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
b2206021@TuongB2206021: ~/CT106H/lab1/exercises3

```
b2206021@TuongB2206021:~/CT106H/lab1/exercises3$ kathara lclean
                         Stopping Network Scenario
[Deleting devices]         ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4/4
[Deleting collision domains]     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3/3
b2206021@TuongB2206021:~/CT106H/lab1/exercises3$
```

Exercise 4: Study arp protocol

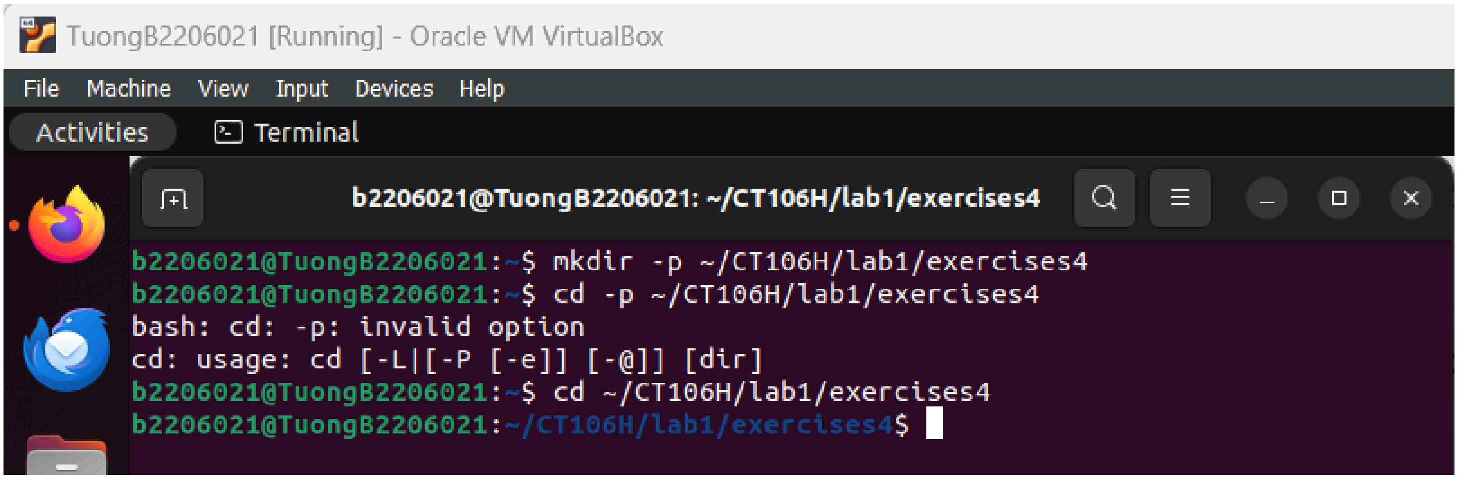


Answer:

Create exercises4 directory

$ mkdir -p ~/CT106H/lab1/exercises4

$ cd ~/CT106H/lab1/exercises4


TuongB2206021 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
b2206021@TuongB2206021: ~/CT106H/lab1/exercises4

```
b2206021@TuongB2206021:~$ mkdir -p ~/CT106H/lab1/exercises4
b2206021@TuongB2206021:~$ cd -p ~/CT106H/lab1/exercises4
bash: cd: -p: invalid option
cd: usage: cd [-L|[-P [-e]] [-@]] [dir]
b2206021@TuongB2206021:~$ cd ~/CT106H/lab1/exercises4
b2206021@TuongB2206021:~/CT106H/lab1/exercises4$
```

Prepare the lab

$ gedit lab.conf

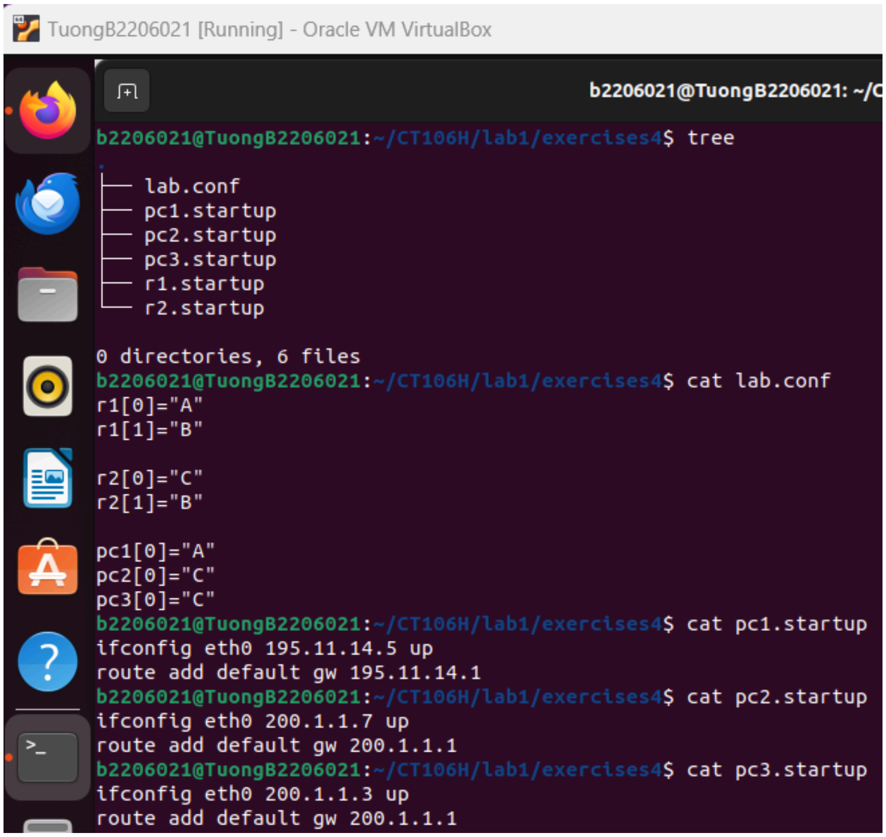
$ gedit pc1.startup

$ gedit pc2.startup

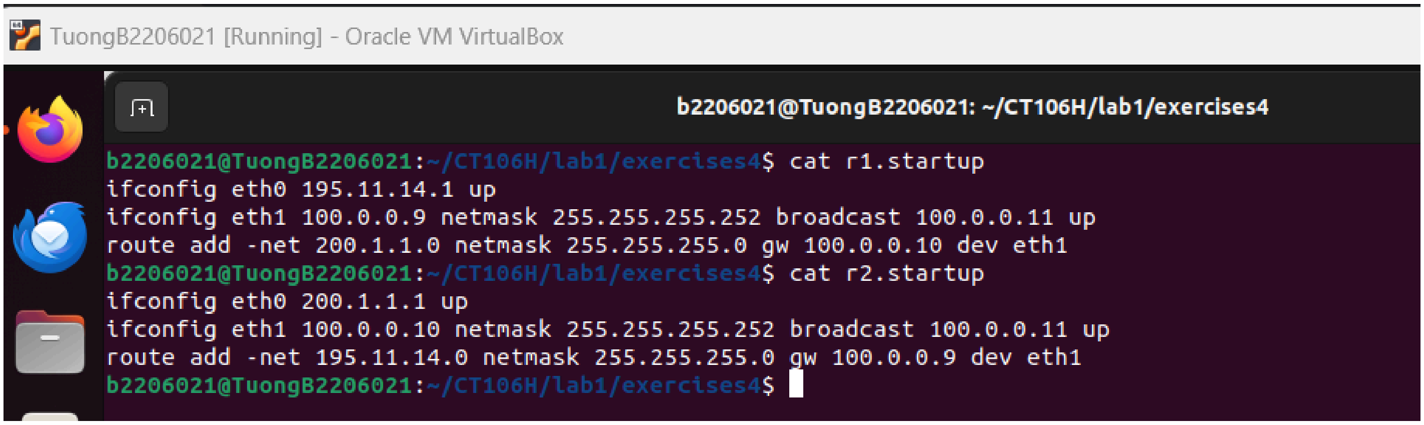
$ gedit pc2.startup

$ gedit r1.startup

$ gedit r2.startup


TuongB2206021 [Running] - Oracle VM VirtualBox
b2206021@TuongB2206021: ~/C

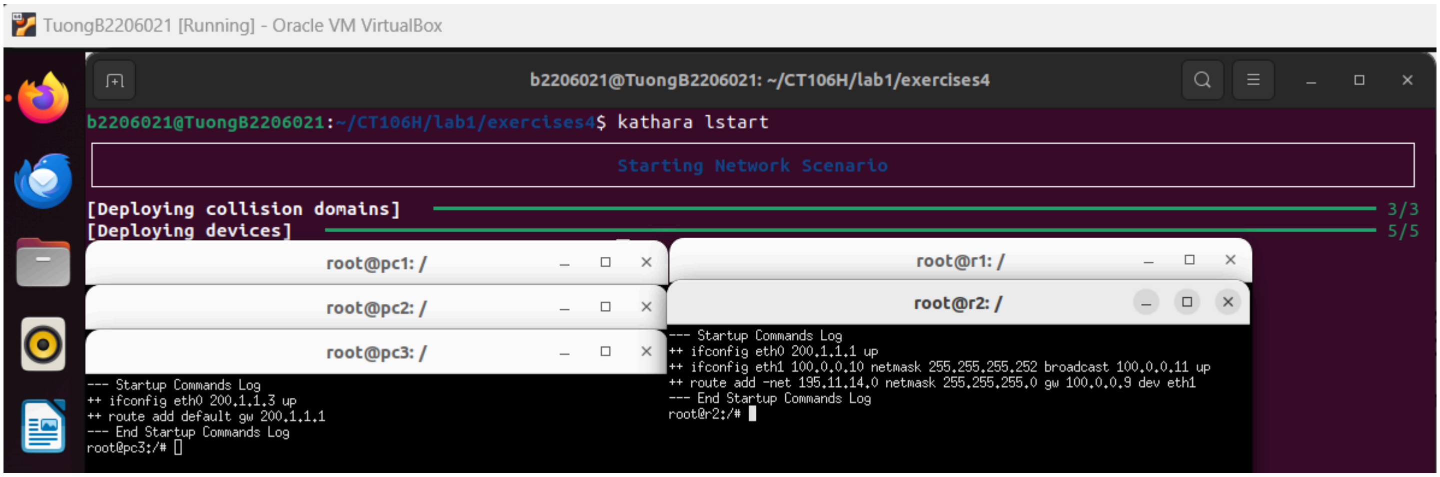
```
b2206021@TuongB2206021:~/CT106H/lab1/exercises4$ tree
.
├── lab.conf
├── pc1.startup
├── pc2.startup
├── pc3.startup
├── r1.startup
└── r2.startup

0 directories, 6 files
b2206021@TuongB2206021:~/CT106H/lab1/exercises4$ cat lab.conf
r1[0]="A"
r1[1]="B"

r2[0]="C"
r2[1]="B"

pc1[0]="A"
pc2[0]="C"
pc3[0]="C"
b2206021@TuongB2206021:~/CT106H/lab1/exercises4$ cat pc1.startup
ifconfig eth0 195.11.14.5 up
route add default gw 195.11.14.1
b2206021@TuongB2206021:~/CT106H/lab1/exercises4$ cat pc2.startup
ifconfig eth0 200.1.1.7 up
route add default gw 200.1.1.1
b2206021@TuongB2206021:~/CT106H/lab1/exercises4$ cat pc3.startup
ifconfig eth0 200.1.1.3 up
route add default gw 200.1.1.1
```


TuongB2206021 [Running] - Oracle VM VirtualBox
b2206021@TuongB2206021: ~/CT106H/lab1/exercises4

```
b2206021@TuongB2206021:~/CT106H/lab1/exercises4$ cat r1.startup
ifconfig eth0 195.11.14.1 up
ifconfig eth1 100.0.0.9 netmask 255.255.255.252 broadcast 100.0.0.11 up
route add -net 200.1.1.0 netmask 255.255.255.0 gw 100.0.0.10 dev eth1
b2206021@TuongB2206021:~/CT106H/lab1/exercises4$ cat r2.startup
ifconfig eth0 200.1.1.1 up
ifconfig eth1 100.0.0.10 netmask 255.255.255.252 broadcast 100.0.0.11 up
route add -net 195.11.14.0 netmask 255.255.255.0 gw 100.0.0.9 dev eth1
b2206021@TuongB2206021:~/CT106H/lab1/exercises4$
```

Start the lab

$ kathara lstart



TuongB2206021 [Running] - Oracle VM VirtualBox
b2206021@TuongB2206021: ~/CT106H/lab1/exercises4
root@pc1: /
root@pc2: /
root@pc3: /
root@r1: /
root@r2: /

```
b2206021@TuongB2206021:~/CT106H/lab1/exercises4$ kathara lstart
                         Starting Network Scenario
[Deploying collision domains] ──────────────────────────────── 3/3
[Deploying devices] ─────────────────────────────────────────── 5/5
```

```
--- Startup Commands Log
++ ifconfig eth0 200.1.1.3 up
++ route add default gw 200.1.1.1
--- End Startup Commands Log
root@pc3:/# 
```

```
--- Startup Commands Log
++ ifconfig eth0 200.1.1.1 up
++ ifconfig eth1 100.0.0.10 netmask 255.255.255.252 broadcast 100.0.0.11 up
++ route add -net 195.11.14.0 netmask 255.255.255.0 gw 100.0.0.9 dev eth1
--- End Startup Commands Log
root@r2:/# 
```

