

**LAB 1**  
**CONSTRUCT A SIMPLE NETWORK**



Name: Huỳnh Tú Phương

ID: B2206005

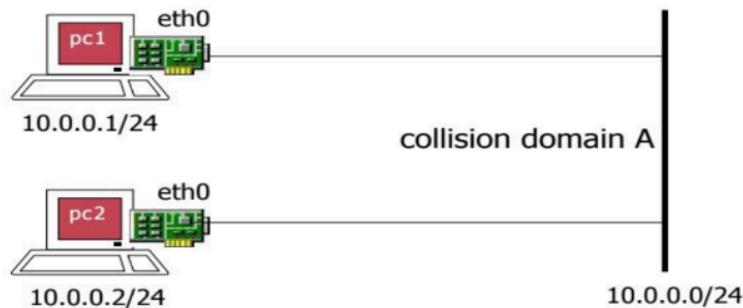
Group: M01

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

## Exercise 1

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

Solution: 003-kathara-lab\_two-hosts.pdf



**Answer:**

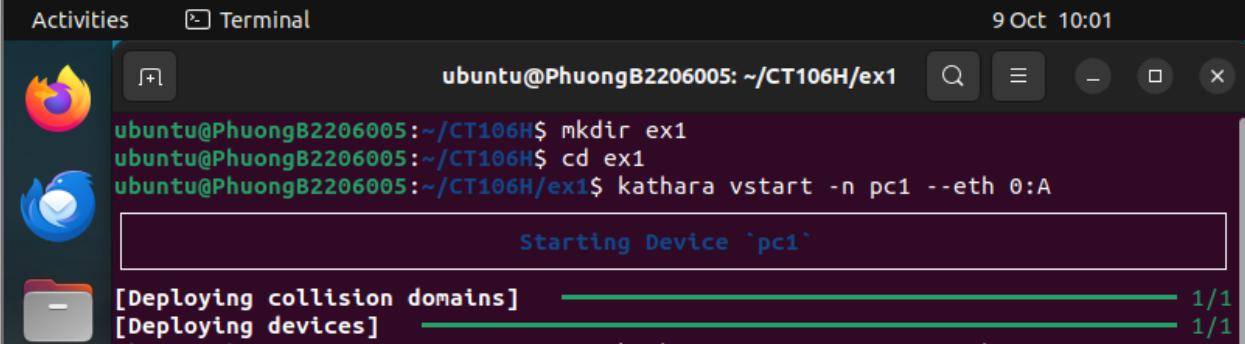
**Step 1: Creating the vms**

/CT106H\$ mkdir ex1

/CT106H\$ cd ex1

Start a new device called pc1 and connected to the virtual collision domain A

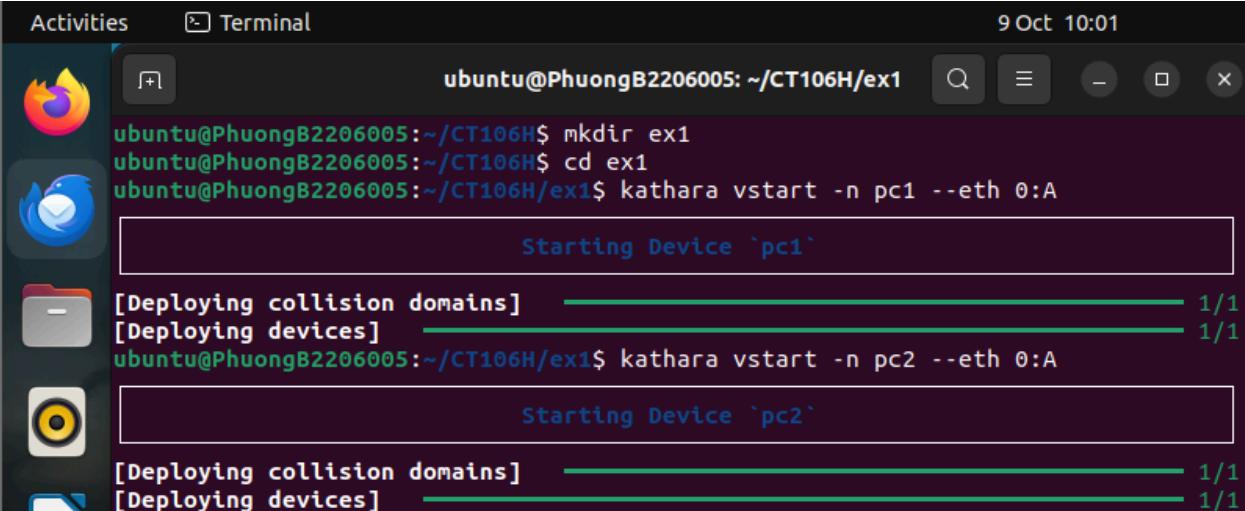
/CT106H/ex1 \$kathara vstart -n pc1 --eth 0:A



```
Activities Terminal 9 Oct 10:01
ubuntu@PhuongB2206005:~/CT106H/ex1$ mkdir ex1
ubuntu@PhuongB2206005:~/CT106H/ex1$ cd ex1
ubuntu@PhuongB2206005:~/CT106H/ex1$ kathara vstart -n pc1 --eth 0:A
Starting Device `pc1'
[Deploying collision domains] 1/1
[Deploying devices] 1/1
```

Start another device called pc2 and connected to the virtual collision domain A

```
/CT106H/ex1$ kathara vstart -n pc2 --eth 0:A
```

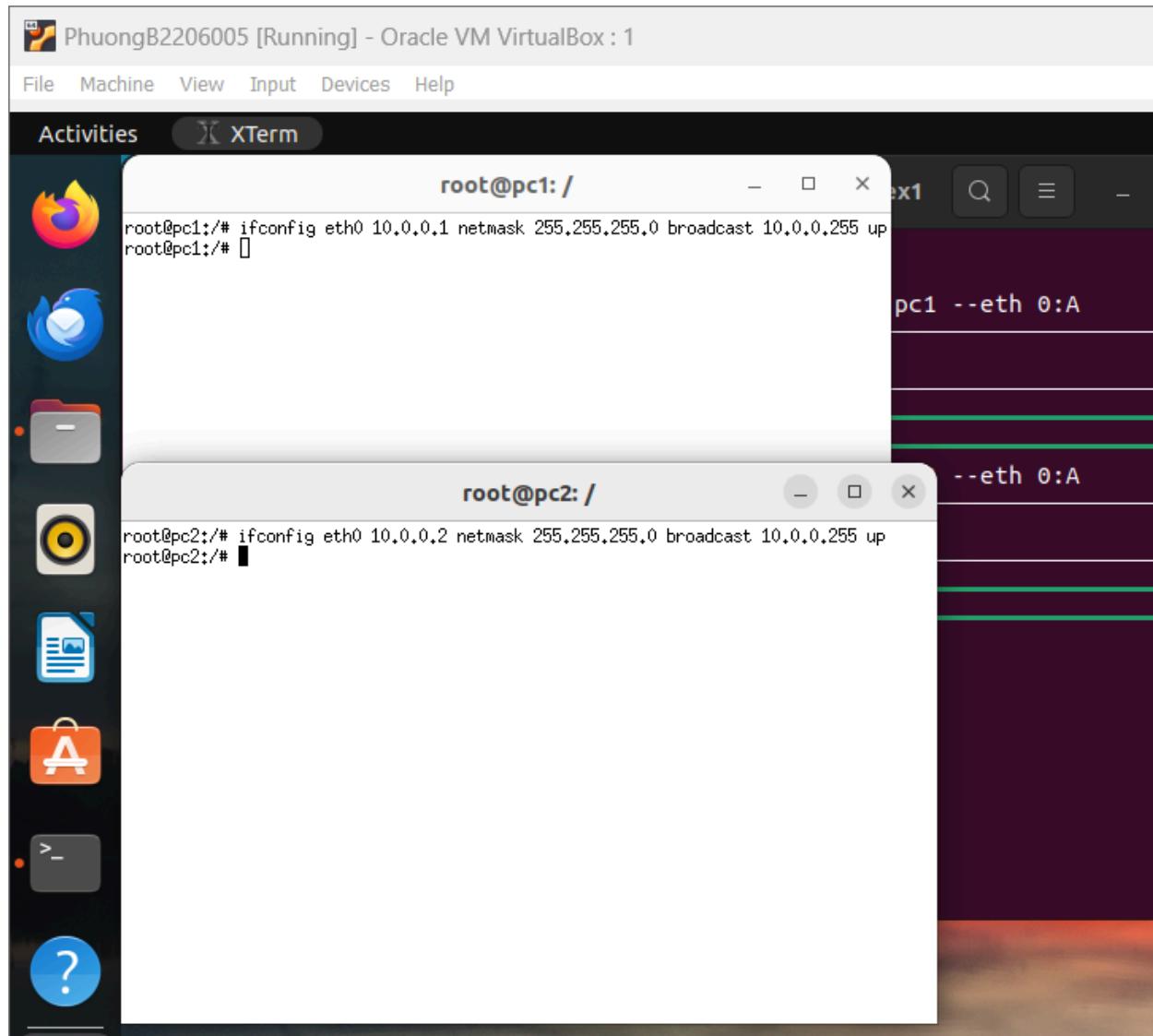


```
Activities Terminal 9 Oct 10:01
ubuntu@PhuongB2206005:~/CT106H/ex1$ mkdir ex1
ubuntu@PhuongB2206005:~/CT106H/ex1$ cd ex1
ubuntu@PhuongB2206005:~/CT106H/ex1$ kathara vstart -n pc1 --eth 0:A
Starting Device `pc1'
[Deploying collision domains] 1/1
[Deploying devices] 1/1
ubuntu@PhuongB2206005:~/CT106H/ex1$ kathara vstart -n pc2 --eth 0:A
Starting Device `pc2'
[Deploying collision domains] 1/1
[Deploying devices] 1/1
```

## Step 2: configuring network interfaces

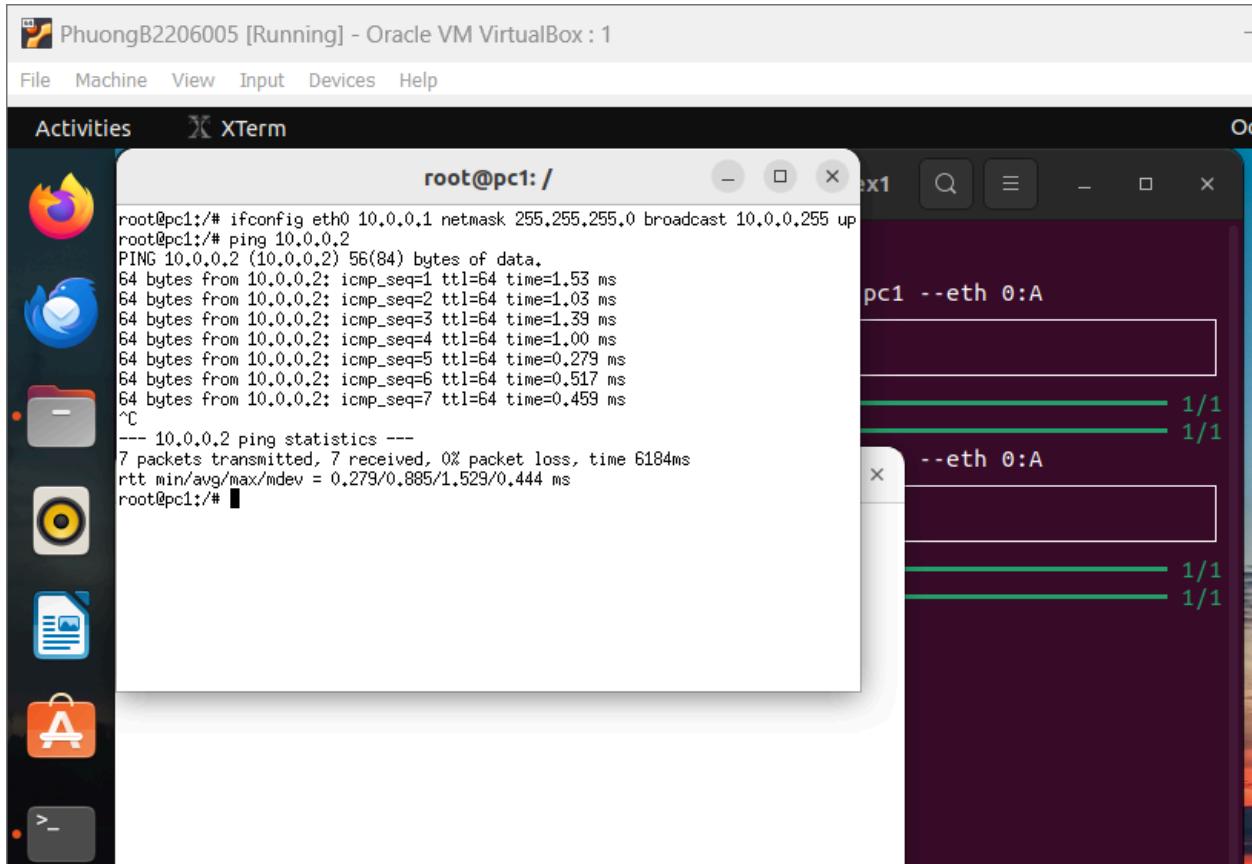
```
#ifconfig eth0 10.0.0.1 netmask 255.255.255.0 broadcast 10.0.0.255 up
```

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



### Step 3: Ping

```
#ping 10.0.0.2 ( Ctrl + C)
```

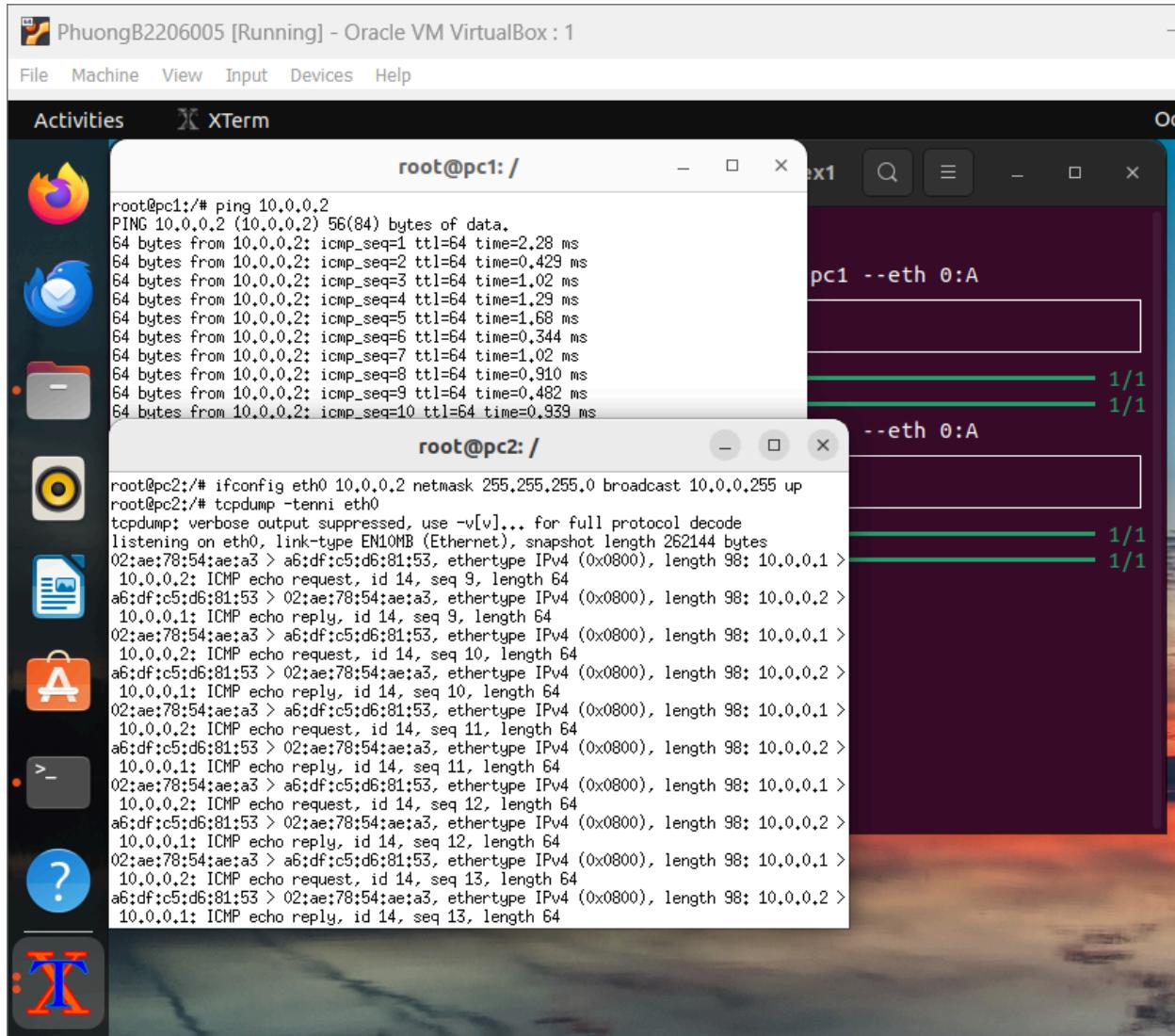


*Pc1 and pc2 can read each other*

**Step 4: a look at the packets**

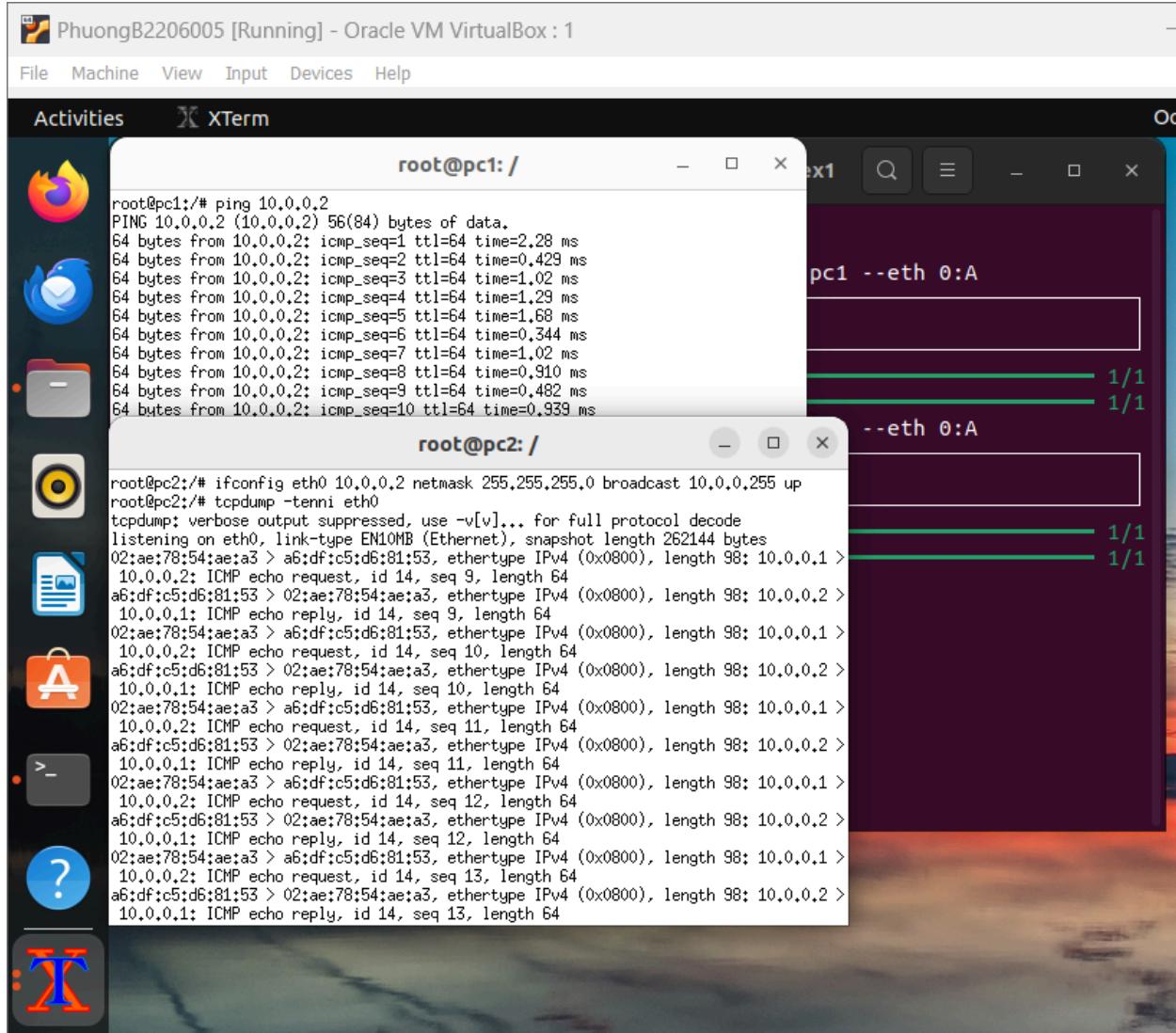
+ *ping from pc1*

#ping 10.0.0.2 ( Ctrl + C)

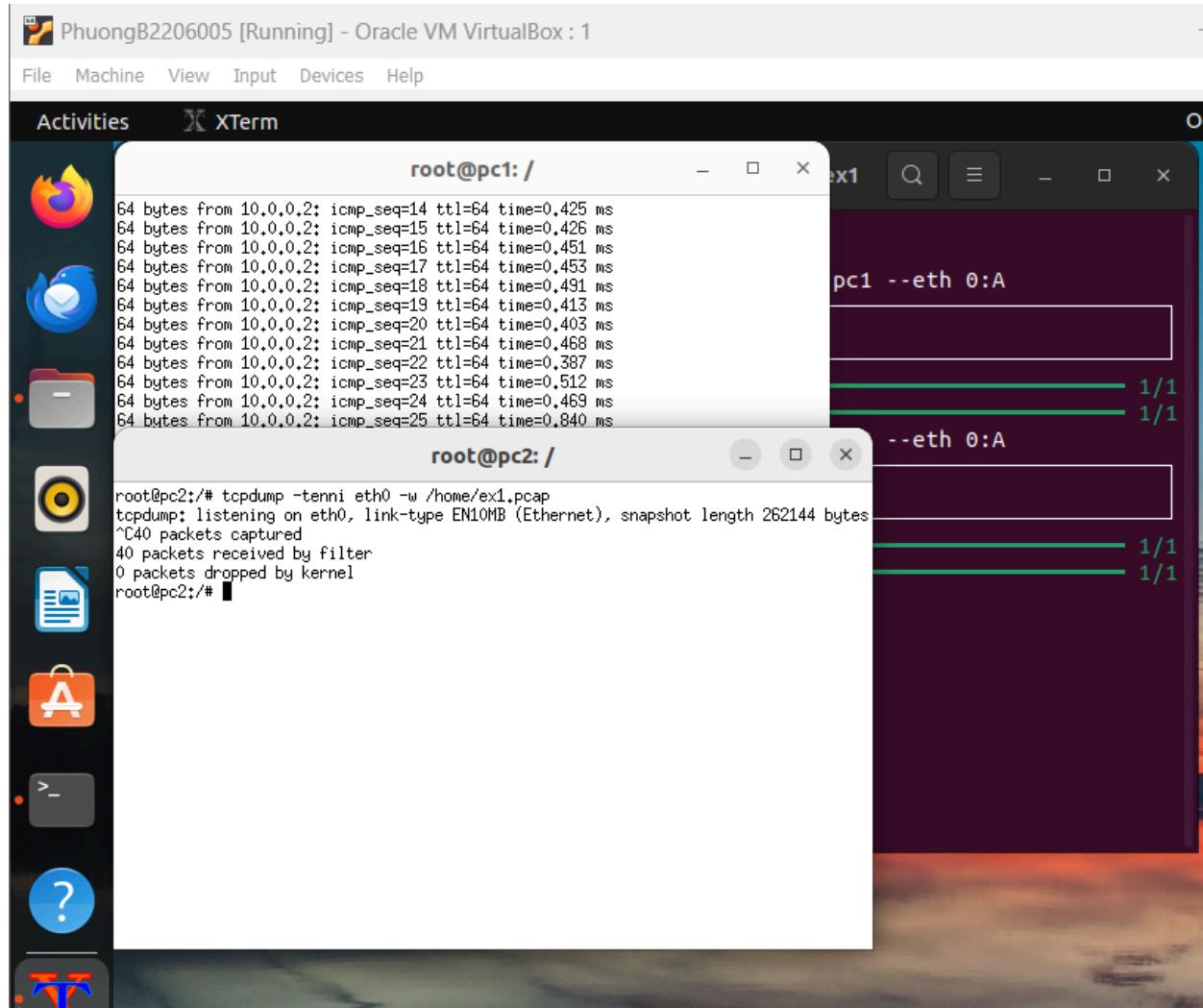


+ *at the same time, sniff from pc2*

#tcpdump -tetti eth0



- + same as before, but store sniffed packets into file capture.pcap
- ```
#tcpdump -t -n -w /home/ex1.pcap
```

