katLAB 1

CONSTRUCT A SIMPLE NETWORK

Name: Lê Hoàng Minh Quân

ID: B2206008

Group: M02

Exercise 1:

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

kathara vstart -n pc2 --eth 0:A

Wireshark ~/capture.pcap

pc1:

Ifconfig eth0 10.0.0.1 netmask 255.255.255.0 broadcast 10.0.0.255 up

Ping 10.0.0.2

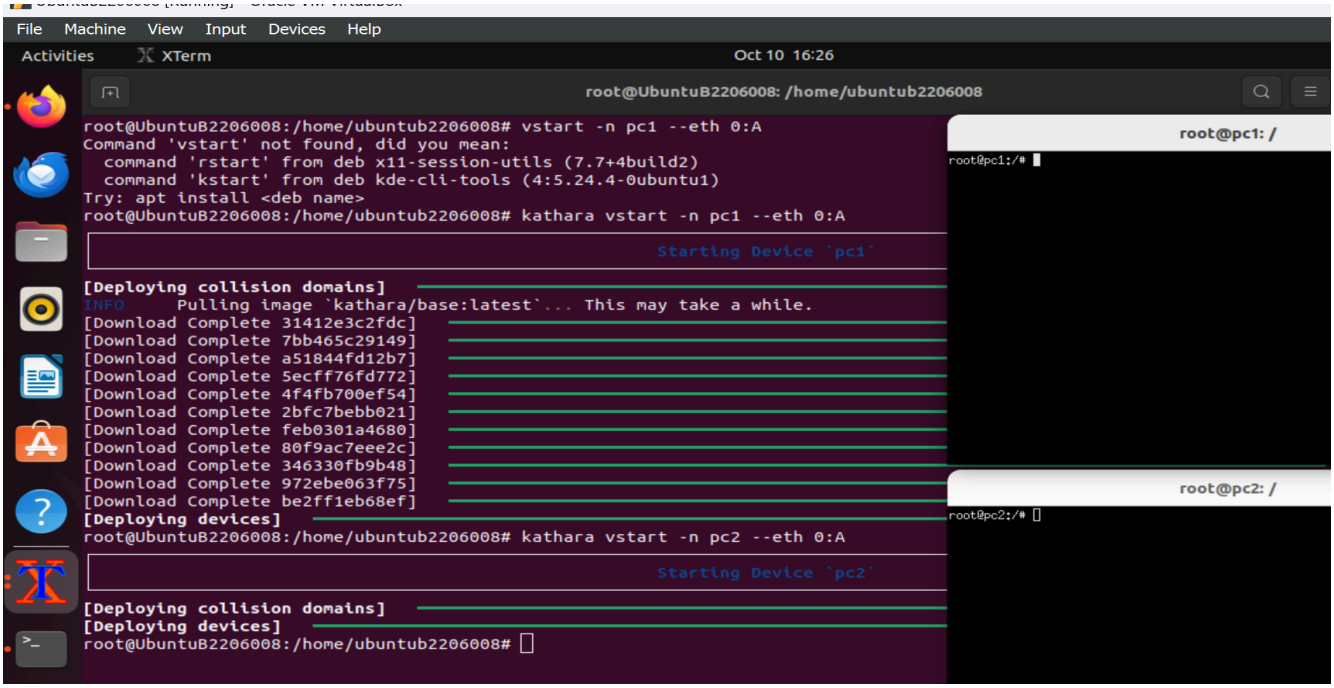
pc2:

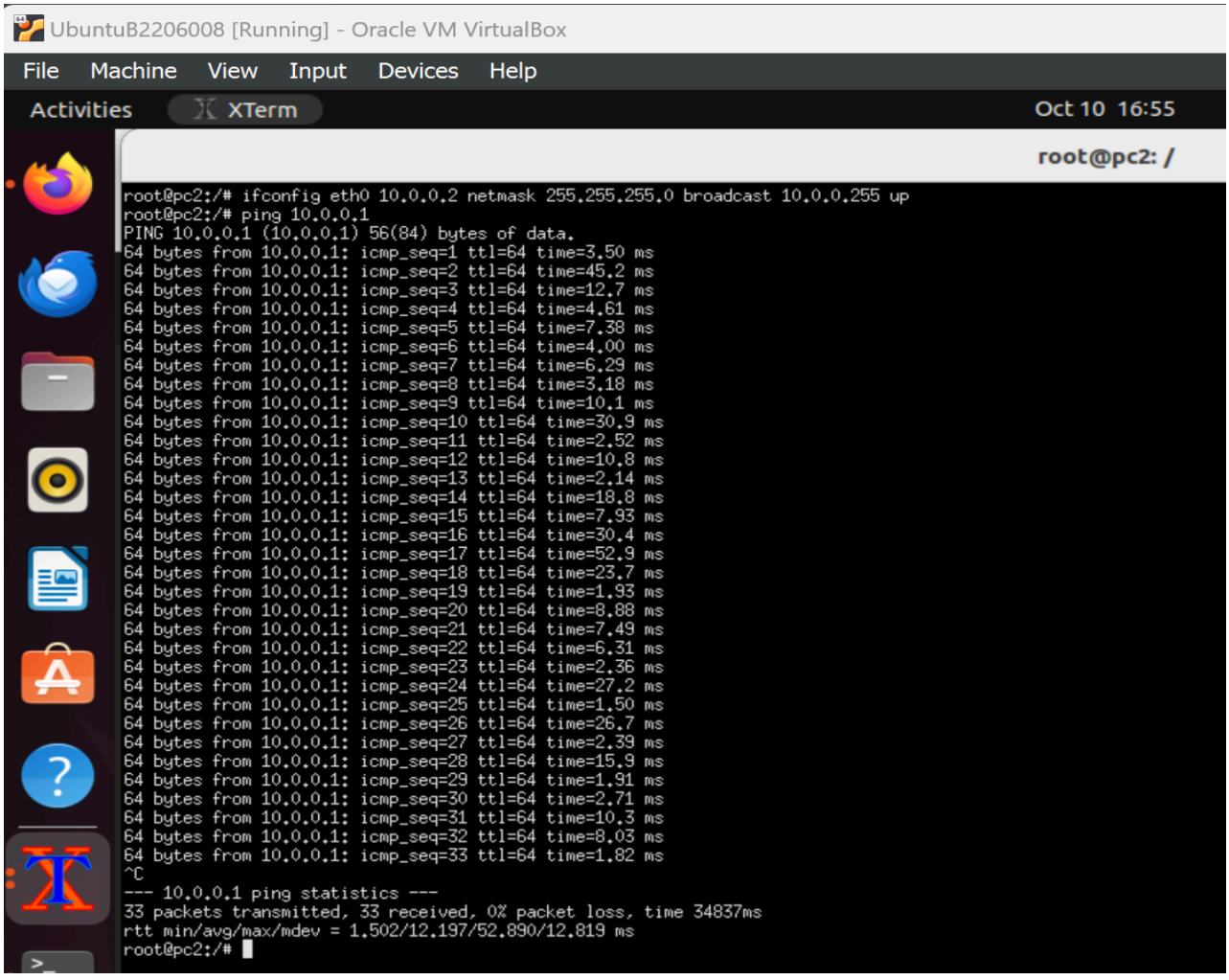
Ifconfig eth0 10.0.0.2 netmask 255.255.255.0 broadcast 10.0.0.255 up

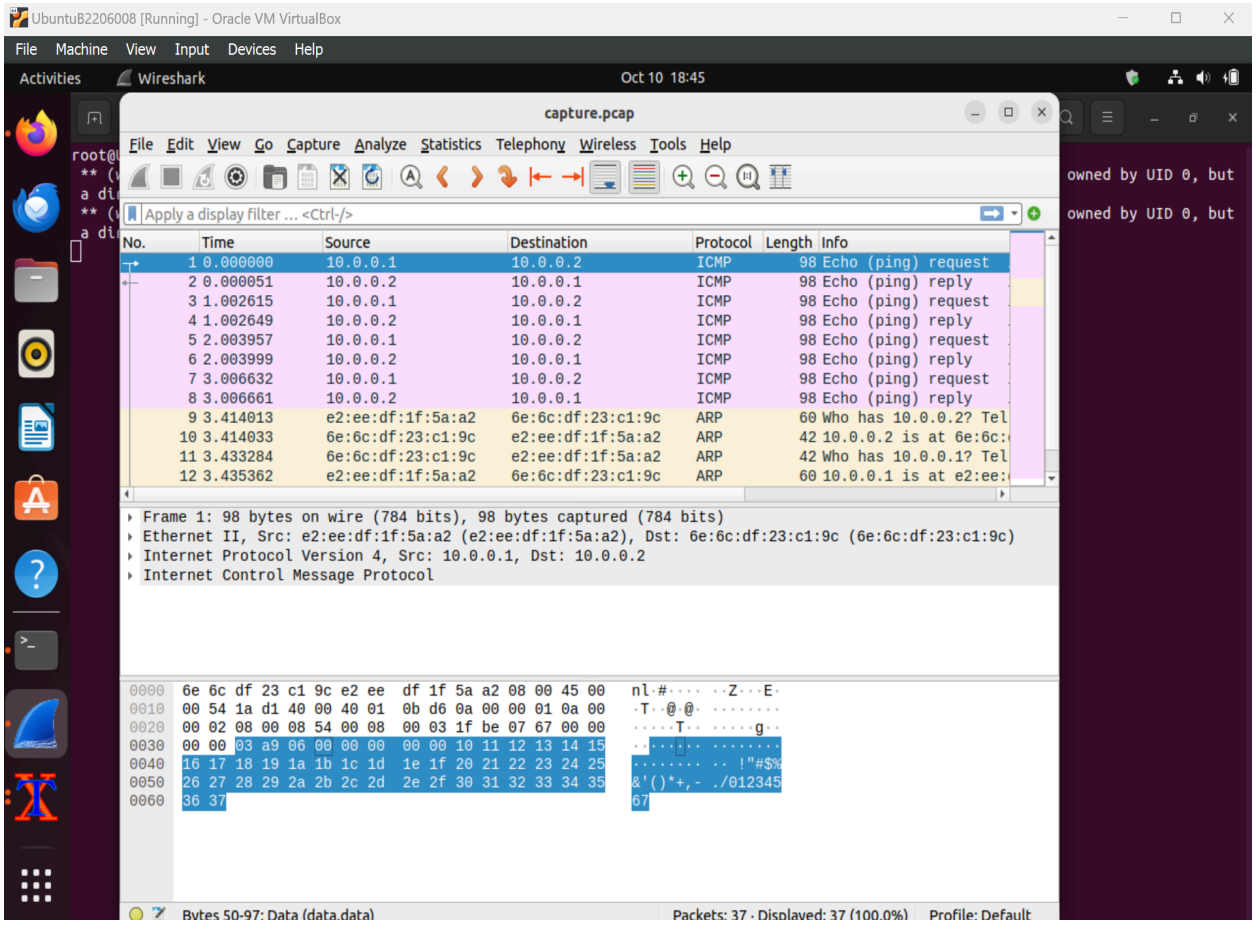
Ping 10.0.0.1

Tcpdump -tenni eth0

Tcpdump -tenni eth0 -w /hosthome/capture.pcap


File Machine View Input Devices Help
Activities XTerm Oct 10 16:26
root@UbuntuB2206008: /home/ubuntub2206008
root@UbuntuB2206008:/home/ubuntub2206008# vstart -n pc1 --eth 0:A
Command 'vstart' not found, did you mean:
command 'rstart' from deb x11-session-utils (7.7+4build2)
command 'kstart' from deb kde-cli-tools (4:5.24.4-0ubuntu1)
Try: apt install <deb name>
root@UbuntuB2206008:/home/ubuntub2206008# kathara vstart -n pc1 --eth 0:A
Starting Device `pc1`
[Deploying collision domains]
INFO Pulling image `kathara/base:latest`... This may take a while.
[Download Complete 31412e3c2fdc]
[Download Complete 7bb465c29149]
[Download Complete a51844fd12b7]
[Download Complete 5ecff76fd772]
[Download Complete 4f4fb700ef54]
[Download Complete 2bfc7bebb021]
[Download Complete feb0301a4680]
[Download Complete 80f9ac7eee2c]
[Download Complete 346330fb9b48]
[Download Complete 972ebe063f75]
[Download Complete be2ff1eb68ef]
[Deploying devices]
root@UbuntuB2206008:/home/ubuntub2206008# kathara vstart -n pc2 --eth 0:A
Starting Device `pc2`
[Deploying collision domains]
[Deploying devices]
root@UbuntuB2206008:/home/ubuntub2206008#
root@pc1: /
root@pc1:/#
root@pc2: /
root@pc2:/#


UbuntuB2206008 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities XTerm Oct 10 16:55
root@pc2: /
root@pc2:/# ifconfig eth0 10.0.0.2 netmask 255.255.255.0 broadcast 10.0.0.255 up
root@pc2:/# ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=3.50 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=45.2 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=12.7 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=4.61 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=7.38 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=4.00 ms
64 bytes from 10.0.0.1: icmp_seq=7 ttl=64 time=6.29 ms
64 bytes from 10.0.0.1: icmp_seq=8 ttl=64 time=3.18 ms
64 bytes from 10.0.0.1: icmp_seq=9 ttl=64 time=10.1 ms
64 bytes from 10.0.0.1: icmp_seq=10 ttl=64 time=30.9 ms
64 bytes from 10.0.0.1: icmp_seq=11 ttl=64 time=2.52 ms
64 bytes from 10.0.0.1: icmp_seq=12 ttl=64 time=10.8 ms
64 bytes from 10.0.0.1: icmp_seq=13 ttl=64 time=2.14 ms
64 bytes from 10.0.0.1: icmp_seq=14 ttl=64 time=18.8 ms
64 bytes from 10.0.0.1: icmp_seq=15 ttl=64 time=7.93 ms
64 bytes from 10.0.0.1: icmp_seq=16 ttl=64 time=30.4 ms
64 bytes from 10.0.0.1: icmp_seq=17 ttl=64 time=52.9 ms
64 bytes from 10.0.0.1: icmp_seq=18 ttl=64 time=23.7 ms
64 bytes from 10.0.0.1: icmp_seq=19 ttl=64 time=1.93 ms
64 bytes from 10.0.0.1: icmp_seq=20 ttl=64 time=8.88 ms
64 bytes from 10.0.0.1: icmp_seq=21 ttl=64 time=7.49 ms
64 bytes from 10.0.0.1: icmp_seq=22 ttl=64 time=6.31 ms
64 bytes from 10.0.0.1: icmp_seq=23 ttl=64 time=2.36 ms
64 bytes from 10.0.0.1: icmp_seq=24 ttl=64 time=27.2 ms
64 bytes from 10.0.0.1: icmp_seq=25 ttl=64 time=1.50 ms
64 bytes from 10.0.0.1: icmp_seq=26 ttl=64 time=26.7 ms
64 bytes from 10.0.0.1: icmp_seq=27 ttl=64 time=2.39 ms
64 bytes from 10.0.0.1: icmp_seq=28 ttl=64 time=15.9 ms
64 bytes from 10.0.0.1: icmp_seq=29 ttl=64 time=1.91 ms
64 bytes from 10.0.0.1: icmp_seq=30 ttl=64 time=2.71 ms
64 bytes from 10.0.0.1: icmp_seq=31 ttl=64 time=10.3 ms
64 bytes from 10.0.0.1: icmp_seq=32 ttl=64 time=8.03 ms
64 bytes from 10.0.0.1: icmp_seq=33 ttl=64 time=1.82 ms
^C
--- 10.0.0.1 ping statistics ---
33 packets transmitted, 33 received, 0% packet loss, time 34837ms
rtt min/avg/max/mdev = 1.502/12.197/52.890/12.819 ms
root@pc2:/#



Exercise 2:
Solution:
Command: kathara lstart
File:

Lab.conf:

pc1[0]=A
pc3[0]=A
pc2[0]=B
pc4[0]=B
router1[0]=A
router1[1]=B

Pc1:

Ifconfig eth0 10.0.0.101/24 up
Route add default gw 10.0.0.1

Pc2:

Ifconfig eth0 10.0.1.101/24 up
Route add default gw 10.0.1.1

Pc3:

Ifconfig eth0 10.0.0.102/24 up

Route add default gw 10.0.0.1

Pc4:

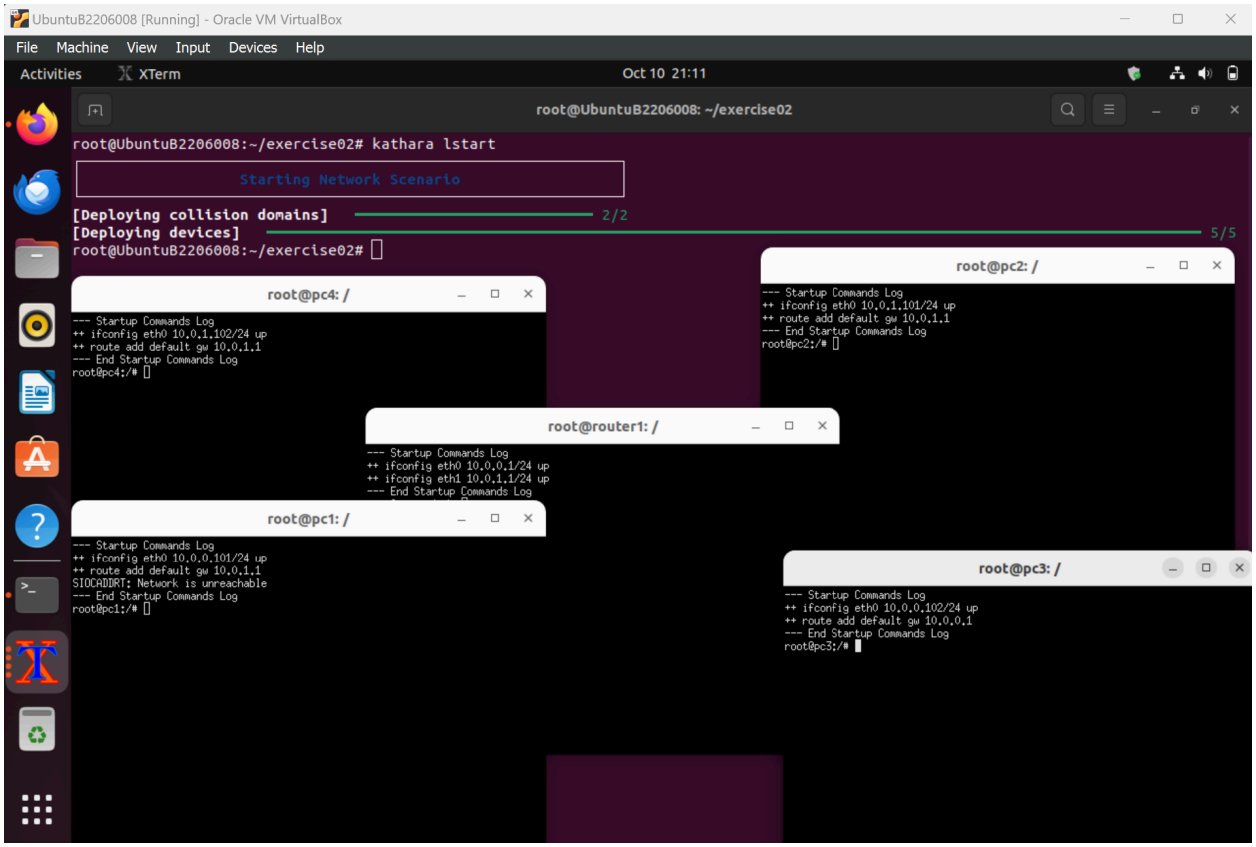
Ifconfig eth0 10.0.1.102/24 up

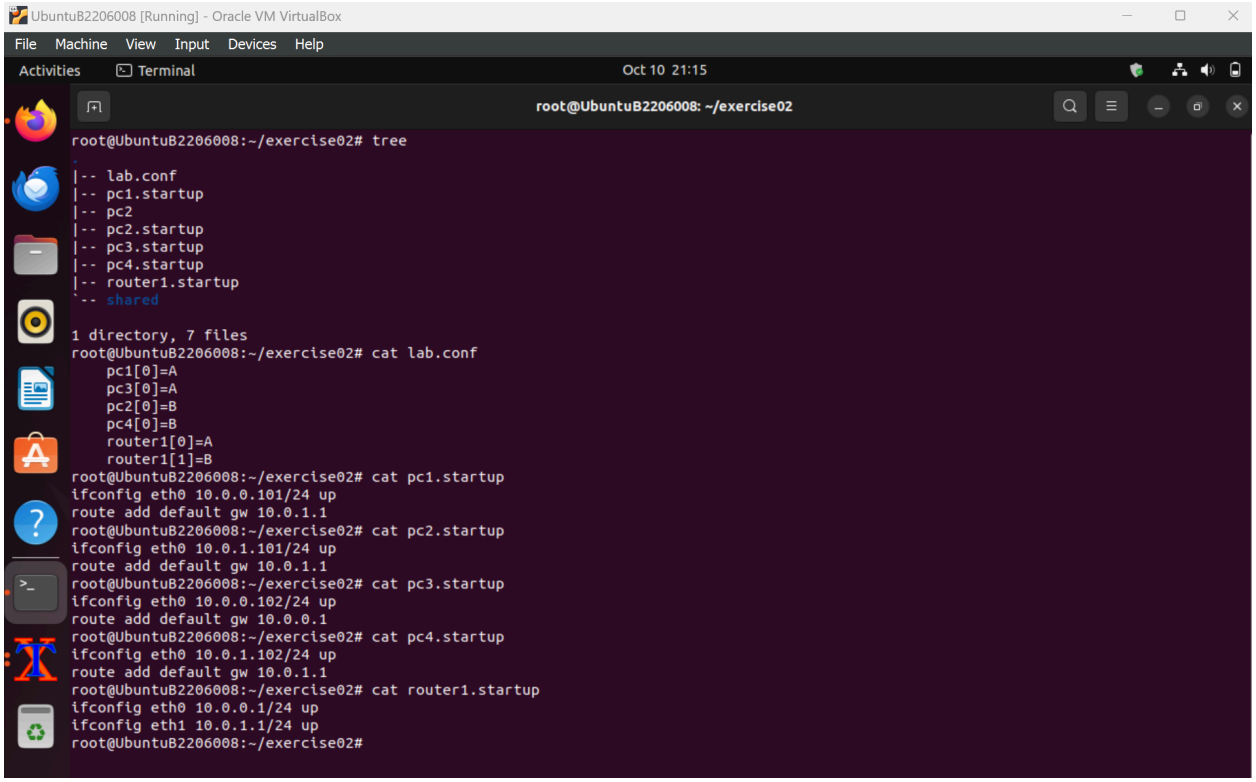
Route add default gw 10.0.1.1

Router1.startup:

Ifconfig eth0 10.0.0.1/24 up

Ifconfig eth0 10.0.1.1/24 up





Exercise 3:

Solution:

File:

lab.conf:

r1[0]=A

r1[1]=B

r2[0]=C

r2[1]=B

pc1[0]=A

pc2[0]=C

Pc1.startup:

ifconfig eth0 195.11.14.5/24 up

Pc2.startup:

ifconfig eth0 200.1.1.7/24 up

R1.startup:

ifconfig eth0 195.11.14.1/24 up

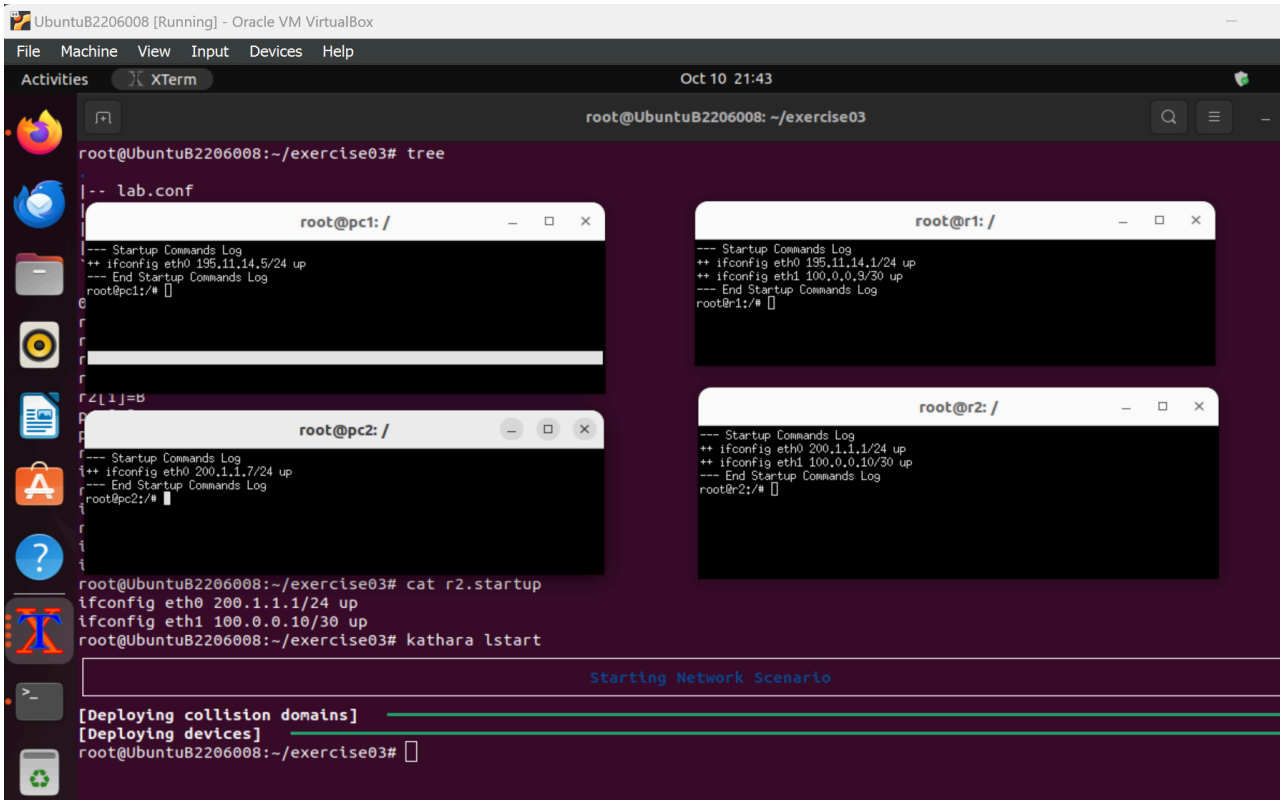
ifconfig eth1 100.0.0.9/30 up

R2.startup

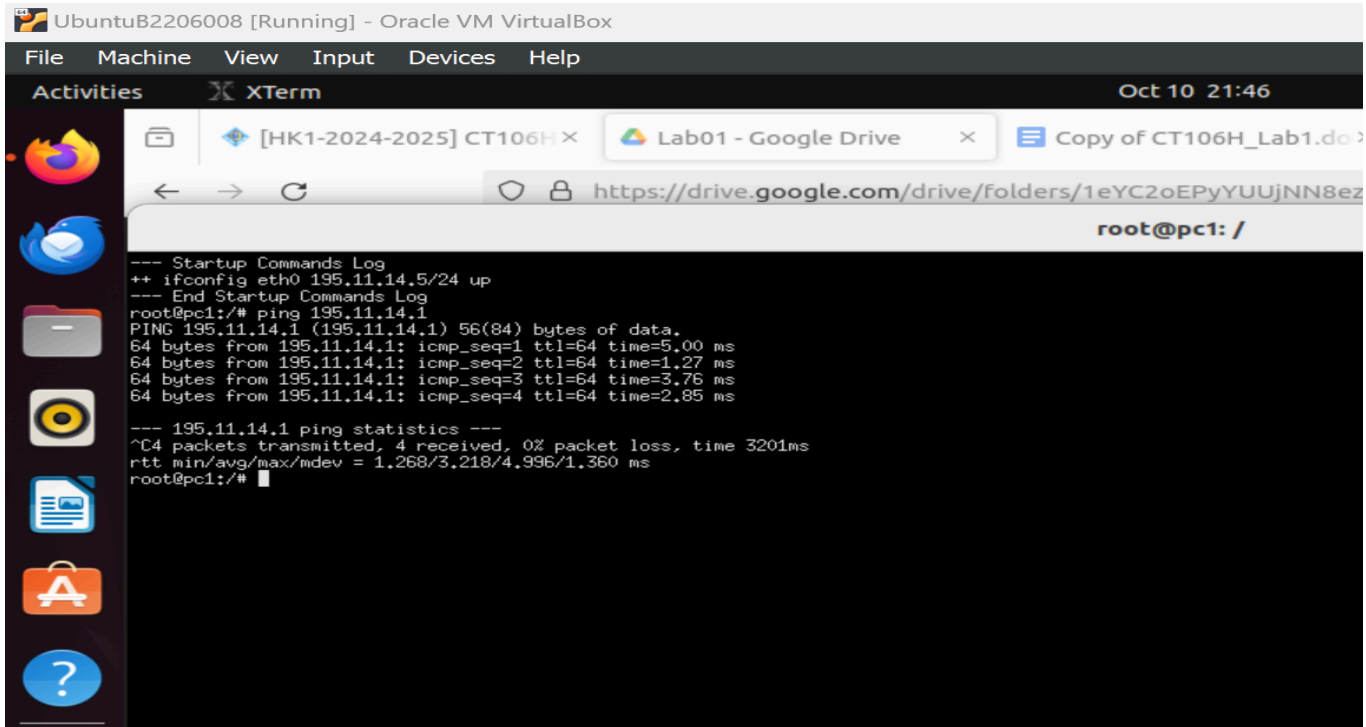
ifconfig eth0 200.1.1.1/24 up

ifconfig eth1 100.0.0.10/30 up

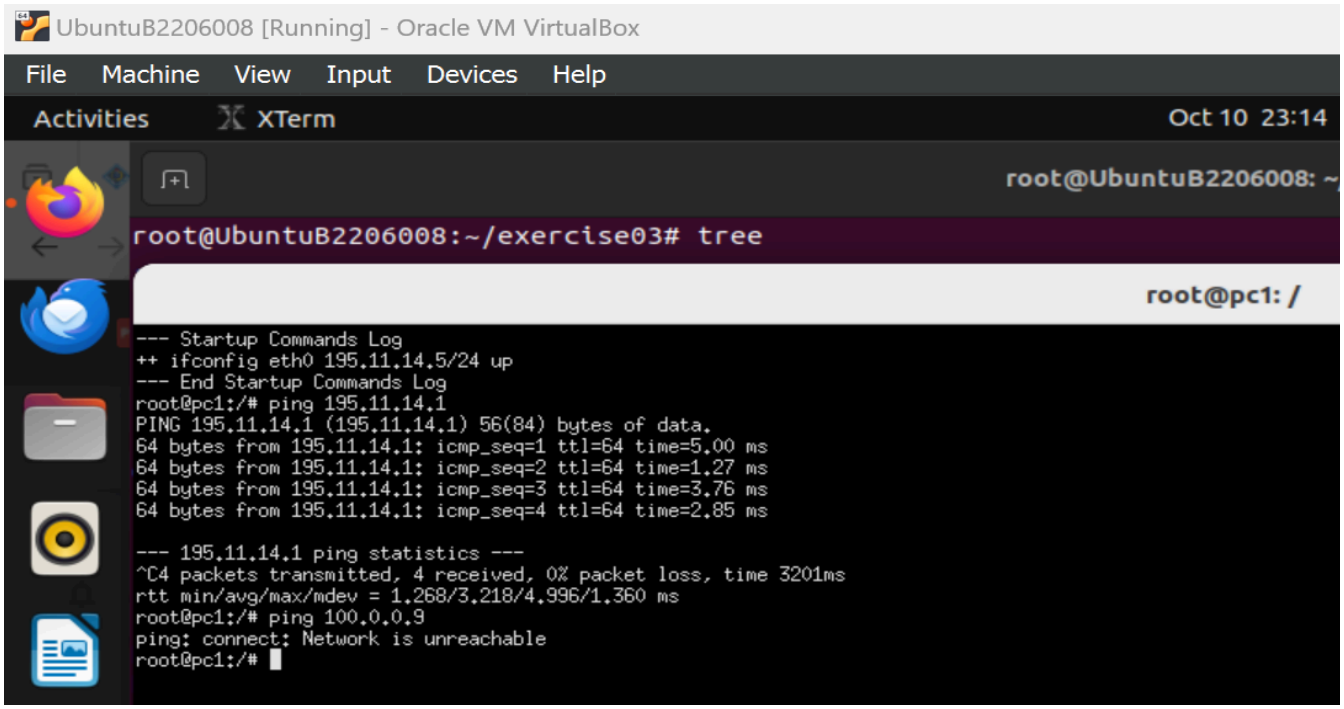
- **Start Scenario**
  Kathara lstart



- **Ping from pc1 to r1 eth0 (195.11.14.1)**
  Ping 195.11.14.1



- **Ping from pc1 to r1 eth1(100.0.0.9)**
  Ping 100.0.0.9



- **Interfaces on different domains cannot be reached. Can you tell why?**
  - **Both routers and PCs don't know how to reach networks that are not directly connected to them.**
  - **Directly connected networks are automatically inserted into the routing table when the corresponding interface is brought up.**

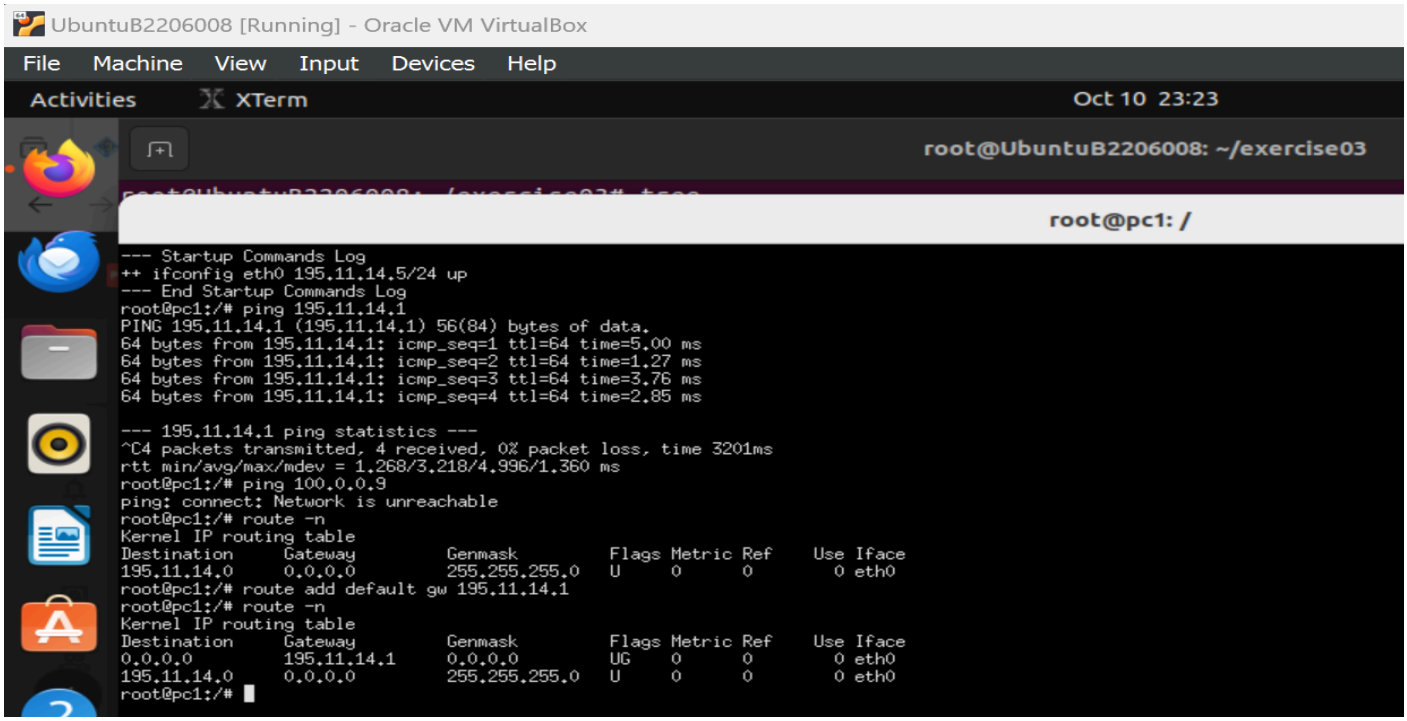
- Add default route to pc1, from pc1 to r1 eth1