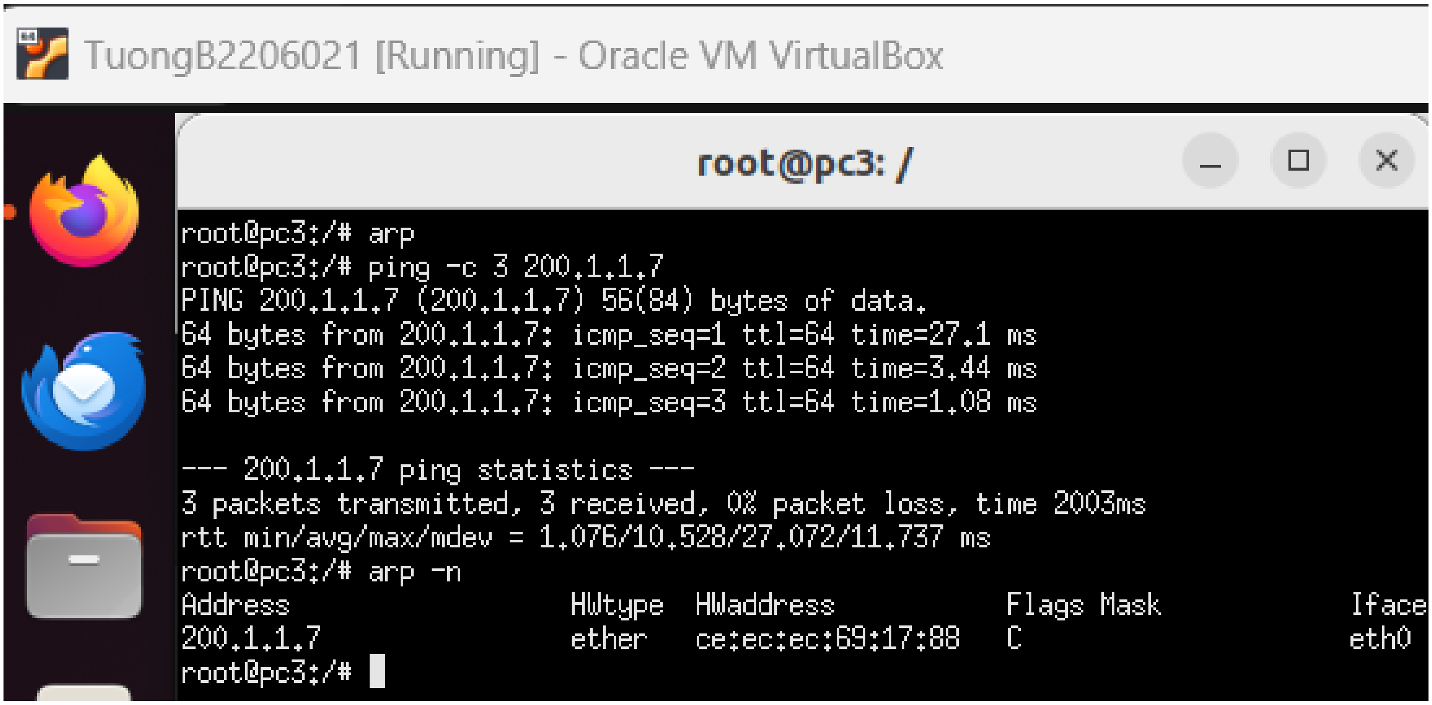
Inspecting the arp cache (local traffic)

pc2 ping pc3 (local network) ⇒ Address resolution results are stored in the arp cache

# arp *(check the current ARP cache)*

# ping -c 3 200.1.1.7 **(***After the ping, the resolved MAC address for PC3 will be stored in the ARP cache)*

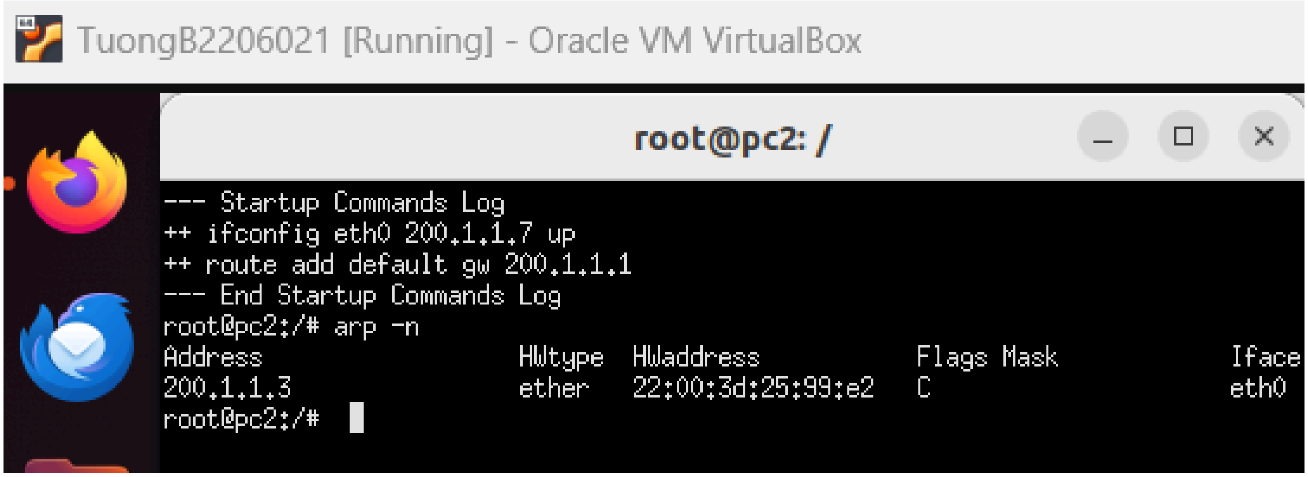
# arp -n *(The "-n" option makes arp display numeric addresses)*



```
root@pc3:/# arp
root@pc3:/# ping -c 3 200.1.1.7
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=27.1 ms
64 bytes from 200.1.1.7: icmp_seq=2 ttl=64 time=3.44 ms
64 bytes from 200.1.1.7: icmp_seq=3 ttl=64 time=1.08 ms

--- 200.1.1.7 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 1.076/10.528/27.072/11.737 ms
root@pc3:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
200.1.1.7                ether   ce:ec:ec:69:17:88   C                     eth0
root@pc3:/# 
```

Communications are usually bi-directional. The receiver of the arp request learns the mac address of the other party, to avoid a new arp in opposite direction

# arp -n *(on pc2)*



```
--- Startup Commands Log
++ ifconfig eth0 200.1.1.7 up
++ route add default gw 200.1.1.1
--- End Startup Commands Log
root@pc2:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
200.1.1.3                ether   22:00:3d:25:99:e2   C                     eth0
root@pc2:/#
```

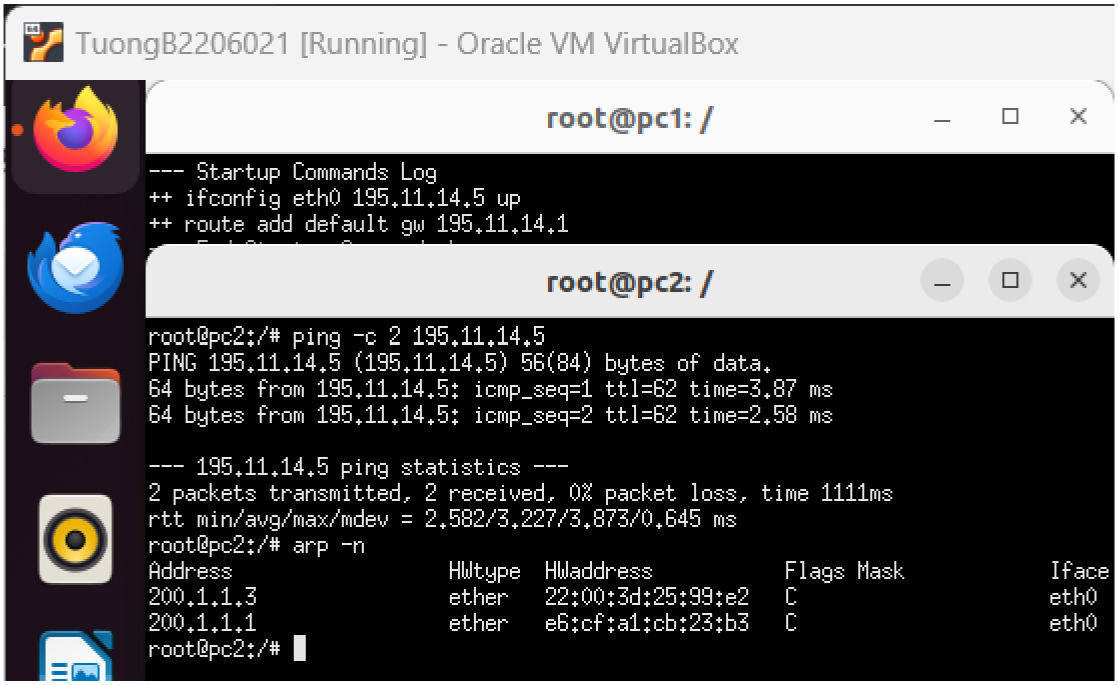
Inspecting the arp cache (non local traffic)

pc2 ping pc1 (non local network)

⇒ *When ip traffic is addressed outside the local network, the sender needs the mac address of the router, arp requests can get replies only within the local network*

# ping -c 2 195.11.14.5

# arp -n



```
--- Startup Commands Log
++ ifconfig eth0 195.11.14.5 up
++ route add default gw 195.11.14.1
```

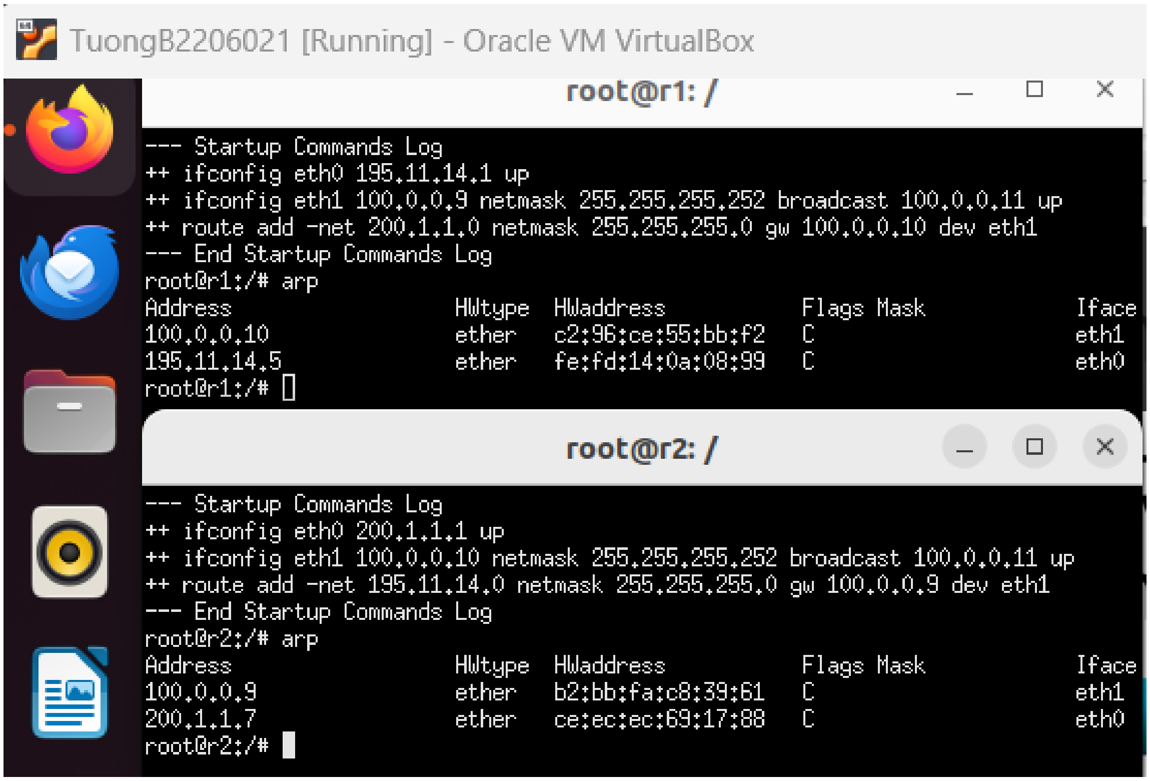
```
root@pc2:/# ping -c 2 195.11.14.5
PING 195.11.14.5 (195.11.14.5) 56(84) bytes of data.
64 bytes from 195.11.14.5: icmp_seq=1 ttl=62 time=3.87 ms
64 bytes from 195.11.14.5: icmp_seq=2 ttl=62 time=2.58 ms

--- 195.11.14.5 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1111ms
rtt min/avg/max/mdev = 2.582/3.227/3.873/0.645 ms
root@pc2:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
200.1.1.3                ether   22:00:3d:25:99:e2   C                     eth0
200.1.1.1                ether   e6:cf:a1:cb:23:b3   C                     eth0
root@pc2:/#
```

Inspect arp of the router

- Routers perform arp too (hence have arp caches) anytime they have to send ip packets on an ethernet LAN

# arp *(r1)*

# arp *(r2)*



```
--- Startup Commands Log
++ ifconfig eth0 195.11.14.1 up
++ ifconfig eth1 100.0.0.9 netmask 255.255.255.252 broadcast 100.0.0.11 up
++ route add -net 200.1.1.0 netmask 255.255.255.0 gw 100.0.0.10 dev eth1
--- End Startup Commands Log
root@r1:/# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
100.0.0.10               ether   c2:96:ce:55:bb:f2   C                     eth1
195.11.14.5              ether   fe:fd:14:0a:08:99   C                     eth0
root@r1:/#
```