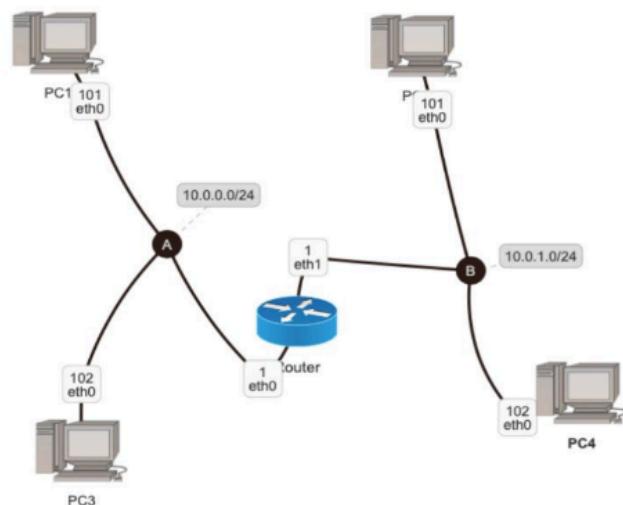


```
/CT106H/ex1 $kathara vclean -n pc1  
/CT106H/ex1 $kathara vclean -n pc2
```

```
ubuntu@PhuongB2206005:~/CT106H/ex1$ kathara vstart -n pc1 --eth 0:A
   Starting Device `pc1'
[Deploying collision domains] 1/1
[Deploying devices] 1/1
ubuntu@PhuongB2206005:~/CT106H/ex1$ kathara vstart -n pc2 --eth 0:A
   Starting Device `pc2'
[Deploying collision domains] 1/1
[Deploying devices] 1/1
ubuntu@PhuongB2206005:~/CT106H/ex1$ kathara vclean -n pc1
   Stopping Device `pc1'
[Deleting devices] 1/1
ubuntu@PhuongB2206005:~/CT106H/ex1$ kathara vclean -n pc2
   Stopping Device `pc2'
[Deleting devices] 1/1
[Deleting collision domains] 1/1
```

## Exercise 2

Construct the following network



```
$mkdir -p CT106H/exercise02
$cd CT106H/exercise02
/CT106H/exercise02$mkdir pc1 pc2 pc3 pc4 router1 shared
```

```
/CT106H/exercise02$touch lab.conf pc1.startup pc2.startup pc3.startup pc4.startup  
router1.startup  
/CT106H/exercise02$tree
```

The screenshot shows a terminal window titled "Terminal" running on an Ubuntu system. The terminal session starts with the user navigating to the directory "/CT106H/exercise02". They then run several commands to create a directory structure and files:

```
ubuntu@PhuongB2206005:~$ mkdir -p CT106H/exercise02  
ubuntu@PhuongB2206005:~$ cd CT106H/exercise02  
ubuntu@PhuongB2206005:~/CT106H/exercise02$ mkdir pc1 pc2 pc3 route1 shared  
ubuntu@PhuongB2206005:~/CT106H/exercise02$ touch lab.conf pc1.startup pc2.startup pc3.startup pc4.startup router1.startup  
ubuntu@PhuongB2206005:~/CT106H/exercise02$ mkdir pc4  
ubuntu@PhuongB2206005:~/CT106H/exercise02$ tree
```

The output of the "tree" command shows the following directory structure:

```
.
```

- lab.conf
- pc1
  - pc1.startup
  - pc2
    - pc2.startup
    - pc3
      - pc3.startup
      - pc4
        - pc4.startup
        - route1
          - router1.startup
  - shared

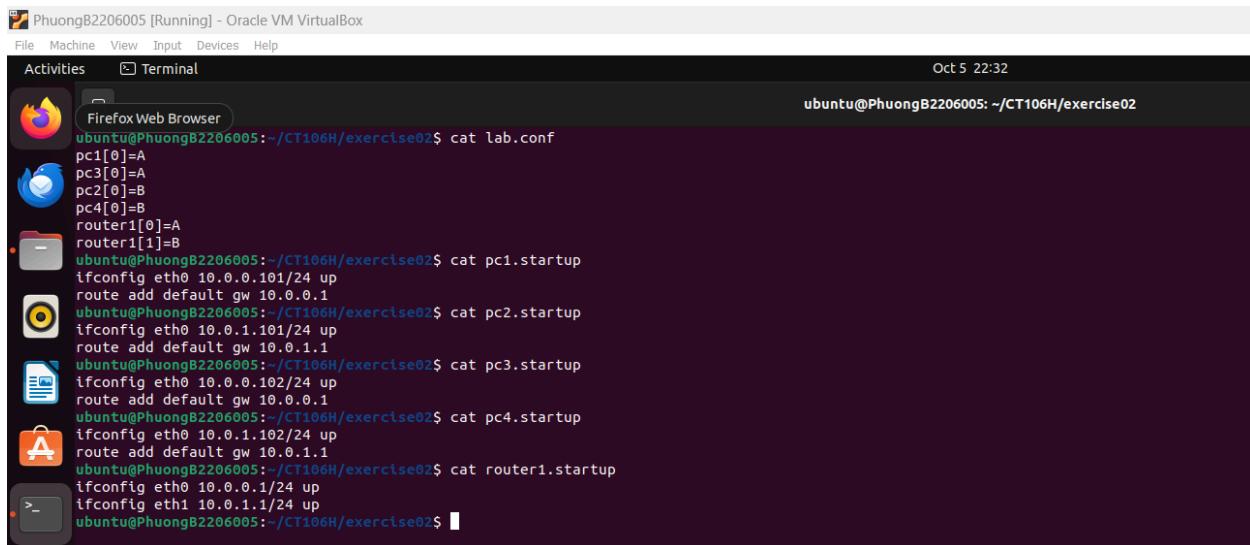
6 directories, 6 files

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

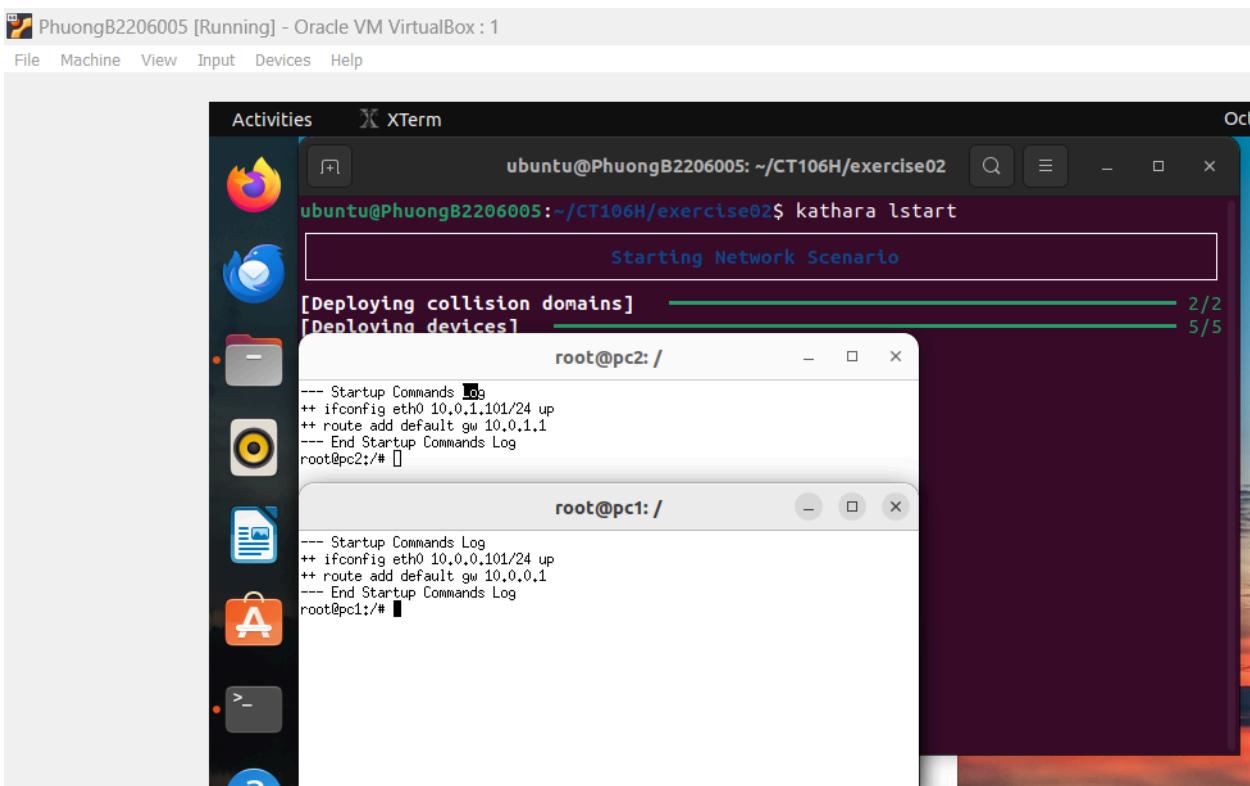
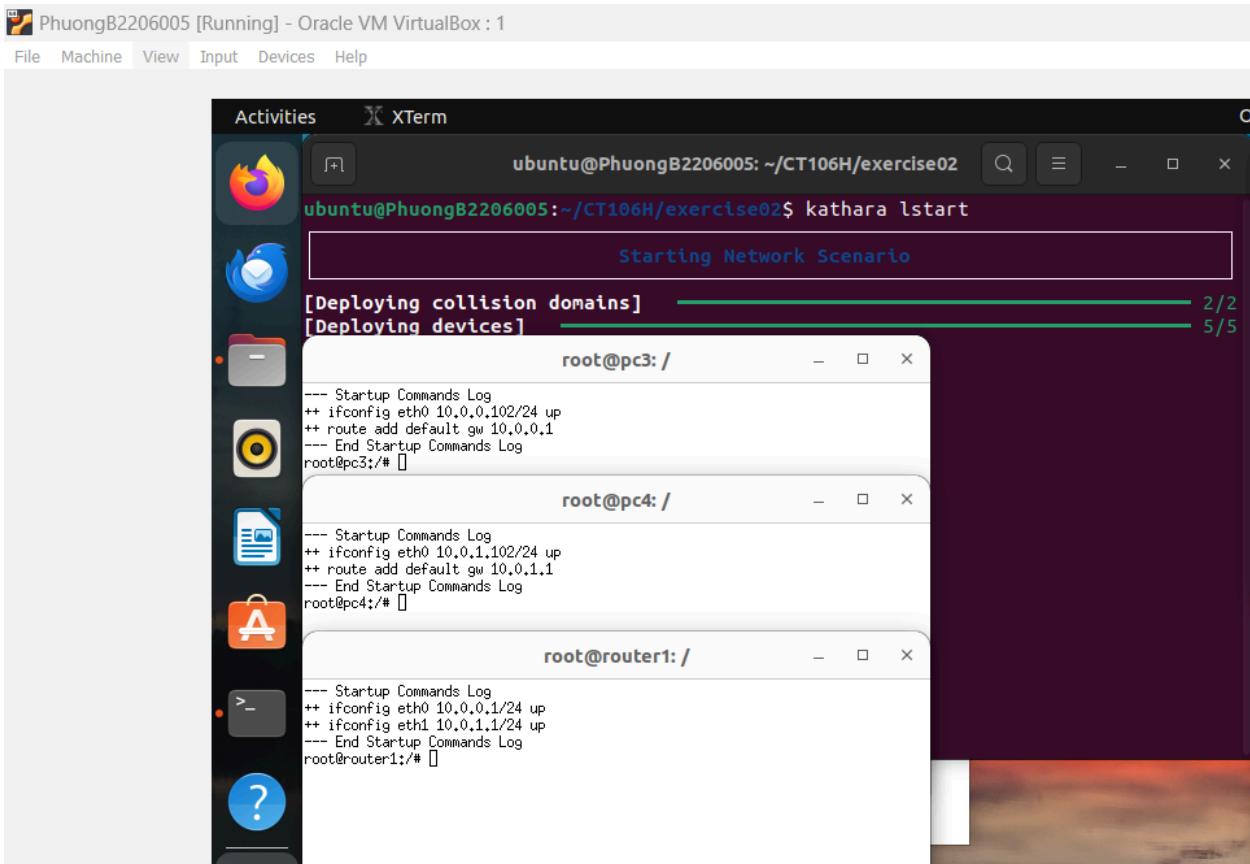
```
/CT106H/exercise02$nano lab.conf
```

```
pc1[0]=A  
pc2[0]=B  
pc3[0]=A  
pc4[0]=B  
router1[0]=A  
router1[1]=B  
/CT106H/exercise02$nano pc1.startup  
ifconfig eth0 10.0.0.101/24 up  
Route add default gw 10.0.0.1  
/CT106H/exercise02$nano pc2.startup  
ifconfig eth0 10.0.1.101/24 up  
Route add default gw 10.0.1.1  
/CT106H/exercise02$nano pc3.startup  
ifconfig eth0 10.0.0.102/24 up  
Route add default gw 10.0.0.1  
/CT106H/exercise02$nano pc4.startup
```

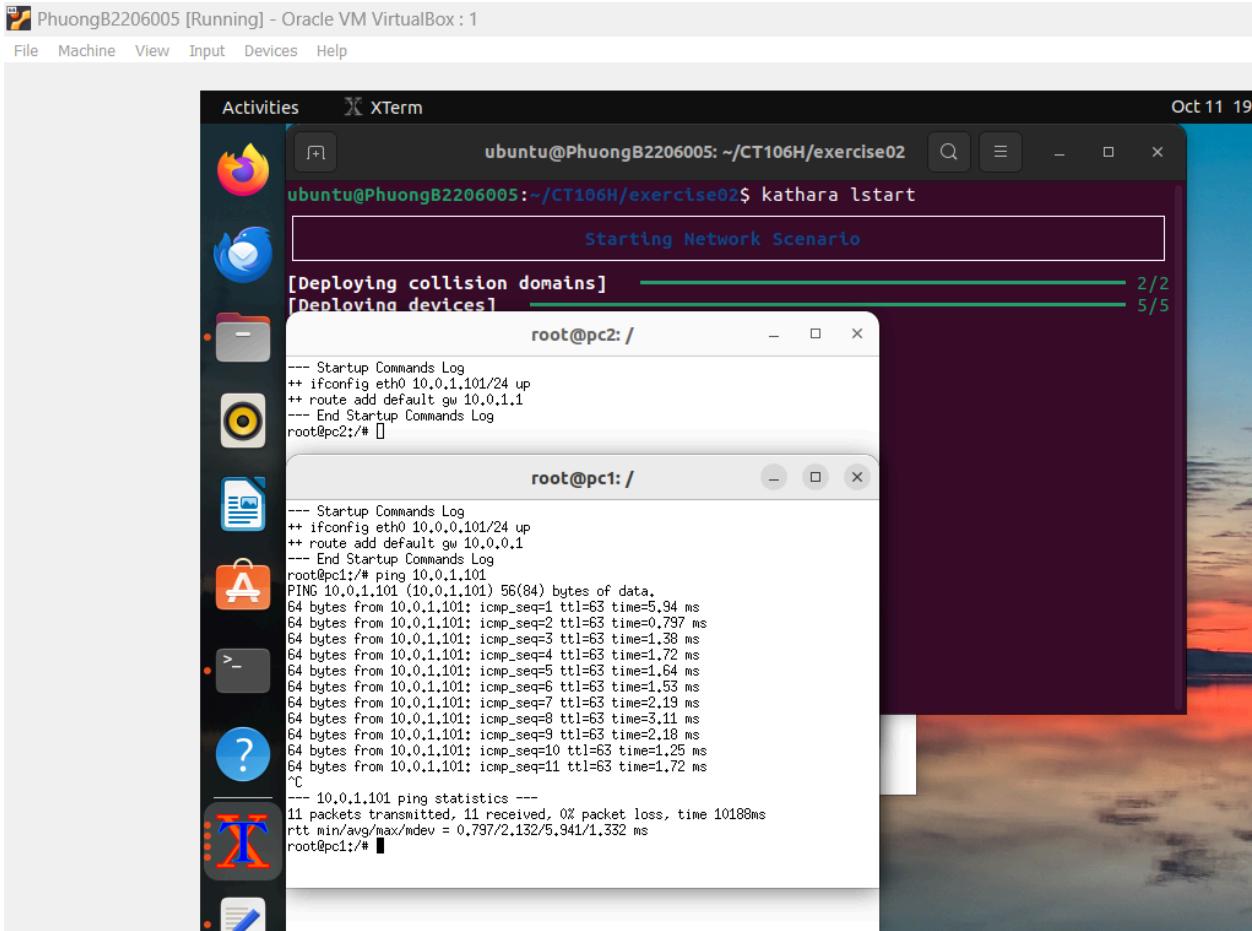
```
ifconfig eth0 10.0.1.102/24 up
Route add default gw 10.0.1.1
/CT106H/exercise02$ nano router1.startup
ifconfig eth0 10.0.0.1/24 up
ifconfig eth0 10.0.1.1/24 up
/CT106H/exercise02$ cat lab.conf
/CT106H/exercise02$ cat pc1.startup
/CT106H/exercise02$ cat pc2.startup
/CT106H/exercise02$ cat pc3.startup
/CT106H/exercise02$ cat pc4.startup
/CT106H/exercise02$ cat router1.startup
```



/CT106H/exercise02 \$ kathara lstart

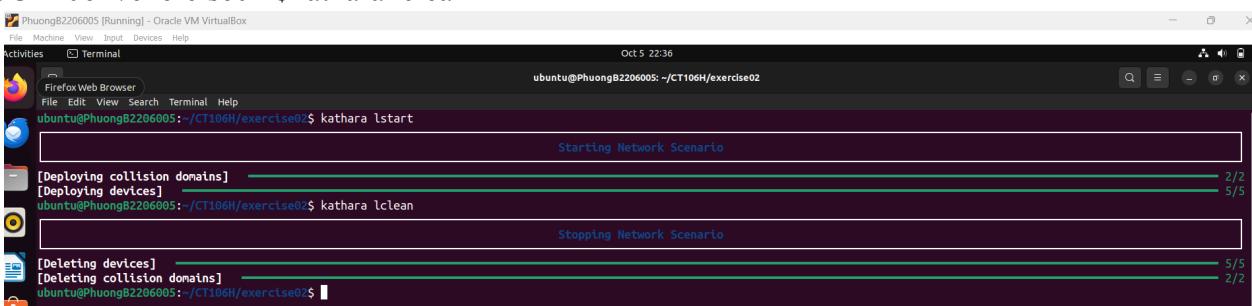


Test connecting



*All pc can read each other*

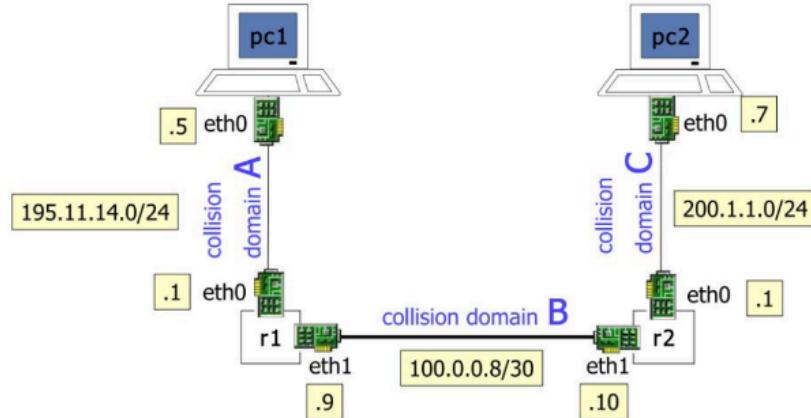
Before leaving, remember to halts all the devices of this lab:  
`/CT106H/exercise02 $kathara lclean`



## Exercise 3

Construct the following network

Solution: 004-kathara-lab\_static-routing.pdf

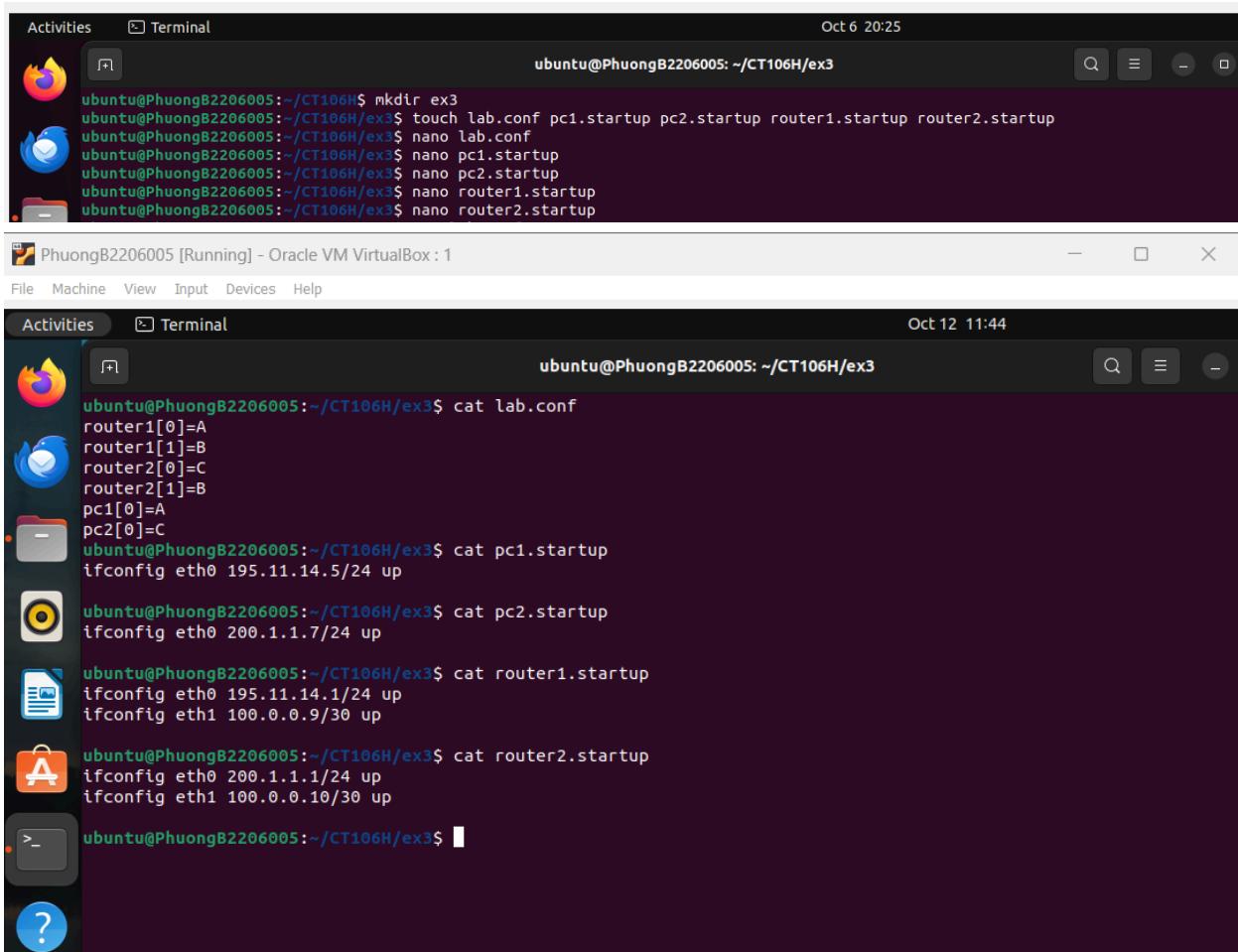


\$mkdir ex3

**Prepared for ex3**

```
/CT106H/ex3 $nano lab.conf
pc1[0]=A
pc2[0]=C
router1[0]=A
router1[1]=B
router2[0]=C
router2[1]=B
/CT106H/ex3$nano pc1.startup
ifconfig eth0 195.11.14.5/24 up
Route add default gw 195.11.14..1
/CT106H/ex3$nano pc2.startup
ifconfig eth0 200.1.1.7/24 up
Route add default gw 200.1.1.1
/CT106H/ex3$nano router1.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
/CT106H/ex3$nano router2.startup
ifconfig eth0 200.1.1.1/24 up
ifconfig eth1 100.0.0.10/30 up
Route add -net 195.11.14.0/24 gw 100.0.0.9 dev eth1
/CT106H/ex3$cat lab.conf
/CT106H/ex3$cat pc1.startup
/CT106H/ex3$cat pc2.startup
/CT106H/ex3$cat router1.startup
/CT106H/ex3$cat router2.startup
```