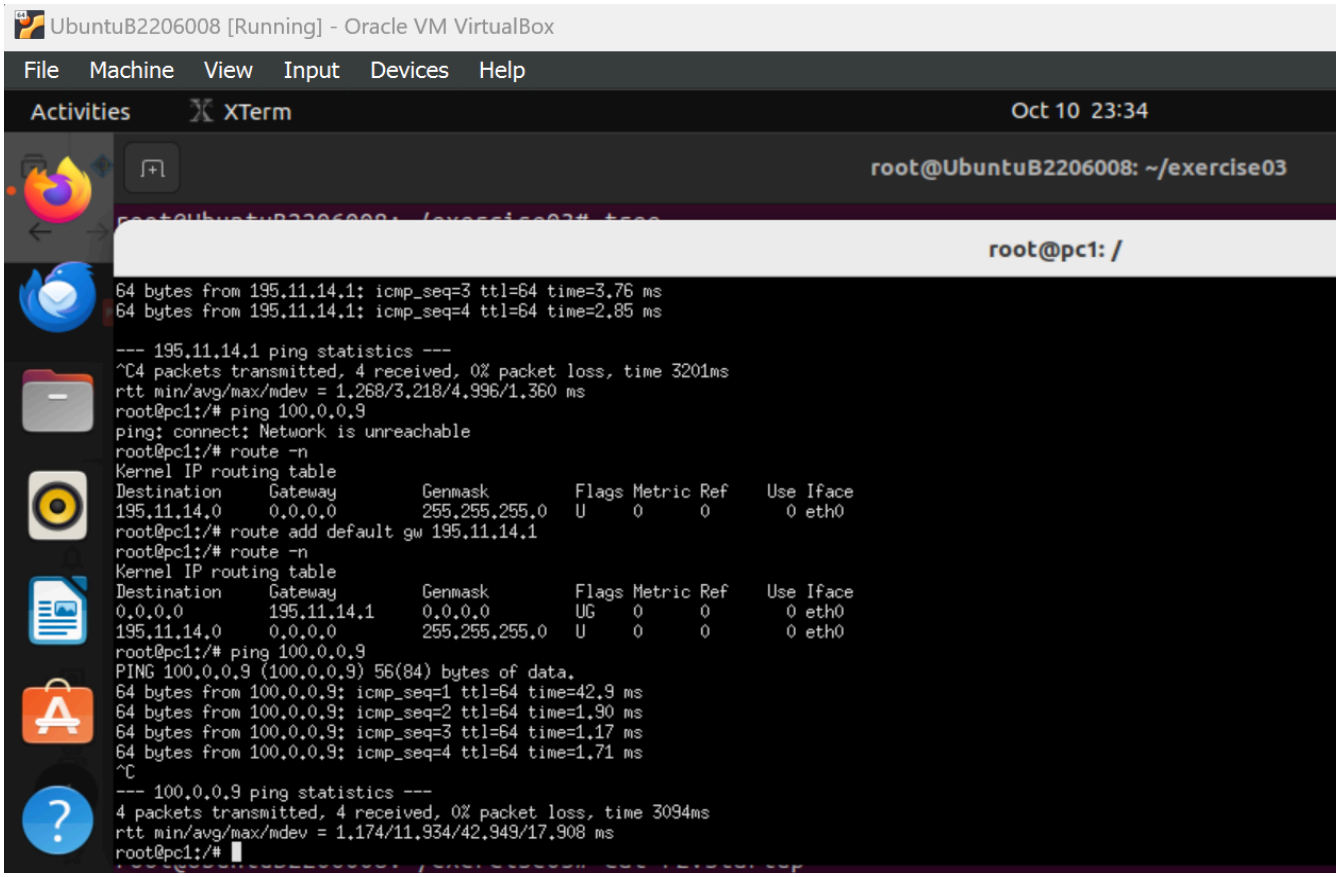
Route -n

Route add default gw 195.11.14.1

Route –n

Ping 100.0.0.9

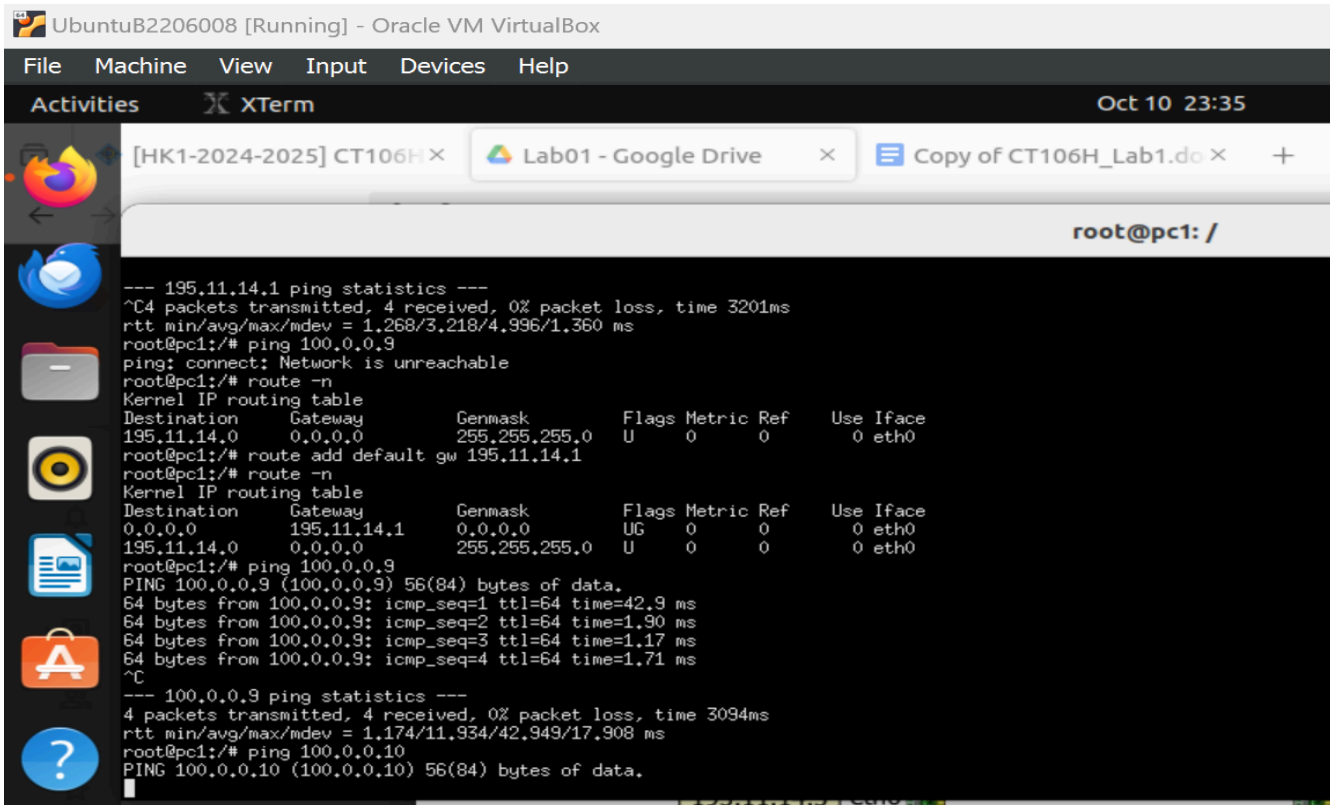


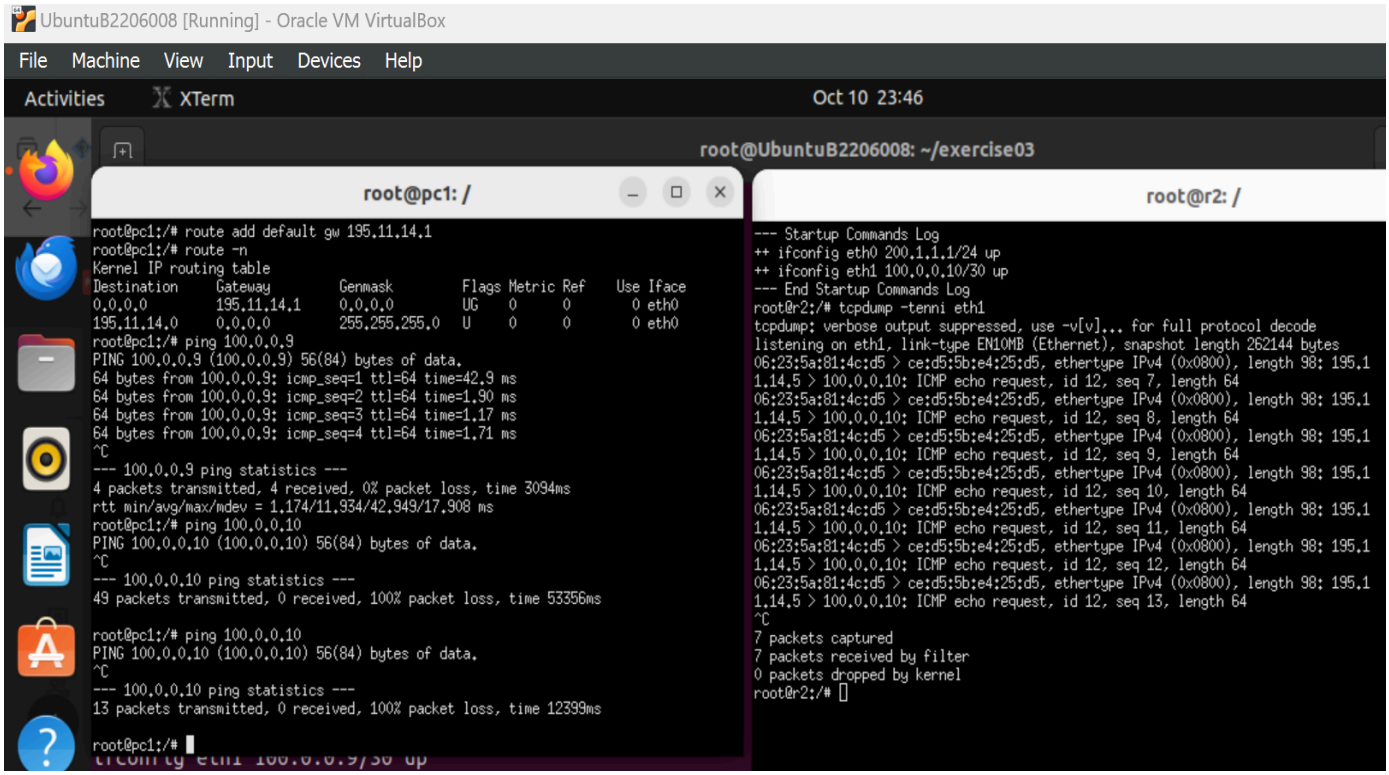


- **Ping from pc1 to r2 eth1 and sniff at r2**

  Ping 100.0.0.10

  Tcpdump -tenni eth1





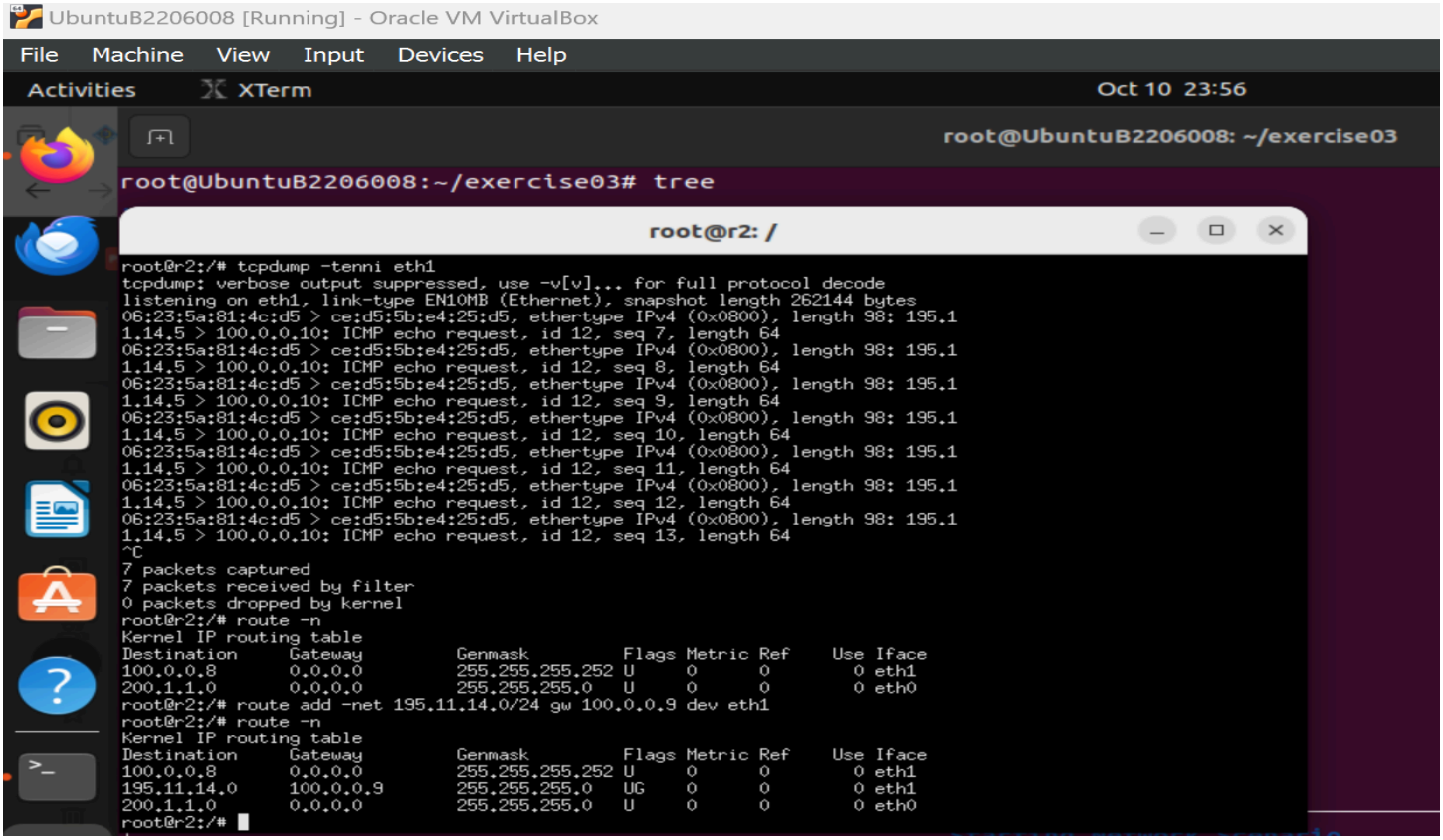
- **Interfaces on r2 seem unreachable, can you tell why?**
  - **R2 only echo requests.**
  - **Because we just added a gateway for pc1 so r2 still can't know where to forward the replies when it receives packets.**

- **Add route r2 - r1 eth0, gateway r1 eth1**

  Route -n

  Route add–net 195.11.14.0/24 gw 100.0.0.9 dev eth1

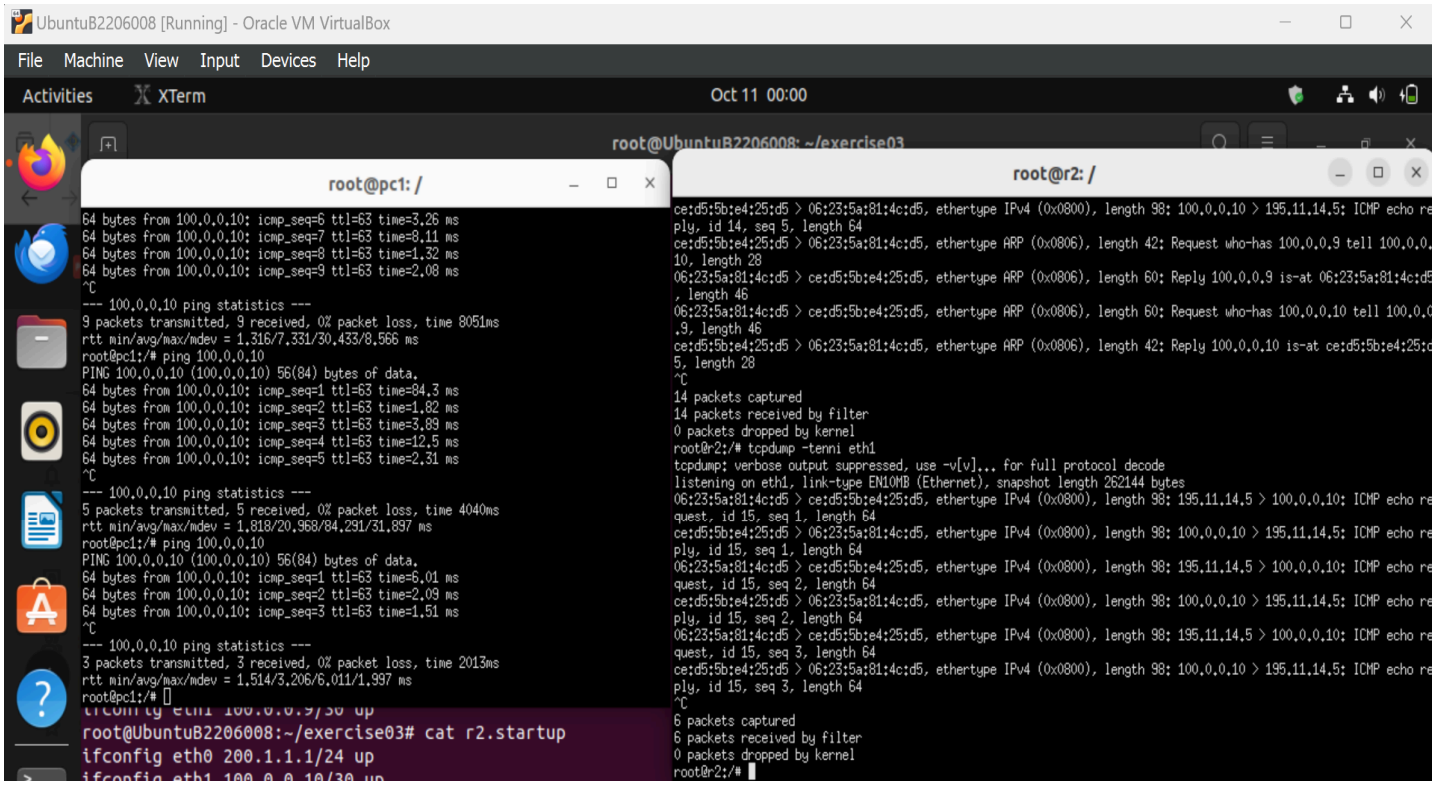
  Route -n



- **Ping from pc1 to r2 eth1**

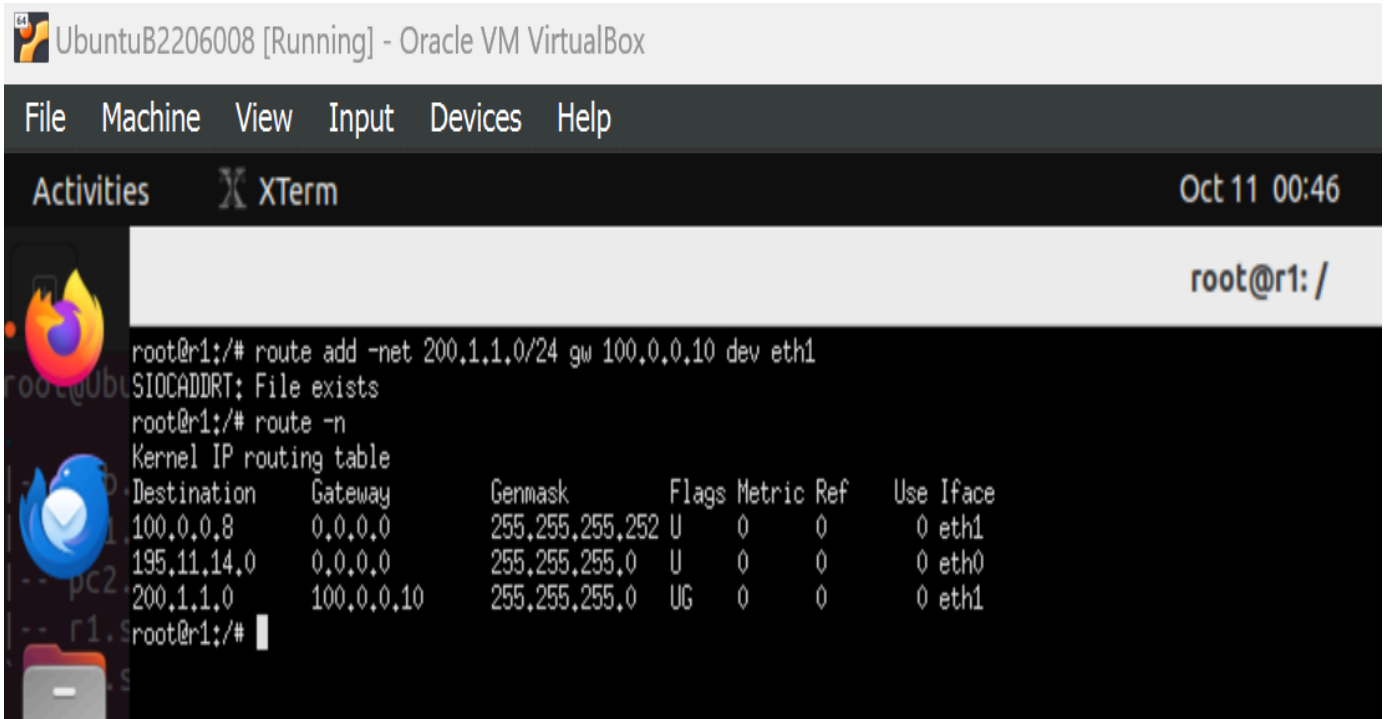
  ping 100.0.0.10
- **Sniff packets sent from pc1 to r2 eth1**

  Tcpdump -tenni eth1

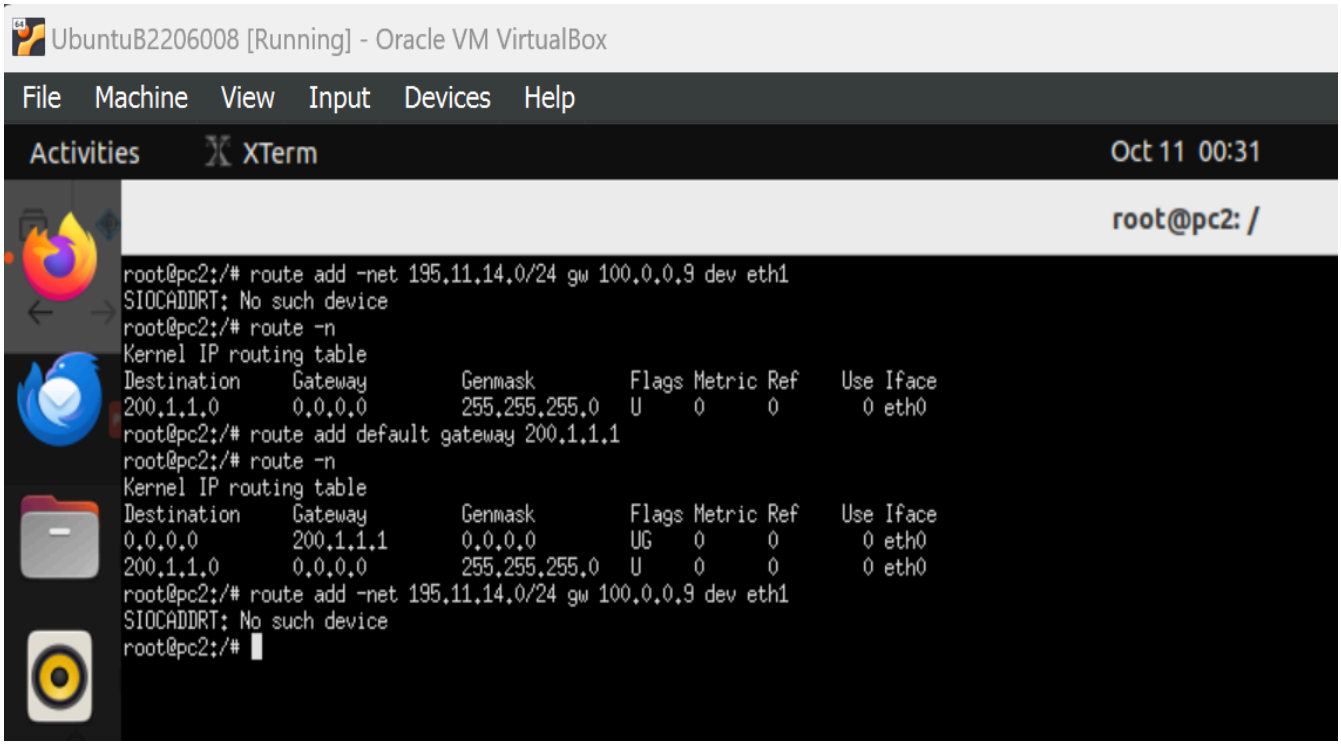


- **Add route r1 to r2 eth0, through r1 eth1**

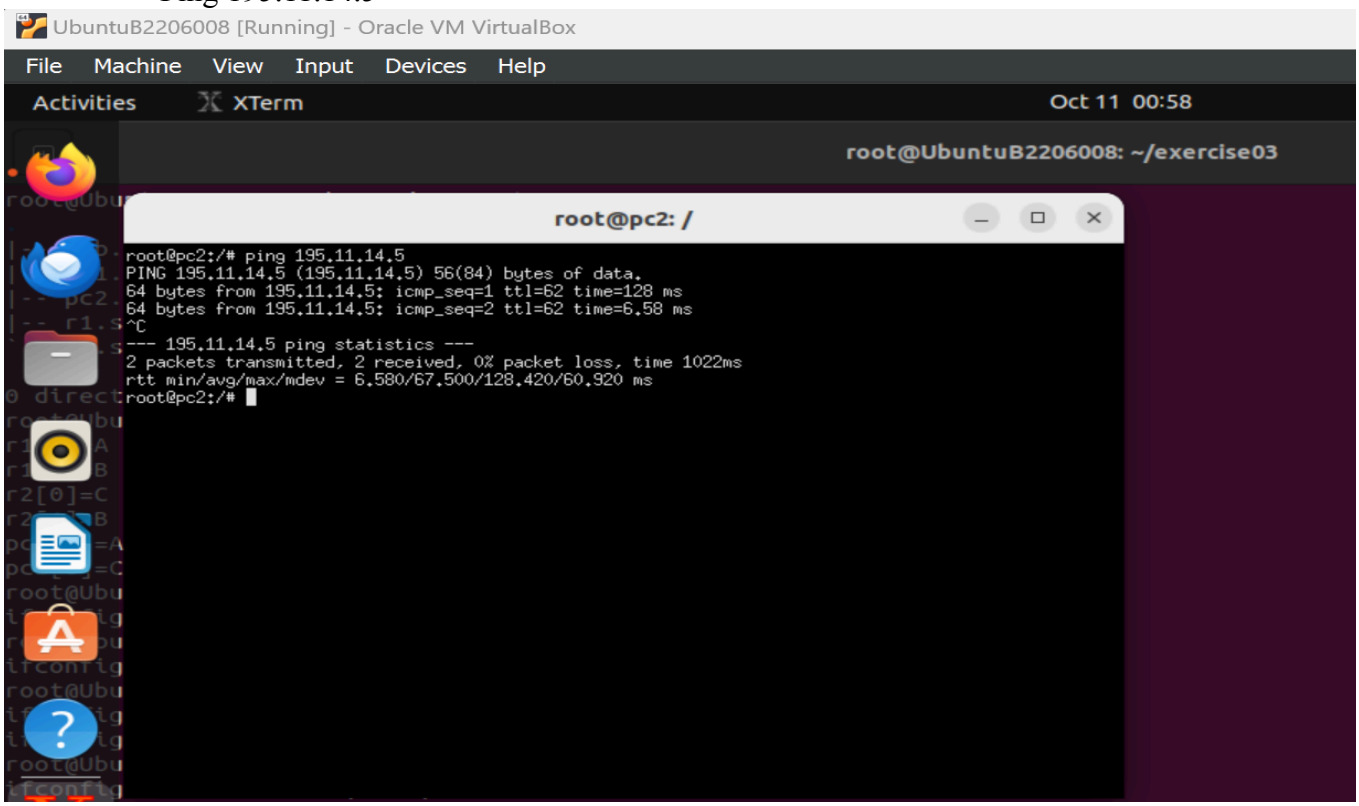
  Route add –net 200.1.1.0/24 gw 100.0.0.10 dev eth1