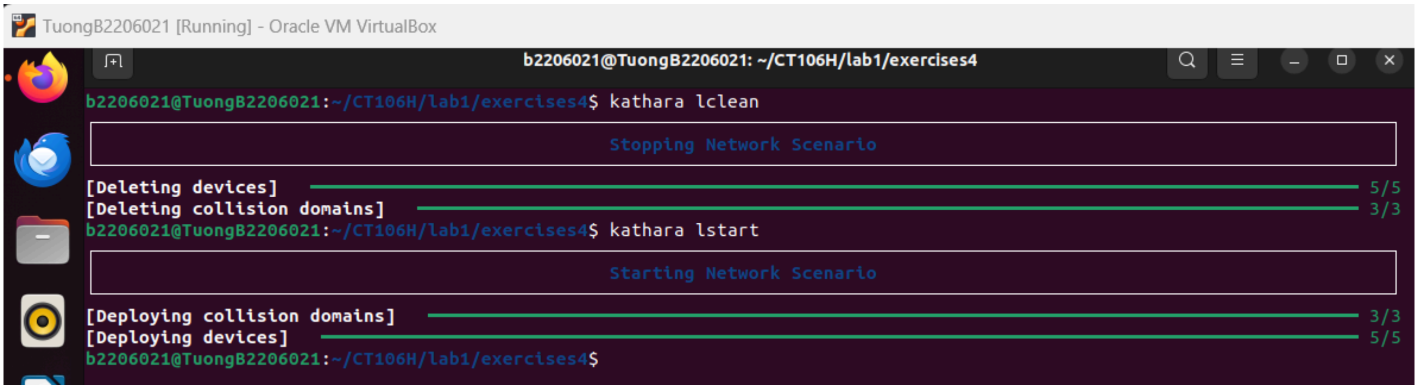
```
--- Startup Commands Log
++ ifconfig eth0 200.1.1.1 up
++ ifconfig eth1 100.0.0.10 netmask 255.255.255.252 broadcast 100.0.0.11 up
++ route add -net 195.11.14.0 netmask 255.255.255.0 gw 100.0.0.9 dev eth1
--- End Startup Commands Log
root@r2:/# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
100.0.0.9                ether   b2:bb:fa:c8:39:61   C                     eth1
200.1.1.7                ether   ce:ec:ec:69:17:88   C                     eth0
root@r2:/#
```

Sniff arp traffic

Restart the lab to clear arp caches

$ kathara lclean

$ kathara lstart

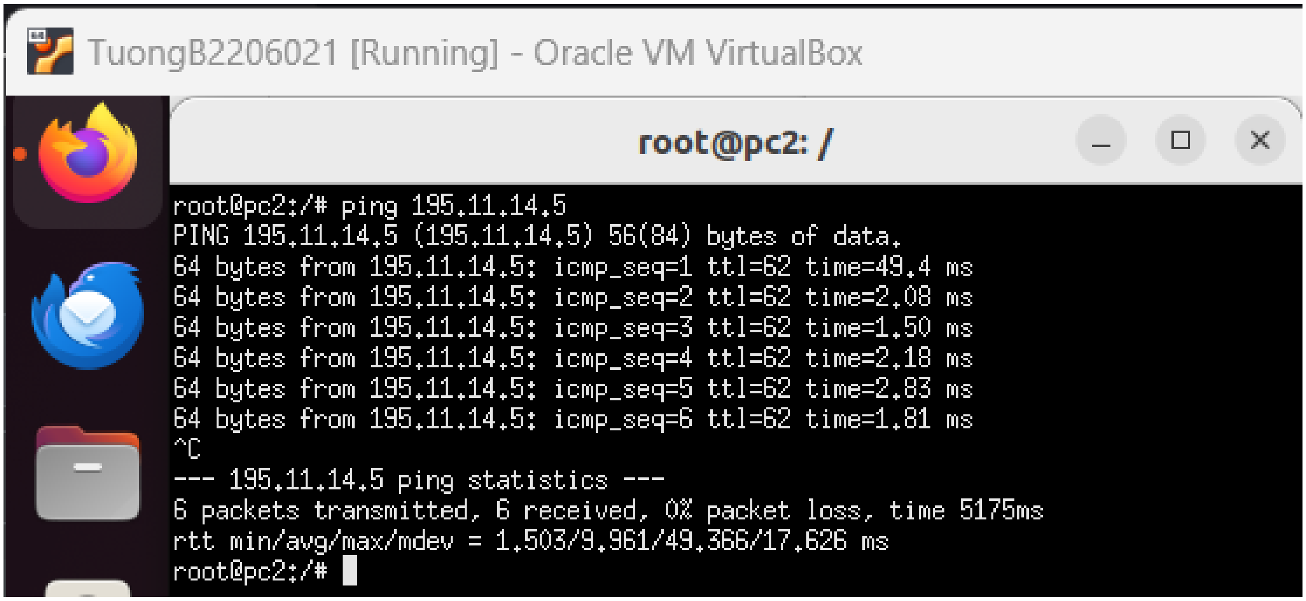


TuongB2206021 [Running] - Oracle VM VirtualBox
b2206021@TuongB2206021: ~/CT106H/lab1/exercises4

```
b2206021@TuongB2206021:~/CT106H/lab1/exercises4$ kathara lclean
                         Stopping Network Scenario
[Deleting devices]            ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5/5
[Deleting collision domains]  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3/3
b2206021@TuongB2206021:~/CT106H/lab1/exercises4$ kathara lstart
                         Starting Network Scenario
[Deploying collision domains] ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3/3
[Deploying devices]           ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5/5
b2206021@TuongB2206021:~/CT106H/lab1/exercises4$
```

Start sniffing

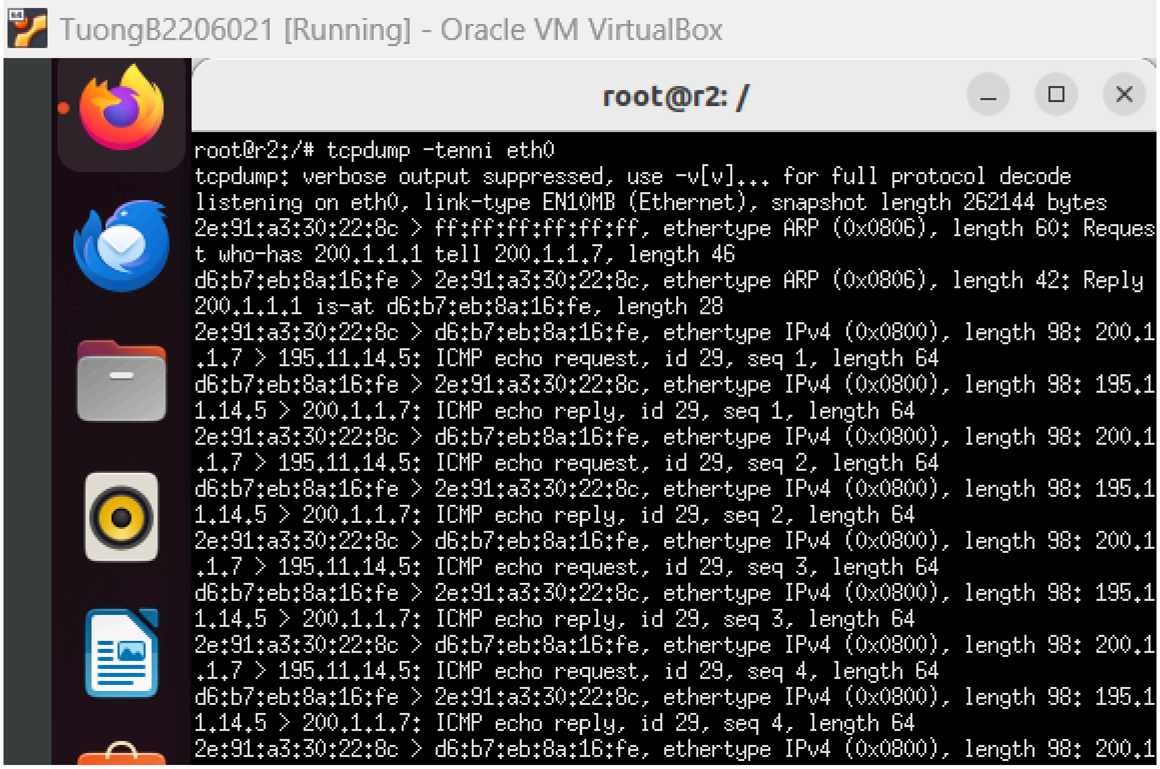
# ping 195.11.14.5 *(pc2 ping pc1)*



```
root@pc2:/# 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=49.4 ms
64 bytes from 195.11.14.5: icmp_seq=2 ttl=62 time=2.08 ms
64 bytes from 195.11.14.5: icmp_seq=3 ttl=62 time=1.50 ms
64 bytes from 195.11.14.5: icmp_seq=4 ttl=62 time=2.18 ms
64 bytes from 195.11.14.5: icmp_seq=5 ttl=62 time=2.83 ms
64 bytes from 195.11.14.5: icmp_seq=6 ttl=62 time=1.81 ms
^C
--- 195.11.14.5 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5175ms
rtt min/avg/max/mdev = 1.503/9.961/49.366/17.626 ms
root@pc2:/# 
```

On collision domain C

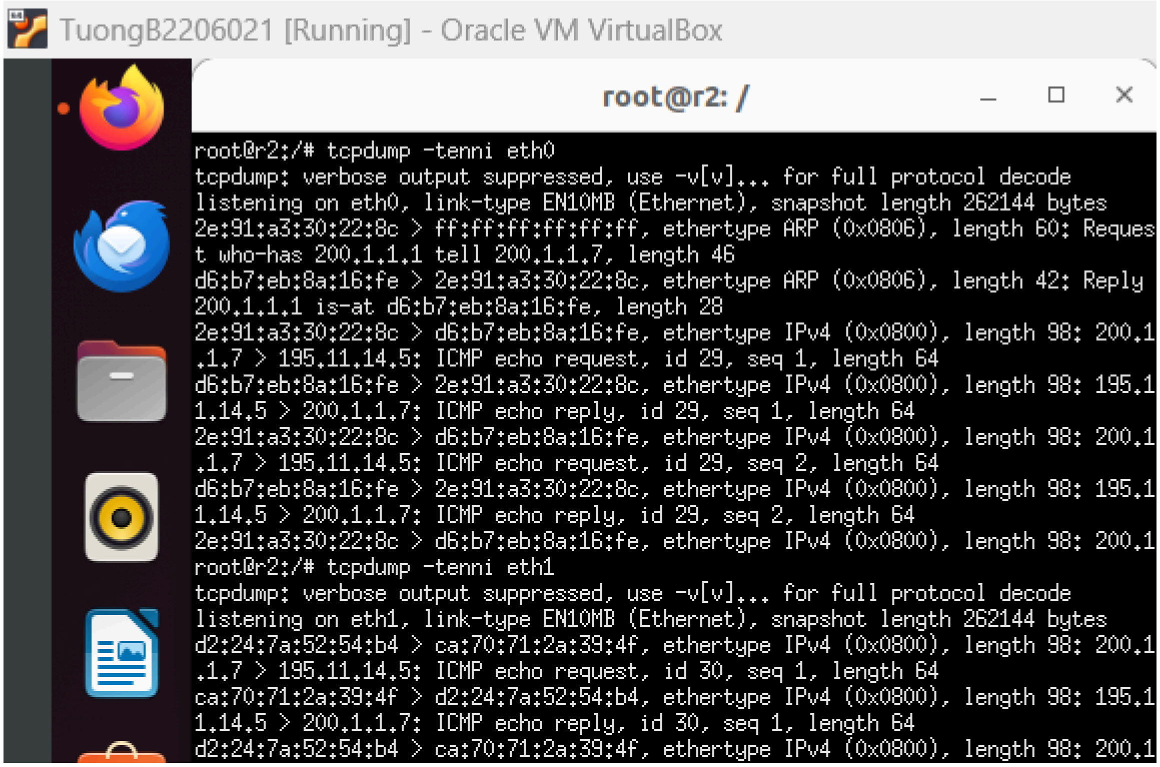
# tcpdump -tenni eth0



```
root@r2:/# tcpdump -tenni eth0
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
2e:91:a3:30:22:8c > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 60: Reques
t who-has 200.1.1.1 tell 200.1.1.7, length 46
d6:b7:eb:8a:16:fe > 2e:91:a3:30:22:8c, ethertype ARP (0x0806), length 42: Reply
200.1.1.1 is-at d6:b7:eb:8a:16:fe, length 28
2e:91:a3:30:22:8c > d6:b7:eb:8a:16:fe, ethertype IPv4 (0x0800), length 98: 200.1
.1.7 > 195.11.14.5: ICMP echo request, id 29, seq 1, length 64
d6:b7:eb:8a:16:fe > 2e:91:a3:30:22:8c, ethertype IPv4 (0x0800), length 98: 195.1
1.14.5 > 200.1.1.7: ICMP echo reply, id 29, seq 1, length 64
2e:91:a3:30:22:8c > d6:b7:eb:8a:16:fe, ethertype IPv4 (0x0800), length 98: 200.1
.1.7 > 195.11.14.5: ICMP echo request, id 29, seq 2, length 64
d6:b7:eb:8a:16:fe > 2e:91:a3:30:22:8c, ethertype IPv4 (0x0800), length 98: 195.1
1.14.5 > 200.1.1.7: ICMP echo reply, id 29, seq 2, length 64
2e:91:a3:30:22:8c > d6:b7:eb:8a:16:fe, ethertype IPv4 (0x0800), length 98: 200.1
.1.7 > 195.11.14.5: ICMP echo request, id 29, seq 3, length 64
d6:b7:eb:8a:16:fe > 2e:91:a3:30:22:8c, ethertype IPv4 (0x0800), length 98: 195.1
1.14.5 > 200.1.1.7: ICMP echo reply, id 29, seq 3, length 64
2e:91:a3:30:22:8c > d6:b7:eb:8a:16:fe, ethertype IPv4 (0x0800), length 98: 200.1
.1.7 > 195.11.14.5: ICMP echo request, id 29, seq 4, length 64
d6:b7:eb:8a:16:fe > 2e:91:a3:30:22:8c, ethertype IPv4 (0x0800), length 98: 195.1
1.14.5 > 200.1.1.7: ICMP echo reply, id 29, seq 4, length 64
2e:91:a3:30:22:8c > d6:b7:eb:8a:16:fe, ethertype IPv4 (0x0800), length 98: 200.1
```