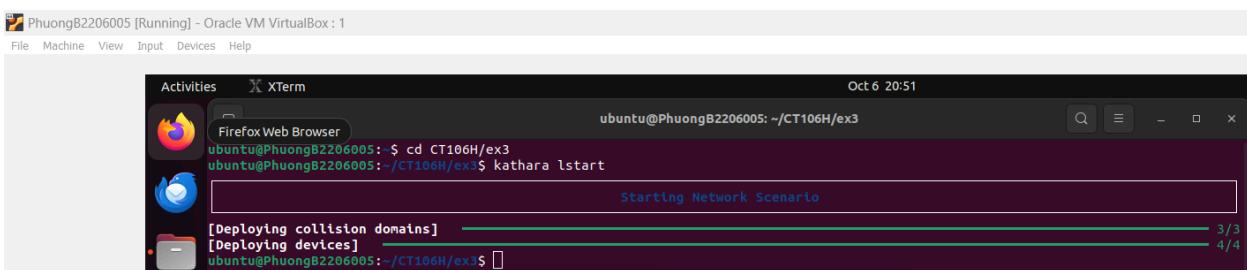
## CT106H – Computer Network



```
Activities Terminal Oct 6 20:25
ubuntu@PhuongB2206005:~/CT106H/ex3$ mkdir ex3
ubuntu@PhuongB2206005:~/CT106H/ex3$ touch lab.conf pc1.startup pc2.startup router1.startup router2.startup
ubuntu@PhuongB2206005:~/CT106H/ex3$ nano lab.conf
ubuntu@PhuongB2206005:~/CT106H/ex3$ nano pc1.startup
ubuntu@PhuongB2206005:~/CT106H/ex3$ nano pc2.startup
ubuntu@PhuongB2206005:~/CT106H/ex3$ nano router1.startup
ubuntu@PhuongB2206005:~/CT106H/ex3$ nano router2.startup

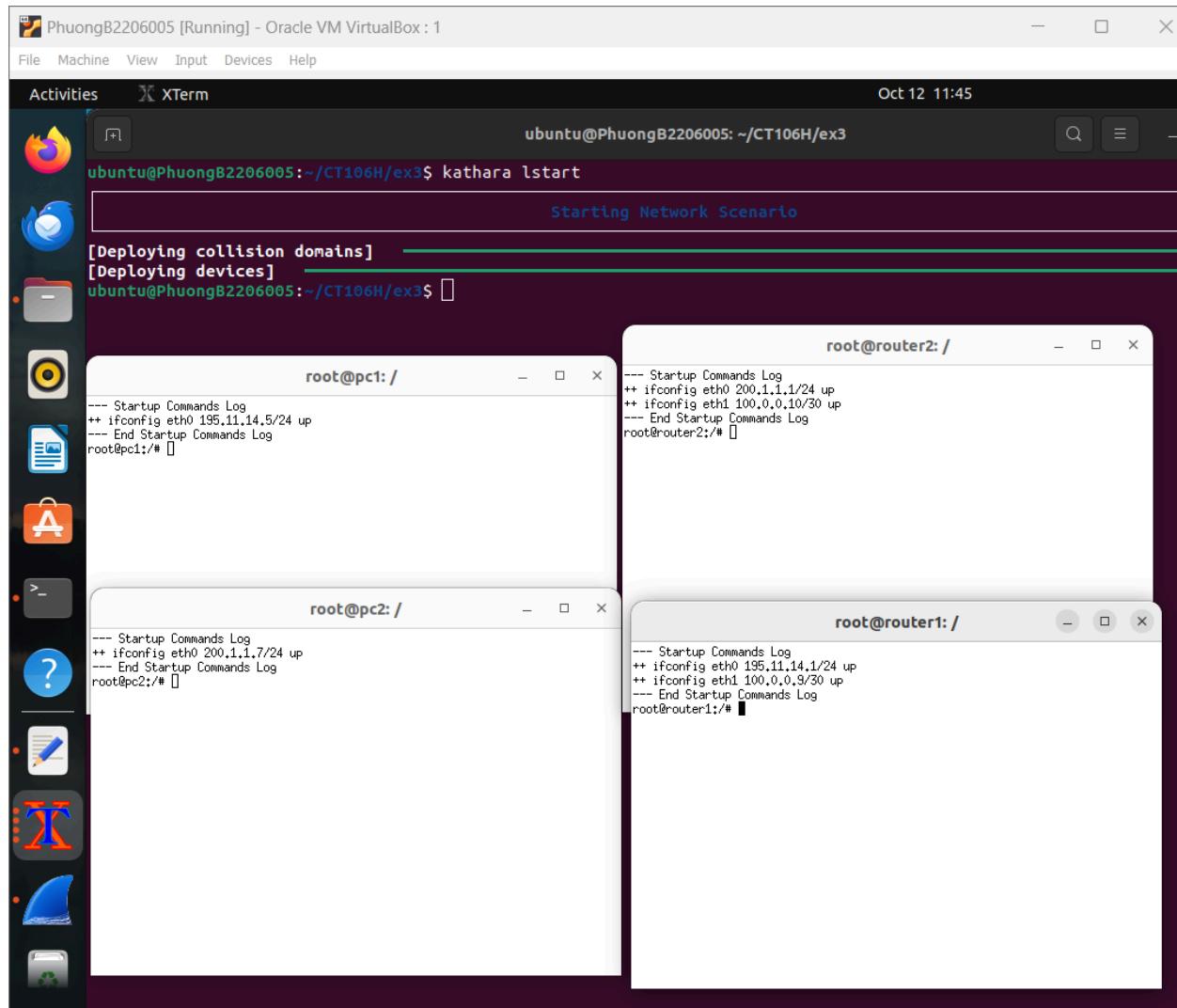
PhuongB2206005 [Running] - Oracle VM VirtualBox : 1
File Machine View Input Devices Help
Activities Terminal Oct 12 11:44
ubuntu@PhuongB2206005:~/CT106H/ex3$ cat lab.conf
router1[0]=A
router1[1]=B
router2[0]=C
router2[1]=B
pc1[0]=A
pc2[0]=C
ubuntu@PhuongB2206005:~/CT106H/ex3$ cat pc1.startup
ifconfig eth0 195.11.14.5/24 up
ubuntu@PhuongB2206005:~/CT106H/ex3$ cat pc2.startup
ifconfig eth0 200.1.1.7/24 up
ubuntu@PhuongB2206005:~/CT106H/ex3$ cat router1.startup
ifconfig eth0 195.11.14.1/24 up
ifconfig eth1 100.0.0.9/30 up
ubuntu@PhuongB2206005:~/CT106H/ex3$ cat router2.startup
ifconfig eth0 200.1.1.1/24 up
ifconfig eth1 100.0.0.10/30 up
ubuntu@PhuongB2206005:~/CT106H/ex3$
```

/CT106H/ex3\$kathara lstart

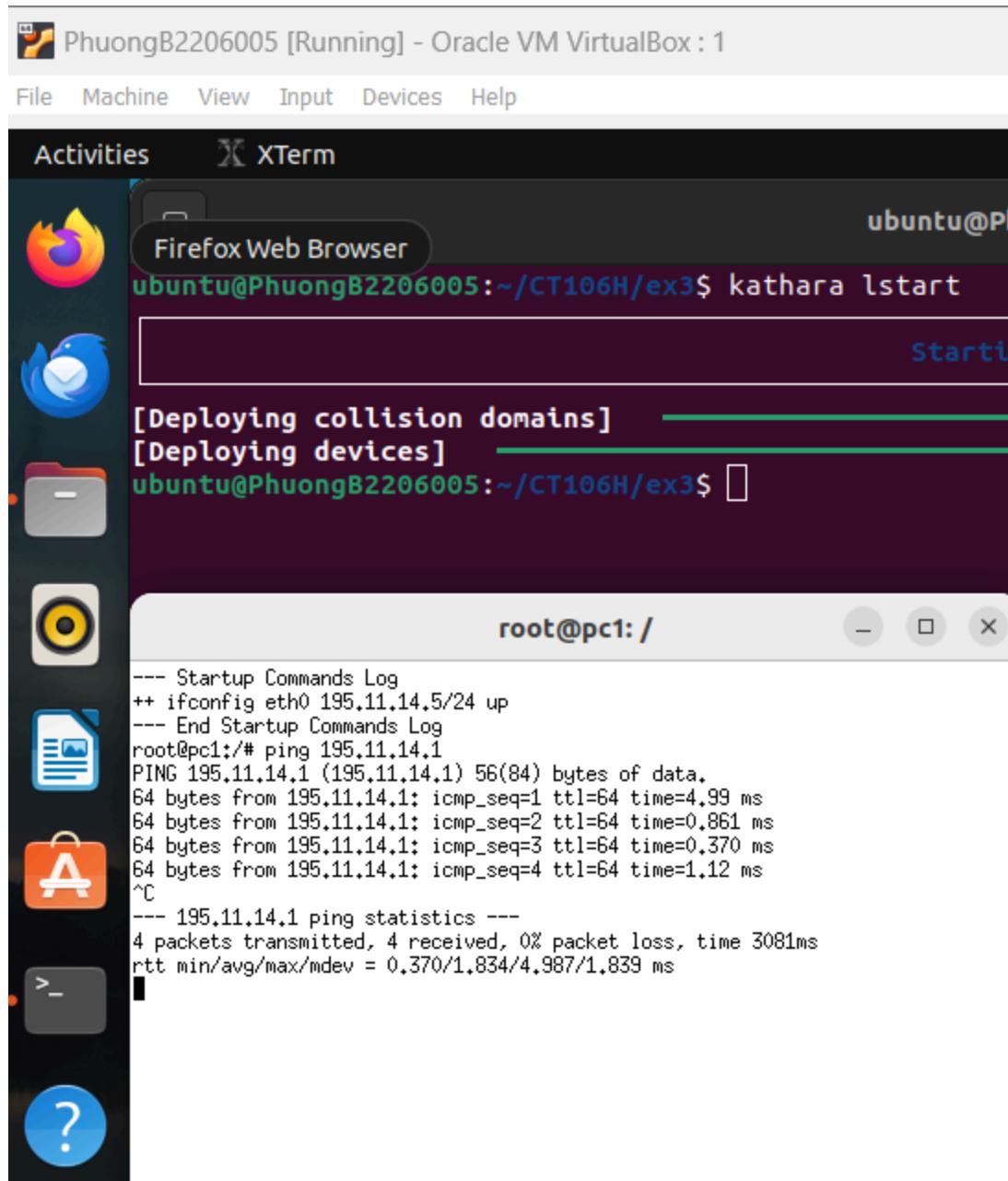


```
Activities XTerm Oct 6 20:51
ubuntu@PhuongB2206005:~/CT106H/ex3$ cd CT106H/ex3
ubuntu@PhuongB2206005:~/CT106H/ex3$ kathara lstart
Starting Network Scenario
[Deploying collision domains] _____ 3/3
[Deploying devices] _____ 4/4
ubuntu@PhuongB2206005:~/CT106H/ex3$
```

## CT106H – Computer Network



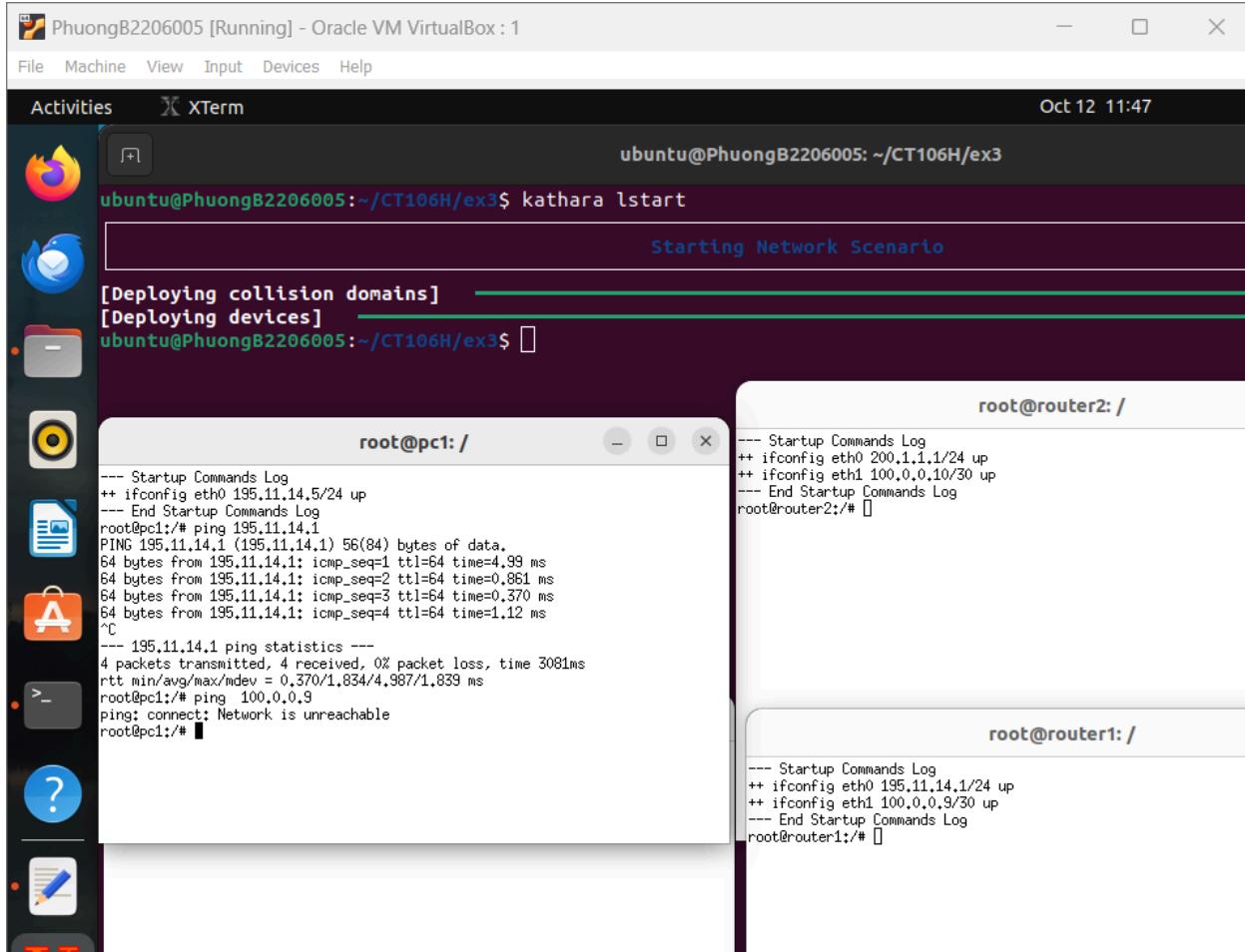
**Test connecting**



### Inspecting routing tables

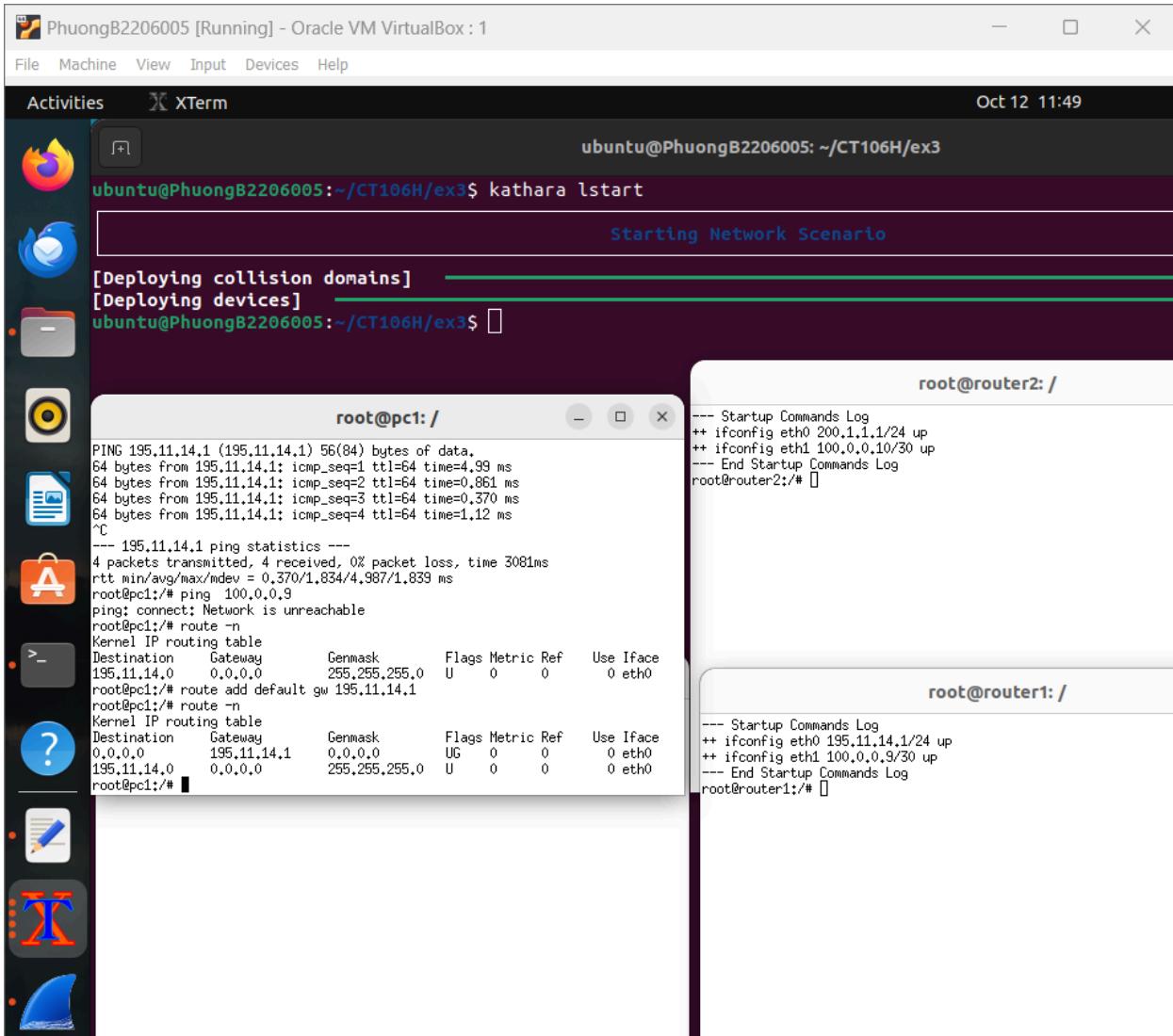
Both routers and PCs don't know how to reach networks that are not directly connected to them

#ping 100.0.0.9 ->Network is unreachable

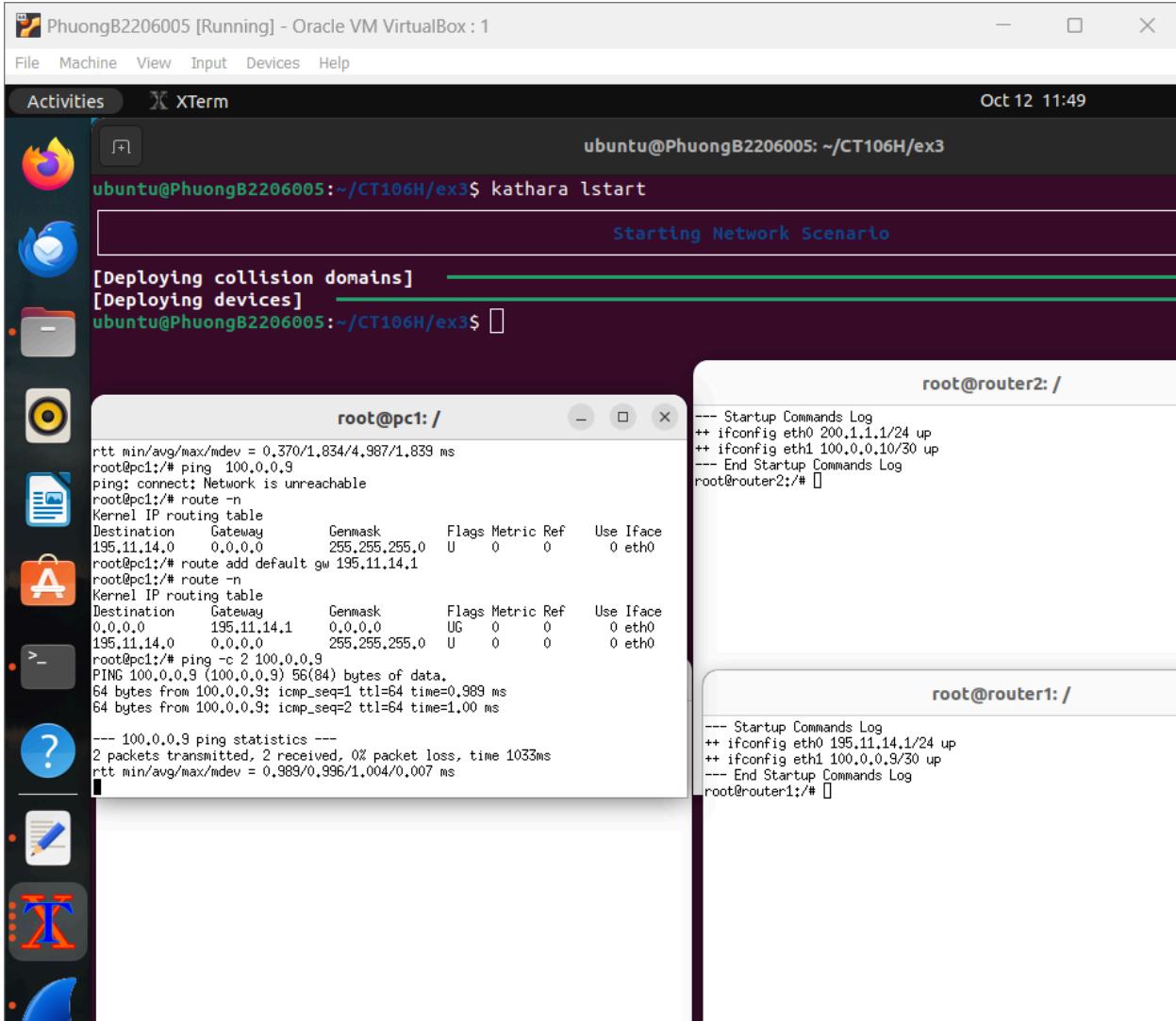


To fix the problem we could specify the default route on the pcs: “through this gateway (IP number) you can reach all the other networks”

```
# route add default gw 195.11.14.1
```



**Success after add default route on the pc1**



### r2's routing table

pc1's address is 195.11.14.5

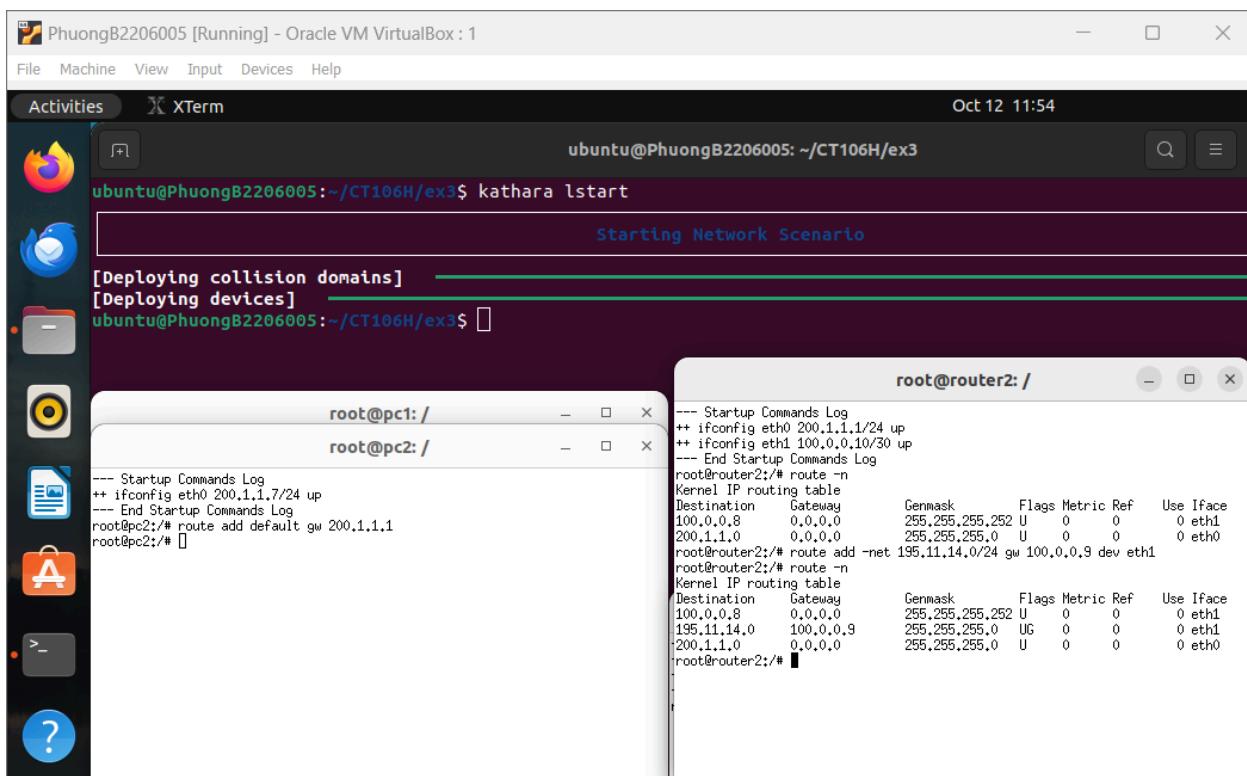
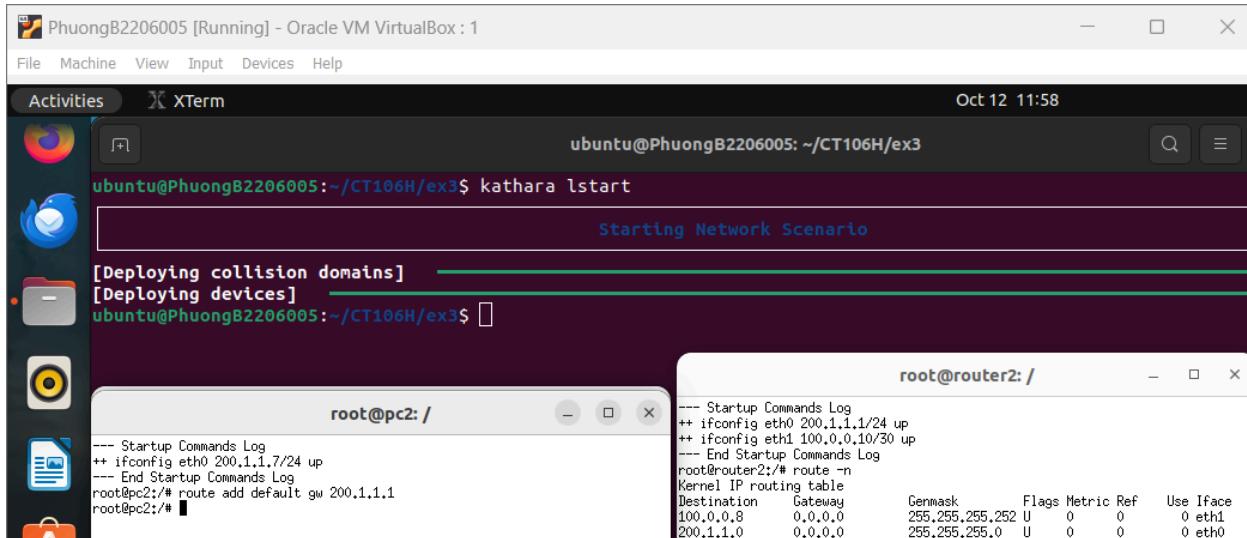
r2 does not know how to reach such an address.