- **Add default gateway r2 - pc2**
  route –n
  route add default gateway 200.1.1.1
  route –n



- **Ping from p2 to p1**
  Ping 195.11.14.5



Proposed Exercise:

- A command to configure a static route that is equivalent to the default route (route add default gw 195.11.14.1 dev eth0):

```
route add -net 0.0.0.0/0 gw 195.11.14.1 dev eth0
```

Exercise 4:

Solution:

- Create exercise04 directory and config the lab

File:

Lab.conf:

```
r1[0]="A"
r1[1]="B"
r2[0]="C"
r2[1]="B"
pc1[0]="A"
pc2[0]="C"
pc3[0]="C"
```

Pc1.startup:

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

Pc2.startup:

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

Pc3.startup:

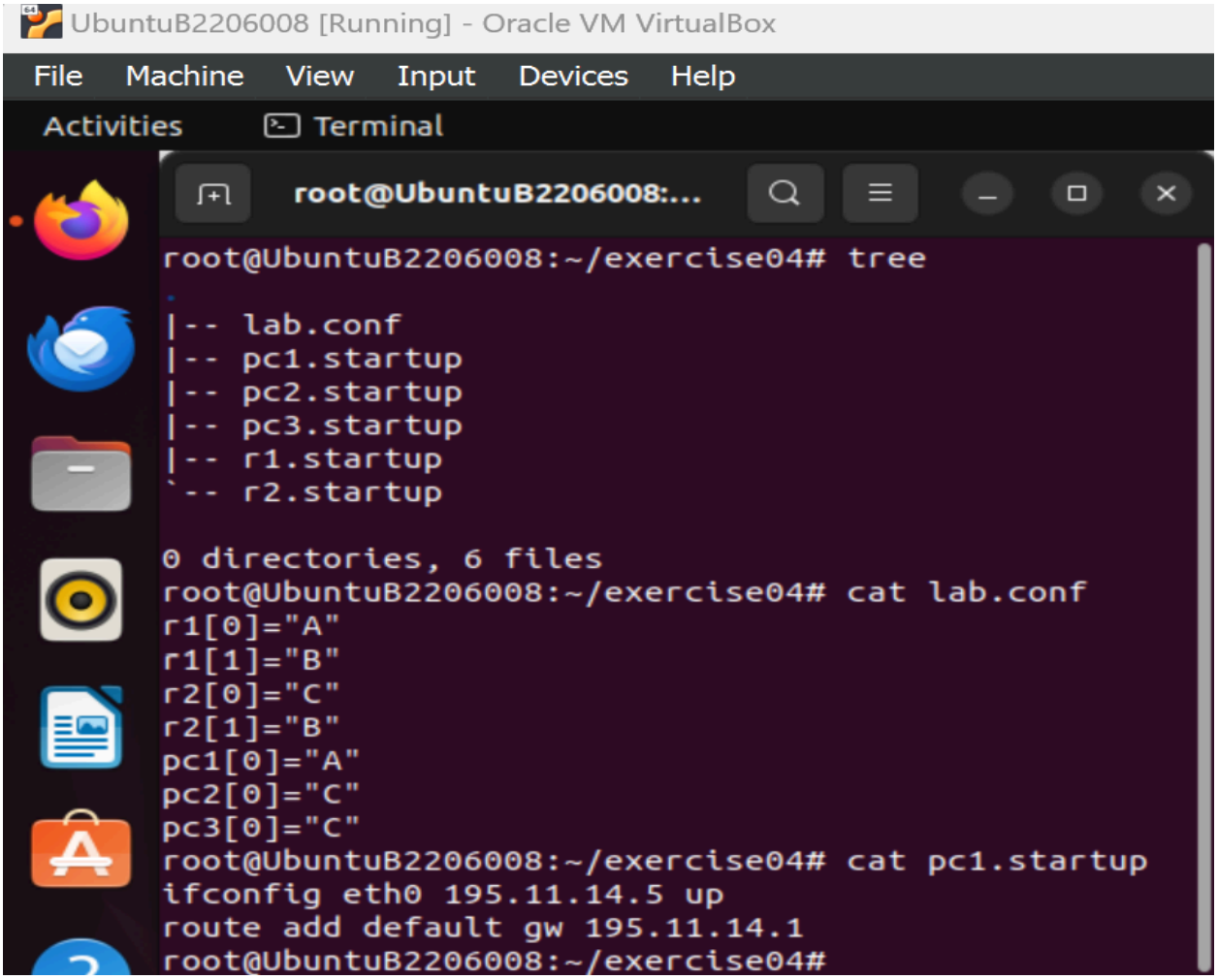
```
ifconfig eth0 200.1.1.3 up
route add default gw 200.1.1.1
```

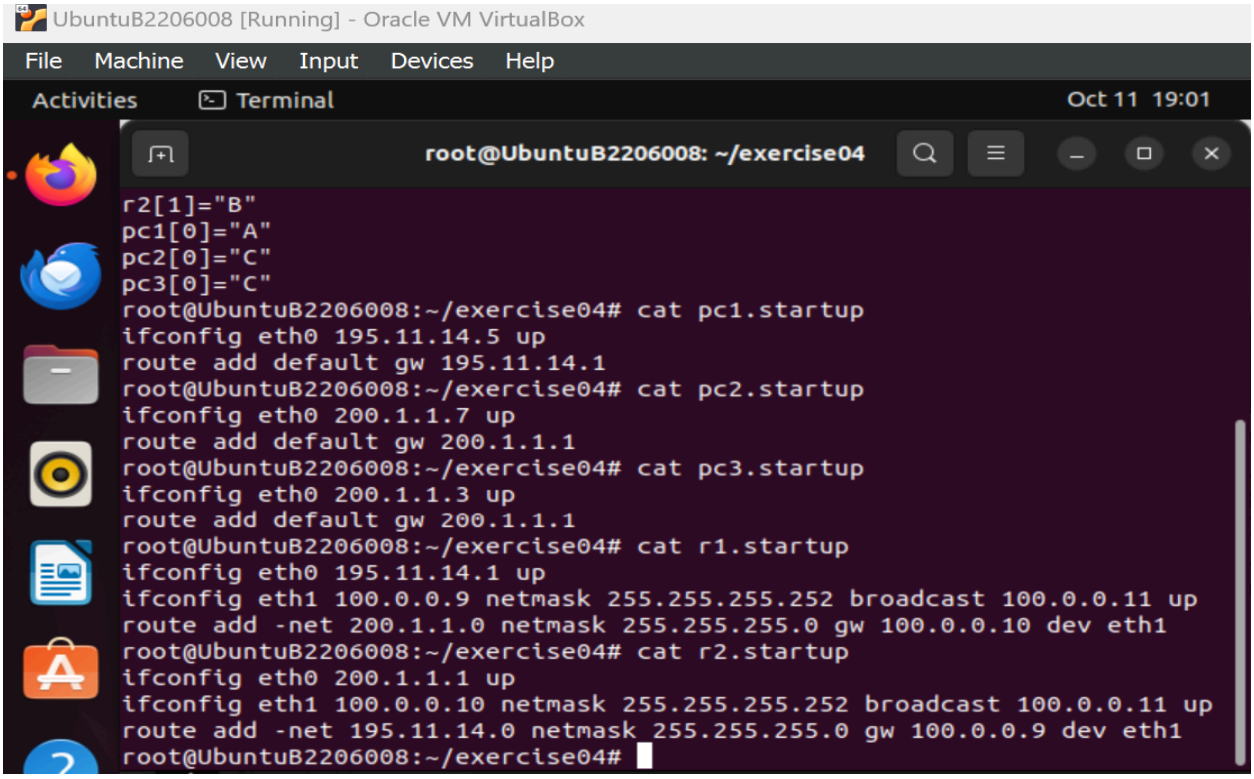
R1.startup:

```
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
```

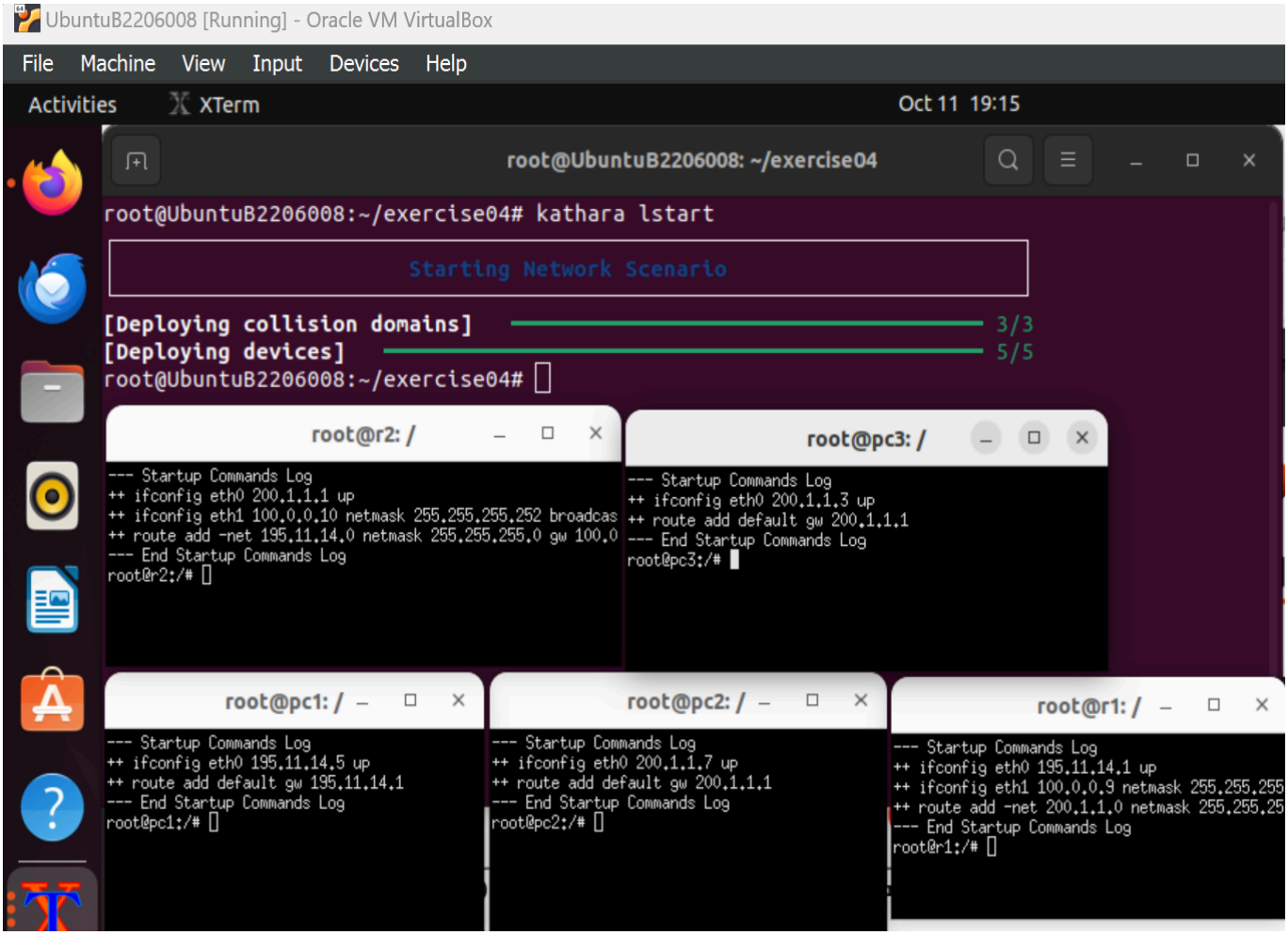
R2.startup:

```
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 eth1ls
```


UbuntuB2206008 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
root@UbuntuB2206008:...
root@UbuntuB2206008:~/exercise04# tree
.
|-- lab.conf
|-- pc1.startup
|-- pc2.startup
|-- pc3.startup
|-- r1.startup
`-- r2.startup
0 directories, 6 files
root@UbuntuB2206008:~/exercise04# cat lab.conf
r1[0]="A"
r1[1]="B"
r2[0]="C"
r2[1]="B"
pc1[0]="A"
pc2[0]="C"
pc3[0]="C"
root@UbuntuB2206008:~/exercise04# cat pc1.startup
ifconfig eth0 195.11.14.5 up
route add default gw 195.11.14.1
root@UbuntuB2206008:~/exercise04#


UbuntuB2206008 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Oct 11 19:01
root@UbuntuB2206008: ~/exercise04
r2[1]="B"
pc1[0]="A"
pc2[0]="C"
pc3[0]="C"
root@UbuntuB2206008:~/exercise04# cat pc1.startup
ifconfig eth0 195.11.14.5 up
route add default gw 195.11.14.1
root@UbuntuB2206008:~/exercise04# cat pc2.startup
ifconfig eth0 200.1.1.7 up
route add default gw 200.1.1.1
root@UbuntuB2206008:~/exercise04# cat pc3.startup
ifconfig eth0 200.1.1.3 up
route add default gw 200.1.1.1
root@UbuntuB2206008:~/exercise04# 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
root@UbuntuB2206008:~/exercise04# 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
root@UbuntuB2206008:~/exercise04#

- **Start the scenario**
  Kathara lstart



-

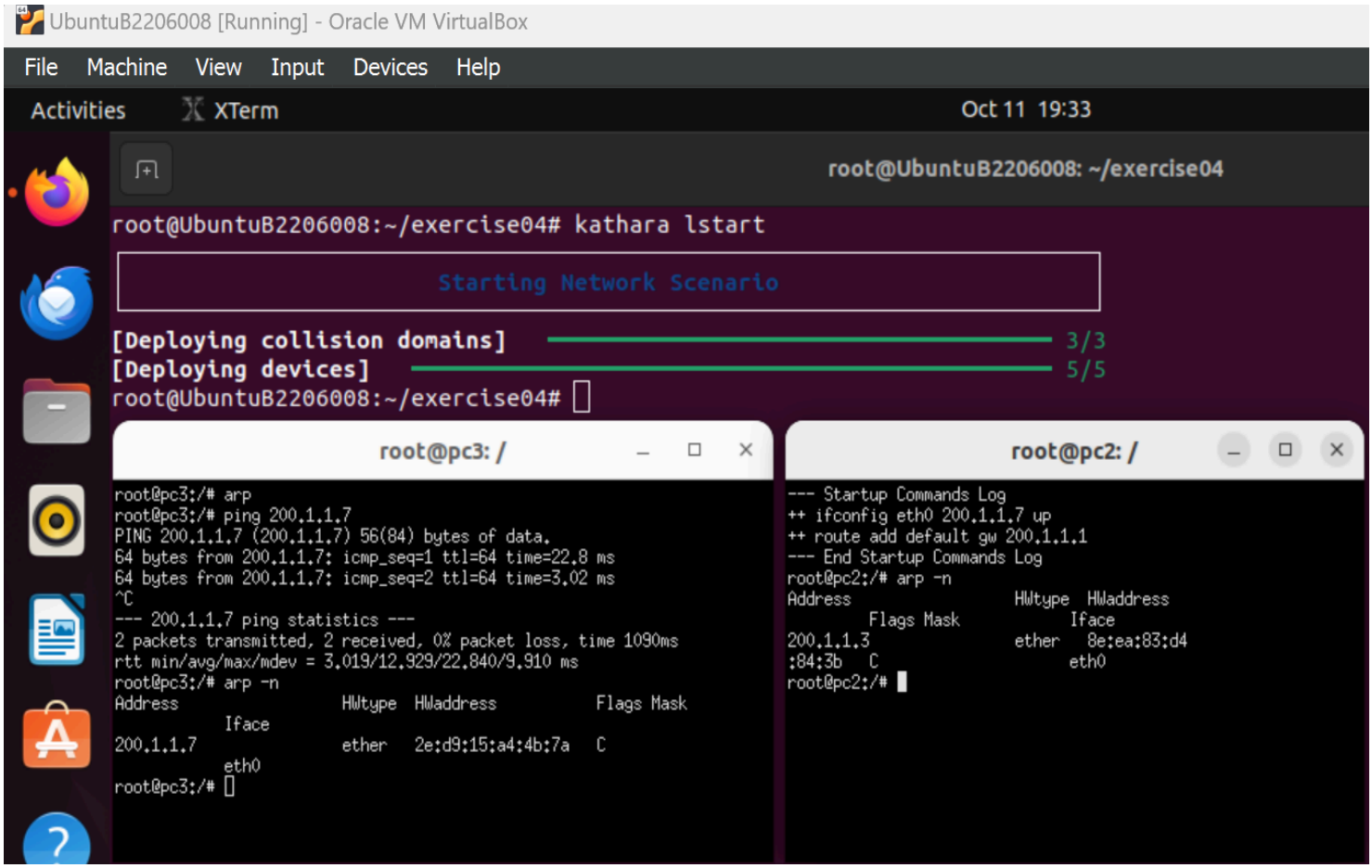
- **Inspect arp cache (local traffic)**

  Pc3:

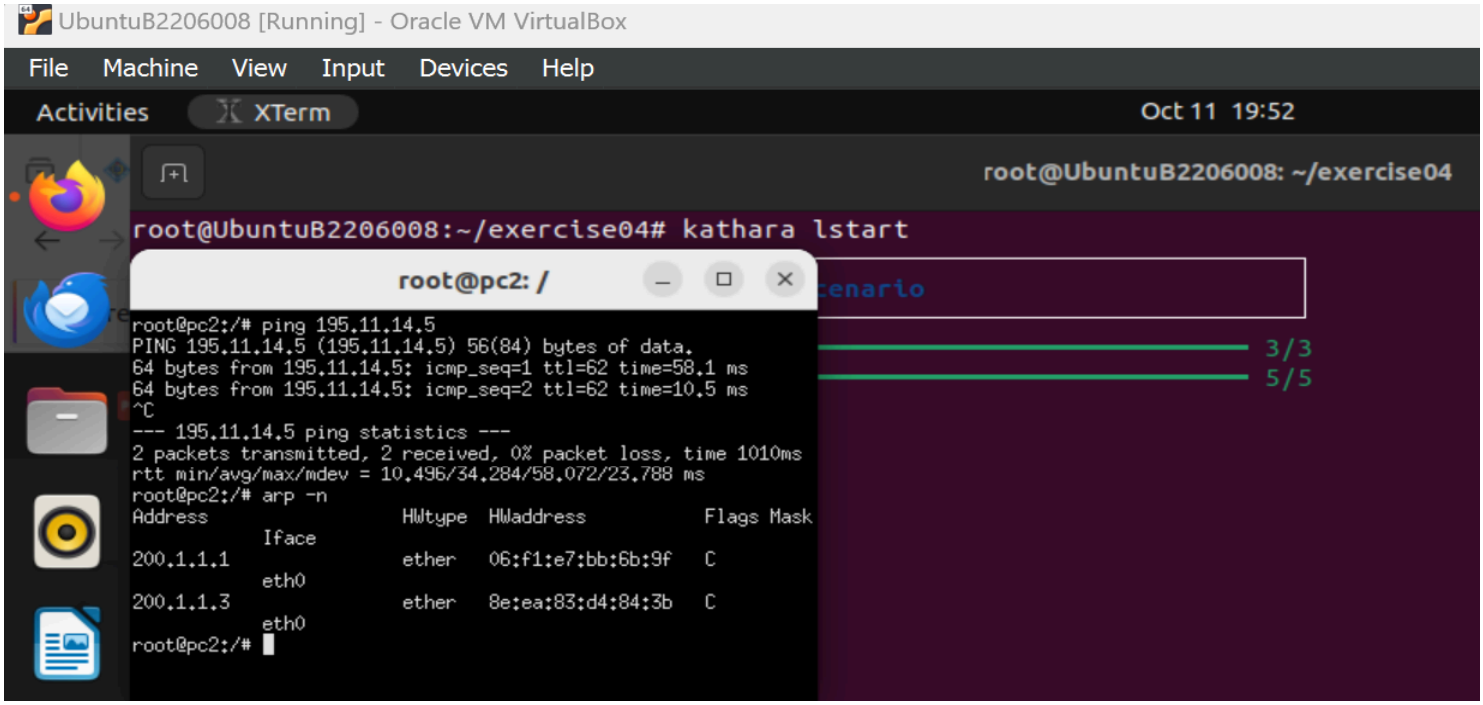
  Arp
  Ping 200.1.1.7 (pc2)
  Arp -n

  Pc2:

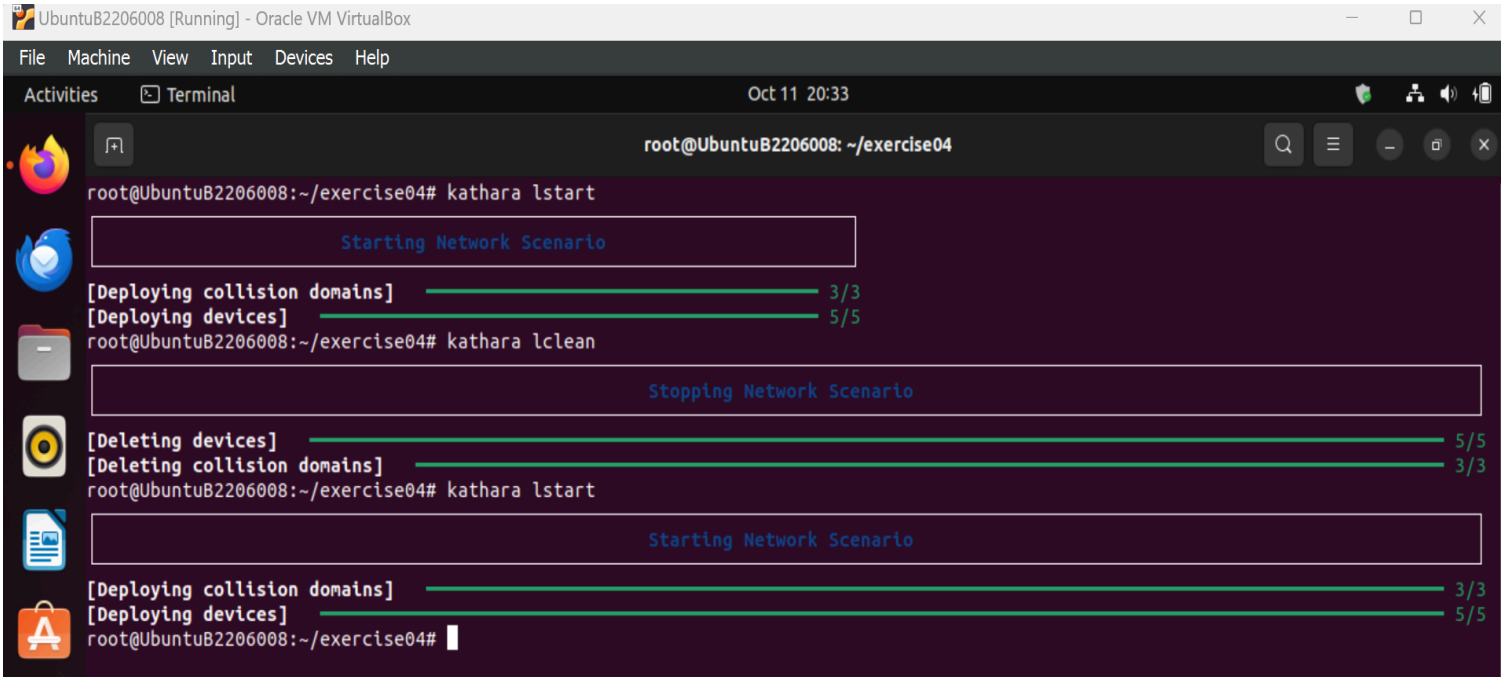
  Arp -n



- **Inspect arp cache (non-local traffic)**
  Pc2:
  Ping 195.11.14.5 (pc1)
  Arp -n (r2 eth0 MAC address added instead of pc1 MAC address)



- **Restart scenario to clear arp caches**
  Kathara lclean
  Kathara lstart



- **Sniffing arp traffic**

  Pc2:

  Ping 195.11.14.5

  R2:

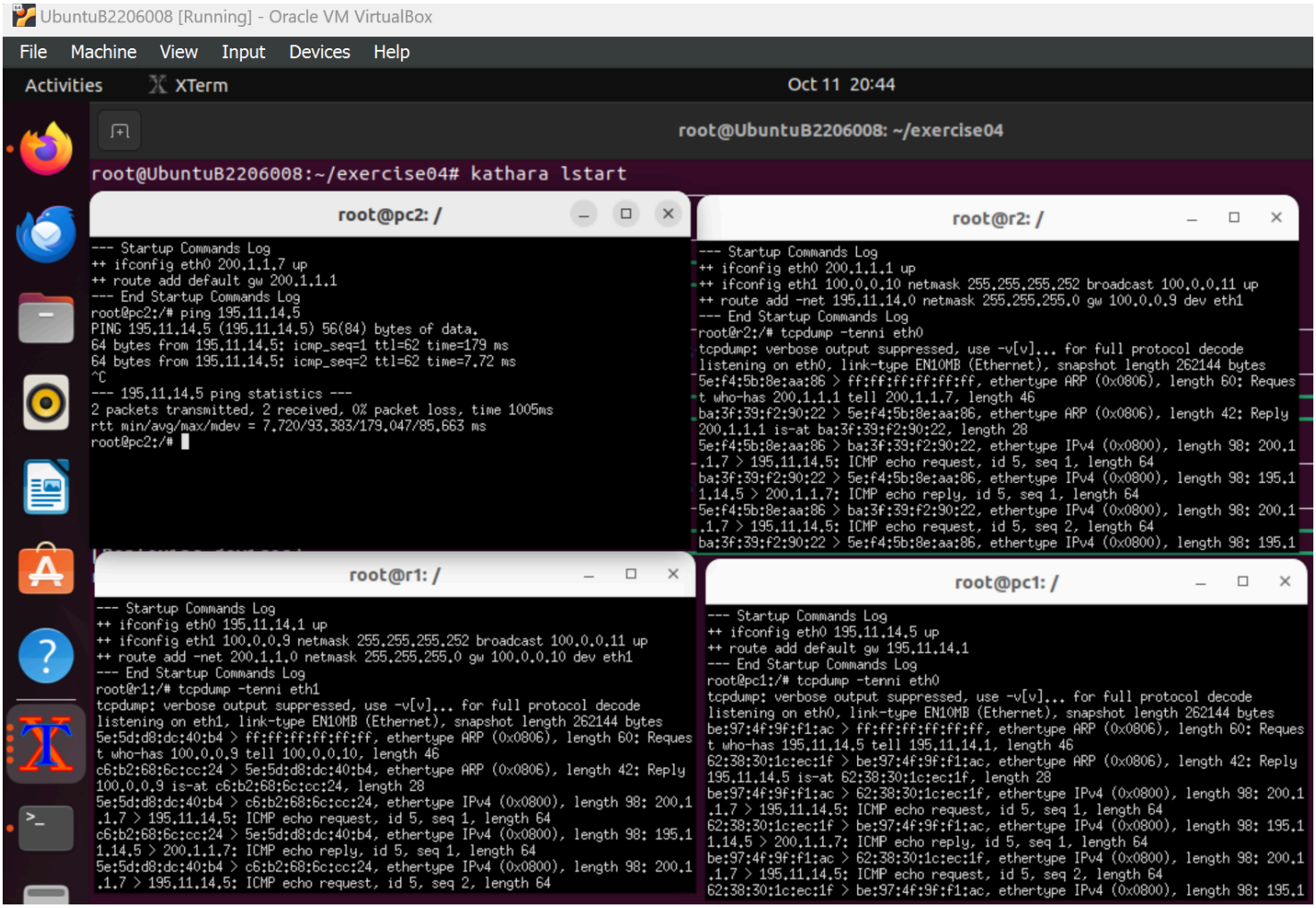
  Tcpdump -tenni eth0

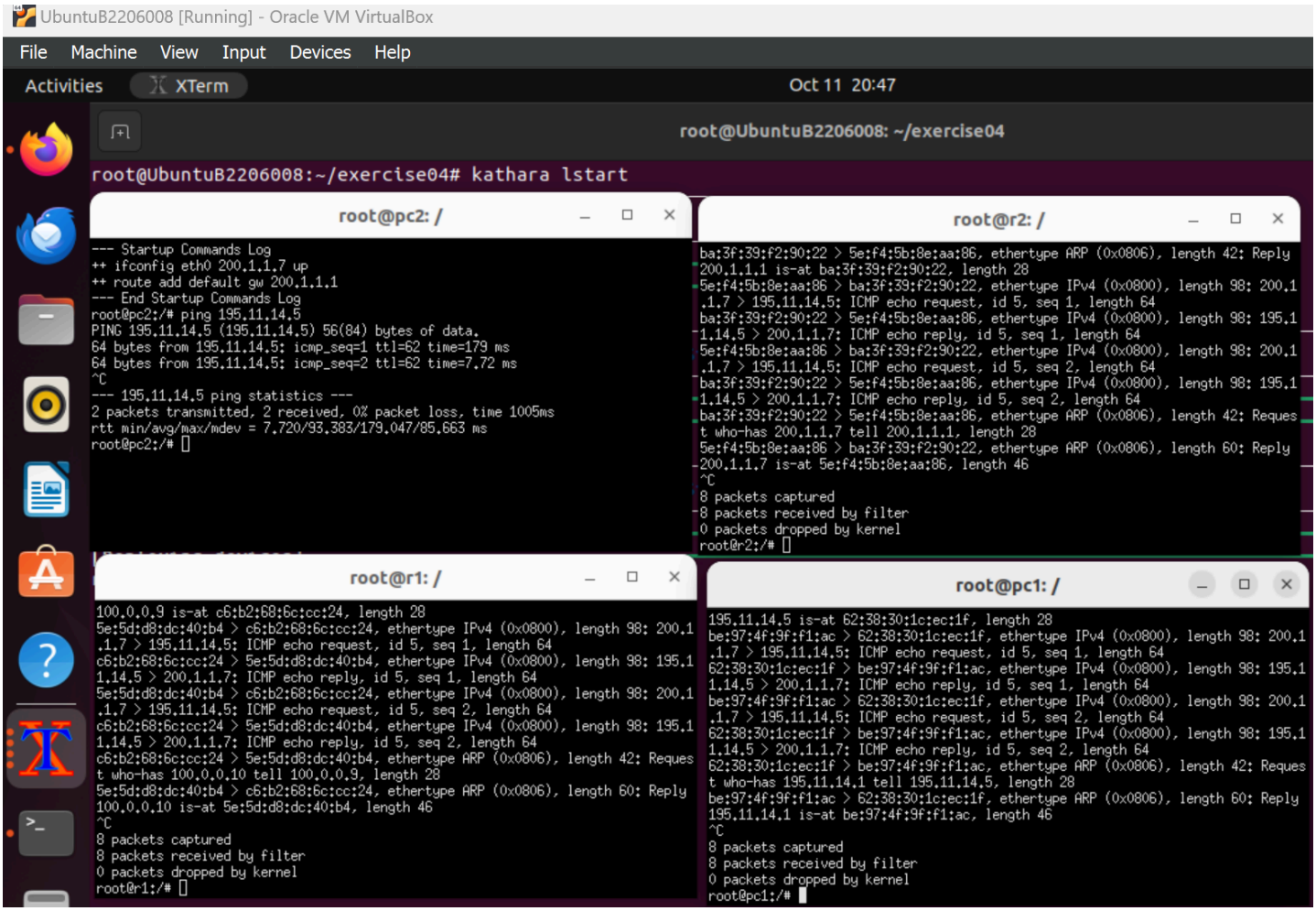
  R1:

  Tcpdump -tenni eth1

  Pc1:

  Tcpdump -tenni eth0





- **Proposed Exercise:**
  If we keep pinging from pc2 to pc1, there are only unicast packets since the ARP resolution from pc2 to pc1 has been completed, so there's no need of broadcast transmitting.
- **Proposed Exercise**
  Local destination: the source device keep broadcasting ask for the MAC address of destination device, but no device reply since they don't know.
  Non-local destination: Error: Destination net unreachable
  Packages captured:
  Local destination: just ARP requests but no ARP replies.
  Non-local destination: no packet is exchanged in local domain

Exercise 5:

Solution:

File:

lab.conf:

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

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
```

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
```

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
```

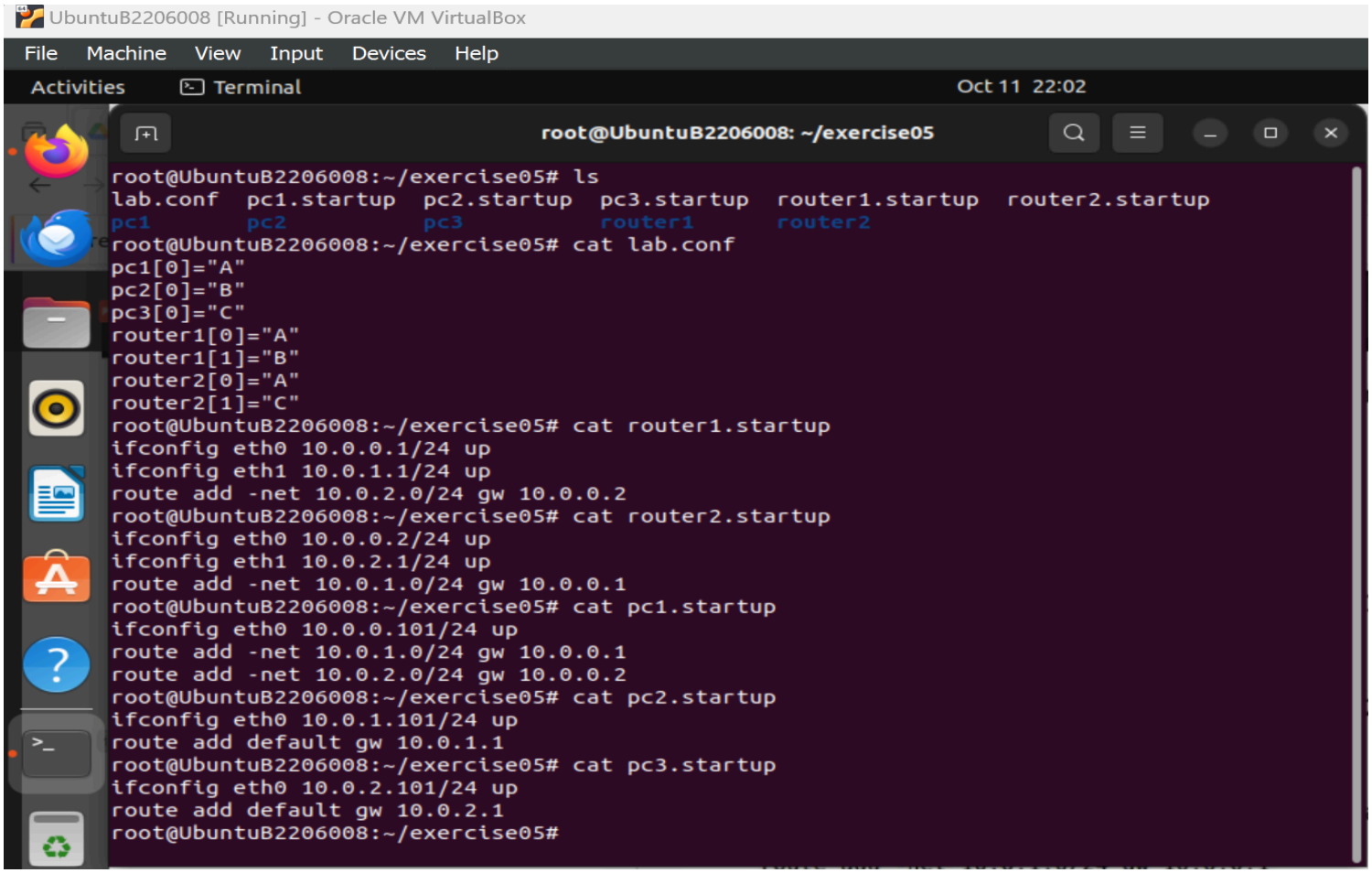
pc2.startup:

```
ifconfig eth0 10.0.1.101/24 up
route add default gw 10.0.1.1
```

pc3.startup:

```
ifconfig eth0 10.0.2.101/24 up
route add default gw 10.0.2.1
```

- **Config scenario**



- **Ping check the network**

  Pc1:

  Route -n

  Ping 10.0.1.101

  Ping 10.0.2.101

  Pc2:

  Route -n

  Ping 10.0.2.101

  Pc3:

  Route -n

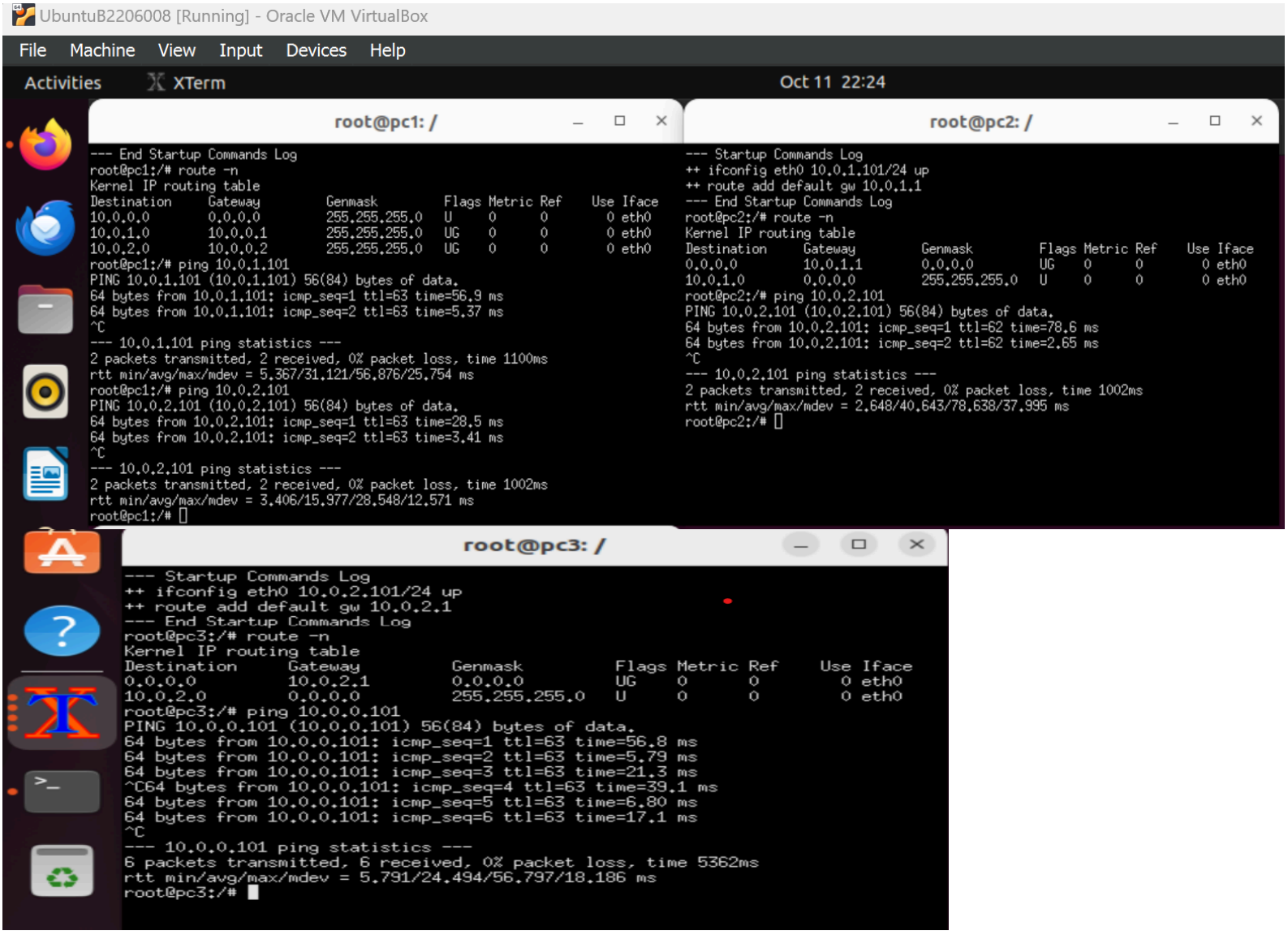
  Ping 10.0.0.101

  Router1:

  Router -n

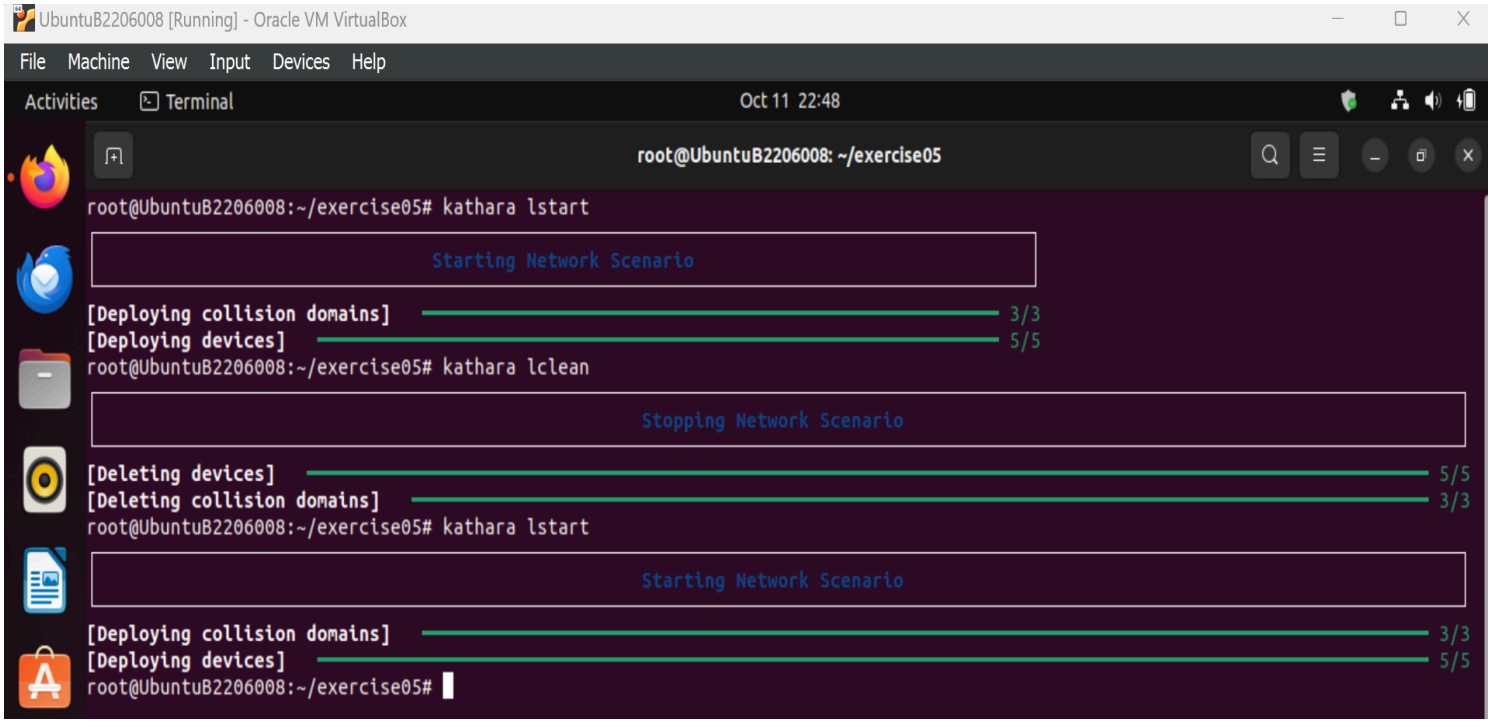
  Router2:

  Router -n



Wireshark

- **Start the scenario**



- **Ping from pc3 and sniffing at pc2, router1, router2**

  Pc2:

  Tcpdump -s 1536 -w /shared/Ex5_pc2.pcap

  Router1:

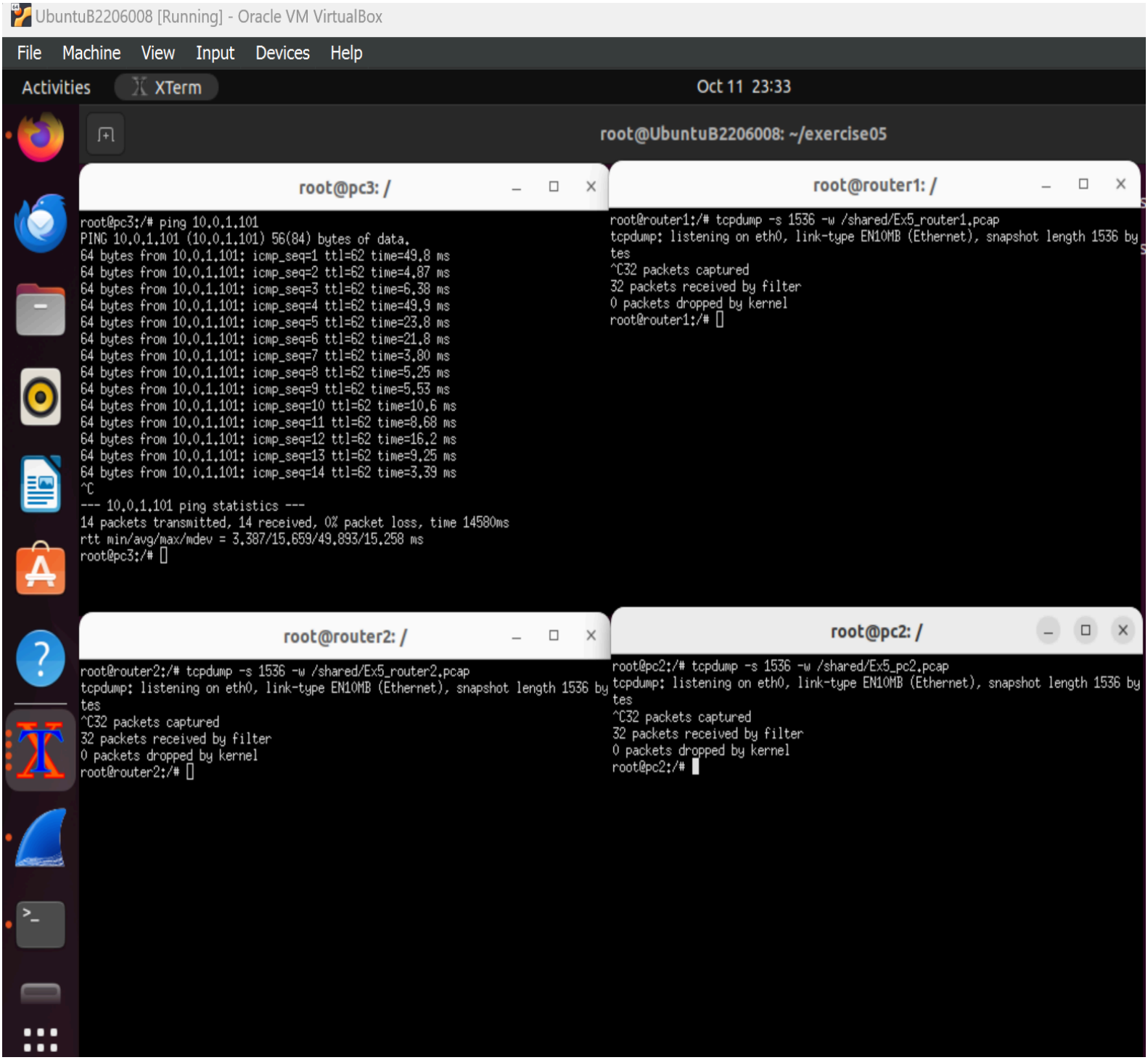
  Tcpdump -s 1536 -w /shared/Ex5_router1.pcap

  Router2:

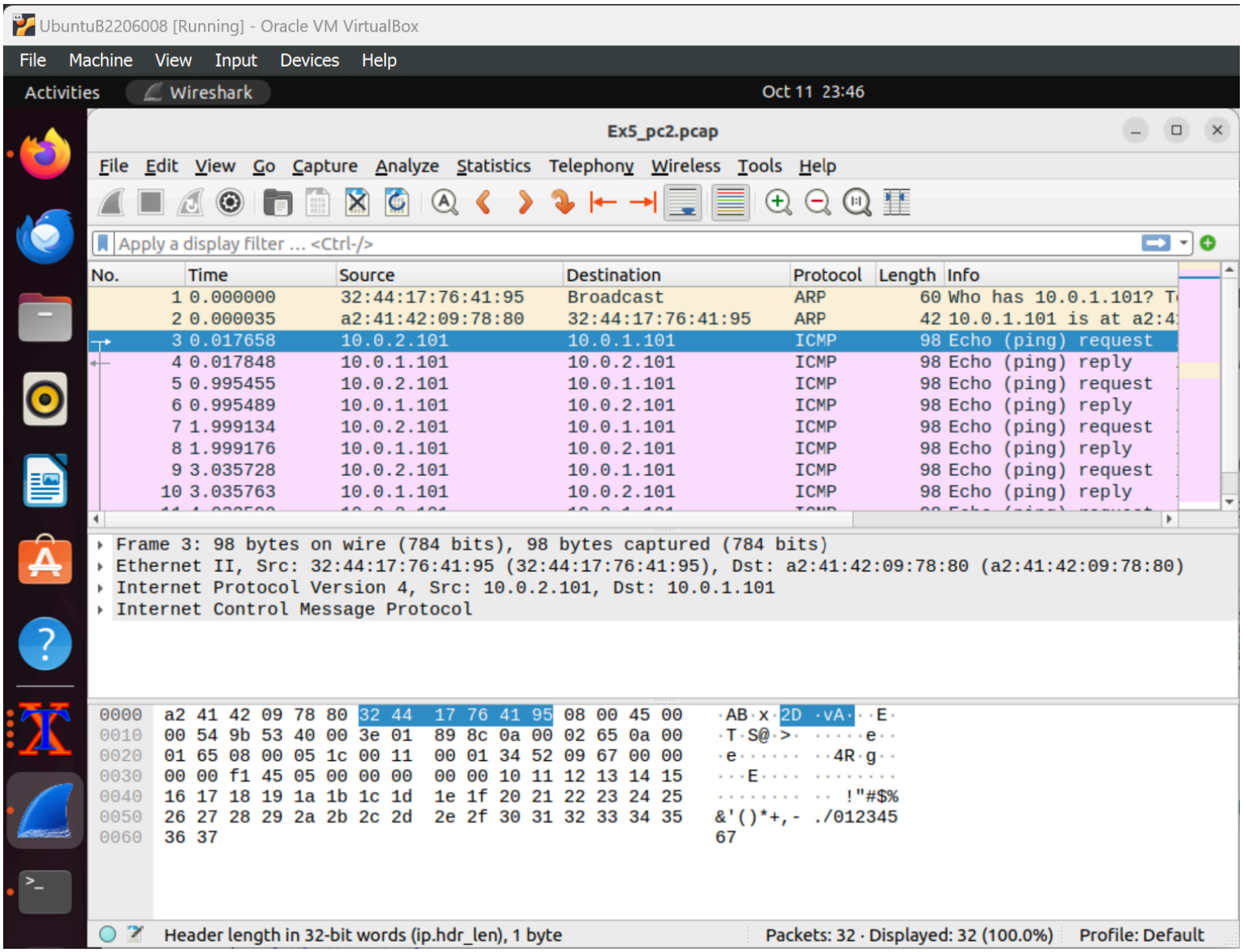
  Tcpdump -s 1536 -w /shared/Ex5_router2.pcap

  Pc3:

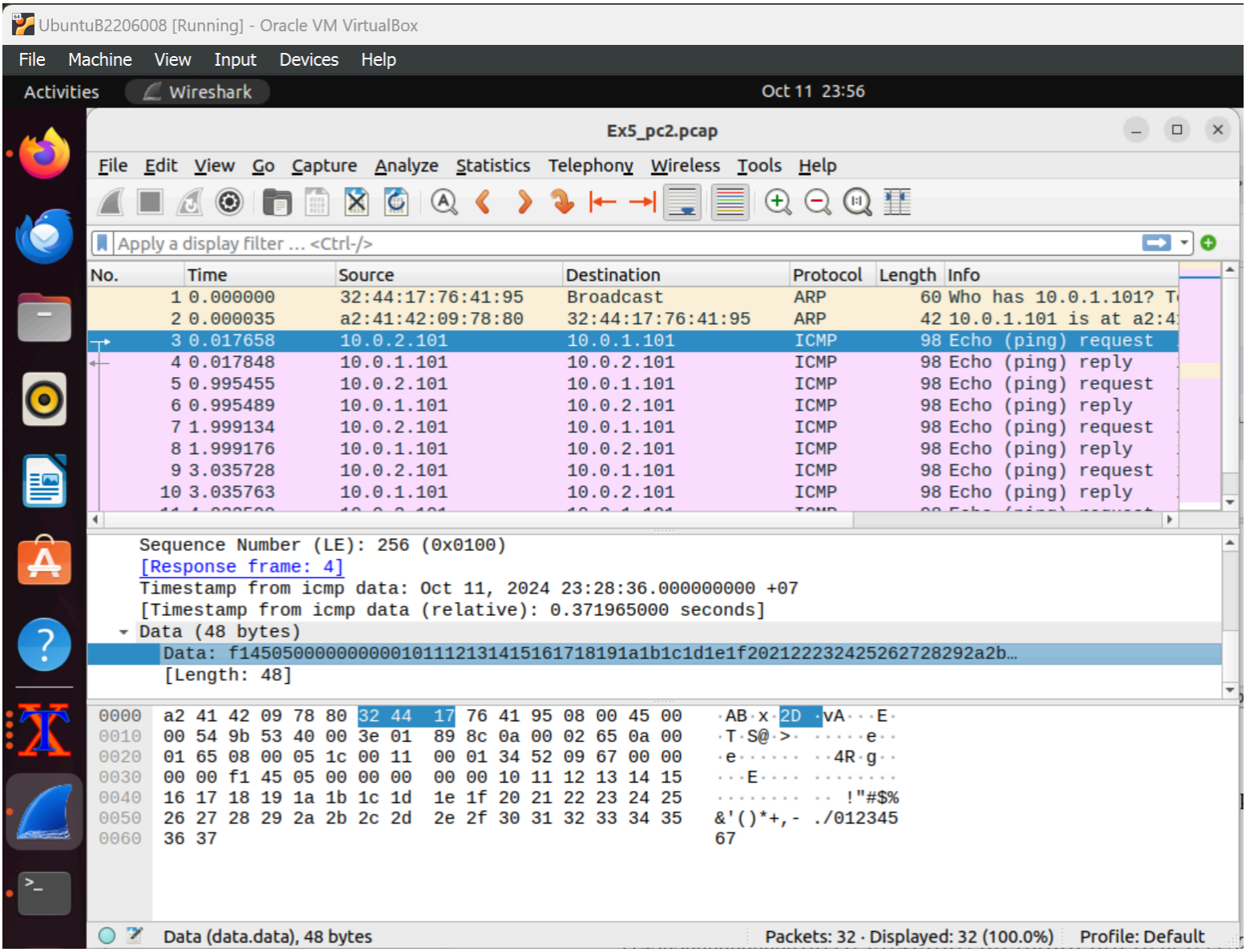
  Ping 10.0.1.101