On collision domain B

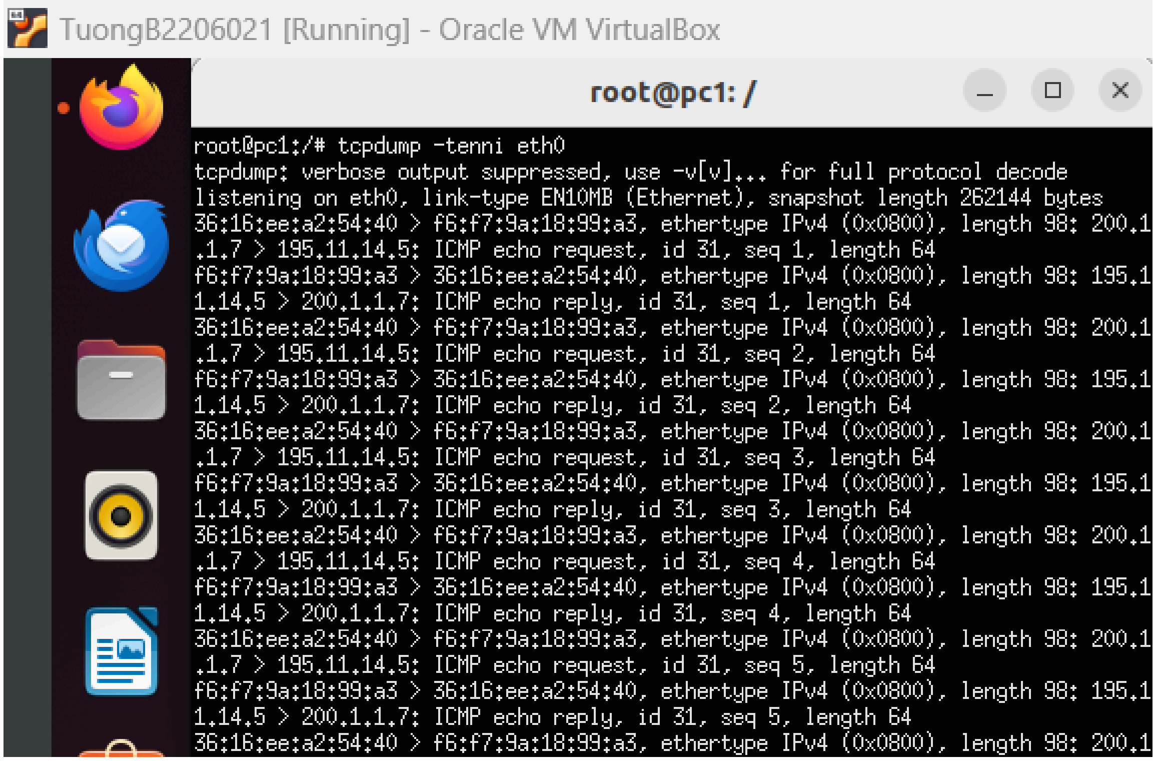
# tcpdump -tenni eth0



```
root@r2:/# tcpdump -tenni eth0
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
2e:91:a3:30:22:8c > ff:ff:ff:ff:ff:ff, ethertype ARP (0x0806), length 60: Reques
t who-has 200.1.1.1 tell 200.1.1.7, length 46
d6:b7:eb:8a:16:fe > 2e:91:a3:30:22:8c, ethertype ARP (0x0806), length 42: Reply
200.1.1.1 is-at d6:b7:eb:8a:16:fe, length 28
2e:91:a3:30:22:8c > d6:b7:eb:8a:16:fe, ethertype IPv4 (0x0800), length 98: 200.1
.1.7 > 195.11.14.5: ICMP echo request, id 29, seq 1, length 64
d6:b7:eb:8a:16:fe > 2e:91:a3:30:22:8c, ethertype IPv4 (0x0800), length 98: 195.1
1.14.5 > 200.1.1.7: ICMP echo reply, id 29, seq 1, length 64
2e:91:a3:30:22:8c > d6:b7:eb:8a:16:fe, ethertype IPv4 (0x0800), length 98: 200.1
.1.7 > 195.11.14.5: ICMP echo request, id 29, seq 2, length 64
d6:b7:eb:8a:16:fe > 2e:91:a3:30:22:8c, ethertype IPv4 (0x0800), length 98: 195.1
1.14.5 > 200.1.1.7: ICMP echo reply, id 29, seq 2, length 64
2e:91:a3:30:22:8c > d6:b7:eb:8a:16:fe, ethertype IPv4 (0x0800), length 98: 200.1
root@r2:/# tcpdump -tenni eth1
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), snapshot length 262144 bytes
d2:24:7a:52:54:b4 > ca:70:71:2a:39:4f, ethertype IPv4 (0x0800), length 98: 200.1
.1.7 > 195.11.14.5: ICMP echo request, id 30, seq 1, length 64
ca:70:71:2a:39:4f > d2:24:7a:52:54:b4, ethertype IPv4 (0x0800), length 98: 195.1
1.14.5 > 200.1.1.7: ICMP echo reply, id 30, seq 1, length 64
d2:24:7a:52:54:b4 > ca:70:71:2a:39:4f, ethertype IPv4 (0x0800), length 98: 200.1
```

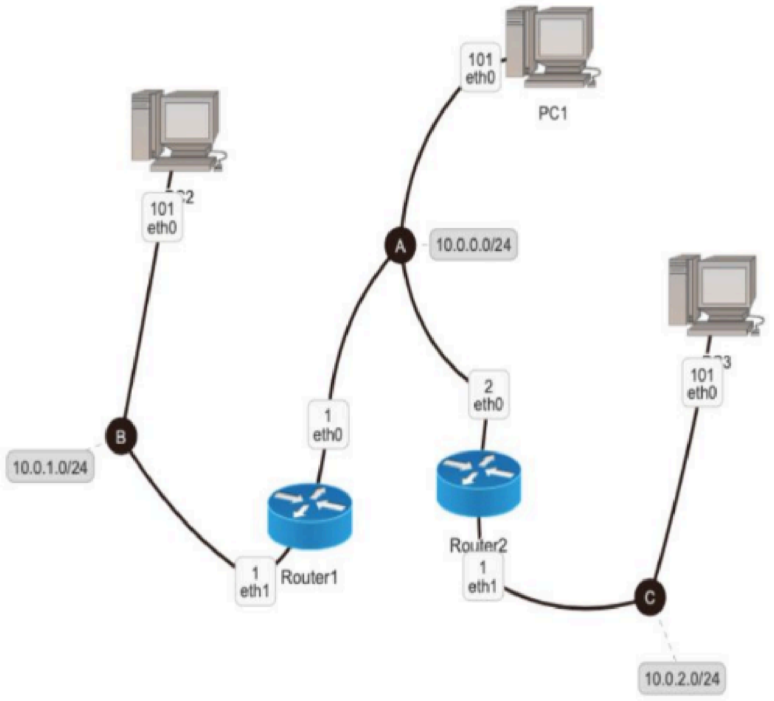
On collision domain A

# tcpdump -tenni eth0



```
root@pc1:/# tcpdump -tenni eth0
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
36:16:ee:a2:54:40 > f6:f7:9a:18:99:a3, ethertype IPv4 (0x0800), length 98: 200.1
.1.7 > 195.11.14.5: ICMP echo request, id 31, seq 1, length 64
f6:f7:9a:18:99:a3 > 36:16:ee:a2:54:40, ethertype IPv4 (0x0800), length 98: 195.1
1.14.5 > 200.1.1.7: ICMP echo reply, id 31, seq 1, length 64
36:16:ee:a2:54:40 > f6:f7:9a:18:99:a3, ethertype IPv4 (0x0800), length 98: 200.1
.1.7 > 195.11.14.5: ICMP echo request, id 31, seq 2, length 64
f6:f7:9a:18:99:a3 > 36:16:ee:a2:54:40, ethertype IPv4 (0x0800), length 98: 195.1
1.14.5 > 200.1.1.7: ICMP echo reply, id 31, seq 2, length 64
36:16:ee:a2:54:40 > f6:f7:9a:18:99:a3, ethertype IPv4 (0x0800), length 98: 200.1
.1.7 > 195.11.14.5: ICMP echo request, id 31, seq 3, length 64
f6:f7:9a:18:99:a3 > 36:16:ee:a2:54:40, ethertype IPv4 (0x0800), length 98: 195.1
1.14.5 > 200.1.1.7: ICMP echo reply, id 31, seq 3, length 64
36:16:ee:a2:54:40 > f6:f7:9a:18:99:a3, ethertype IPv4 (0x0800), length 98: 200.1
.1.7 > 195.11.14.5: ICMP echo request, id 31, seq 4, length 64
f6:f7:9a:18:99:a3 > 36:16:ee:a2:54:40, ethertype IPv4 (0x0800), length 98: 195.1
1.14.5 > 200.1.1.7: ICMP echo reply, id 31, seq 4, length 64
36:16:ee:a2:54:40 > f6:f7:9a:18:99:a3, ethertype IPv4 (0x0800), length 98: 200.1
.1.7 > 195.11.14.5: ICMP echo request, id 31, seq 5, length 64
f6:f7:9a:18:99:a3 > 36:16:ee:a2:54:40, ethertype IPv4 (0x0800), length 98: 195.1
1.14.5 > 200.1.1.7: ICMP echo reply, id 31, seq 5, length 64
36:16:ee:a2:54:40 > f6:f7:9a:18:99:a3, ethertype IPv4 (0x0800), length 98: 200.1
```

Exercise 5: Construct the following network

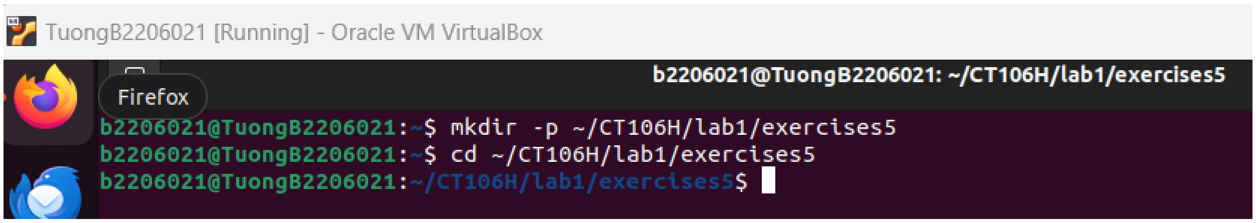


Answer:

Create exercise 2 directory

$ mkdir -p ~/CT106H/lab1/exercises3

$ cd ~/CT106H/lab1/exercises3



```
b2206021@TuongB2206021: ~/CT106H/lab1/exercises5
b2206021@TuongB2206021:~$ mkdir -p ~/CT106H/lab1/exercises5
b2206021@TuongB2206021:~$ cd ~/CT106H/lab1/exercises5
b2206021@TuongB2206021:~/CT106H/lab1/exercises5$ 
```