Echo requests arrive to r2 but r2 does not know where echo replies should be forwarded!

Somebody should teach r2 how to reach pc1

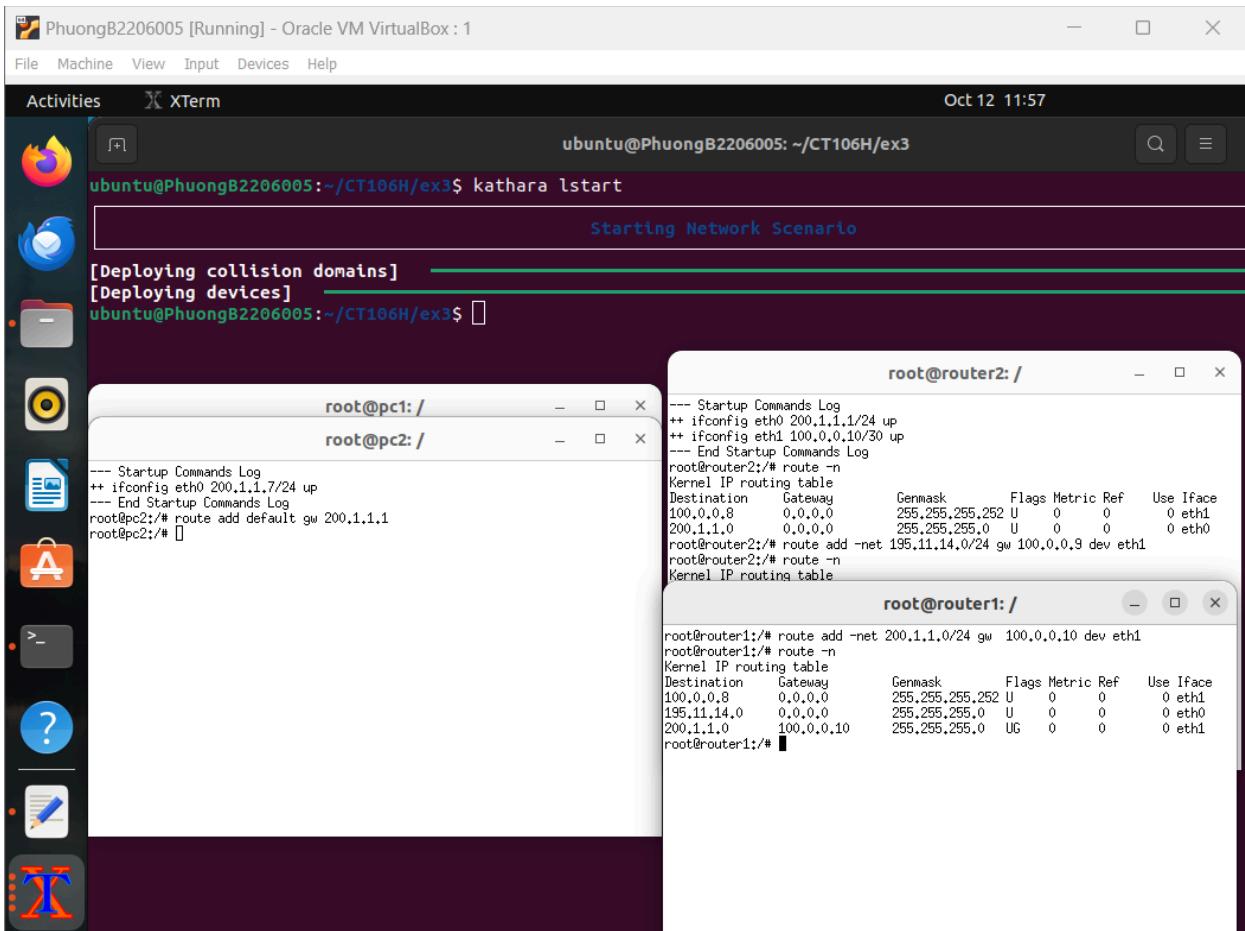
We may insert a static route into the routing table of r2

## CT106H – Computer Network



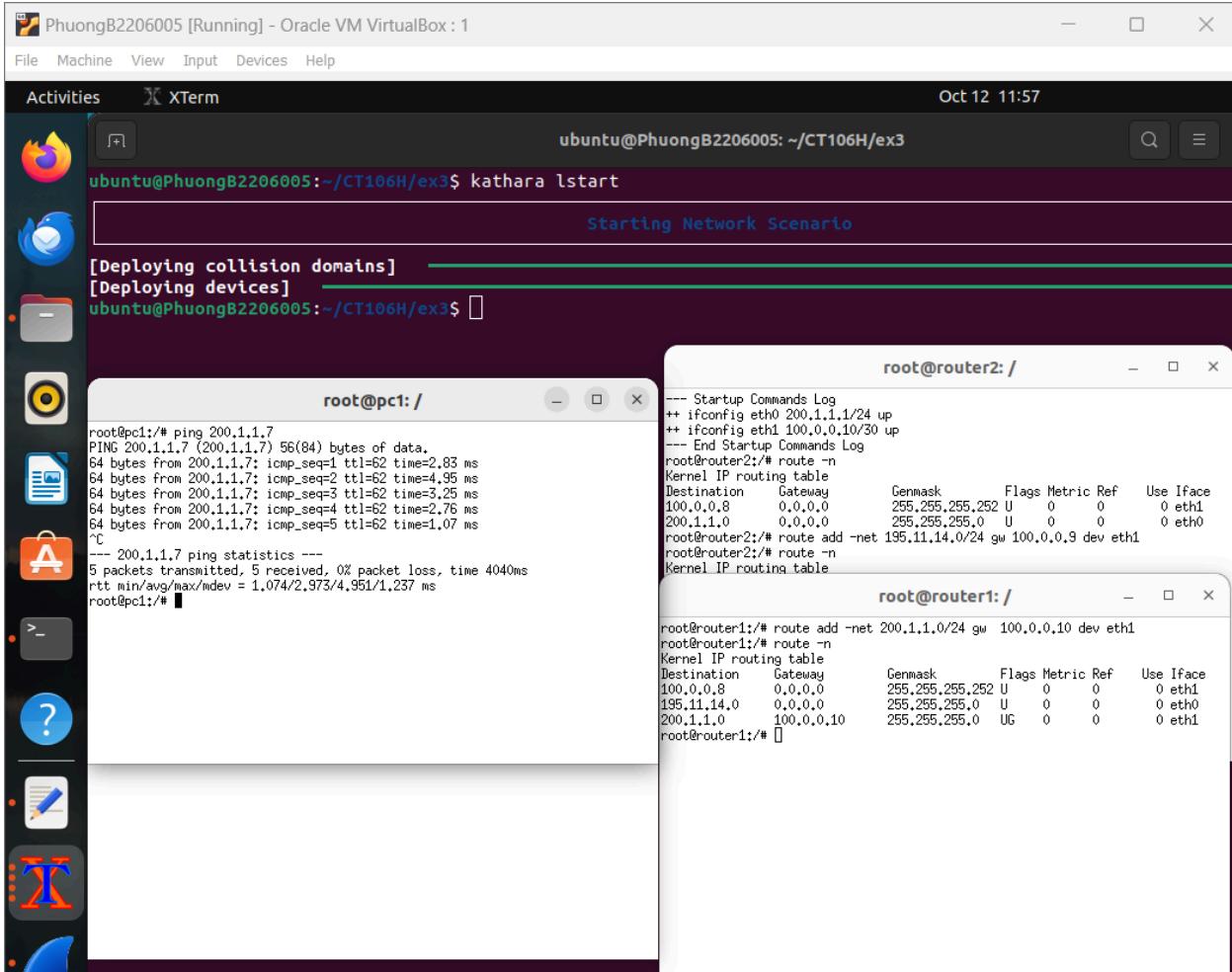
A similar configuration should be deployed on r1

## CT106H – Computer Network

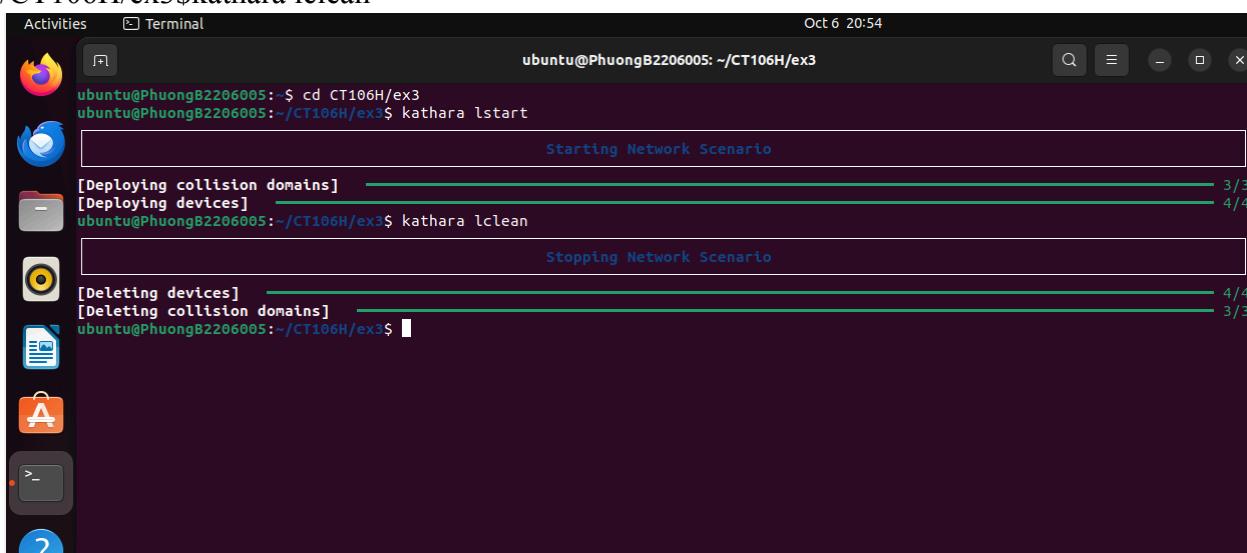


Testing static routes - The PCs can now reach each other

## CT106H – Computer Network



/CT106H/ex3\$ kathara lc



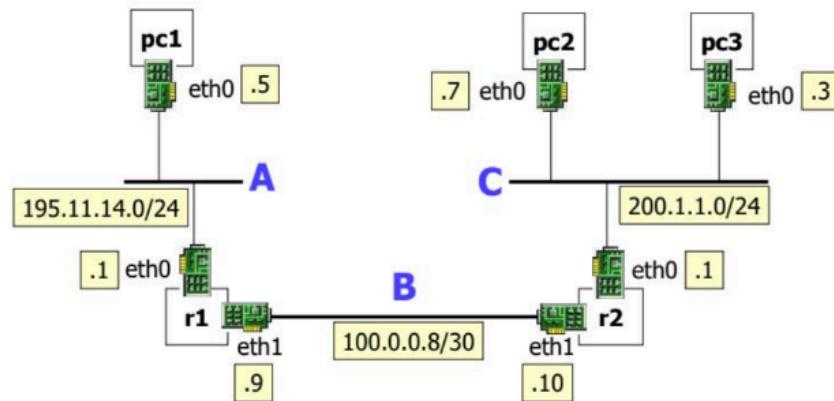
**command to configure a static route that is equivalent to default route:**

route add -net <network> netmask <netmask> gw <gateway\_ip> dev <interface>

## Exercise 4

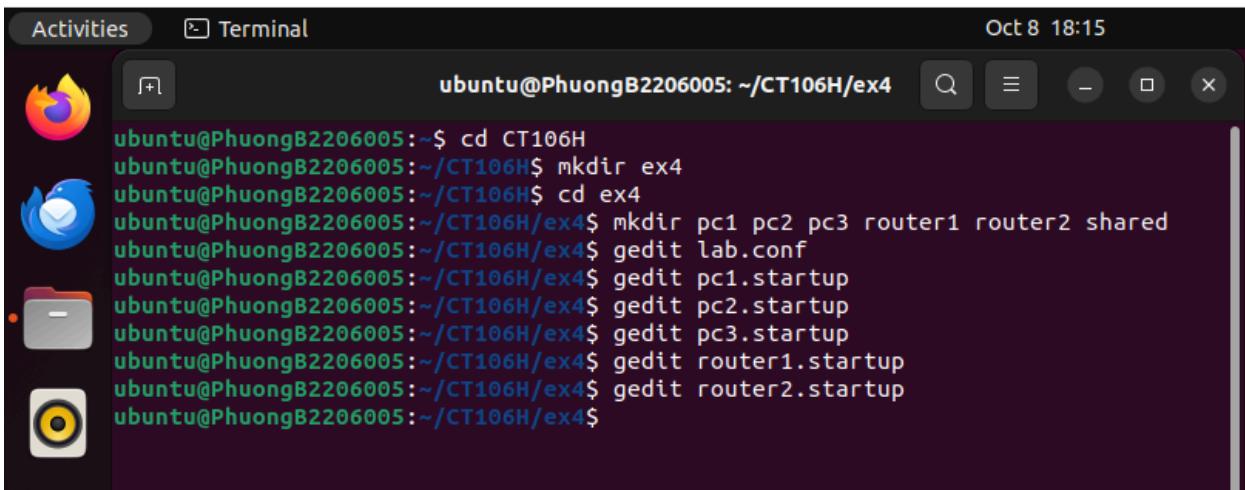
Study arp protocol

Solution: 005-kathara-lab\_arp.pdf



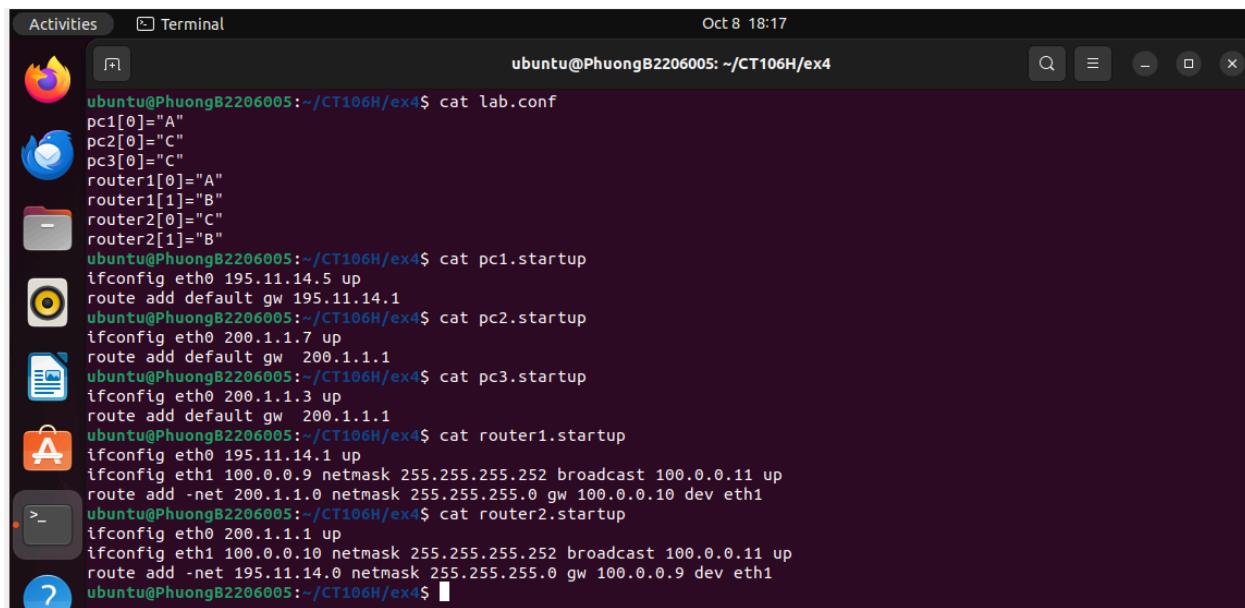
Answer: Prepare a lab for Ex4

```
$cd CT106H  
/CT106H$ mkdir ex4  
/CT106H/ex4$ mkdir pc1 pc2 pc3 router1 router2 shared  
/CT106H/ex4$ gedit lab.conf  
/CT106H/ex4$ gedit pc1.startup  
/CT106H/ex4$ gedit pc2.startup  
/CT106H/ex4$ gedit pc3.startup  
/CT106H/ex4$ gedit router1.startup  
/CT106H/ex4$ gedit router2.startup  
/CT106H/ex4$ cat lab.conf  
/CT106H/ex4$ cat pc1.startup  
/CT106H/ex4$ cat pc2.startup  
/CT106H/ex4$ cat pc3.startup  
/CT106H/ex4$ cat router1.startup  
/CT106H/ex4$ cat router2.startup
```



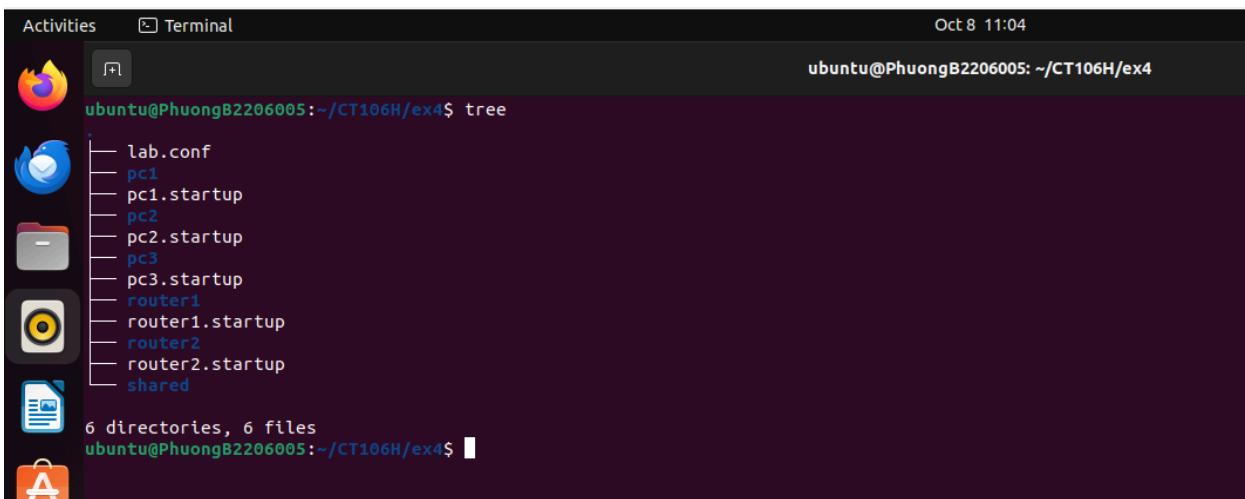
A screenshot of a Ubuntu desktop environment. A terminal window titled "Terminal" is open, showing a command-line session. The session starts with the user navigating to the directory "/CT106H/ex4". The user then creates several directories: "ex4", "pc1", "pc2", "pc3", "router1", "router2", and "shared". After creating these directories, the user runs "gedit" to edit configuration files for each: "lab.conf", "pc1.startup", "pc2.startup", "pc3.startup", "router1.startup", "router2.startup", and "router2.startup". The terminal window has a dark theme and is located in the top panel of the desktop.

```
ubuntu@PhuongB2206005:~$ cd CT106H
ubuntu@PhuongB2206005:~/CT106H$ mkdir ex4
ubuntu@PhuongB2206005:~/CT106H$ cd ex4
ubuntu@PhuongB2206005:~/CT106H/ex4$ mkdir pc1 pc2 pc3 router1 router2 shared
ubuntu@PhuongB2206005:~/CT106H/ex4$ gedit lab.conf
ubuntu@PhuongB2206005:~/CT106H/ex4$ gedit pc1.startup
ubuntu@PhuongB2206005:~/CT106H/ex4$ gedit pc2.startup
ubuntu@PhuongB2206005:~/CT106H/ex4$ gedit pc3.startup
ubuntu@PhuongB2206005:~/CT106H/ex4$ gedit router1.startup
ubuntu@PhuongB2206005:~/CT106H/ex4$ gedit router2.startup
ubuntu@PhuongB2206005:~/CT106H/ex4$
```



A screenshot of a Ubuntu desktop environment. A terminal window titled "Terminal" is open, showing a command-line session. The user runs "cat" to view the contents of "lab.conf", which contains host names: "pc1[0] = "A", "pc2[0] = "C", "pc3[0] = "C", "router1[0] = "A", "router1[1] = "B", "router2[0] = "C", "router2[1] = "B". The user then runs "cat" on each startup file: "pc1.startup", "pc2.startup", "pc3.startup", "router1.startup", and "router2.startup". Each startup file contains specific configuration for its respective host or router, including IP addresses and routing tables. The terminal window has a dark theme and is located in the top panel of the desktop.

```
ubuntu@PhuongB2206005:~/CT106H/ex4$ cat lab.conf
pc1[0] = "A"
pc2[0] = "C"
pc3[0] = "C"
router1[0] = "A"
router1[1] = "B"
router2[0] = "C"
router2[1] = "B"
ubuntu@PhuongB2206005:~/CT106H/ex4$ cat pc1.startup
ifconfig eth0 195.11.14.5 up
route add default gw 195.11.14.1
ubuntu@PhuongB2206005:~/CT106H/ex4$ cat pc2.startup
ifconfig eth0 200.1.1.7 up
route add default gw 200.1.1.1
ubuntu@PhuongB2206005:~/CT106H/ex4$ cat pc3.startup
ifconfig eth0 200.1.1.3 up
route add default gw 200.1.1.1
ubuntu@PhuongB2206005:~/CT106H/ex4$ cat router1.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
ubuntu@PhuongB2206005:~/CT106H/ex4$ cat router2.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
ubuntu@PhuongB2206005:~/CT106H/ex4$
```