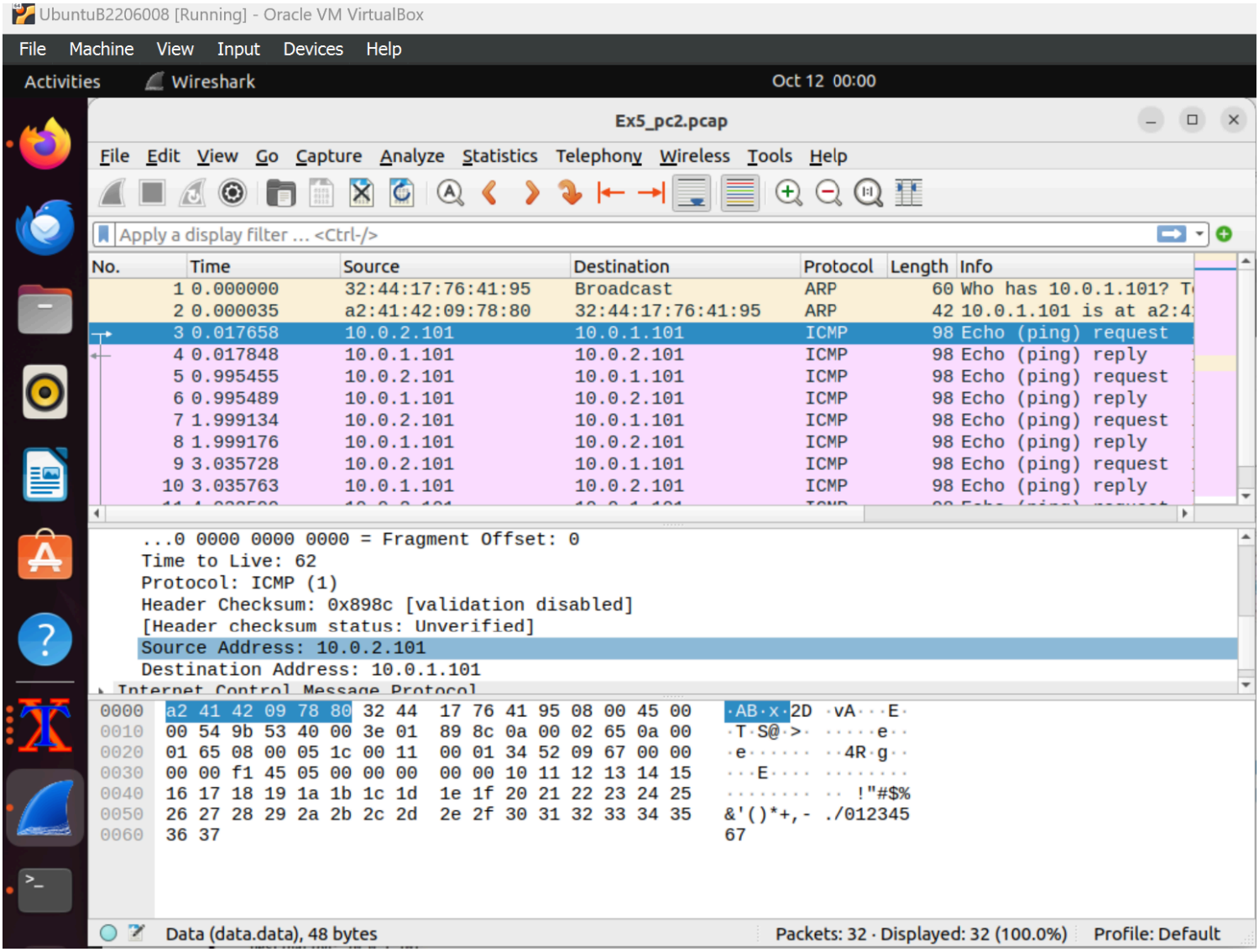
- **Open Ex5_pc2 with Wireshark**
  Wireshark ~/exercise05/shared/Ex5_pc2.pcap



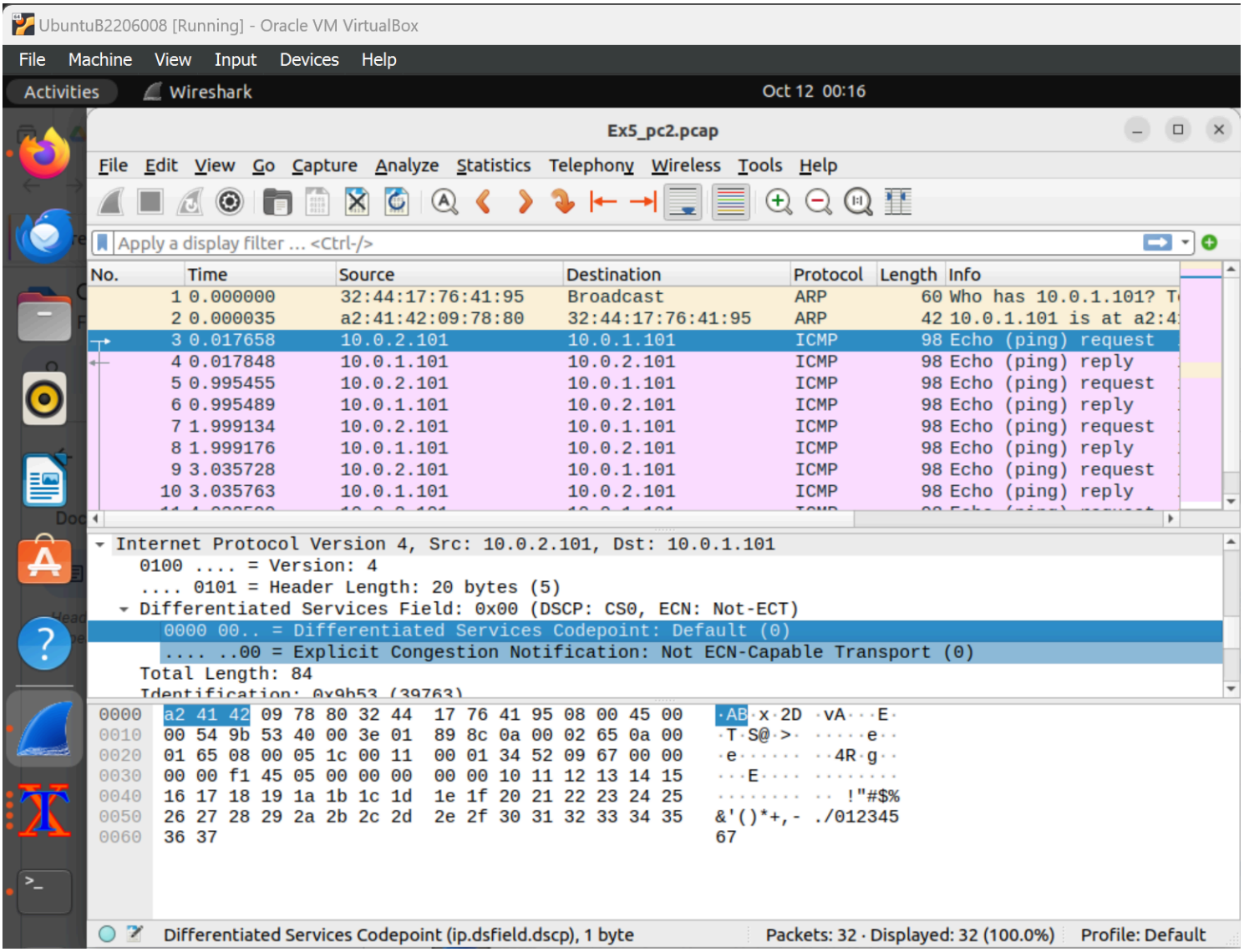
- **Size of frame #3 in bytes? -** 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
- **Protocol is using? -** ICMP
  This protocol operate in which OSI layer? - Network Layer



Content of message? -
f1450500000000001011121314151617181 91a1b1c1d1e1f20212223242526
2728292a2b2c2d2e2f3031323334353637
Size of message in bytes? - 48 bytes

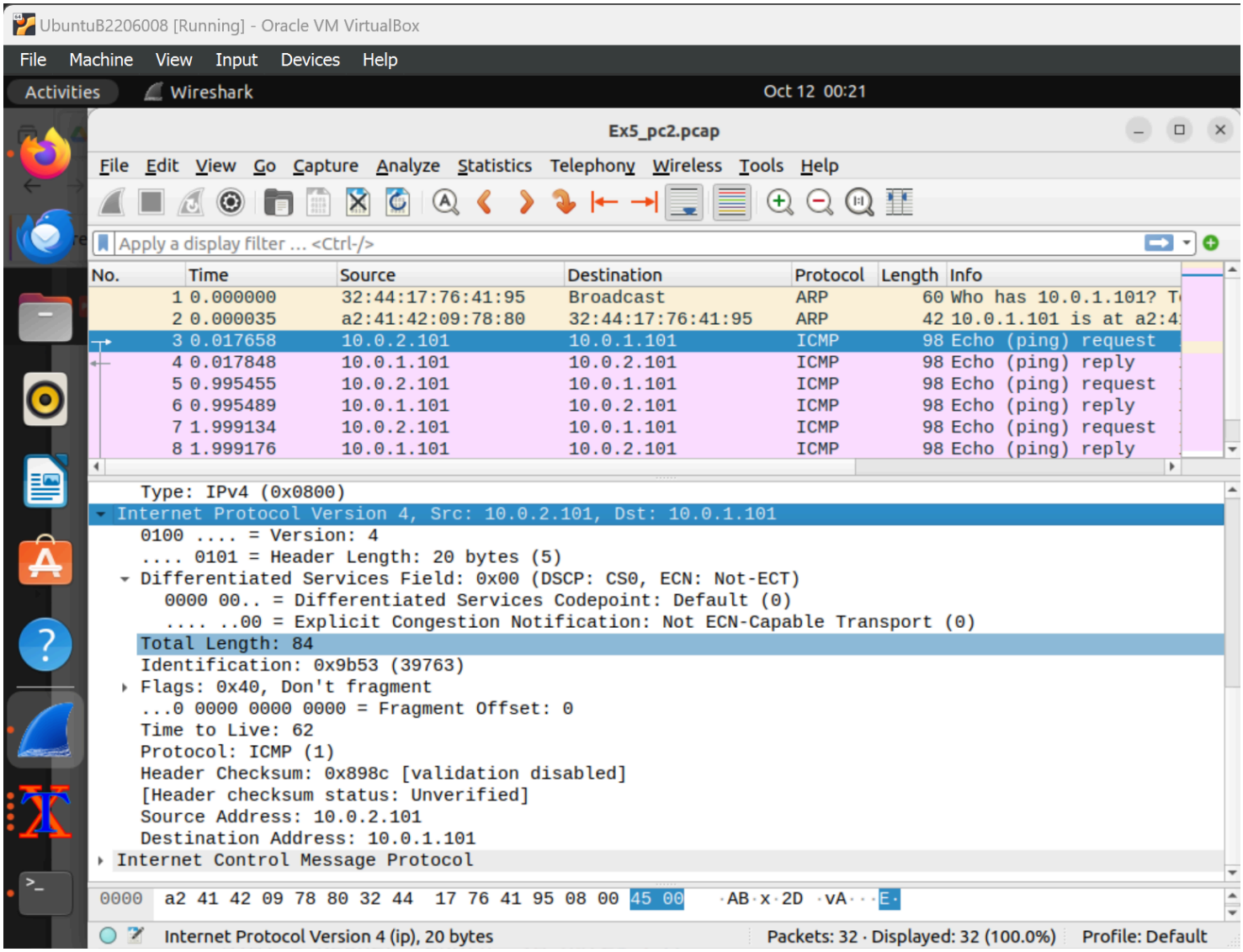


- **IP address of source and destination? -**
  source: 10.0.2.101
  destination: 10.0.1.101

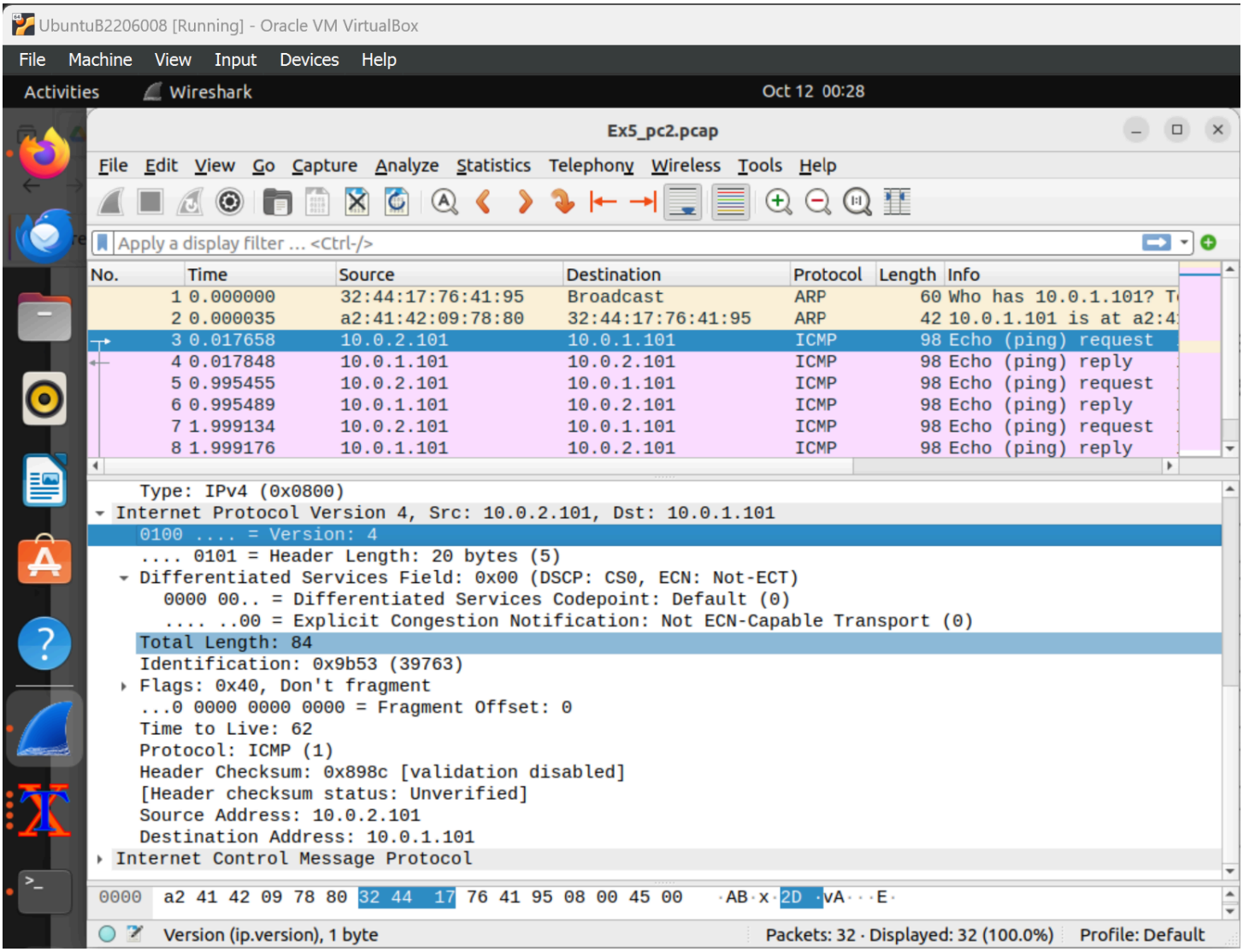


- **What is the length of the IP packet header? -** 20 bytes

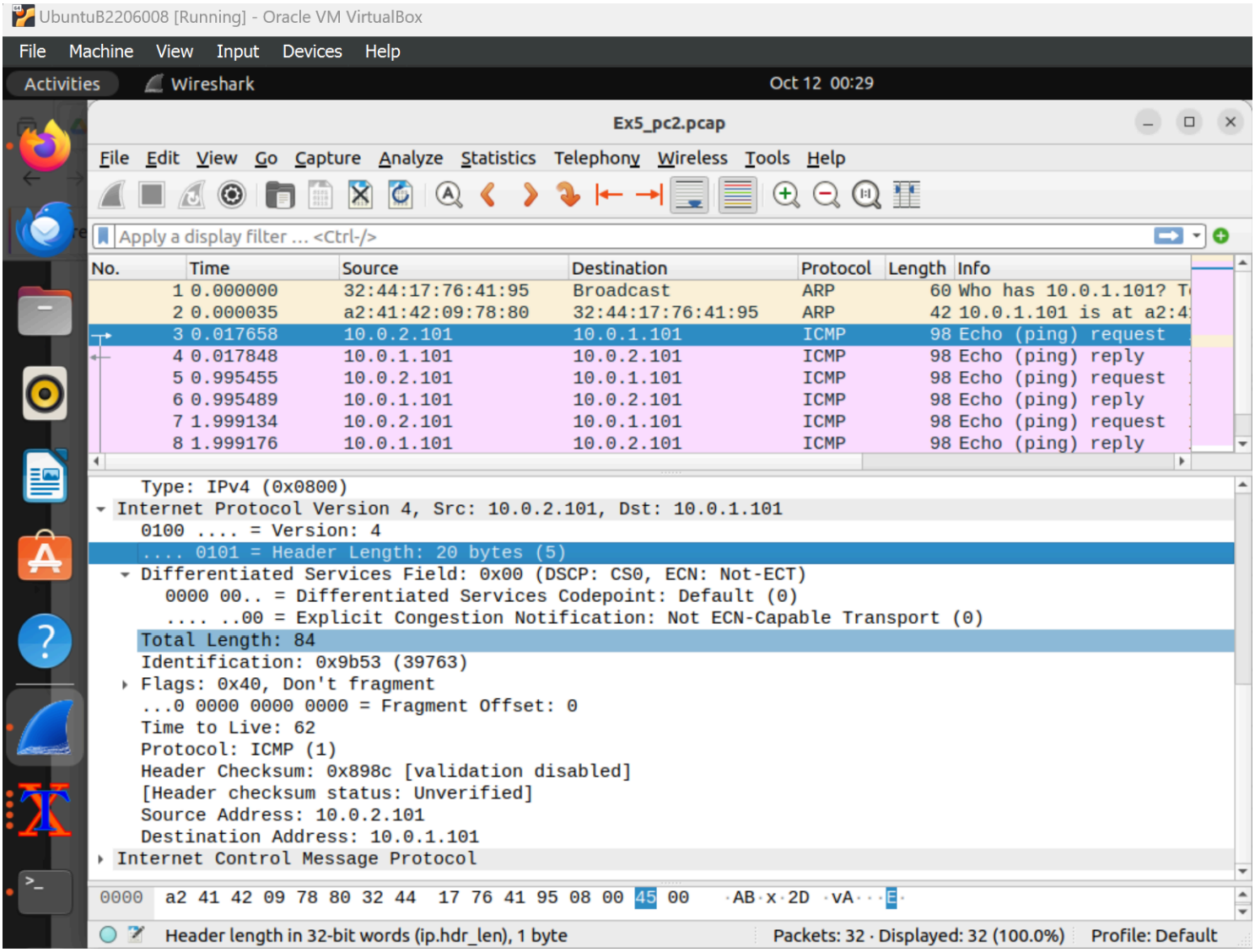