Prepare the lab

$ gedit lab.conf

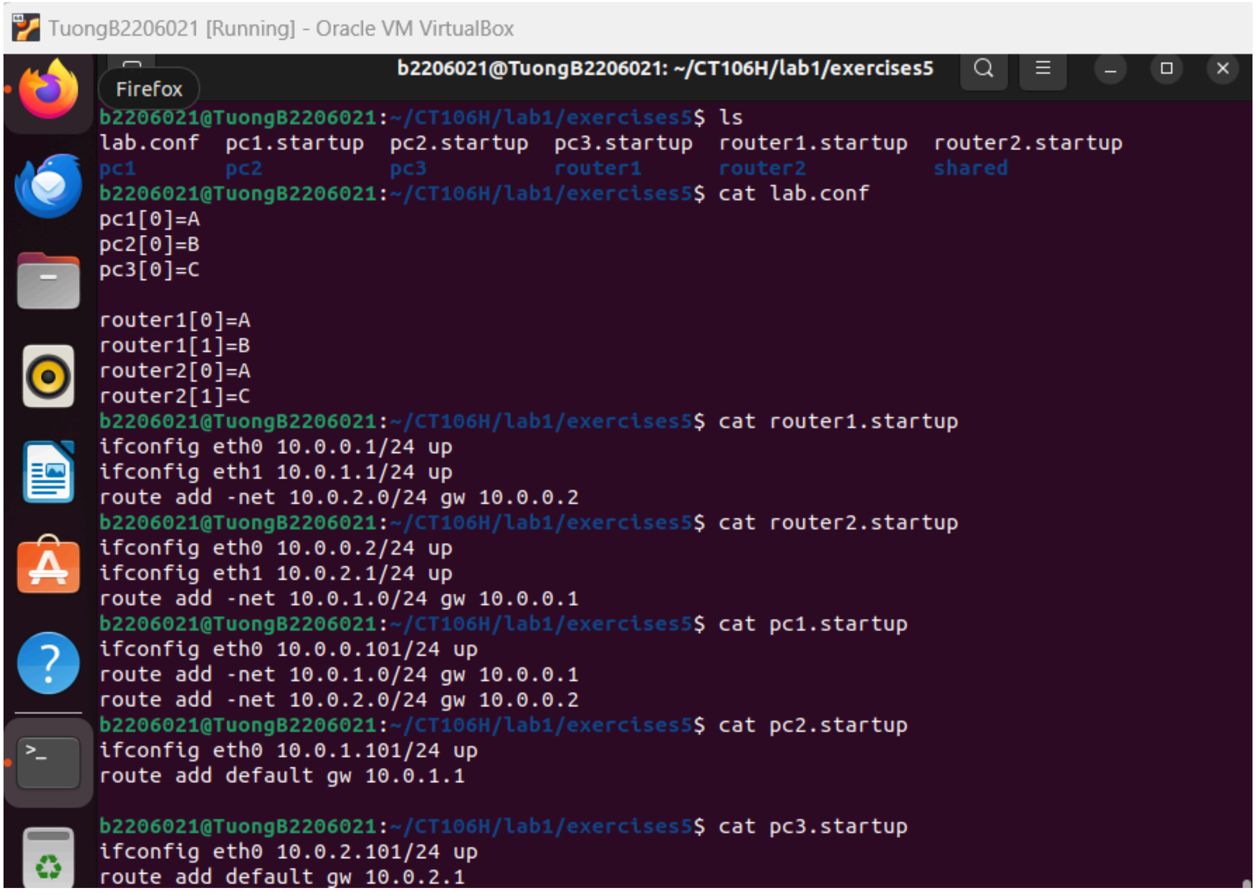
$ gedit pc1.startup

$ gedit pc2.startup

$ gedit pc3.startup

$ gedit router1.startup

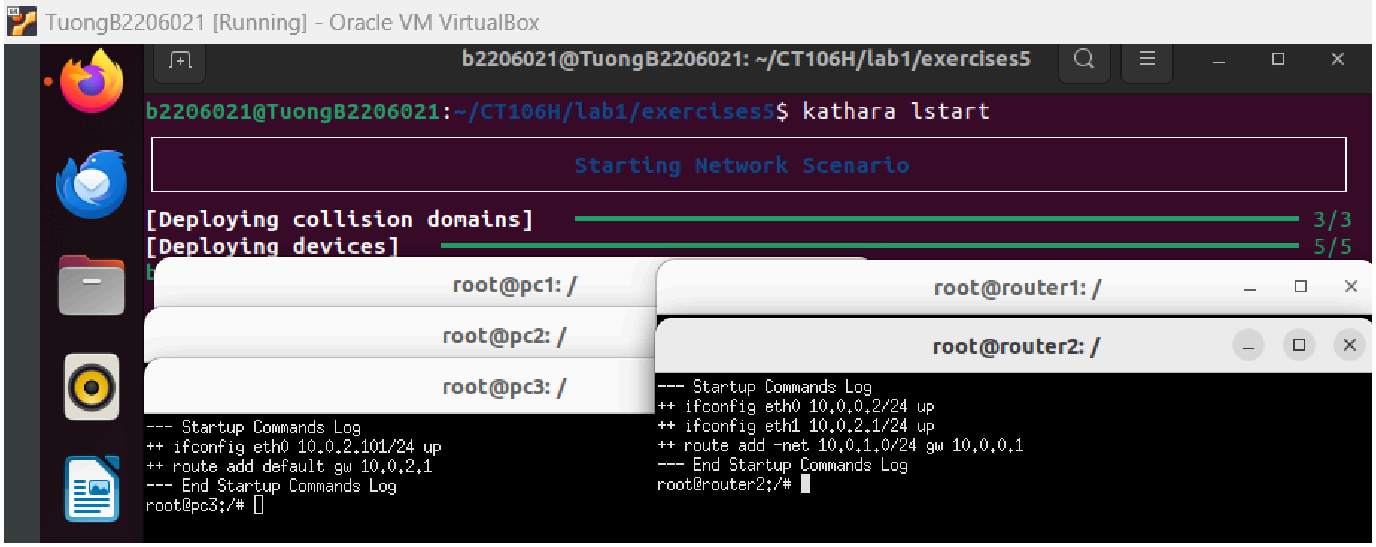
$ gedit router2.startup


TuongB2206021 [Running] - Oracle VM VirtualBox
b2206021@TuongB2206021: ~/CT106H/lab1/exercises5
Firefox

```
b2206021@TuongB2206021:~/CT106H/lab1/exercises5$ ls
lab.conf  pc1.startup  pc2.startup  pc3.startup  router1.startup  router2.startup
pc1       pc2          pc3          router1      router2          shared
b2206021@TuongB2206021:~/CT106H/lab1/exercises5$ cat lab.conf
pc1[0]=A
pc2[0]=B
pc3[0]=C

router1[0]=A
router1[1]=B
router2[0]=A
router2[1]=C
b2206021@TuongB2206021:~/CT106H/lab1/exercises5$ cat router1.startup
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
b2206021@TuongB2206021:~/CT106H/lab1/exercises5$ cat router2.startup
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
b2206021@TuongB2206021:~/CT106H/lab1/exercises5$ cat pc1.startup
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
b2206021@TuongB2206021:~/CT106H/lab1/exercises5$ cat pc2.startup
ifconfig eth0 10.0.1.101/24 up
route add default gw 10.0.1.1

b2206021@TuongB2206021:~/CT106H/lab1/exercises5$ cat pc3.startup
ifconfig eth0 10.0.2.101/24 up
route add default gw 10.0.2.1
```

Start the lab

$ kathara lstart


TuongB2206021 [Running] - Oracle VM VirtualBox
b2206021@TuongB2206021: ~/CT106H/lab1/exercises5
root@pc1: /
root@pc2: /
root@pc3: /
root@router1: /
root@router2: /

```
b2206021@TuongB2206021:~/CT106H/lab1/exercises5$ kathara lstart
                         Starting Network Scenario
[Deploying collision domains]  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3/3
[Deploying devices]  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5/5
```

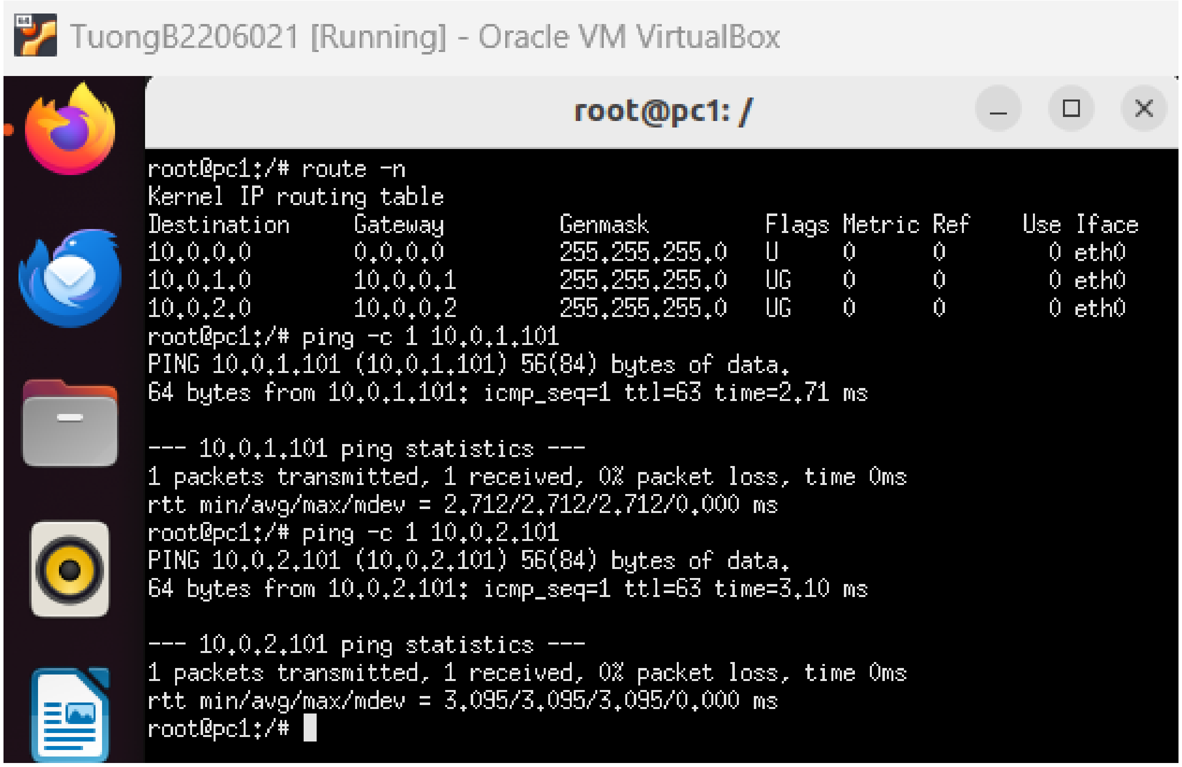
```
--- Startup Commands Log
++ ifconfig eth0 10.0.2.101/24 up
++ route add default gw 10.0.2.1
--- End Startup Commands Log
root@pc3:/# 
```

```
--- 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:/# 
```

Check the results

# route -n *(displays the routing table in a numeric format)*

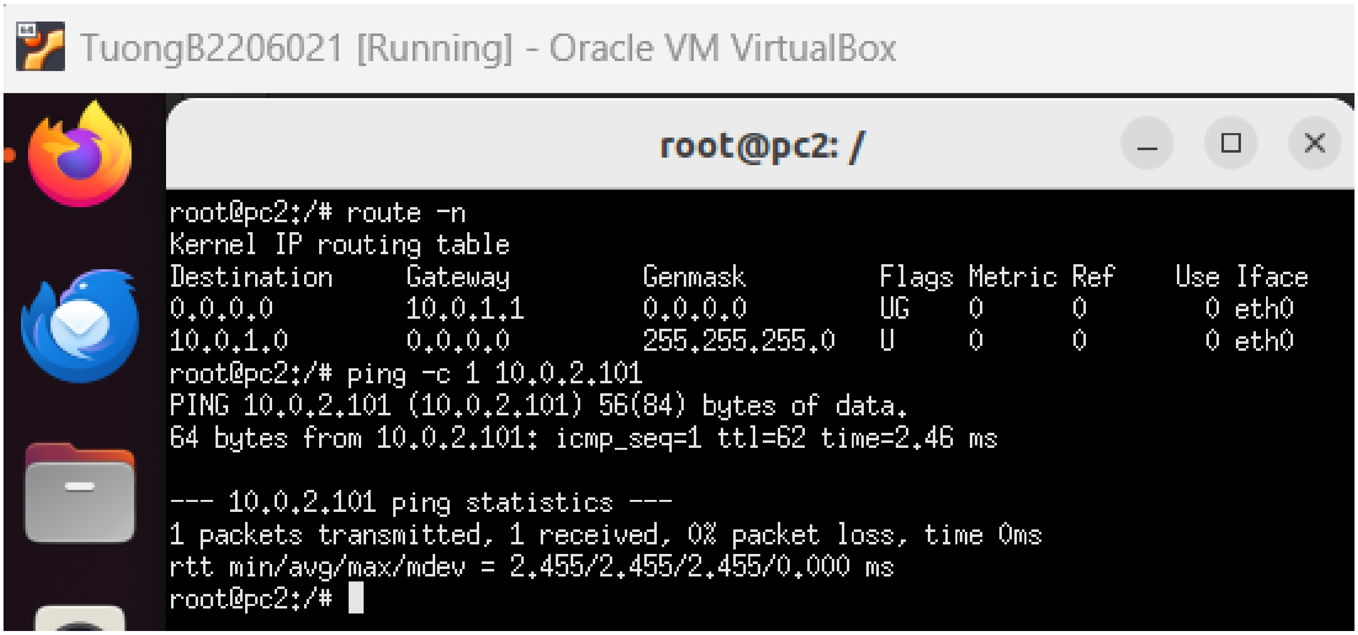
# ping -c 1 10.0.1.101 *(pc1 ping pc2)*



```
root@pc1:/# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        0.0.0.0         255.255.255.0   U     0      0        0 eth0
10.0.1.0        10.0.0.1        255.255.255.0   UG    0      0        0 eth0
10.0.2.0        10.0.0.2        255.255.255.0   UG    0      0        0 eth0
root@pc1:/# ping -c 1 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=63 time=2.71 ms

--- 10.0.1.101 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.712/2.712/2.712/0.000 ms
root@pc1:/# ping -c 1 10.0.2.101
PING 10.0.2.101 (10.0.2.101) 56(84) bytes of data.
64 bytes from 10.0.2.101: icmp_seq=1 ttl=63 time=3.10 ms

--- 10.0.2.101 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 3.095/3.095/3.095/0.000 ms
root@pc1:/#
```

# route -n

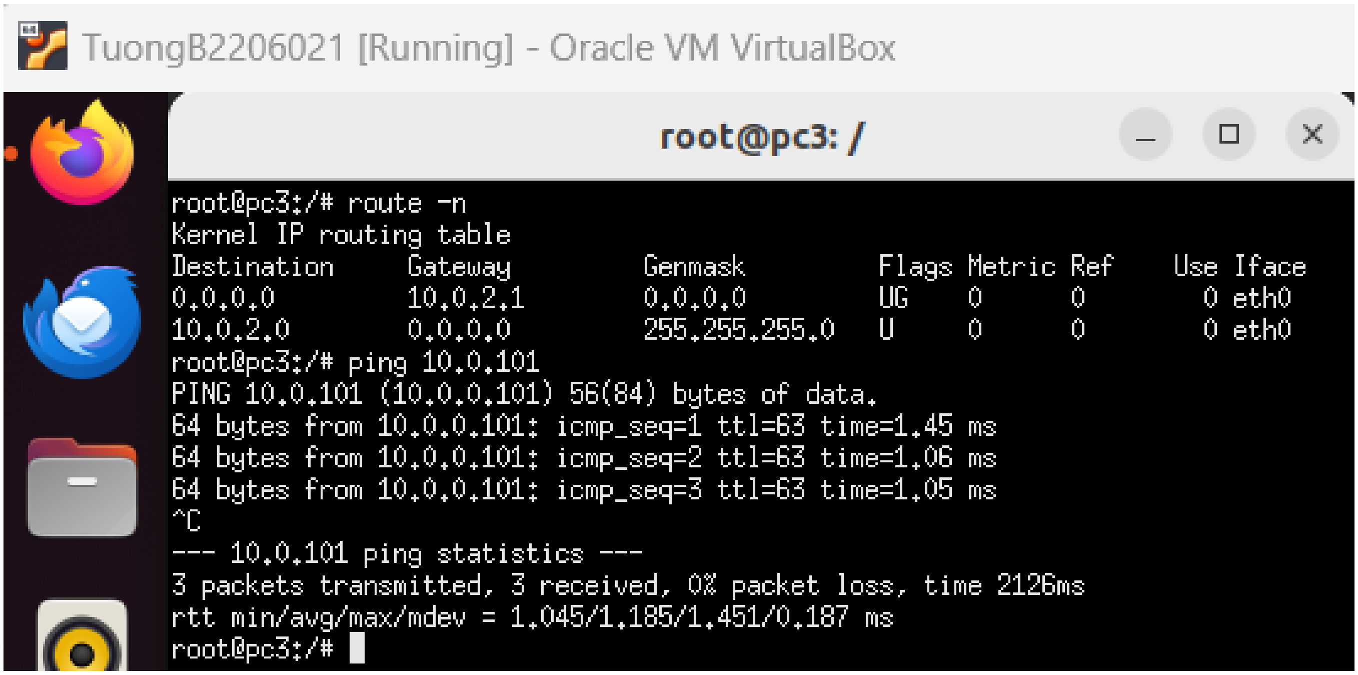
# ping -c 1 10.0.2.101 *(pc2 ping pc3)*



```
root@pc2:/# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.0.1.1        0.0.0.0         UG    0      0        0 eth0
10.0.1.0        0.0.0.0         255.255.255.0   U     0      0        0 eth0
root@pc2:/# ping -c 1 10.0.2.101
PING 10.0.2.101 (10.0.2.101) 56(84) bytes of data.
64 bytes from 10.0.2.101: icmp_seq=1 ttl=62 time=2.46 ms

--- 10.0.2.101 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.455/2.455/2.455/0.000 ms
root@pc2:/#
```