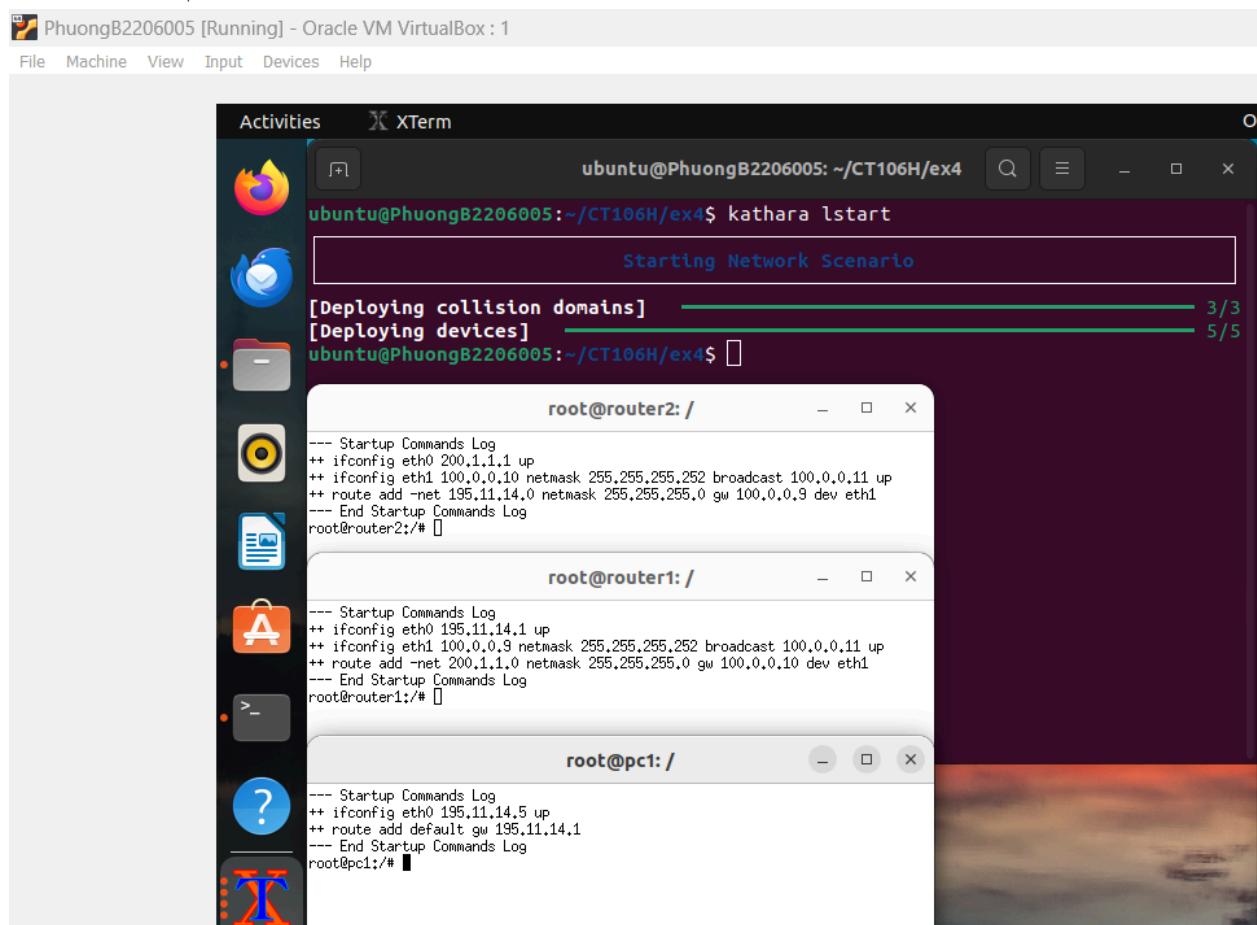


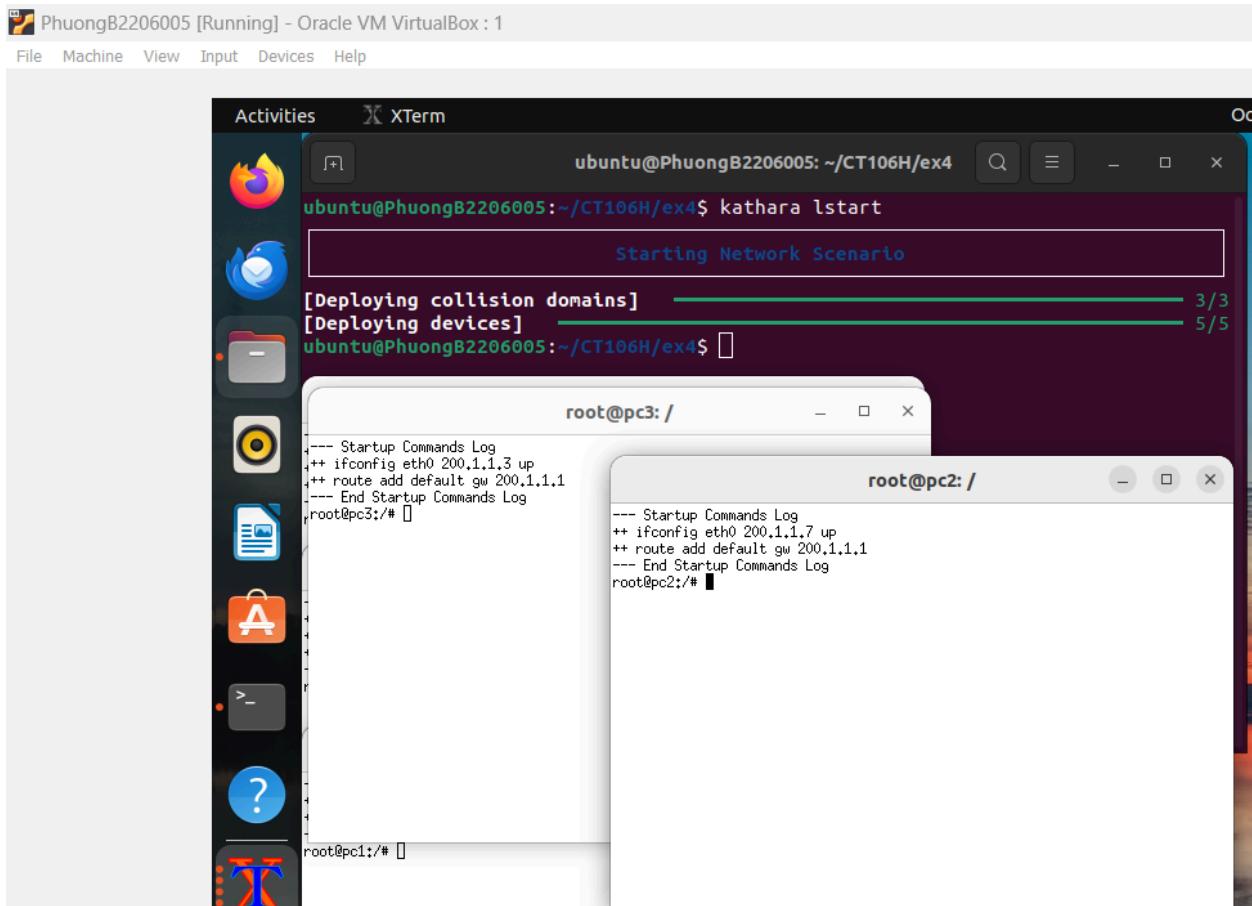
A screenshot of a Ubuntu desktop environment. A terminal window titled "Terminal" is open, showing a command-line session. The user runs "tree" to list the directory structure. The structure shows a root directory with sub-directories "lab.conf", "pc1", "pc2", "pc3", "pc4", "router1", "router2", and "shared". Under "pc1", "pc2", "pc3", and "pc4", there are corresponding ".startup" files. Under "router1" and "router2", there are ".startup" files. The terminal window has a dark theme and is located in the top panel of the desktop.

```
ubuntu@PhuongB2206005:~/CT106H/ex4$ tree
.
├── lab.conf
├── pc1
├── pc2
├── pc3
├── pc4
├── router1
├── router2
└── shared
6 directories, 6 files
ubuntu@PhuongB2206005:~/CT106H/ex4$
```

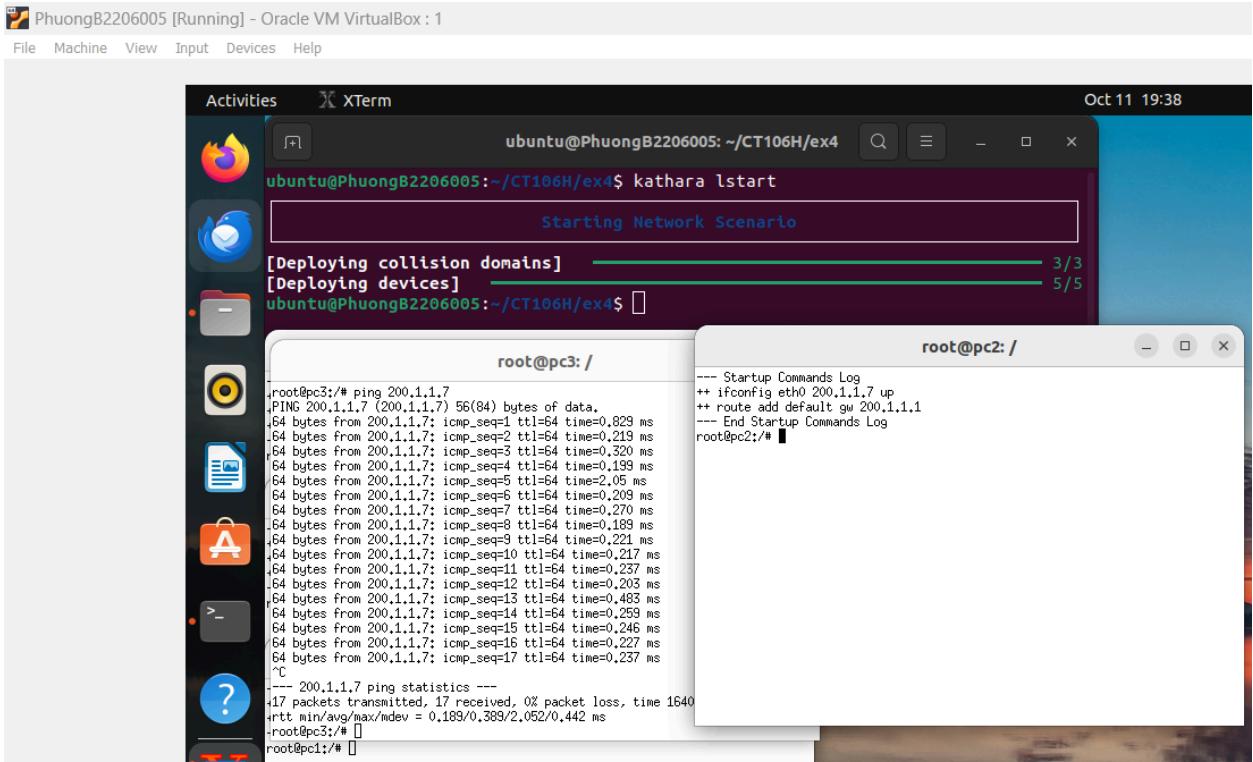
Start the prepared lab:

/CT106H/ex4\$ kathara lstart

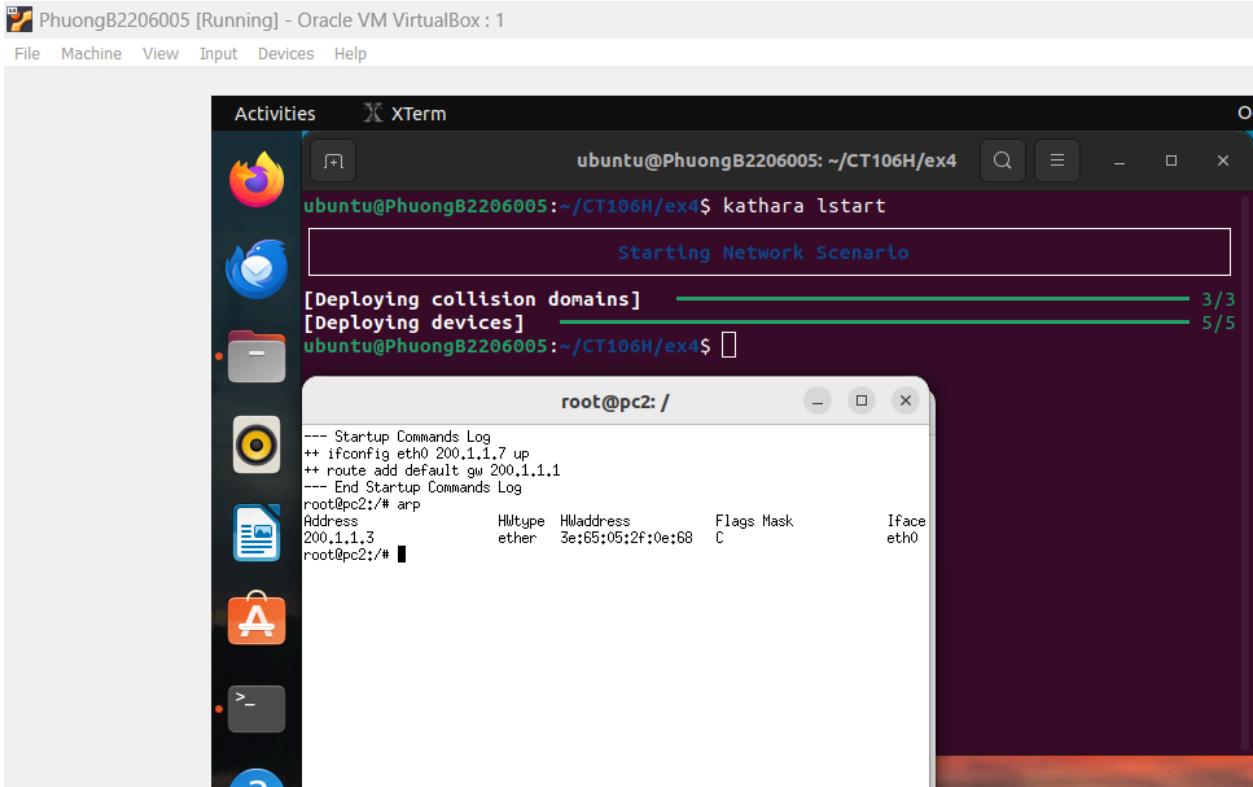




Ping pc2 from pc3



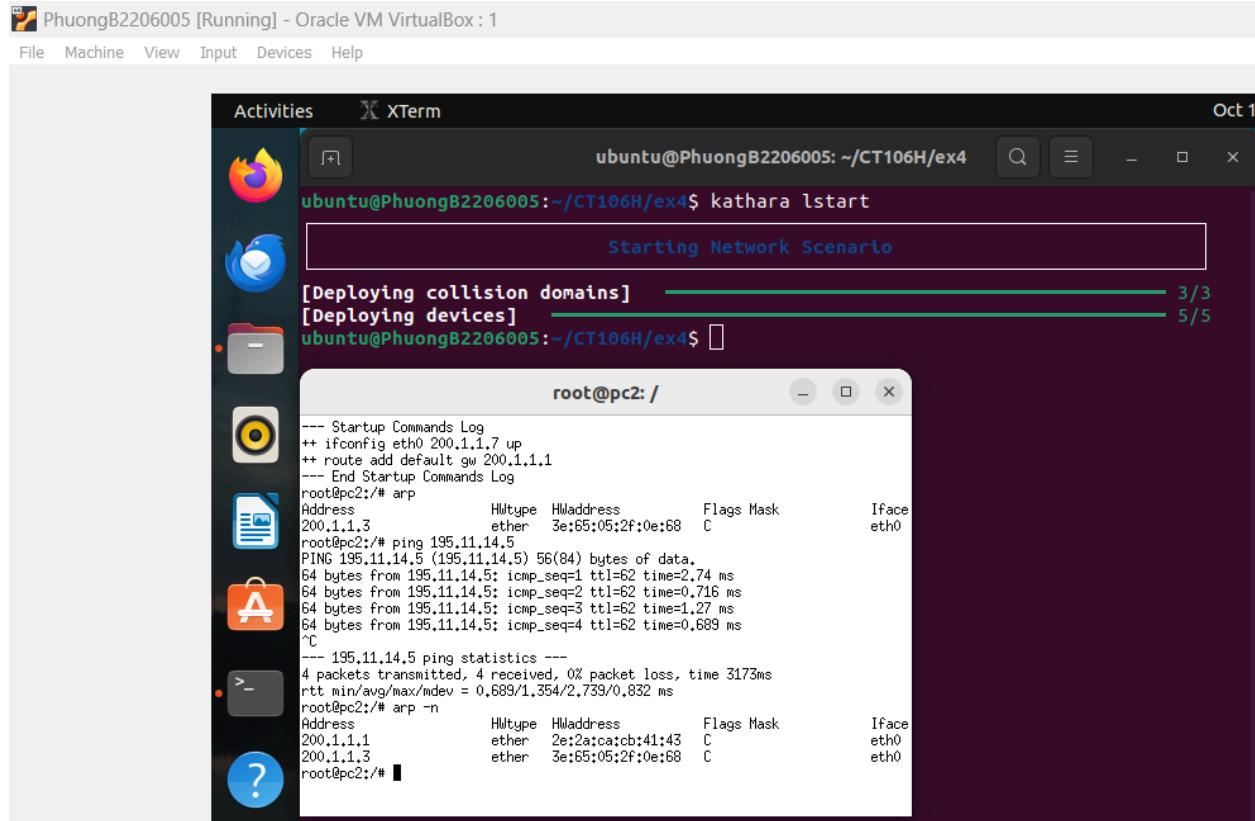
Communications are usually bi-directional. The receiver of the arp request learns the mac address of the other party, to avoid a new arp in opposite direction.



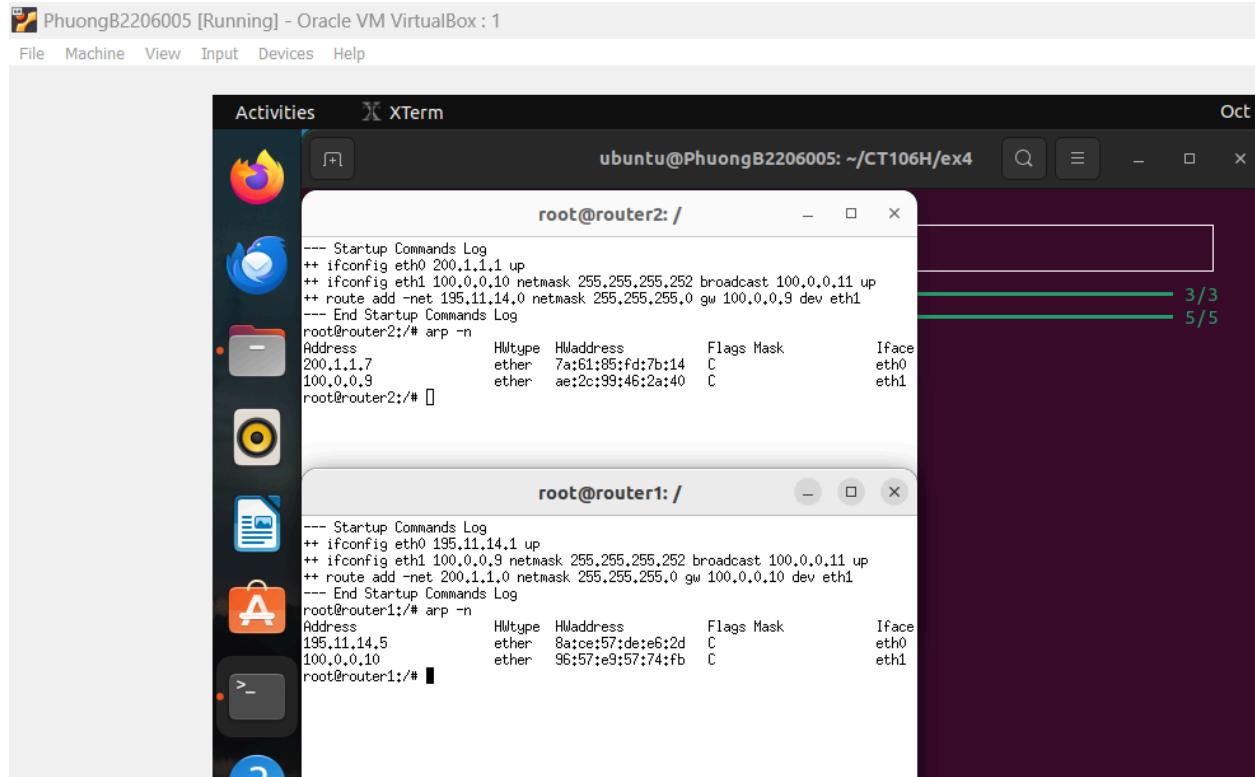
Inspect the arp cache (non local traffic):

Ping pc1 from pc2

```
#ping 195.11.14.5  
#arp -n
```



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.



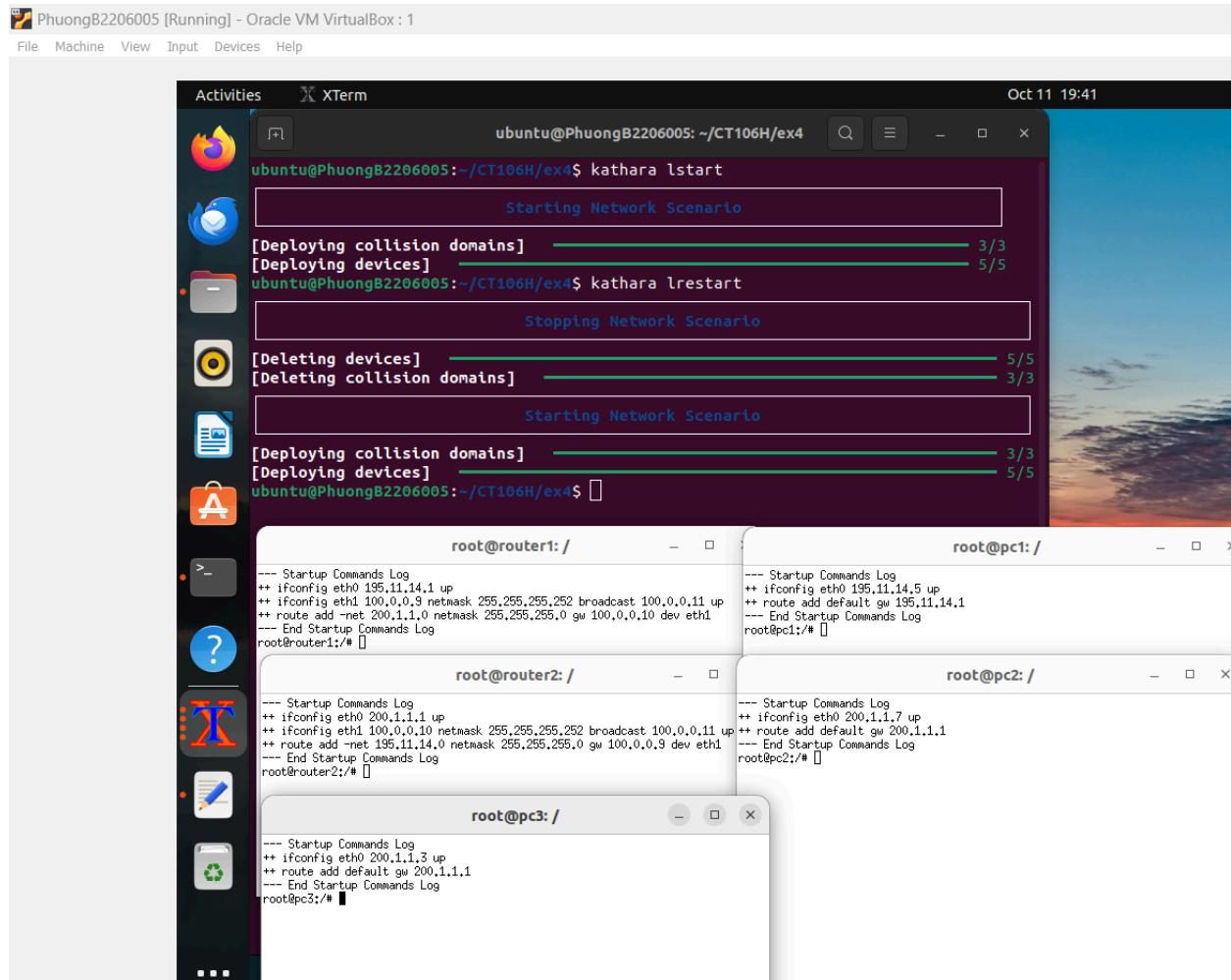
Routers perform arp too (hence have arp caches) anytime they have to send ip packets on an ethernet LAN

### Sniff arp traffic

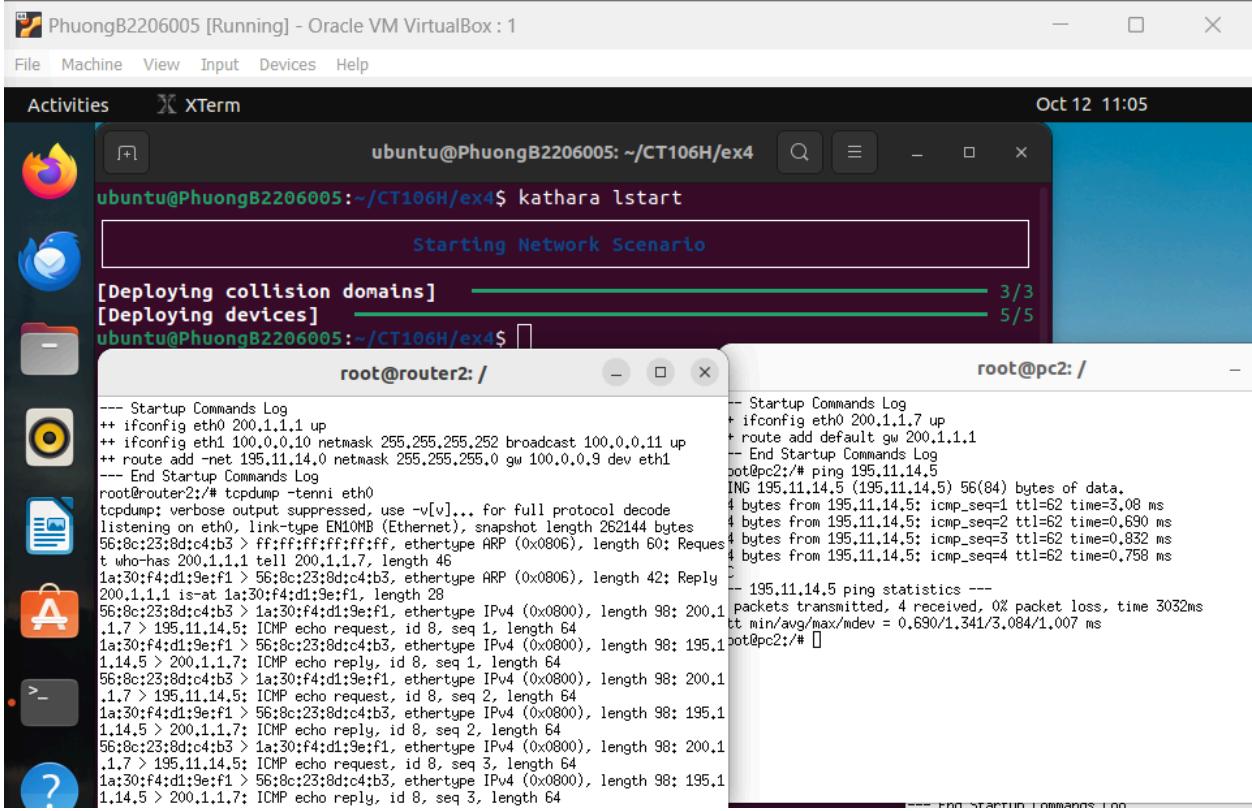
Restart the lab in order to clear arp caches

\$kathara lrestart

## CT106H – Computer Network



**Start to sniff**



On collision domain C:

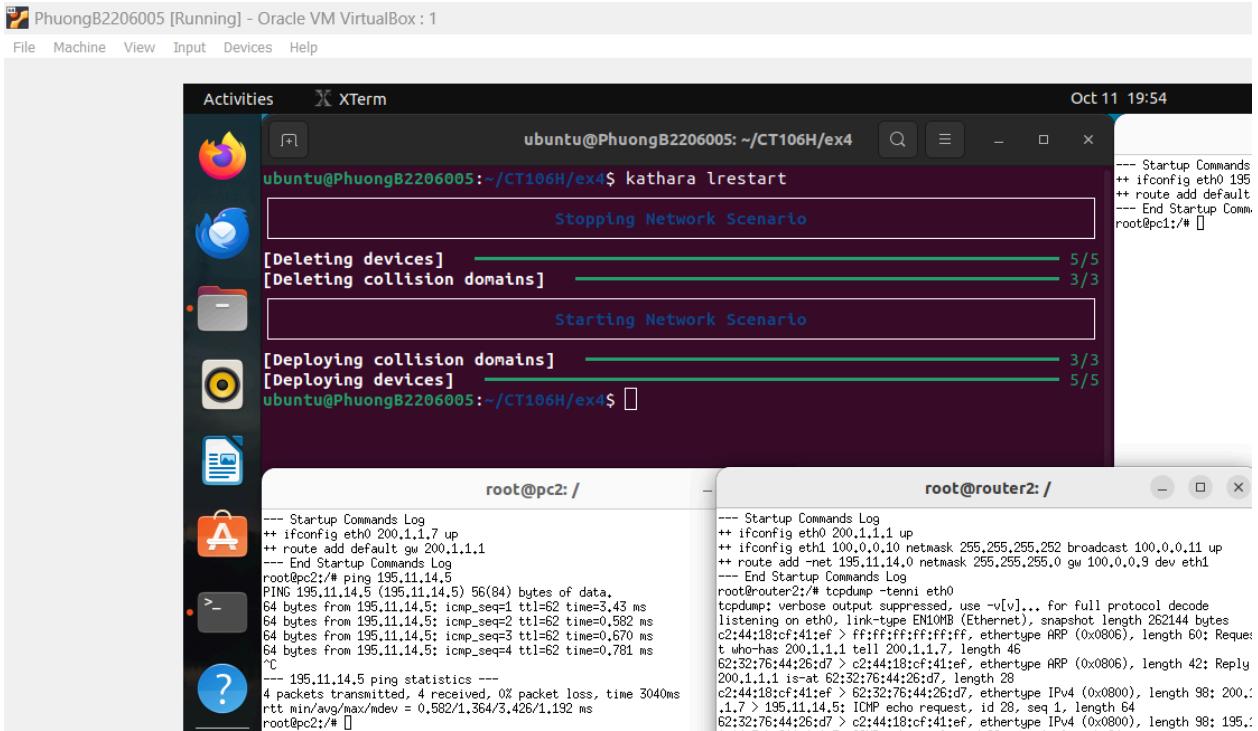
Restart the lab in order to clear arp caches

\$kathara restart

Using pc2 ping pc1]

Router2: #tcpdump -tenni eth0

CT106H – Computer Network



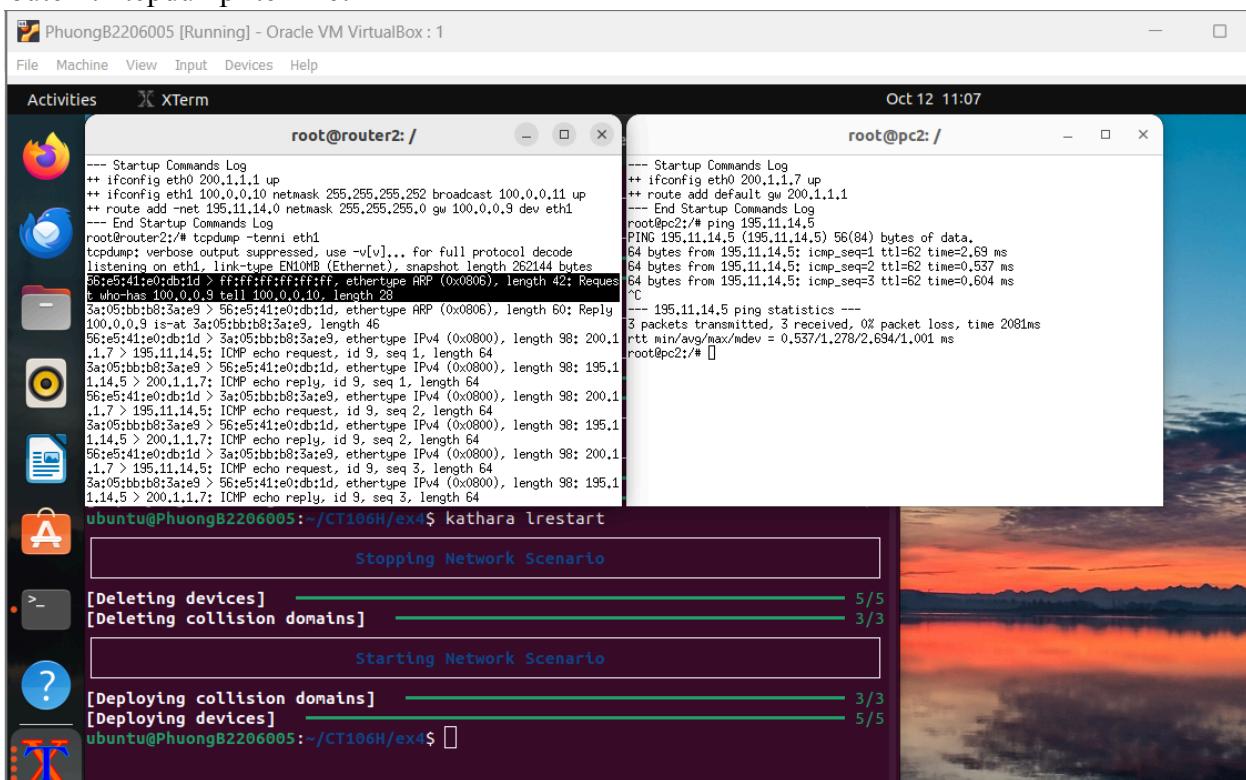
On collision domain B:

Restart the lab in order to clear arp caches

\$kathara restart

Using pc2 ping pc1

Router2: #tcpdump -tetti eth1



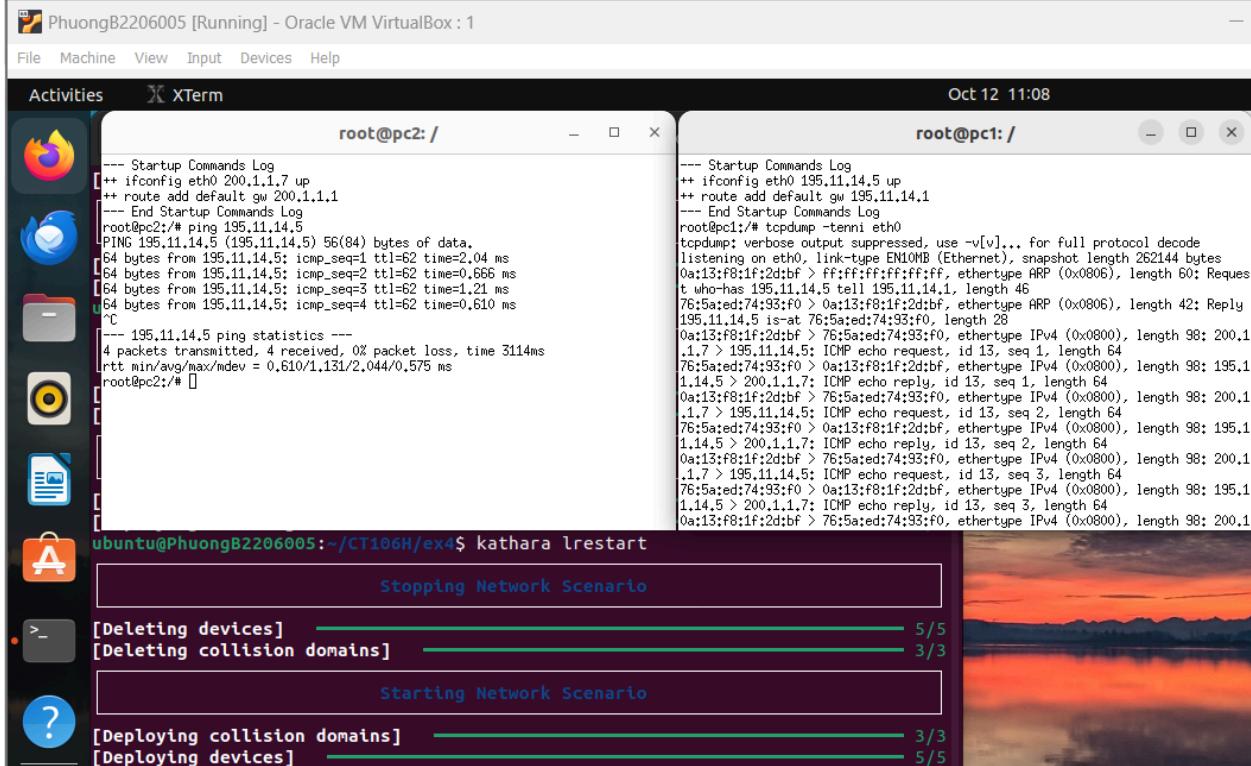
On collision domain A:

Restart the lab in order to clear arp caches

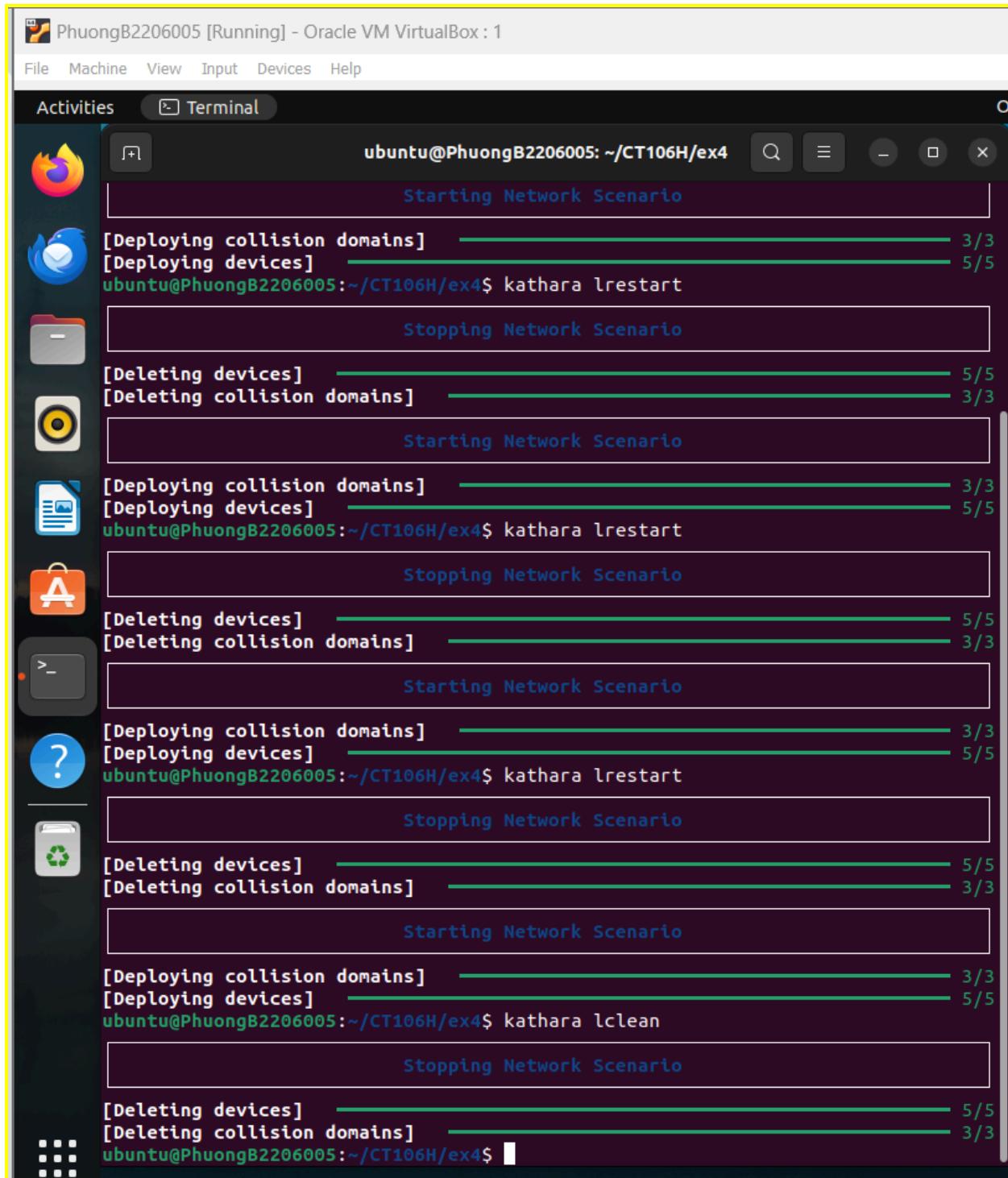
\$kathara lrestart

Using pc2 ping pc1

pc1: #tcpdump -tenui eth0



Before leaving, remember to halts all the devices of this lab



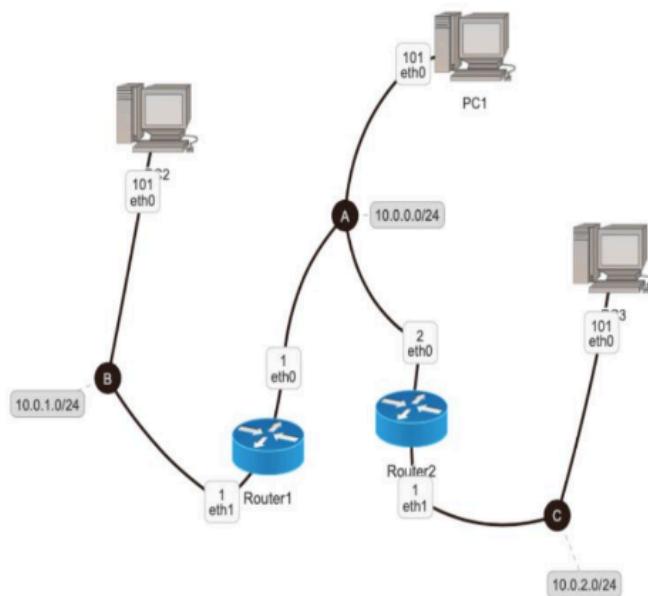
The screenshot shows a terminal window within the Oracle VM VirtualBox interface. The title bar reads "PhuongB2206005 [Running] - Oracle VM VirtualBox : 1". The terminal window has a dark background and displays a series of commands related to network scenario management:

```
ubuntu@PhuongB2206005: ~/CT106H/ex4
Starting Network Scenario
[Deploying collision domains] 3/3
[Deploying devices] 5/5
ubuntu@PhuongB2206005:~/CT106H/ex4$ kathara lrestart
Stopping Network Scenario
[Deleting devices] 5/5
[Deleting collision domains] 3/3
Starting Network Scenario
[Deploying collision domains] 3/3
[Deploying devices] 5/5
ubuntu@PhuongB2206005:~/CT106H/ex4$ kathara lrestart
Stopping Network Scenario
[Deleting devices] 5/5
[Deleting collision domains] 3/3
Starting Network Scenario
[Deploying collision domains] 3/3
[Deploying devices] 5/5
ubuntu@PhuongB2206005:~/CT106H/ex4$ kathara lrestart
Stopping Network Scenario
[Deleting devices] 5/5
[Deleting collision domains] 3/3
Starting Network Scenario
[Deploying collision domains] 3/3
[Deploying devices] 5/5
ubuntu@PhuongB2206005:~/CT106H/ex4$ kathara lclean
Stopping Network Scenario
[Deleting devices] 5/5
[Deleting collision domains] 3/3
ubuntu@PhuongB2206005:~/CT106H/ex4$
```

Ex5:

## Exercise 5

Construct the following network



Prepared for ex5

```
/CT106H/ex5$ mkdir pc1 pc2 pc3 router1 router2 shared  
/CT106H/ex5$ gedit lab.conf  
/CT106H/ex5$ gedit pc1.startup  
/CT106H/ex5$ gedit pc2.startup  
/CT106H/ex5$ gedit pc3.startup  
/CT106H/ex5$ gedit router1.startup  
/CT106H/ex5$ gedit router2.startup  
/CT106H/ex5$ cat lab.conf  
/CT106H/ex5$ cat pc1.startup  
/CT106H/ex5$ cat pc2.startup  
/CT106H/ex5$ cat pc3.startup  
/CT106H/ex5$ cat router1.startup  
/CT106H/ex5$ cat router2.startup
```

The terminal window shows the following command history:

```
Activities Terminal Oct 8 11:58  
ubuntu@PhuongB2206005: ~/CT106H/ex5  
ubuntu@PhuongB2206005:~$ cd CT106H  
ubuntu@PhuongB2206005:~/CT106H$ mkdir ex5  
ubuntu@PhuongB2206005:~/CT106H$ cd ex5  
ubuntu@PhuongB2206005:~/CT106H/ex5$ mkdir pc1 pc2 pc3 router1 router2 shared  
ubuntu@PhuongB2206005:~/CT106H/ex5$ gedit lab.conf  
ubuntu@PhuongB2206005:~/CT106H/ex5$ gedit router1.startup  
ubuntu@PhuongB2206005:~/CT106H/ex5$ gedit router2.startup  
ubuntu@PhuongB2206005:~/CT106H/ex5$ gedit pc1.startup  
ubuntu@PhuongB2206005:~/CT106H/ex5$ gedit pc2.startup  
ubuntu@PhuongB2206005:~/CT106H/ex5$ gedit pc3.startup
```

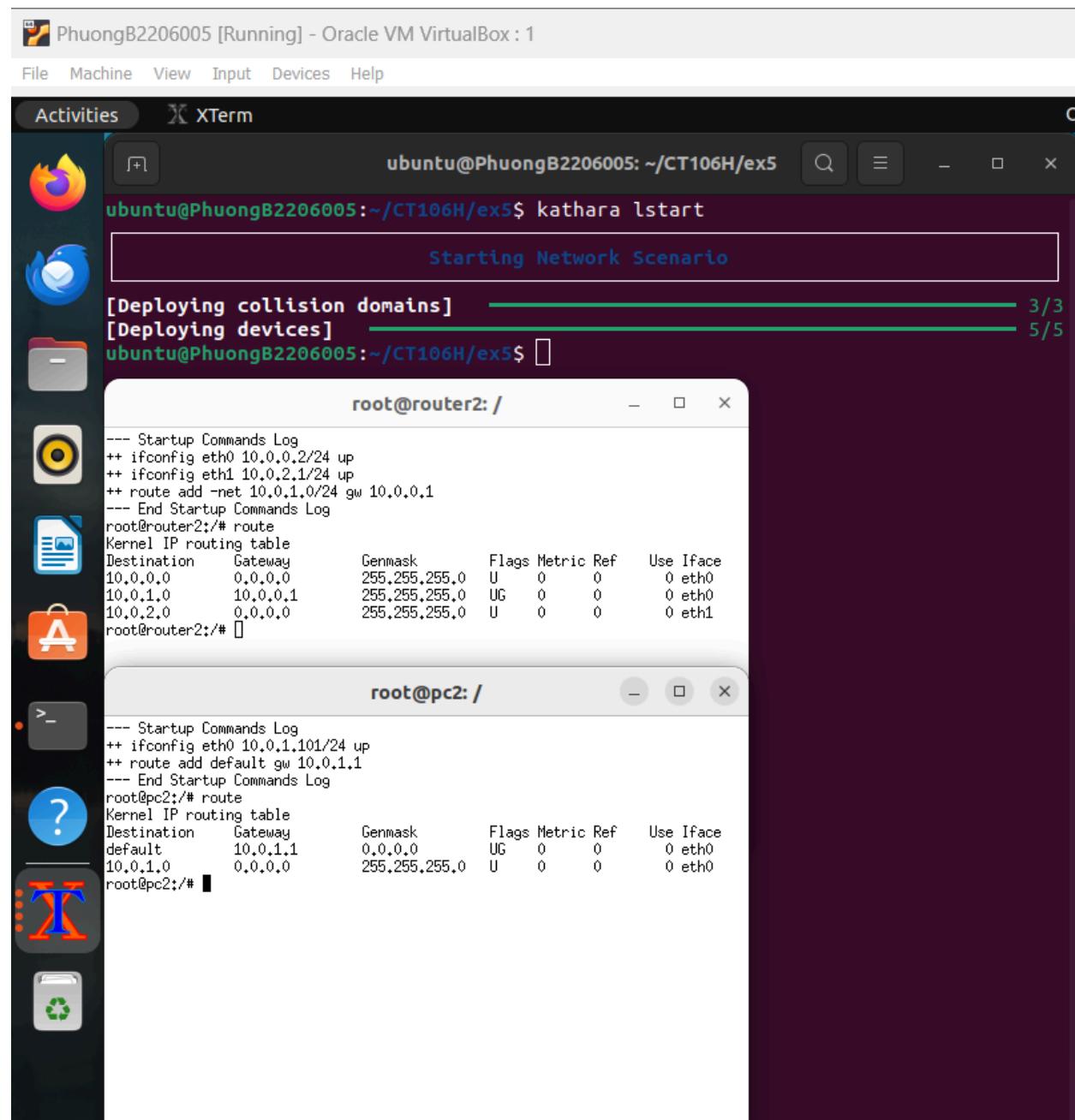
## CT106H – Computer Network

Activities Terminal Oct 8 12:30  
ubuntu@PhuongB2206005:~/CT106H/ex5\$ cat lab.conf  
pc1[0]=A  
pc2[0]=B  
pc3[0]=C  
router1[0]=A  
router1[1]=B  
router2[0]=A  
router2[1]=C  
ubuntu@PhuongB2206005:~/CT106H/ex5\$ 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  
ubuntu@PhuongB2206005:~/CT106H/ex5\$ cat pc2.startup  
ifconfig eth0 10.0.1.101/24 up  
route add default gw 10.0.1.1  
ubuntu@PhuongB2206005:~/CT106H/ex5\$ cat pc3.startup  
ifconfig eth0 10.0.2.101/24 up  
route add default gw 10.0.2.1  
ubuntu@PhuongB2206005:~/CT106H/ex5\$ 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  
ubuntu@PhuongB2206005:~/CT106H/ex5\$ 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  
ubuntu@PhuongB2206005:~/CT106H/ex5\$