# route -n

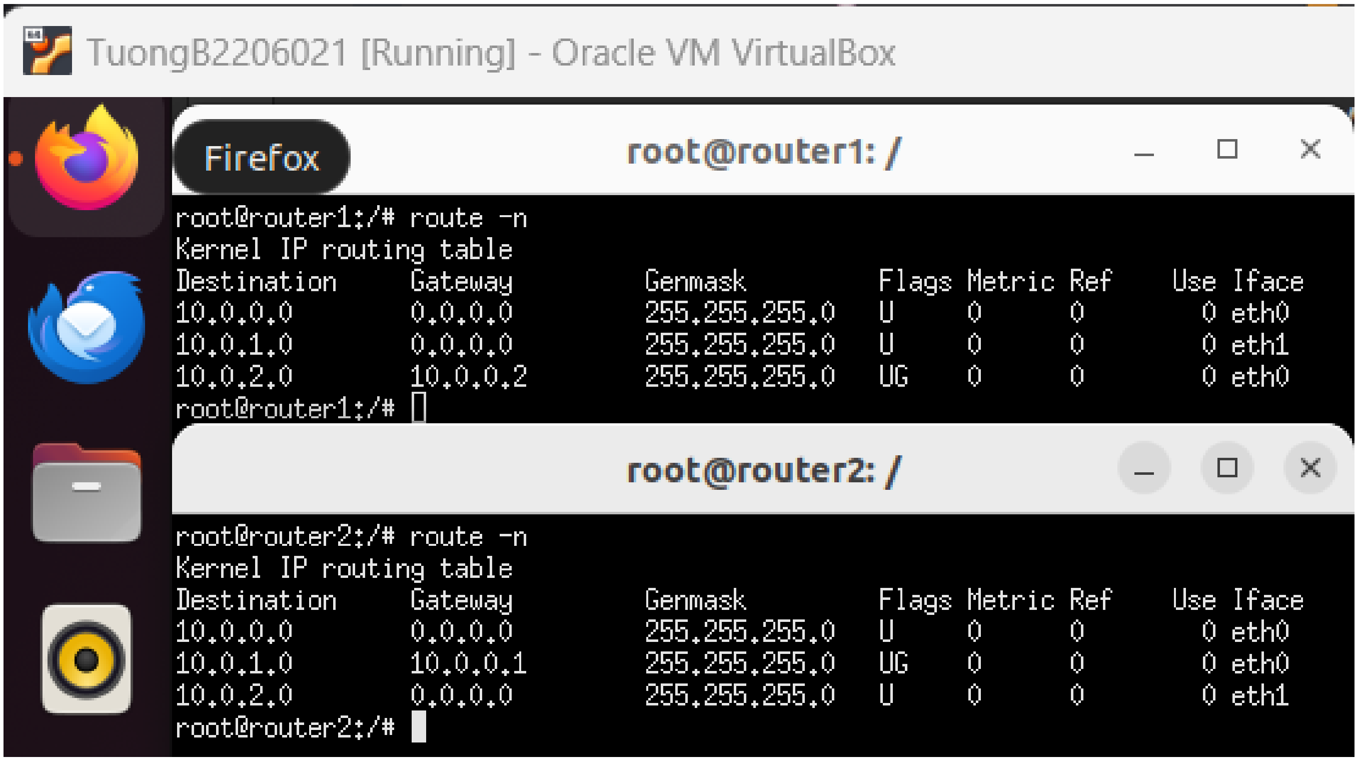
# ping -c 1 10.0.2.101 *(pc3 ping pc1)*



```
root@pc3:/# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.0.2.1        0.0.0.0         UG    0      0        0 eth0
10.0.2.0        0.0.0.0         255.255.255.0   U     0      0        0 eth0
root@pc3:/# ping 10.0.101
PING 10.0.101 (10.0.0.101) 56(84) bytes of data.
64 bytes from 10.0.0.101: icmp_seq=1 ttl=63 time=1.45 ms
64 bytes from 10.0.0.101: icmp_seq=2 ttl=63 time=1.06 ms
64 bytes from 10.0.0.101: icmp_seq=3 ttl=63 time=1.05 ms
^C
--- 10.0.101 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2126ms
rtt min/avg/max/mdev = 1.045/1.185/1.451/0.187 ms
root@pc3:/# 
```

# route -n *(r1)*

# route -n *(r2)*

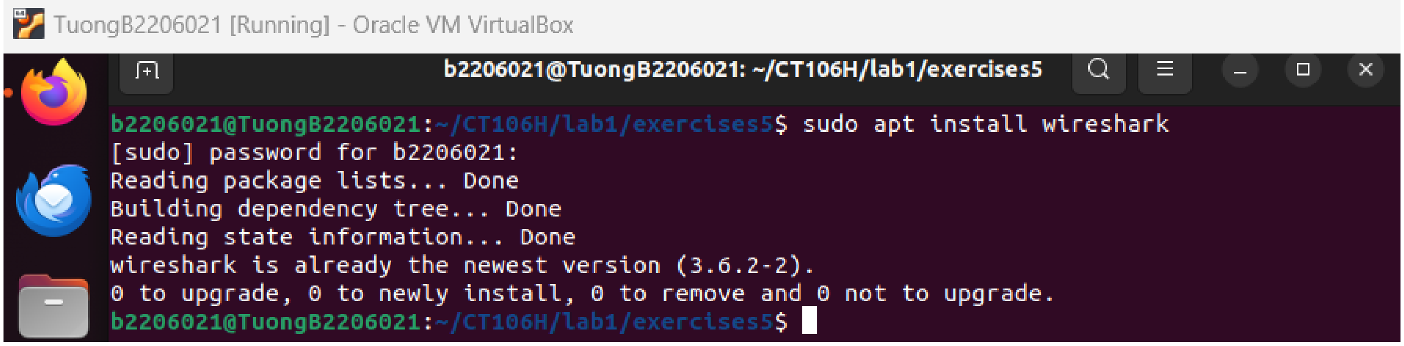


```
root@router1:/# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        0.0.0.0         255.255.255.0   U     0      0        0 eth0
10.0.1.0        0.0.0.0         255.255.255.0   U     0      0        0 eth1
10.0.2.0        10.0.0.2        255.255.255.0   UG    0      0        0 eth0
root@router1:/# 
```

```
root@router2:/# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        0.0.0.0         255.255.255.0   U     0      0        0 eth0
10.0.1.0        10.0.0.1        255.255.255.0   UG    0      0        0 eth0
10.0.2.0        0.0.0.0         255.255.255.0   U     0      0        0 eth1
root@router2:/# 
```

Install Wireshark

$ sudo apt install wireshark



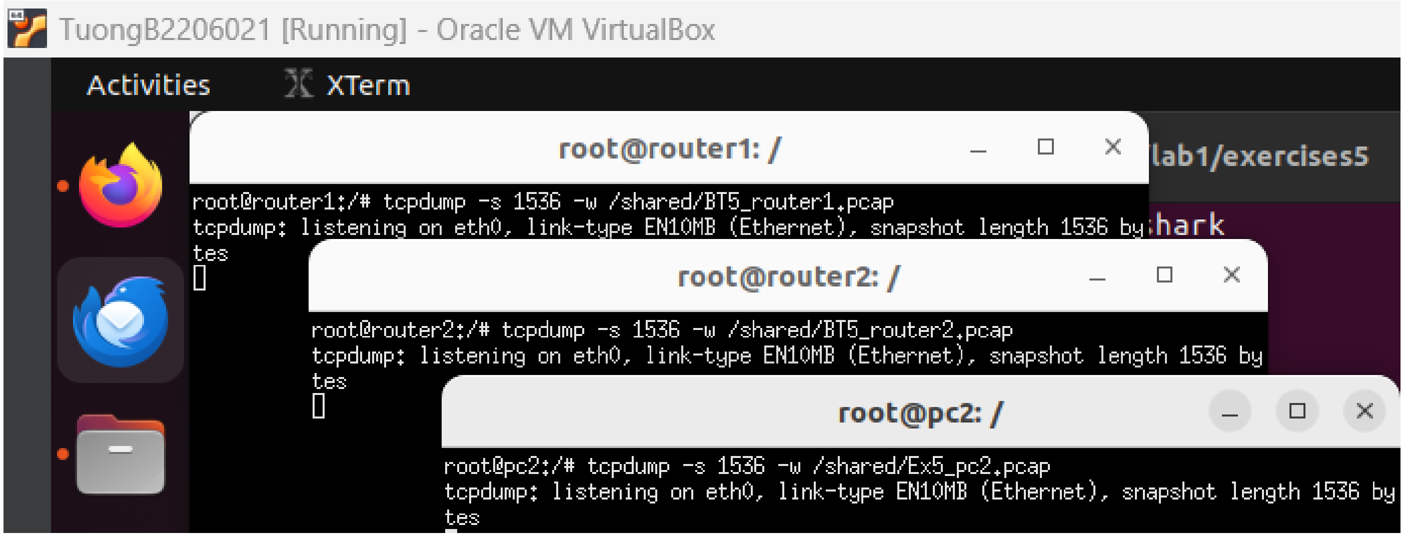
TuongB2206021 [Running] - Oracle VM VirtualBox

b2206021@TuongB2206021: ~/CT106H/lab1/exercises5

```
b2206021@TuongB2206021:~/CT106H/lab1/exercises5$ sudo apt install wireshark
[sudo] password for b2206021:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
wireshark is already the newest version (3.6.2-2).
0 to upgrade, 0 to newly install, 0 to remove and 0 not to upgrade.
b2206021@TuongB2206021:~/CT106H/lab1/exercises5$
```

All packets are save in .pcap files which are in the /shared folder

# tcpdump -s 1536 -w /shared/BT5_router1.pcap *(Store packets in* BT5_router1.pcap file)
# tcpdump -s 1536 -w /shared/BT5_router2.pcap *(Store packets in* BT5_router2.pcap file)
# tcpdump -s 1536 -w /shared/Ex5_pc2.pcap *(Store packets in* Ex5_pc2.pcap file)
(-s 1536: Sets the snapshot length to 1536 bytes)



TuongB2206021 [Running] - Oracle VM VirtualBox

root@router1: /

```
root@router1:/# tcpdump -s 1536 -w /shared/BT5_router1.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 1536 by
tes
```

root@router2: /

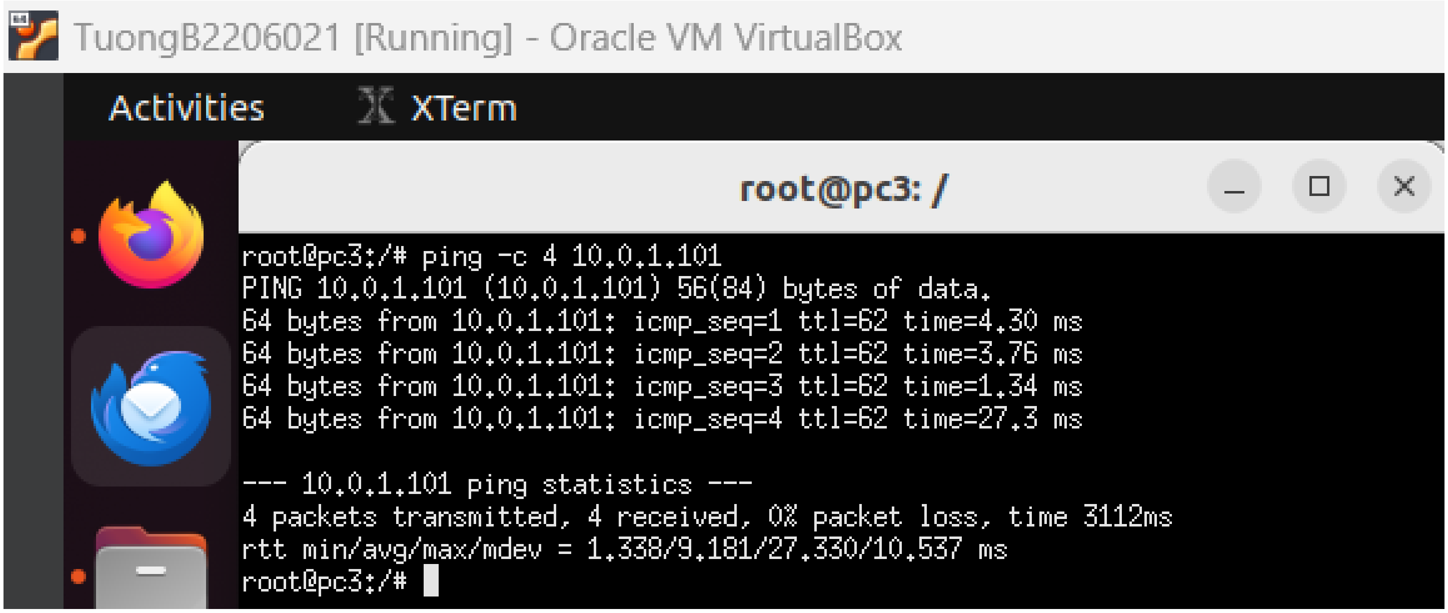
```
root@router2:/# tcpdump -s 1536 -w /shared/BT5_router2.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 1536 by
tes
```

root@pc2: /

```
root@pc2:/# tcpdump -s 1536 -w /shared/Ex5_pc2.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 1536 by
tes
```

On pc3, send packets to pc2

# ping 10.0.1.101



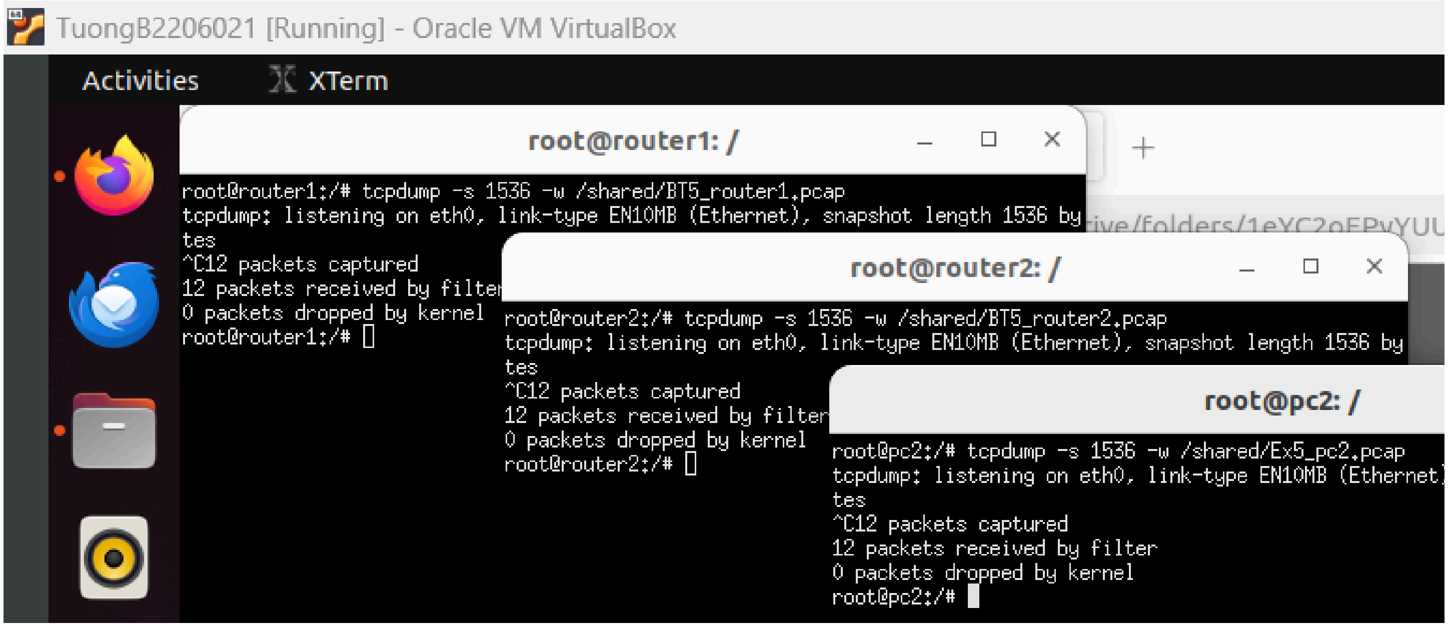
```
root@pc3:/# ping -c 4 10.0.1.101
PING 10.0.1.101 (10.0.1.101) 56(84) bytes of data.
64 bytes from 10.0.1.101: icmp_seq=1 ttl=62 time=4.30 ms
64 bytes from 10.0.1.101: icmp_seq=2 ttl=62 time=3.76 ms
64 bytes from 10.0.1.101: icmp_seq=3 ttl=62 time=1.34 ms
64 bytes from 10.0.1.101: icmp_seq=4 ttl=62 time=27.3 ms

--- 10.0.1.101 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3112ms
rtt min/avg/max/mdev = 1.338/9.181/27.330/10.537 ms
root@pc3:/#
```

Stop tcpdump on pc2, router1 & router2

# tcpdump -s 1536 -w /shared/BT5_router1.pcap (router1)

# tcpdump -s 1536 -w /shared/BT5_router2.pcap (router2)

# tcpdump -s 1536 -w /shared/Ex5_pc2.pcap (pc2)



```
root@router1:/# tcpdump -s 1536 -w /shared/BT5_router1.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 1536 by
tes
^C12 packets captured
12 packets received by filte
0 packets dropped by kernel
root@router1:/#
```

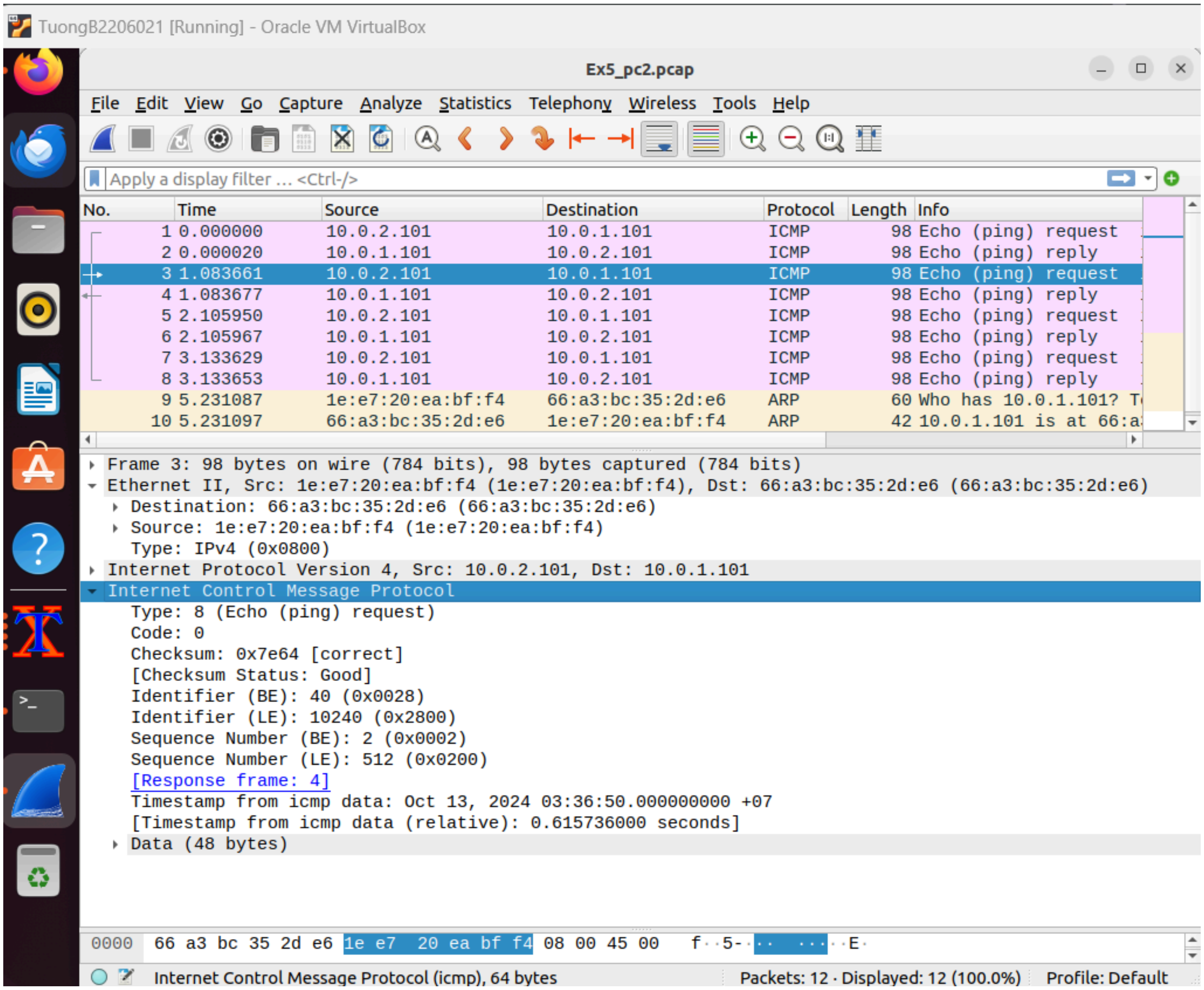
```
root@router2:/# tcpdump -s 1536 -w /shared/BT5_router2.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 1536 by
tes
^C12 packets captured
12 packets received by filter
0 packets dropped by kernel
root@router2:/#
```

```
root@pc2:/# tcpdump -s 1536 -w /shared/Ex5_pc2.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet)
tes
^C12 packets captured
12 packets received by filter
0 packets dropped by kernel
root@pc2:/#
```

On the Ubuntu, open Ex5_pc2.pcap using Wireshark, select the frame #3 and answer the following questions

- Size of frame in bytes?

Answer: ***98 bytes***


TuongB2206021 [Running] - Oracle VM VirtualBox
Ex5_pc2.pcap
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Wireless
Tools
Help
Apply a display filter … <Ctrl-/>

```
No.  Time       Source             Destination        Protocol Length Info
   1 0.000000   10.0.2.101         10.0.1.101         ICMP       98 Echo (ping) request
   2 0.000020   10.0.1.101         10.0.2.101         ICMP       98 Echo (ping) reply
   3 1.083661   10.0.2.101         10.0.1.101         ICMP       98 Echo (ping) request
   4 1.083677   10.0.1.101         10.0.2.101         ICMP       98 Echo (ping) reply
   5 2.105950   10.0.2.101         10.0.1.101         ICMP       98 Echo (ping) request
   6 2.105967   10.0.1.101         10.0.2.101         ICMP       98 Echo (ping) reply
   7 3.133629   10.0.2.101         10.0.1.101         ICMP       98 Echo (ping) request
   8 3.133653   10.0.1.101         10.0.2.101         ICMP       98 Echo (ping) reply
   9 5.231087   1e:e7:20:ea:bf:f4  66:a3:bc:35:2d:e6  ARP        60 Who has 10.0.1.101? T
  10 5.231097   66:a3:bc:35:2d:e6  1e:e7:20:ea:bf:f4  ARP        42 10.0.1.101 is at 66:a
```

```
› Frame 3: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
⌄ Ethernet II, Src: 1e:e7:20:ea:bf:f4 (1e:e7:20:ea:bf:f4), Dst: 66:a3:bc:35:2d:e6 (66:a3:bc:35:2d:e6)
  › Destination: 66:a3:bc:35:2d:e6 (66:a3:bc:35:2d:e6)
  › Source: 1e:e7:20:ea:bf:f4 (1e:e7:20:ea:bf:f4)
    Type: IPv4 (0x0800)
› Internet Protocol Version 4, Src: 10.0.2.101, Dst: 10.0.1.101
⌄ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x7e64 [correct]
    [Checksum Status: Good]
    Identifier (BE): 40 (0x0028)
    Identifier (LE): 10240 (0x2800)
    Sequence Number (BE): 2 (0x0002)
    Sequence Number (LE): 512 (0x0200)
    [Response frame: 4]
    Timestamp from icmp data: Oct 13, 2024 03:36:50.000000000 +07
    [Timestamp from icmp data (relative): 0.615736000 seconds]
  › Data (48 bytes)
```

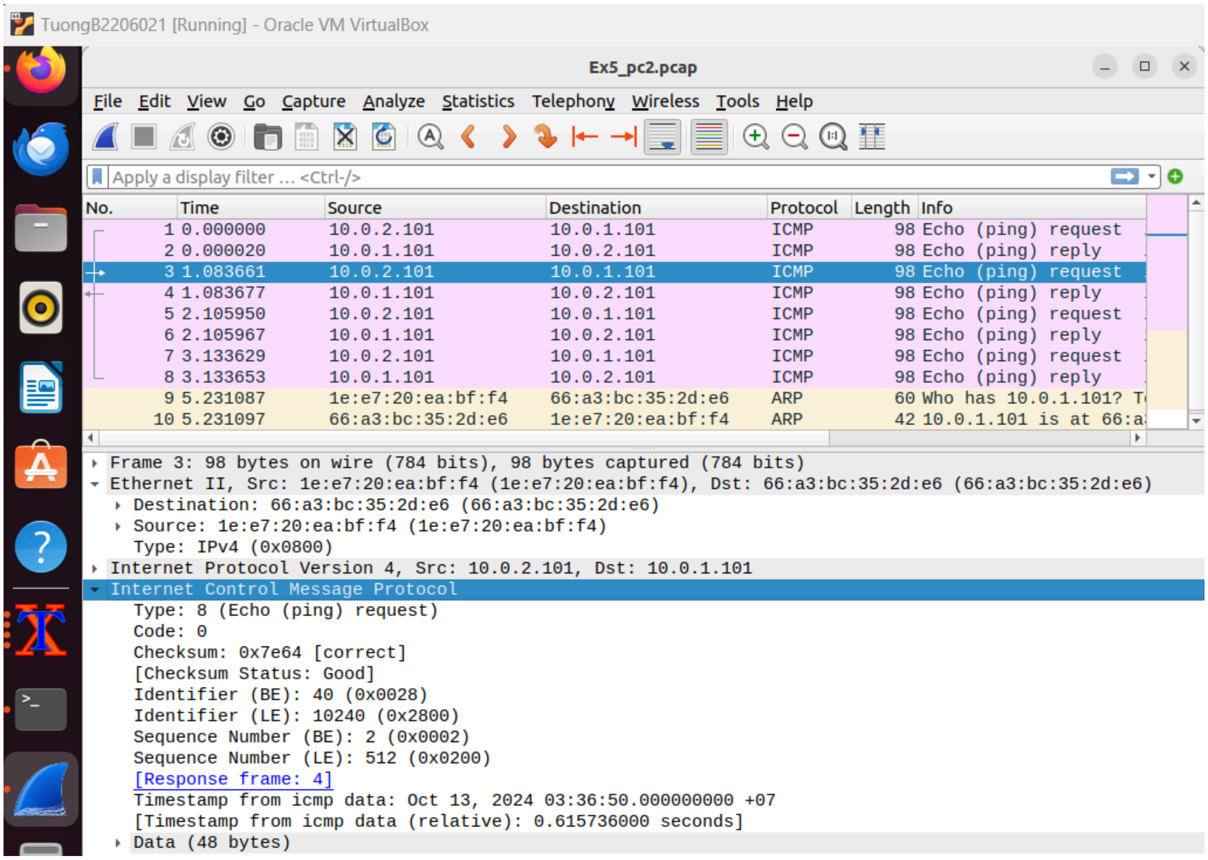
```
0000  66 a3 bc 35 2d e6 1e e7  20 ea bf f4 08 00 45 00   f··5-·······E·
```

```
Internet Control Message Protocol (icmp), 64 bytes     Packets: 12 · Displayed: 12 (100.0%)     Profile: Default
```

- Select Header Internet Control Message Protocol → which protocol is using? On which layer of the OSI model does this protocol operate? What is the content of the message? How long is this message in bytes?