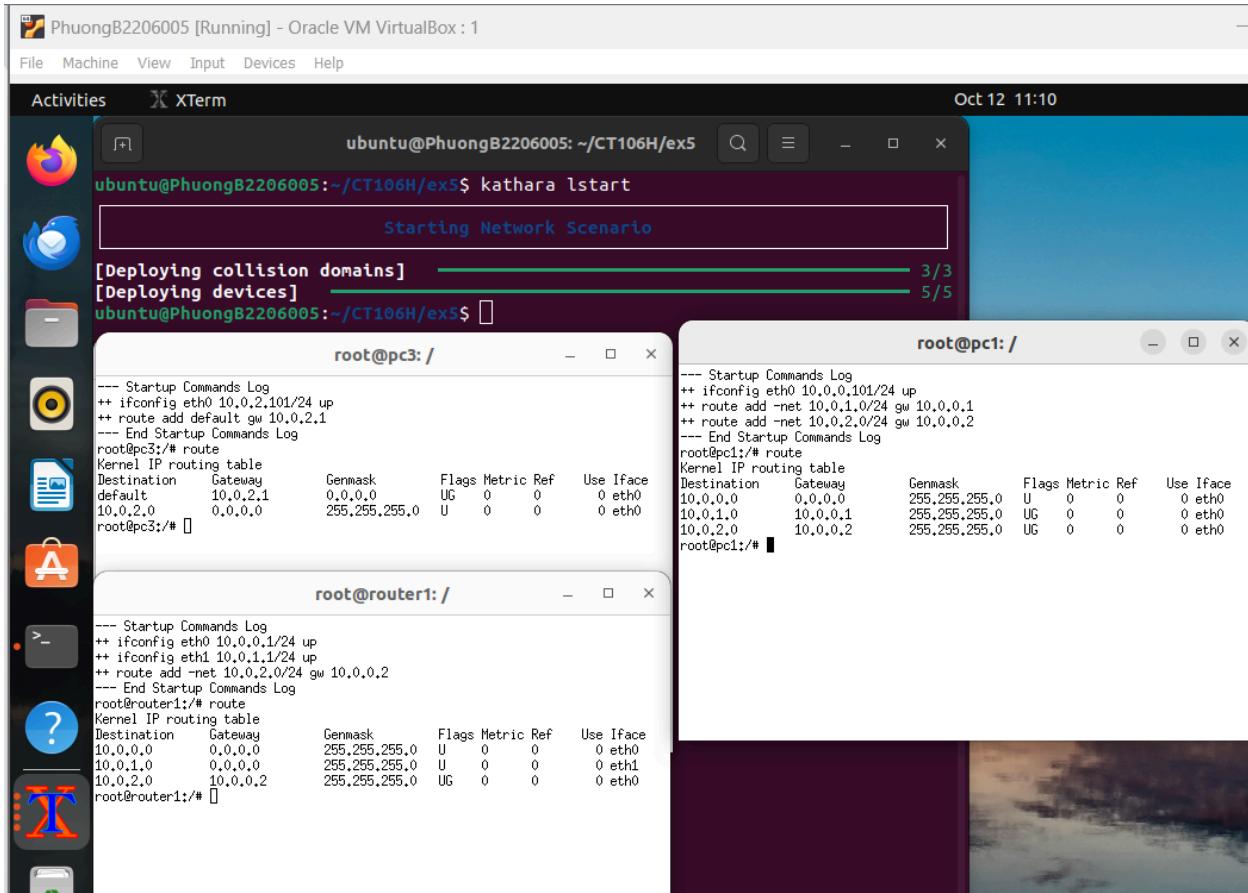
Start kathara  
/CT106H/ex5\$ kathara lstart

Activities XTerm Oct 8 12:31  
ubuntu@PhuongB2206005:~/CT106H/ex5\$ kathara lstart  
Starting Network Scenario  
[Deploying collision domains]  
[Deploying devices]  
ubuntu@PhuongB2206005:~/CT106H/ex5\$  
root@router2: /  
--- 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:/#  
root@pc1: /  
--- 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:/#  
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@pc3: /  
--- 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:/#  
root@router1: /  
--- 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:/#

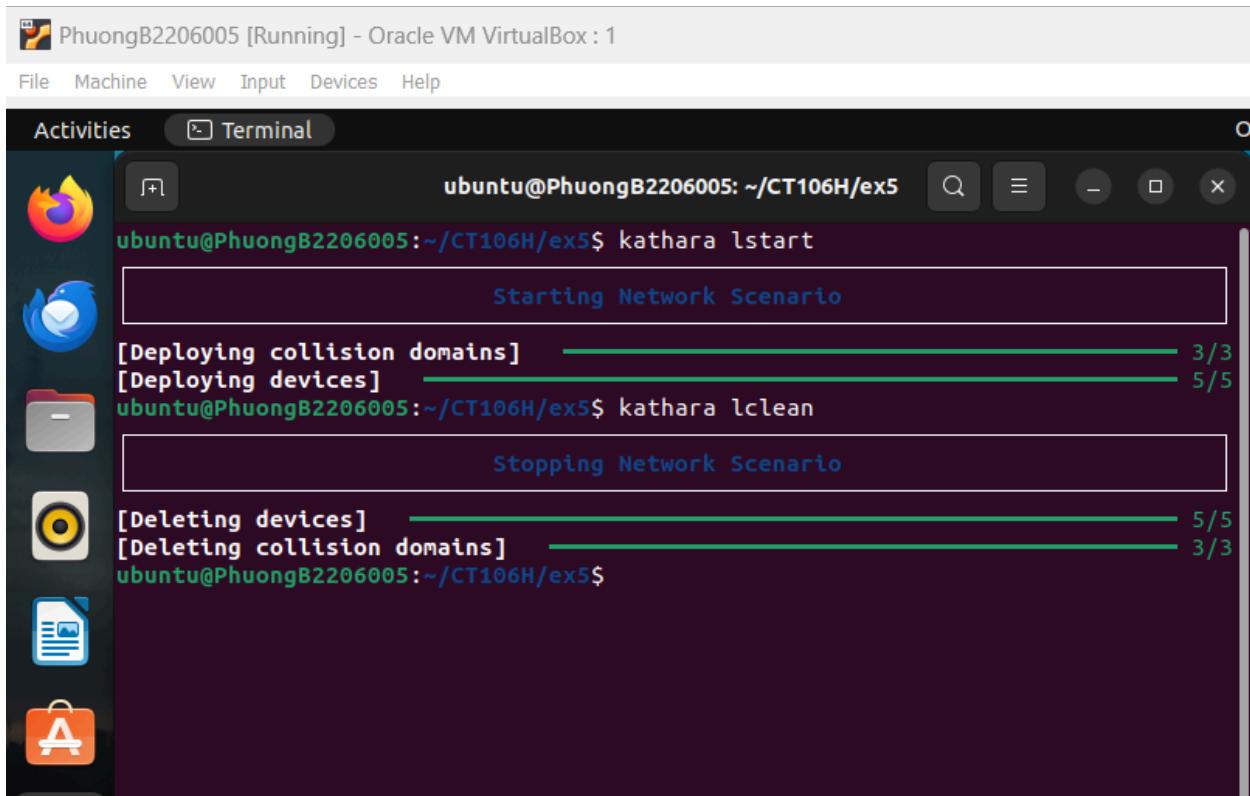
Check result:



## CT106H – Computer Network



Before leaving, remember to halts all the devices of this lab:

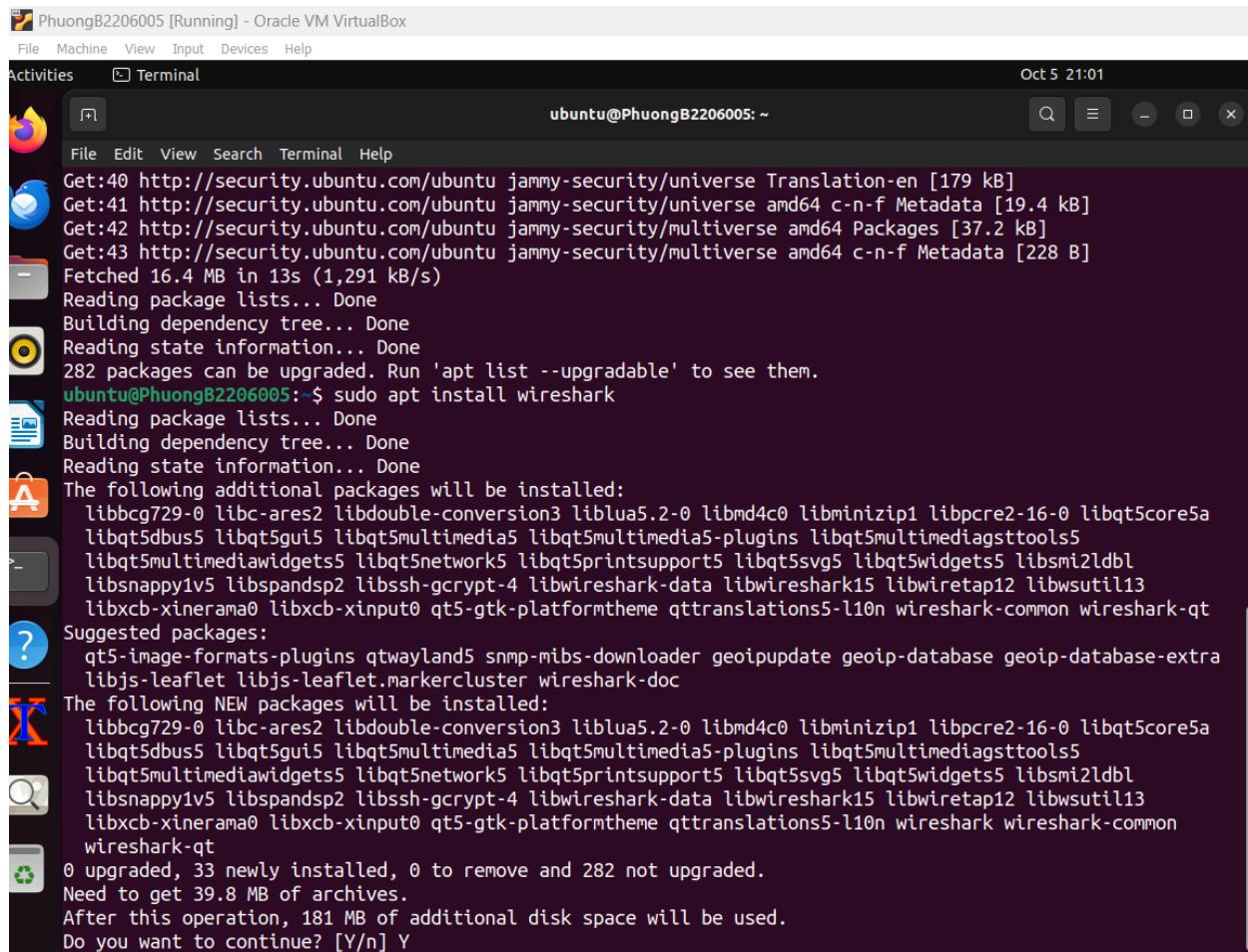


The screenshot shows a terminal window titled "ubuntu@PhuongB2206005: ~/CT106H/ex5\$". The window displays the following command and its execution:

```
ubuntu@PhuongB2206005:~/CT106H/ex5$ kathara lstart
Starting Network Scenario
[Deploying collision domains] 3/3
[Deploying devices] 5/5
ubuntu@PhuongB2206005:~/CT106H/ex5$ kathara lclean
Stopping Network Scenario
[Deleting devices] 5/5
[Deleting collision domains] 3/3
ubuntu@PhuongB2206005:~/CT106H/ex5$
```