Answer:

- The protocol is **ICMP**
- This protocol operates on **the network layer**
- This below capture is the content of the message
- Size of the message is **48 bytes**


TuongB2206021 [Running] - Oracle VM VirtualBox
Ex5_pc2.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

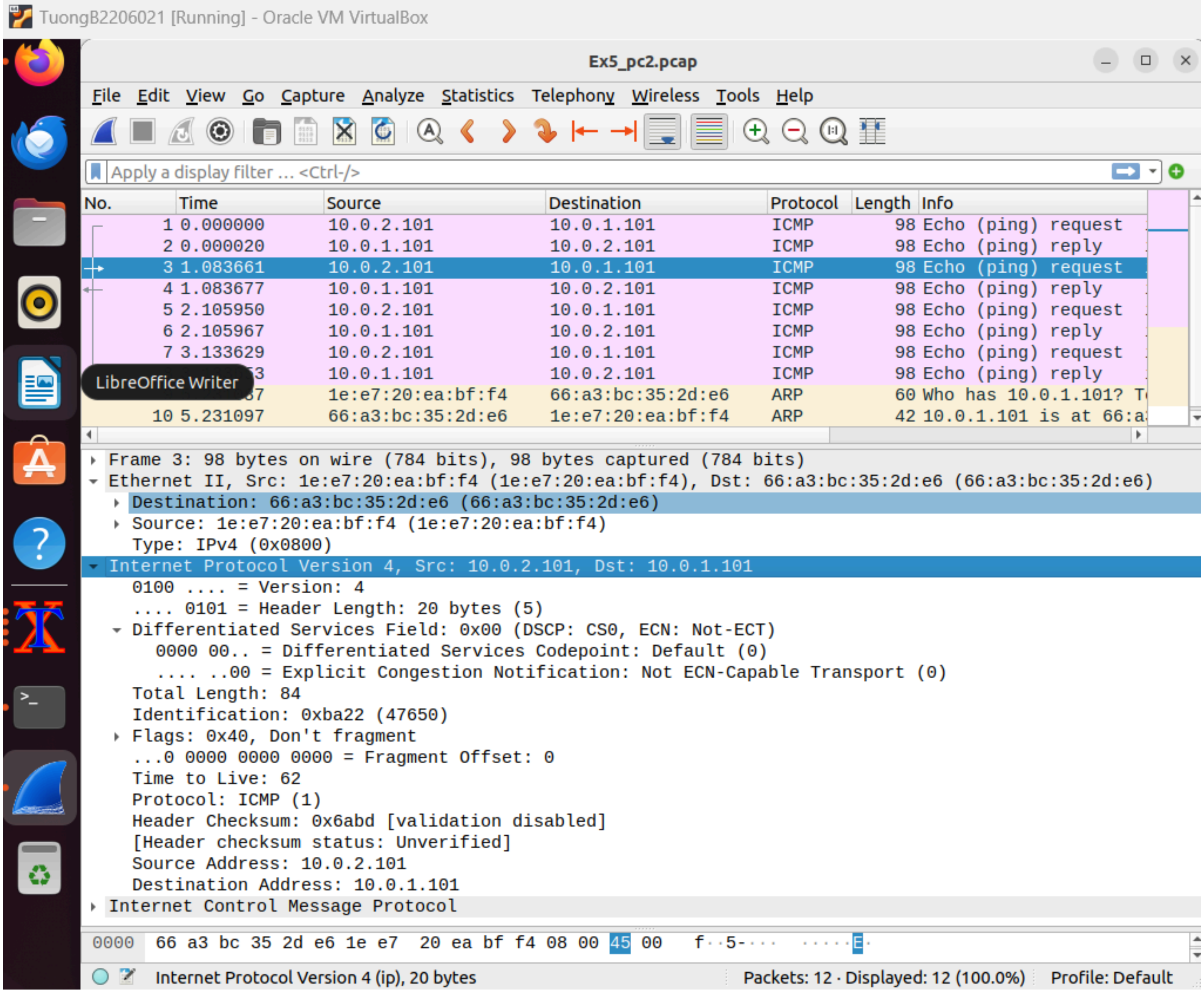
```
No.  Time      Source             Destination        Protocol Length Info
   1 0.000000  10.0.2.101         10.0.1.101         ICMP       98 Echo (ping) request
   2 0.000020  10.0.1.101         10.0.2.101         ICMP       98 Echo (ping) reply
   3 1.083661  10.0.2.101         10.0.1.101         ICMP       98 Echo (ping) request
   4 1.083677  10.0.1.101         10.0.2.101         ICMP       98 Echo (ping) reply
   5 2.105950  10.0.2.101         10.0.1.101         ICMP       98 Echo (ping) request
   6 2.105967  10.0.1.101         10.0.2.101         ICMP       98 Echo (ping) reply
   7 3.133629  10.0.2.101         10.0.1.101         ICMP       98 Echo (ping) request
   8 3.133653  10.0.1.101         10.0.2.101         ICMP       98 Echo (ping) reply
   9 5.231087  1e:e7:20:ea:bf:f4  66:a3:bc:35:2d:e6  ARP        60 Who has 10.0.1.101? T
  10 5.231097  66:a3:bc:35:2d:e6  1e:e7:20:ea:bf:f4  ARP        42 10.0.1.101 is at 66:a
```

```
Frame 3: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
Ethernet II, Src: 1e:e7:20:ea:bf:f4 (1e:e7:20:ea:bf:f4), Dst: 66:a3:bc:35:2d:e6 (66:a3:bc:35:2d:e6)
  Destination: 66:a3:bc:35:2d:e6 (66:a3:bc:35:2d:e6)
  Source: 1e:e7:20:ea:bf:f4 (1e:e7:20:ea:bf:f4)
  Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 10.0.2.101, Dst: 10.0.1.101
Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0x7e64 [correct]
  [Checksum Status: Good]
  Identifier (BE): 40 (0x0028)
  Identifier (LE): 10240 (0x2800)
  Sequence Number (BE): 2 (0x0002)
  Sequence Number (LE): 512 (0x0200)
  [Response frame: 4]
  Timestamp from icmp data: Oct 13, 2024 03:36:50.000000000 +07
  [Timestamp from icmp data (relative): 0.615736000 seconds]
  Data (48 bytes)
```

Select Header Internet Protocol Version 4 → What are the IP addresses of the source and destination hosts? What is the length of the IP packet header? What fields does the Header include? How long is each field (Bytes)? What is the length of the Total Length field (Bytes).

Answer:

- *Ip Address: Source:* ***10.0.2.101*** *Destination:* ***10.0.1.101***
- *Ip packet header's length:* ***20 bytes***
- *Fields of Header:*
    - *Version & Header Length (1 byte)*
    - *Differentiated Services Codepoint & Explicit Congestion Notification (1 Byte)*
    - *Total Length (2 bytes)*
    - *Identification (2 bytes)*
    - *Flags (2 bytes)*
    - *Time to Live (1 byte)*
    - *Protocol (1 byte)*
    - *Header Checksum (2 bytes)*
    - *Source Address (4 bytes)*
    - *Destination Address (4 bytes)*
- *Total Length field is 2 bytes*


TuongB2206021 [Running] - Oracle VM VirtualBox
Ex5_pc2.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
LibreOffice Writer

```
No.  Time        Source              Destination         Protocol  Length Info
   1 0.000000    10.0.2.101          10.0.1.101          ICMP          98 Echo (ping) request
   2 0.000020    10.0.1.101          10.0.2.101          ICMP          98 Echo (ping) reply
   3 1.083661    10.0.2.101          10.0.1.101          ICMP          98 Echo (ping) request
   4 1.083677    10.0.1.101          10.0.2.101          ICMP          98 Echo (ping) reply
   5 2.105950    10.0.2.101          10.0.1.101          ICMP          98 Echo (ping) request
   6 2.105967    10.0.1.101          10.0.2.101          ICMP          98 Echo (ping) reply
   7 3.133629    10.0.2.101          10.0.1.101          ICMP          98 Echo (ping) request
           3     10.0.1.101          10.0.2.101          ICMP          98 Echo (ping) reply
          37     1e:e7:20:ea:bf:f4   66:a3:bc:35:2d:e6   ARP           60 Who has 10.0.1.101? T
  10 5.231097    66:a3:bc:35:2d:e6   1e:e7:20:ea:bf:f4   ARP           42 10.0.1.101 is at 66:a
```

```
Frame 3: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
Ethernet II, Src: 1e:e7:20:ea:bf:f4 (1e:e7:20:ea:bf:f4), Dst: 66:a3:bc:35:2d:e6 (66:a3:bc:35:2d:e6)
  Destination: 66:a3:bc:35:2d:e6 (66:a3:bc:35:2d:e6)
  Source: 1e:e7:20:ea:bf:f4 (1e:e7:20:ea:bf:f4)
  Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 10.0.2.101, Dst: 10.0.1.101
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    0000 00.. = Differentiated Services Codepoint: Default (0)
    .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
  Total Length: 84
  Identification: 0xba22 (47650)
  Flags: 0x40, Don't fragment
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 62
  Protocol: ICMP (1)
  Header Checksum: 0x6abd [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 10.0.2.101
  Destination Address: 10.0.1.101
Internet Control Message Protocol
```

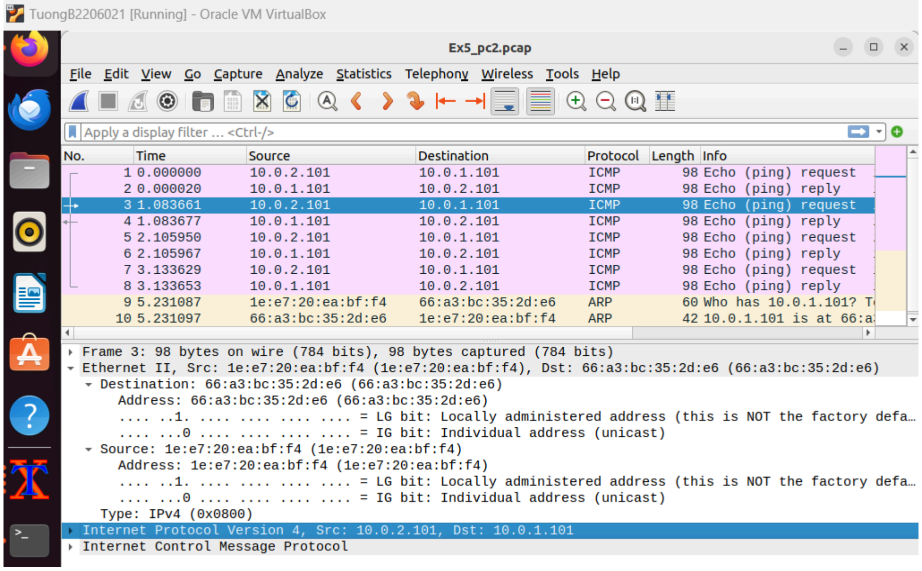
```
0000  66 a3 bc 35 2d e6 1e e7  20 ea bf f4 08 00 45 00   f··5-···  ·····E·
```

Internet Protocol Version 4 (ip), 20 bytes | Packets: 12 · Displayed: 12 (100.0%) | Profile: Default

Select Header Ethernet II → What are the MAC addresses of the source and the destination hosts? What is the Type value?

Answer:

- MAC Address: Source Address: **1e:e7:20:ea:bf:f4** Destination Address: **66:a3:bc:35:2d:e6**
- Type: **IPV4 (0x0800)**


TuongB2206021 [Running] - Oracle VM VirtualBox
Ex5_pc2.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.2.101	10.0.1.101	ICMP	98	Echo (ping) request
2	0.000020	10.0.1.101	10.0.2.101	ICMP	98	Echo (ping) reply
3	1.083661	10.0.2.101	10.0.1.101	ICMP	98	Echo (ping) request
4	1.083677	10.0.1.101	10.0.2.101	ICMP	98	Echo (ping) reply
5	2.105950	10.0.2.101	10.0.1.101	ICMP	98	Echo (ping) request
6	2.105967	10.0.1.101	10.0.2.101	ICMP	98	Echo (ping) reply
7	3.133629	10.0.2.101	10.0.1.101	ICMP	98	Echo (ping) request
8	3.133653	10.0.1.101	10.0.2.101	ICMP	98	Echo (ping) reply
9	5.231087	1e:e7:20:ea:bf:f4	66:a3:bc:35:2d:e6	ARP	60	Who has 10.0.1.101? T
10	5.231097	66:a3:bc:35:2d:e6	1e:e7:20:ea:bf:f4	ARP	42	10.0.1.101 is at 66:a

```
› Frame 3: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
⌄ Ethernet II, Src: 1e:e7:20:ea:bf:f4 (1e:e7:20:ea:bf:f4), Dst: 66:a3:bc:35:2d:e6 (66:a3:bc:35:2d:e6)
  ⌄ Destination: 66:a3:bc:35:2d:e6 (66:a3:bc:35:2d:e6)
      Address: 66:a3:bc:35:2d:e6 (66:a3:bc:35:2d:e6)
      .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NOT the factory defa…
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ⌄ Source: 1e:e7:20:ea:bf:f4 (1e:e7:20:ea:bf:f4)
      Address: 1e:e7:20:ea:bf:f4 (1e:e7:20:ea:bf:f4)
      .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NOT the factory defa…
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
› Internet Protocol Version 4, Src: 10.0.2.101, Dst: 10.0.1.101
› Internet Control Message Protocol
```