Use Wireshark: Install Wireshark

\$sudo apt install wireshark



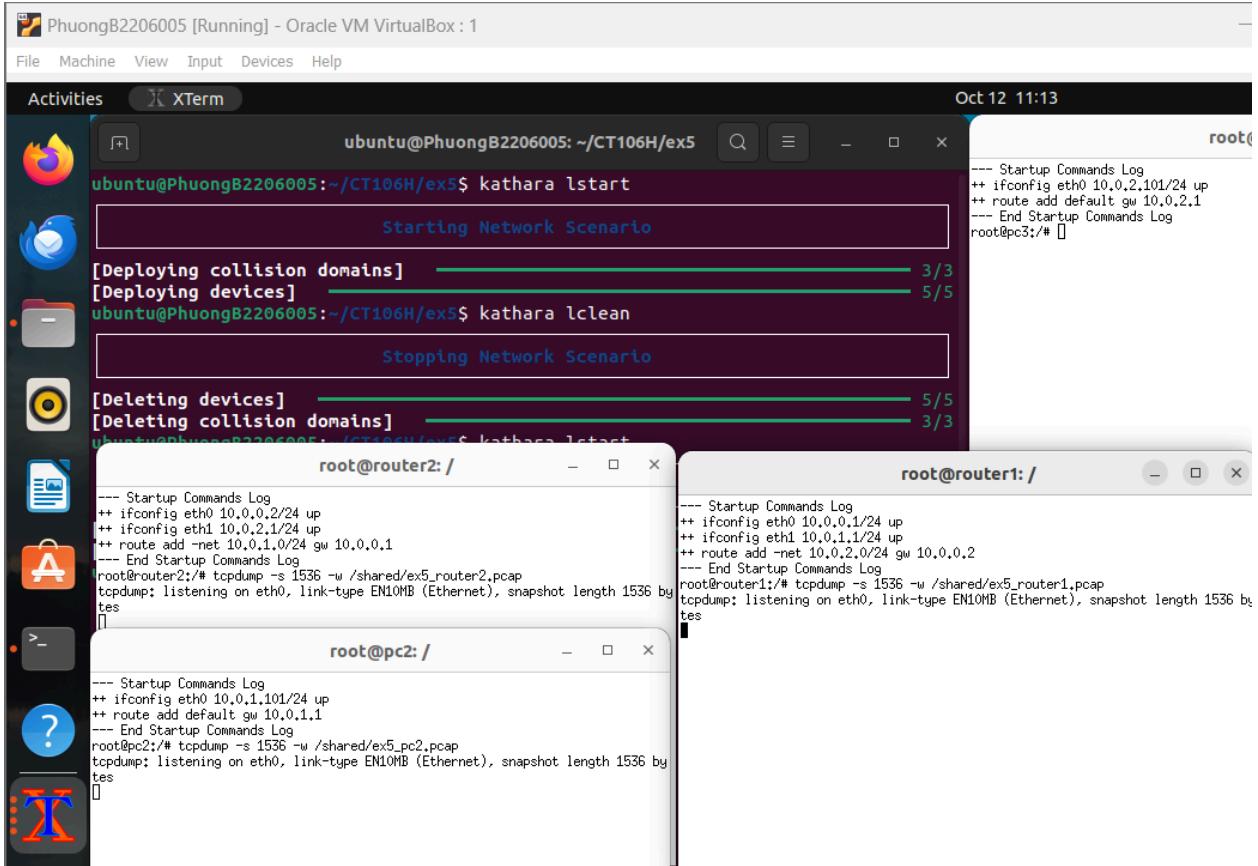
The screenshot shows a terminal window titled "Terminal" running on an Ubuntu system. The terminal output is as follows:

```
ubuntu@PhuongB2206005: ~
Get:40 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [179 kB]
Get:41 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [19.4 kB]
Get:42 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [37.2 kB]
Get:43 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [228 B]
Fetched 16.4 MB in 13s (1,291 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
282 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@PhuongB2206005: $ sudo apt install wireshark
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
 libbcg729-0 libc-ares2 libdouble-conversion3 liblua5.2-0 libmd4c0 libminizip1 libpcre2-16-0 libqt5core5a
 libqt5dbus5 libqt5gui5 libqt5multimedia5 libqt5multimedia5-plugins libqt5multimediasupports5
 libqt5multimediacore5 libqt5multimediamodels5 libqt5multimediamotors5 libqt5multimediacounters5
 libqt5multimediacalculations5 libqt5multimediacalculations5-base libqt5multimediacalculations5-dev
 libsnappy1v5 libspandsp2 libssh-gcrypt-4 libwireshark-data libwireshark15 libwiretap12 libwsutil13
 libxcb-xinerama0 libxcb-xinput0 qt5-gtk-platformtheme qttranslations5-l10n wireshark-common wireshark-qt
Suggested packages:
 qt5-image-formats-plugins qtwayland5 snmp-mibs-downloader geoipupdate geoip-database geoip-database-extra
 libjs-leaflet libjs-leaflet.markercluster wireshark-doc
The following NEW packages will be installed:
 libbcg729-0 libc-ares2 libdouble-conversion3 liblua5.2-0 libmd4c0 libminizip1 libpcre2-16-0 libqt5core5a
 libqt5dbus5 libqt5gui5 libqt5multimedia5 libqt5multimedia5-plugins libqt5multimediasupports5
 libqt5multimediacore5 libqt5multimediamodels5 libqt5multimediamotors5 libqt5multimediacounters5
 libqt5multimediacalculations5 libqt5multimediacalculations5-base libqt5multimediacalculations5-dev
 libsnappy1v5 libspandsp2 libssh-gcrypt-4 libwireshark-data libwireshark15 libwiretap12 libwsutil13
 wireshark-qt
0 upgraded, 33 newly installed, 0 to remove and 282 not upgraded.
Need to get 39.8 MB of archives.
After this operation, 181 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

On pc2, type: tcpdump -s 1536 -w /shared/ex5\_pc2.pcap

On router1, type: tcpdump -s 1536 -w /shared/ex5\_router1.pcap

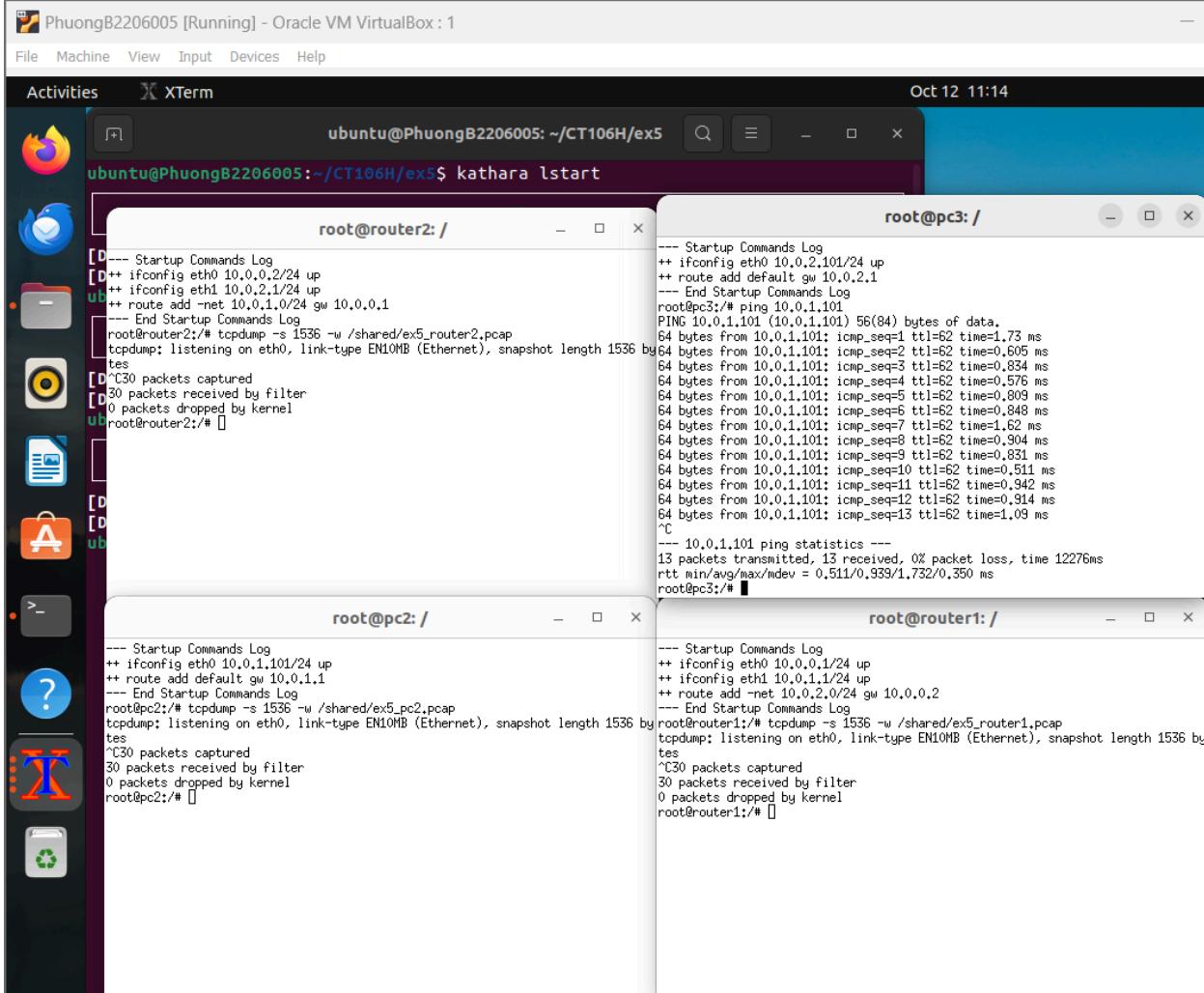
On router2, type: tcpdump -s 1536 -w /shared/ex5\_router2.pcap



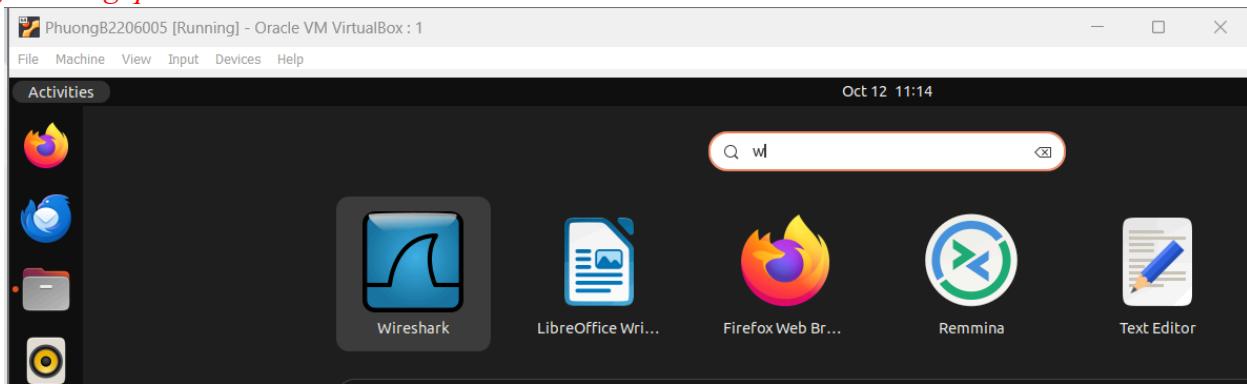
*On pc3, send packets to pc2 using the command ping 10.0.1.101 , wait for about 10 seconds and:*

- Stop the ping command
- Stop the tcpdump on pc2, router1 and router2

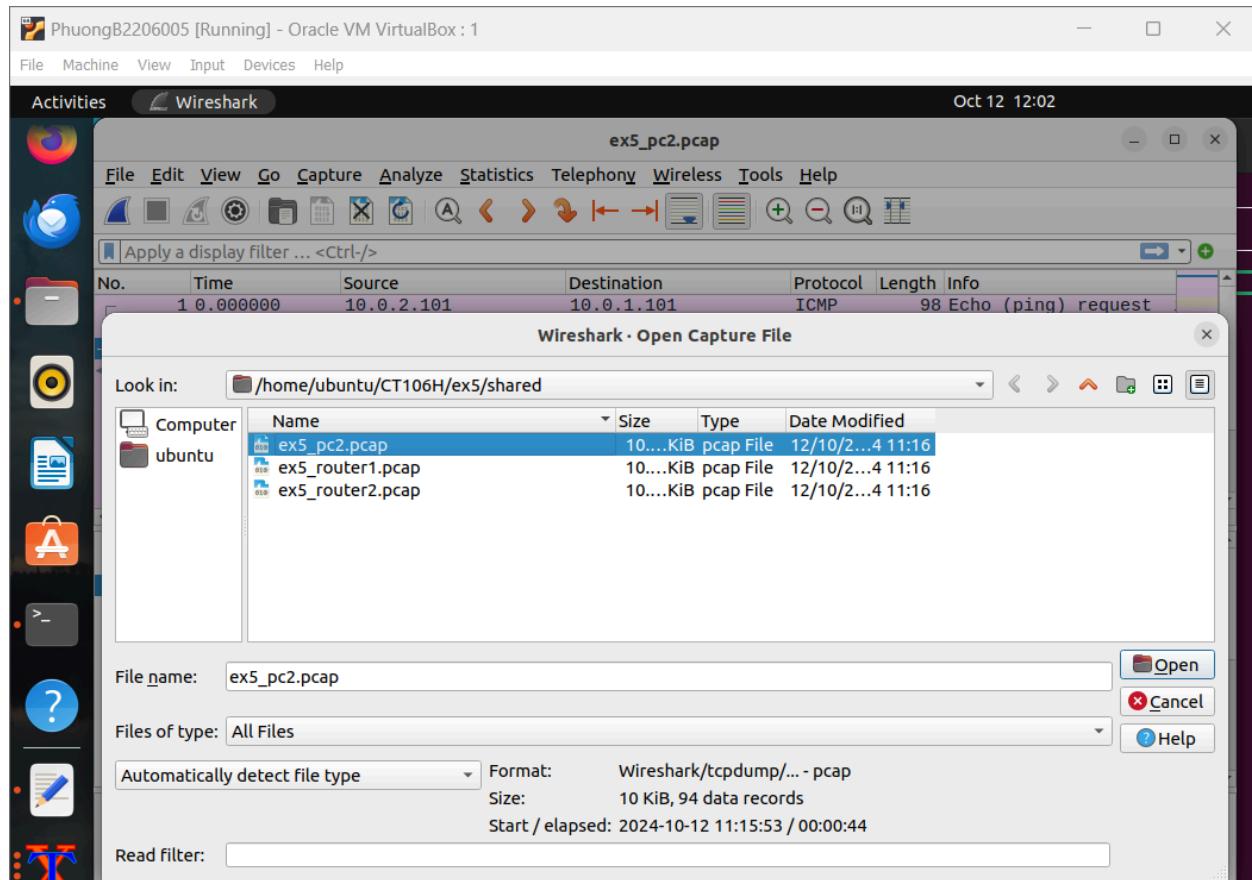
## CT106H – Computer Network

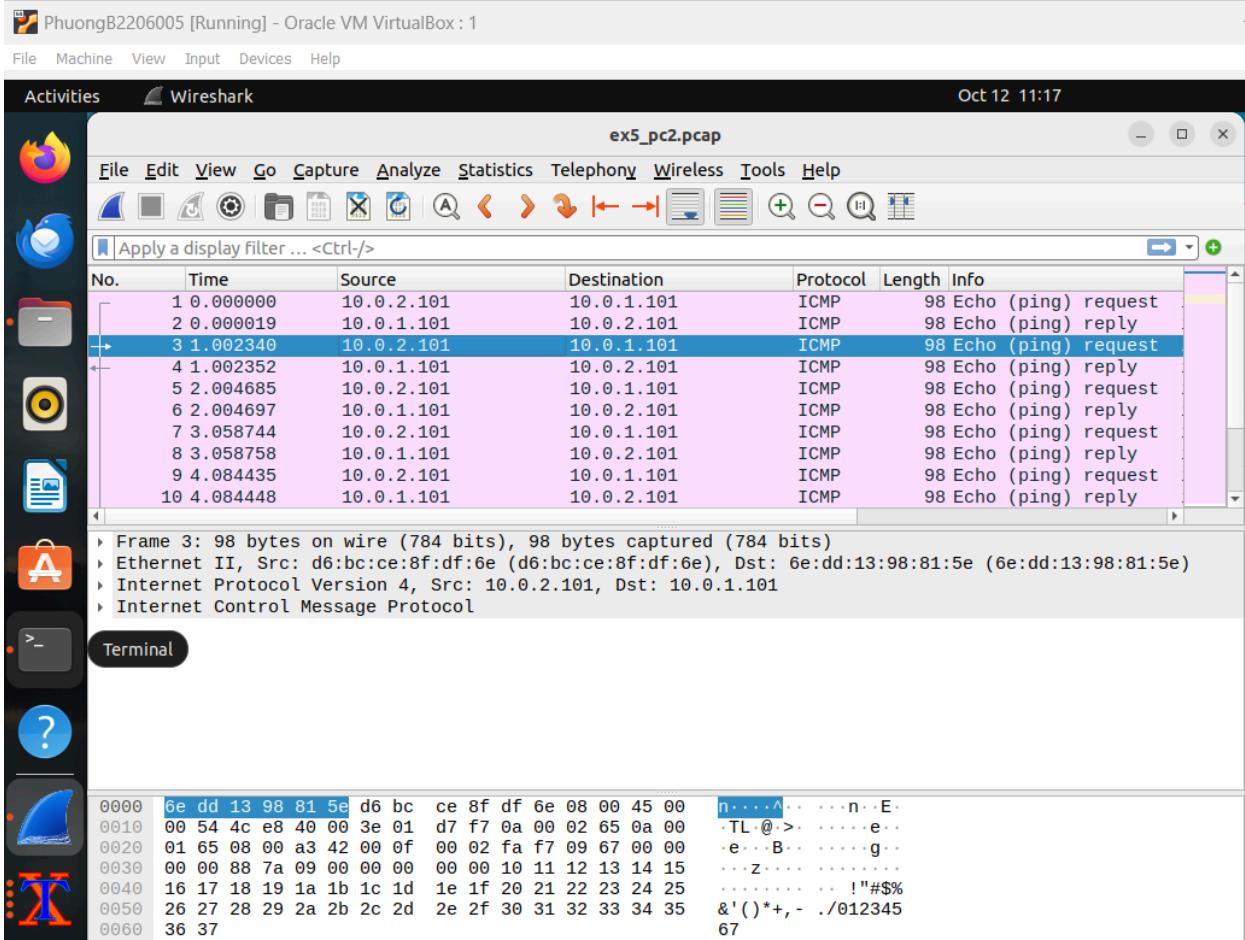


*On the Ubuntu, open Ex5\_pc2.pcap using Wireshark, select the frame #3 and answer the following questions:*



## CT106H – Computer Network

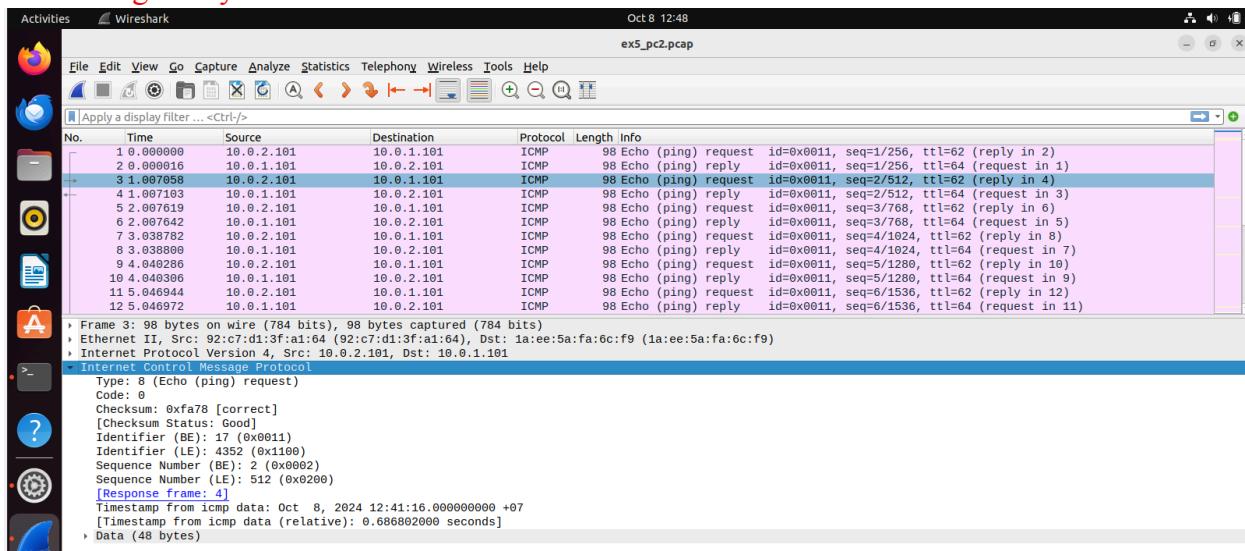




- Size of frame in bytes?

Answer: Size of frame is 98 bytes

- 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:

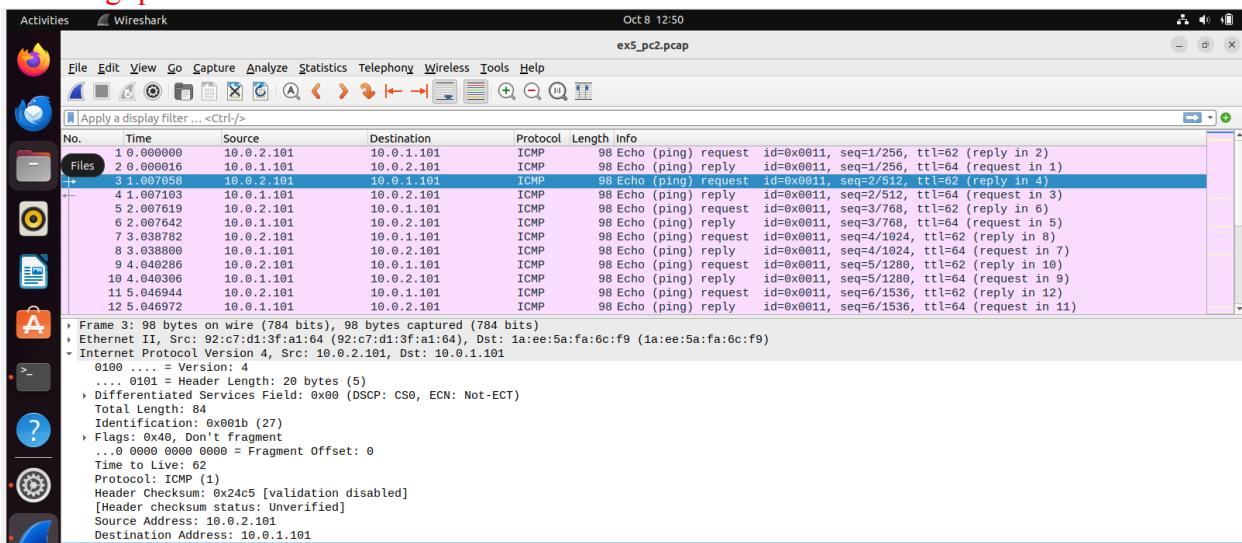
The protocol is ICMP

This protocol operates on network layer

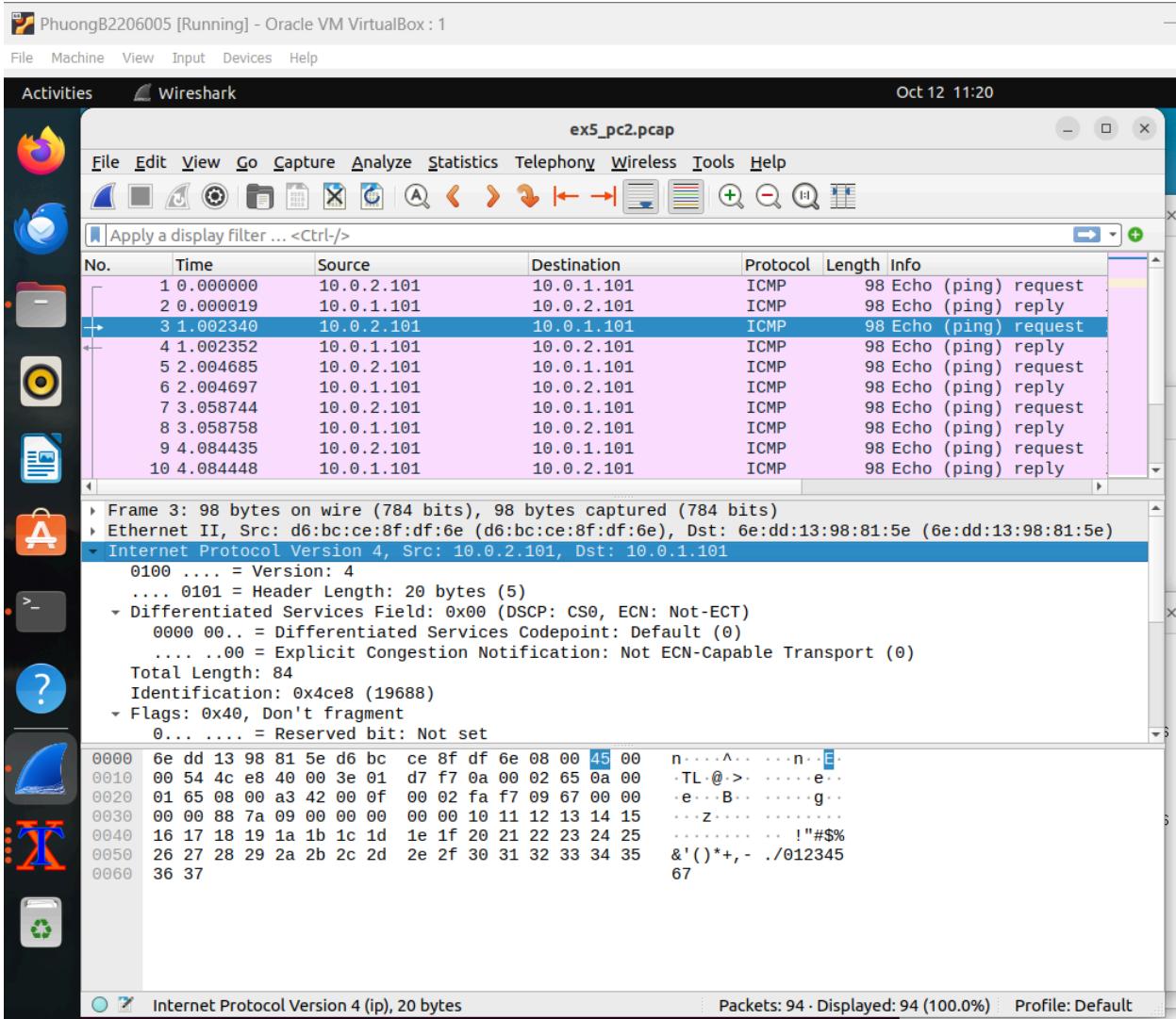
The content of the message is below

Size of this message is 48 bytes

On the Ubuntu, open Ex5\_pc2.pcap using Wireshark, select the frame #3 and answer the following questions:



- Select Header Internet Protocol Version 4



→ what are the IP addresses of the source and destination hosts?

Answer: IP addresses: source: 10.0.2.101 destination: 10.0.1.101

→ What is the length of the IP packet header? What fields does the Header include? How long is each field (Bytes)

Answer:

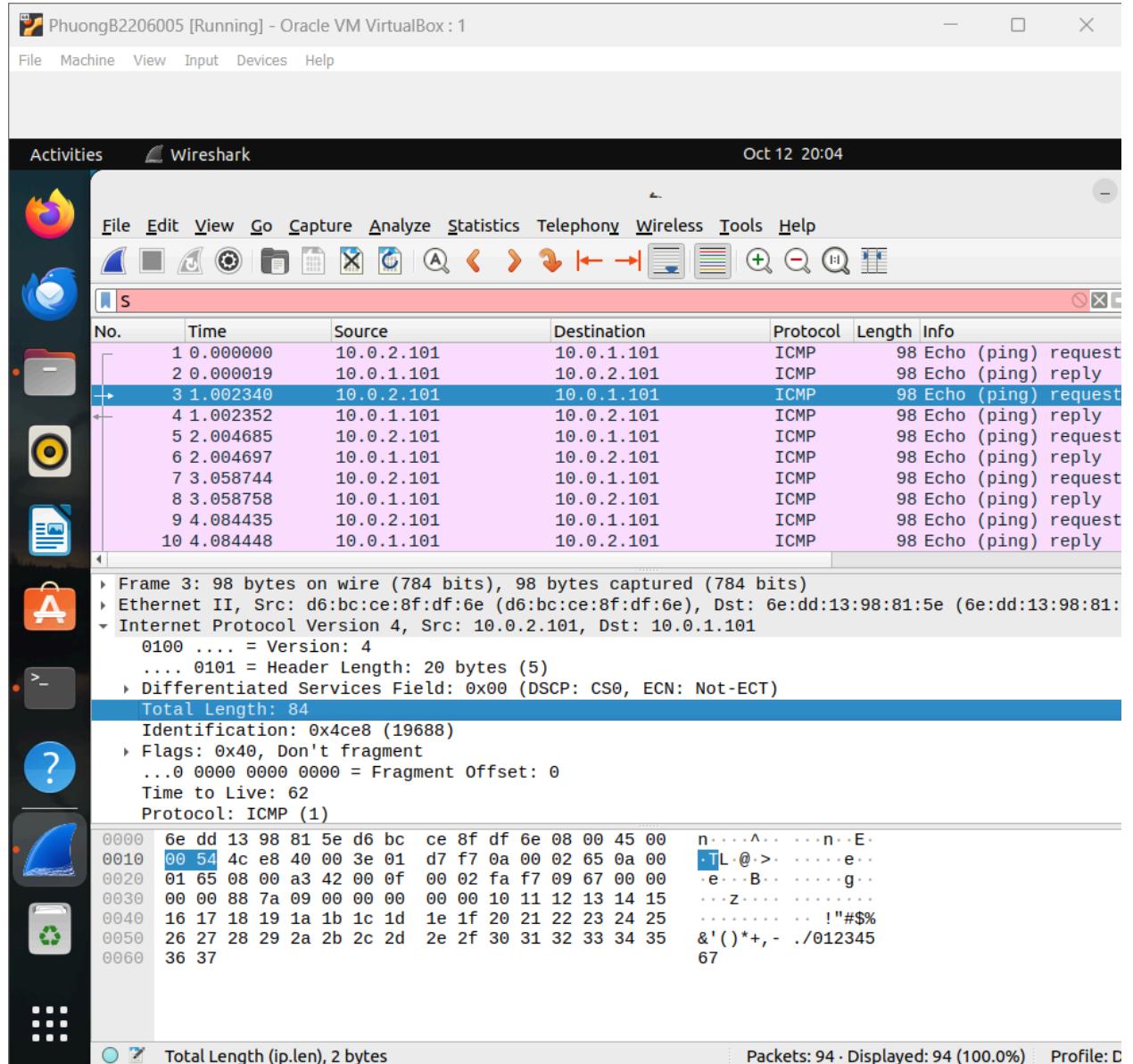
The length of the IP packet header: 20 bytes

Fields of the Header:

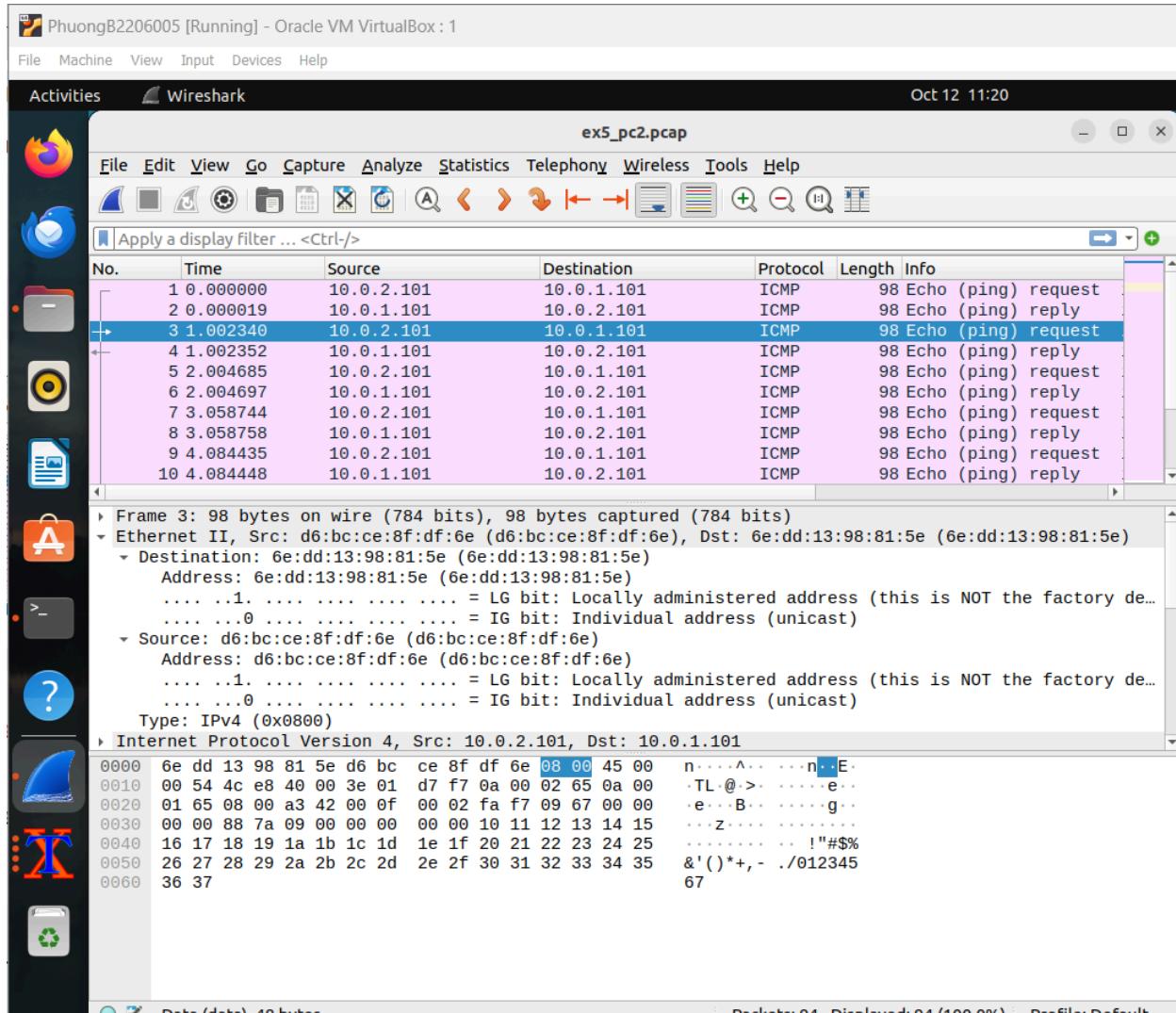
- Version & Header Length (1 byte)
- Differentiated Services Codepoint & Explicit Congestion Notification (1 byte)
- Total Length (2 bytes)
- Identification (2 bytes)
- Flags & Fragment Offset (2 bytes)
- Time To Live (1 byte)
- Protocol (1 byte)
- Header Checksum (2 bytes)
- Source Address (4 bytes)

- Destination Address (4 bytes)

What is the length of the Total Length field (Bytes). →the length of the Total Length field is 2 bytes



Select Header Ethernet II: What are the MAC addresses of the source and the destination hosts?



Answer: MAC addresses:

- source: d6:bc:ce:8f:df:6e
- destination: 6e:dd:13:98:81:5e

**What is the Type value?**

Answer: Type: IPv4 (0x0800)

-----END-----