- **What fields does the Header include? -** from Version field to Destination Address field



- **How long is each field (Bytes) ?**

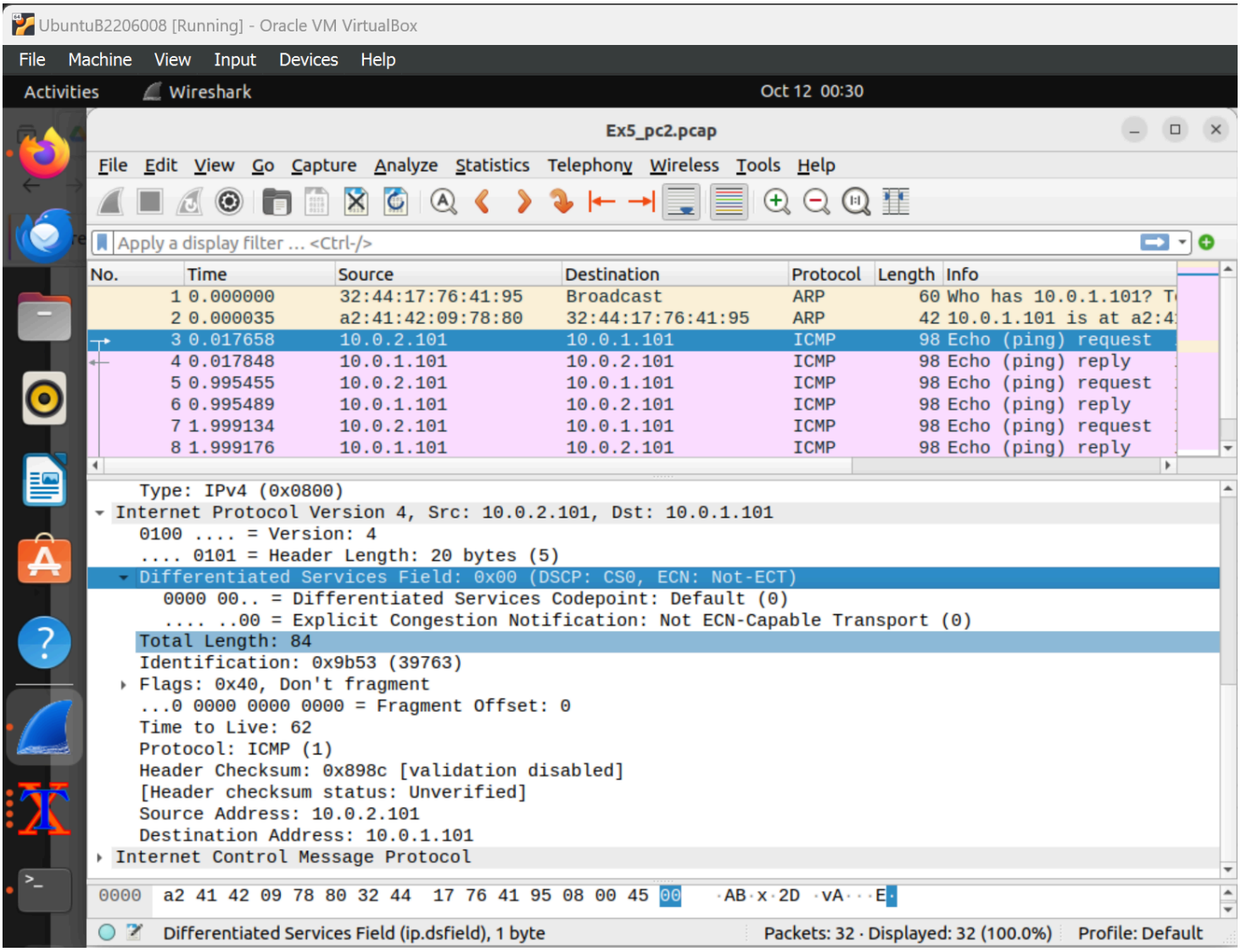


=> Version: 1 byte (4 bits)

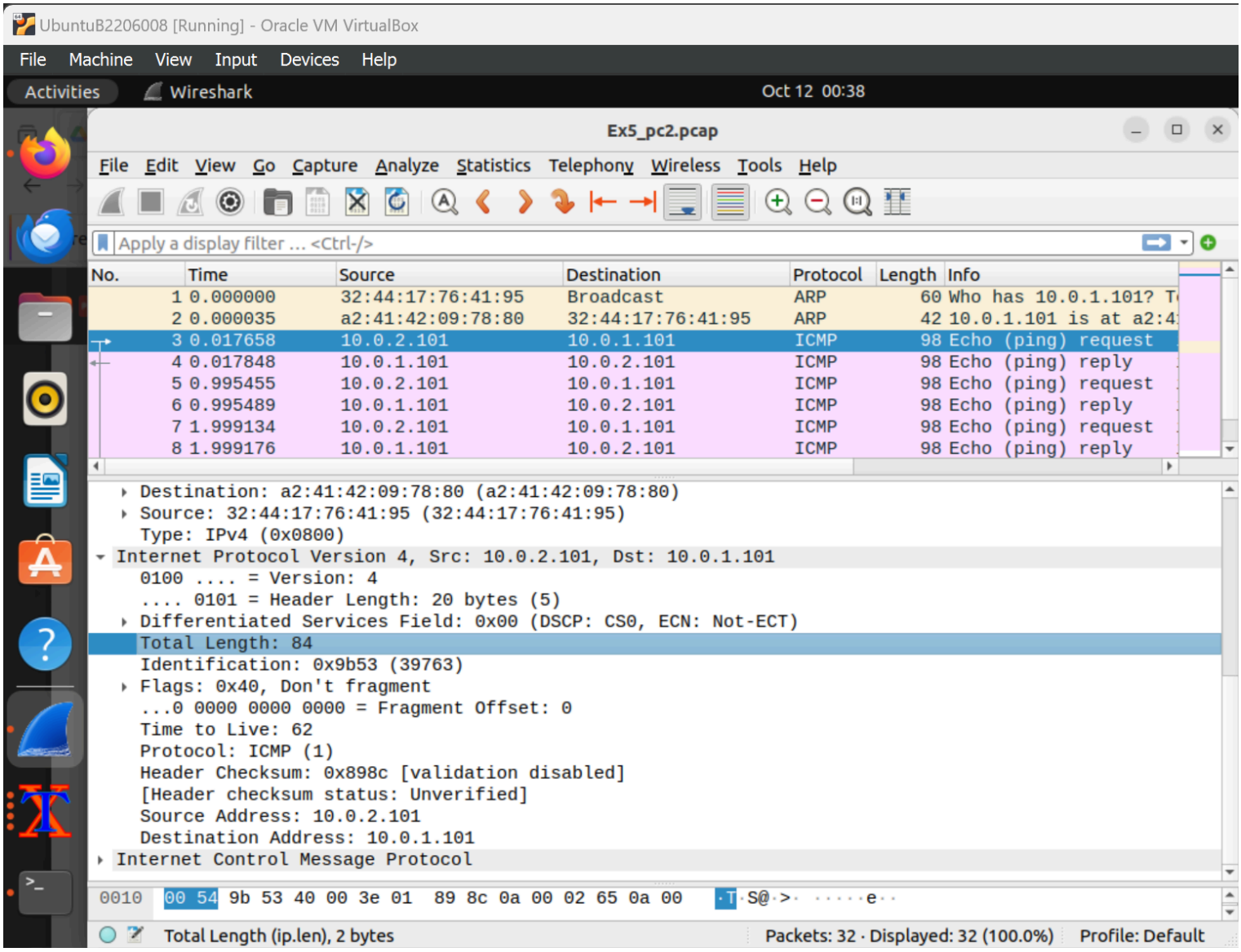


=> Header length in words: 1 byte (4 bits)

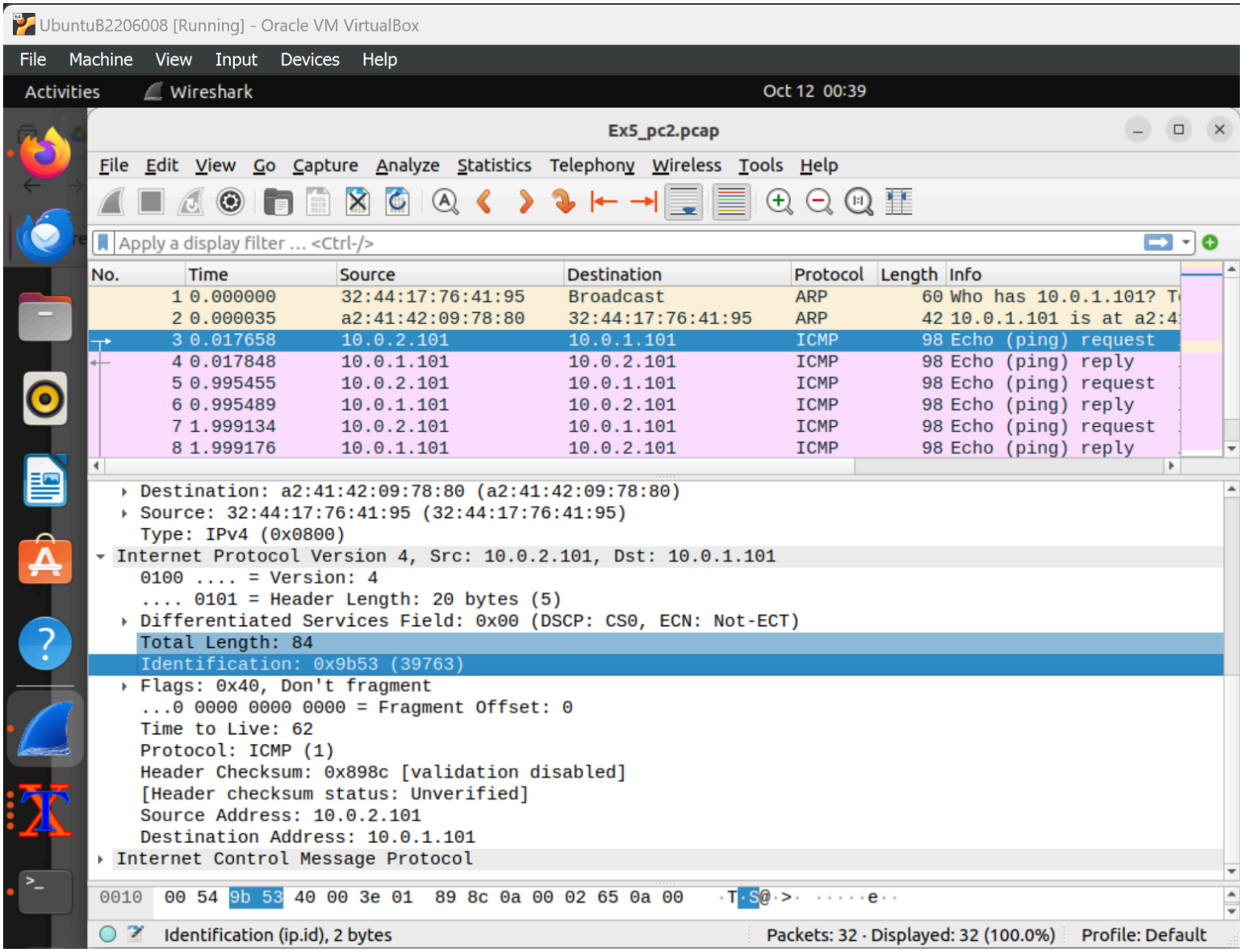
=> Version + Header length in words = 1 byte



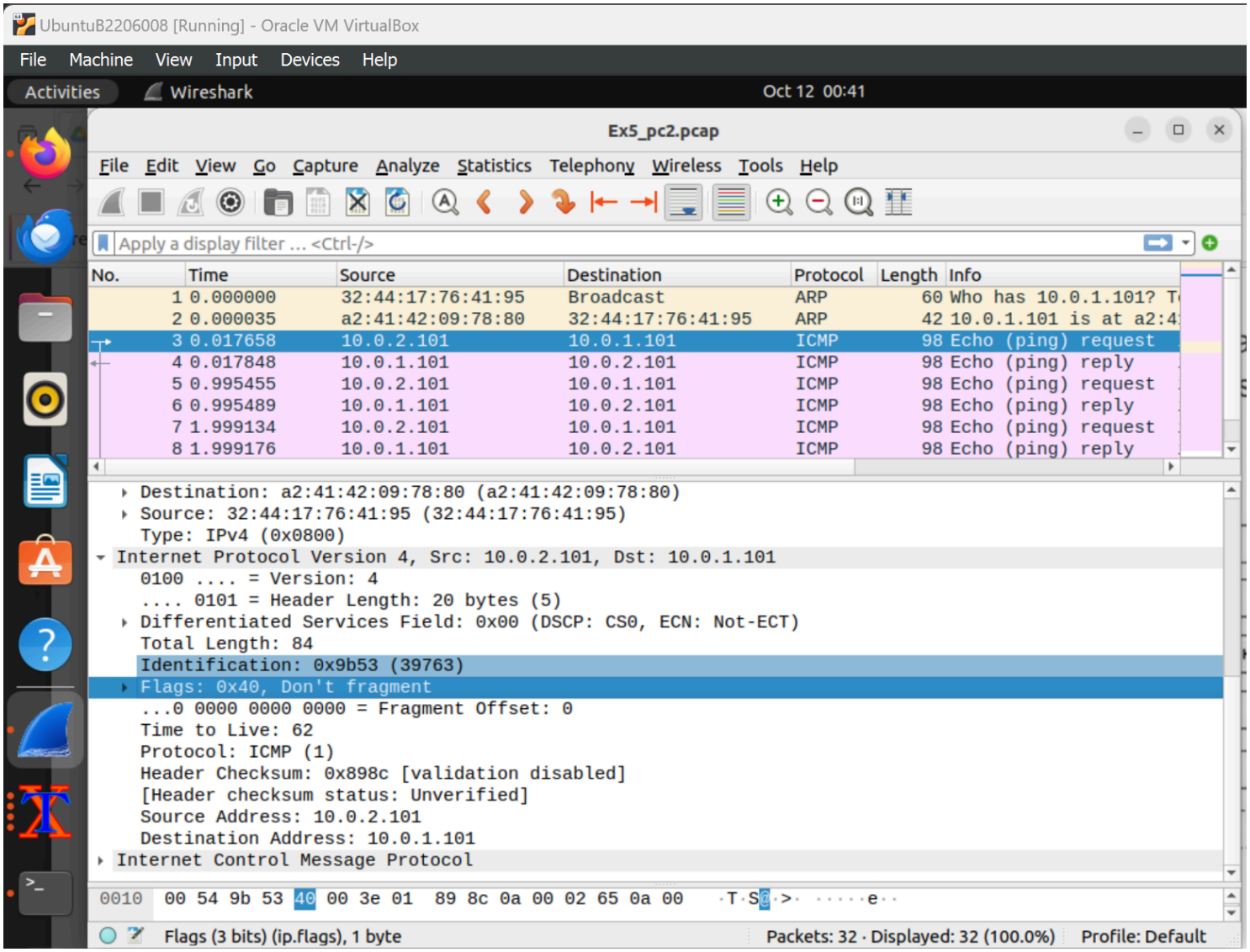
=> Differentiated Services Field: 1 byte



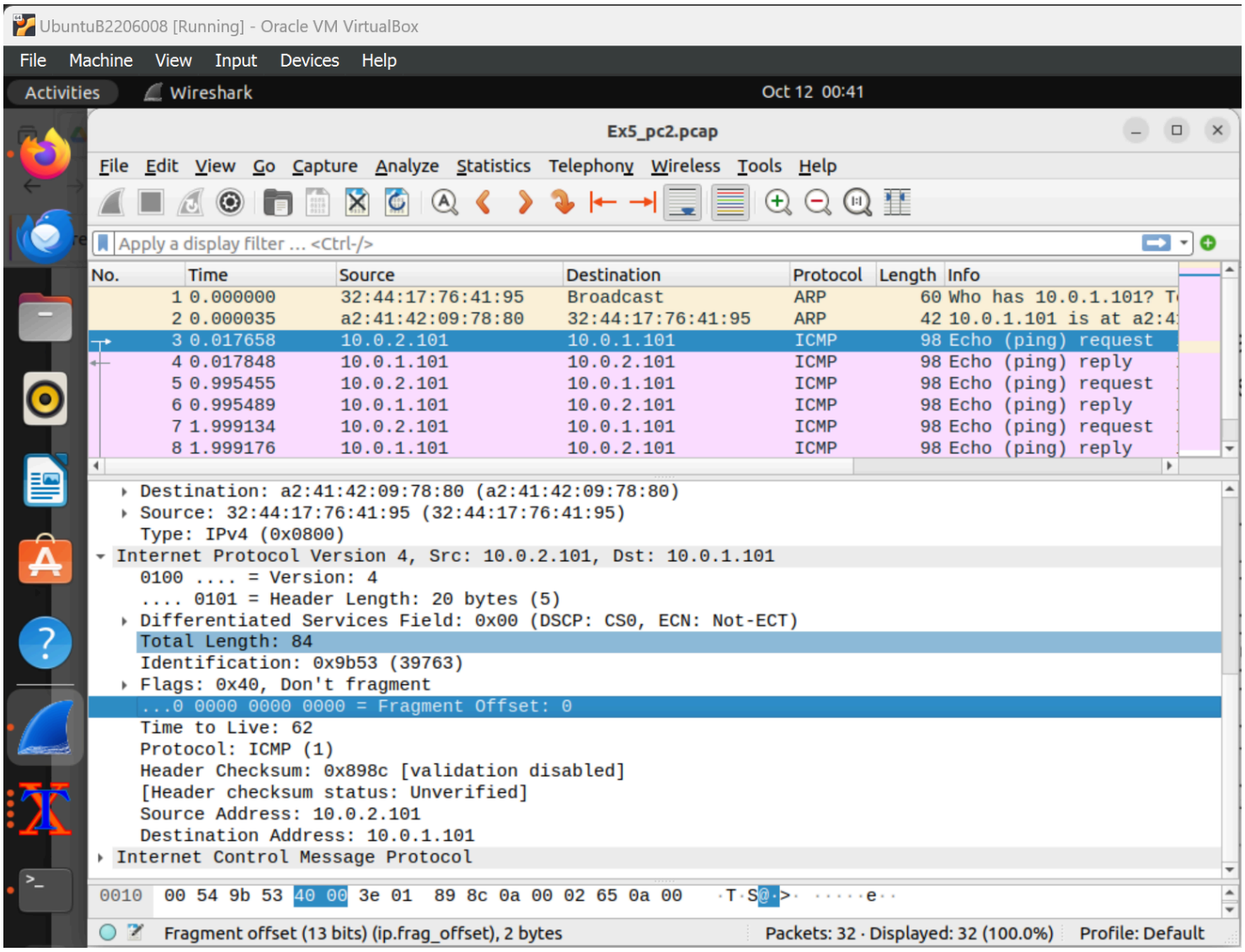
=> Total Length: 2 bytes



=> Identification: 2 bytes

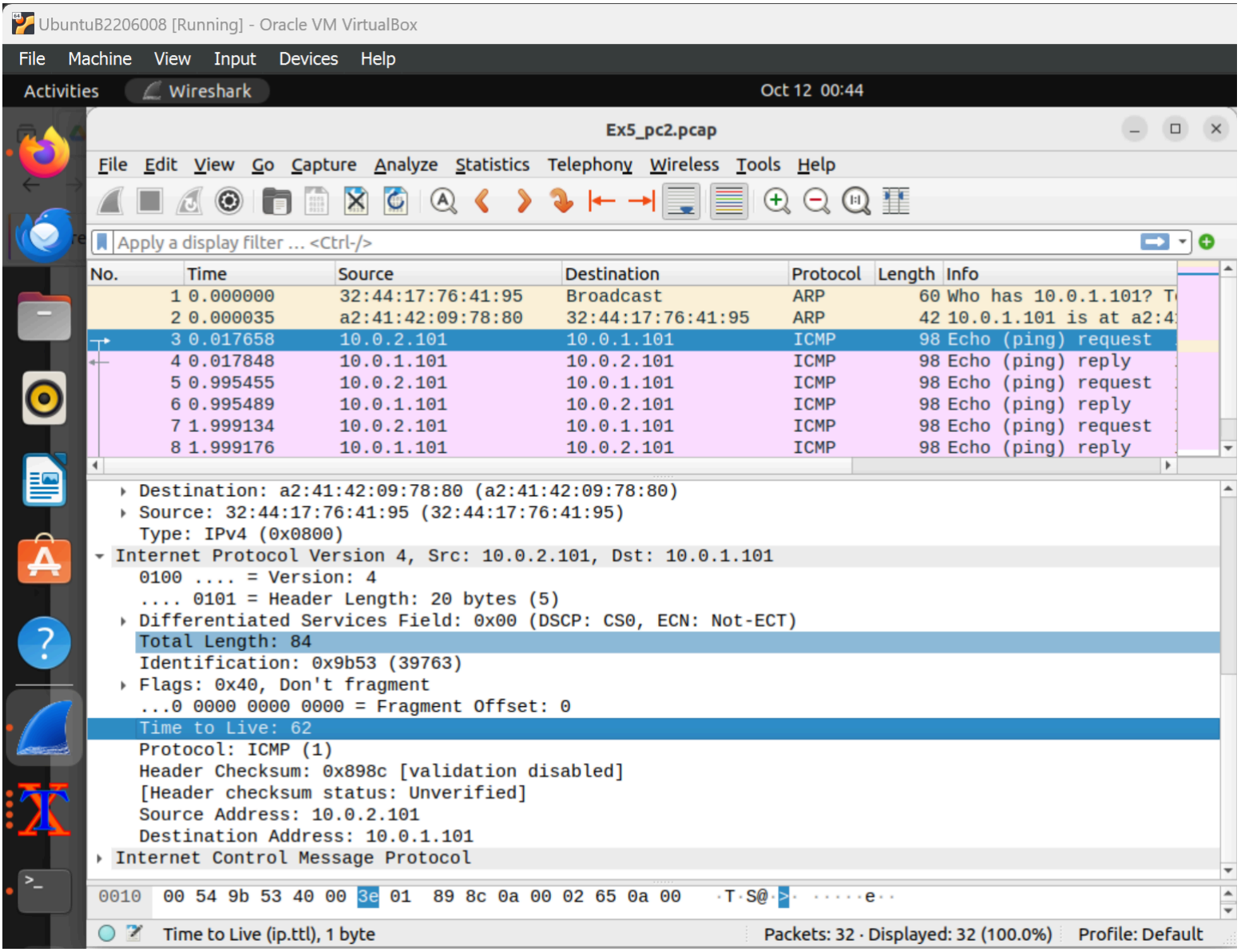


=> Flags: 1 byte (3 bit)

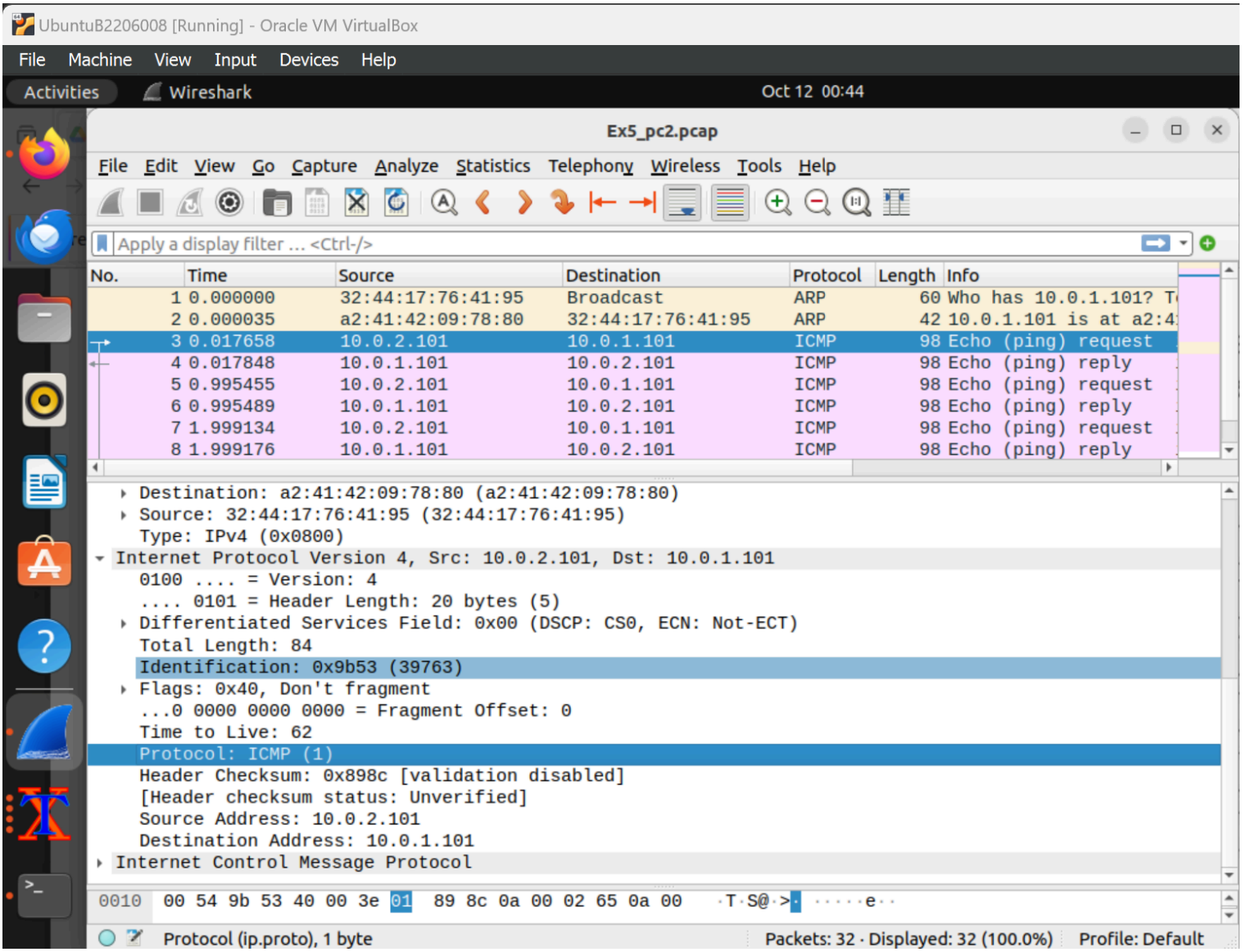


=> Fragment Offset: 1 byte

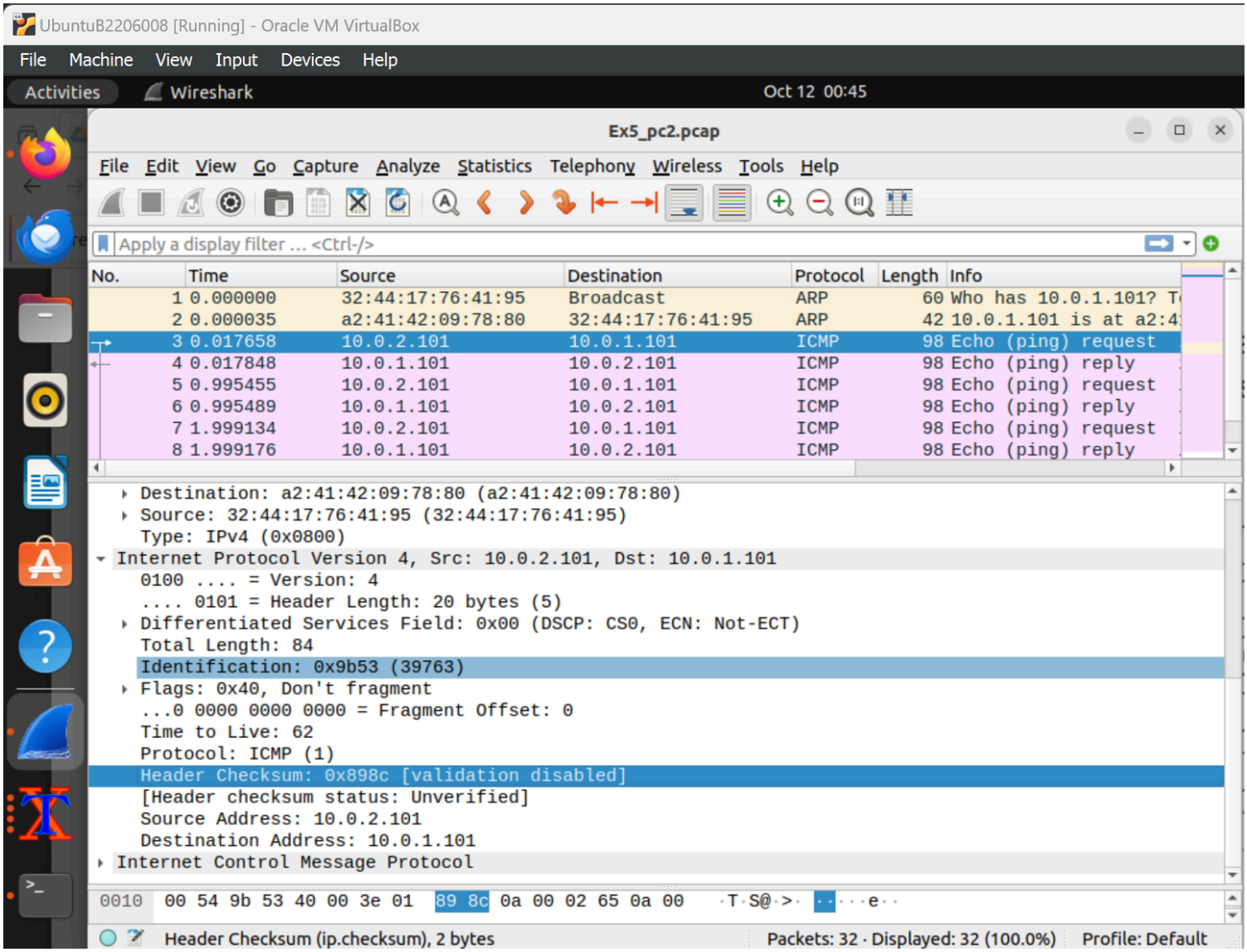
=> Flags(3 bit) and Fragment Offset (13 bit) combine and take 2 bytes.



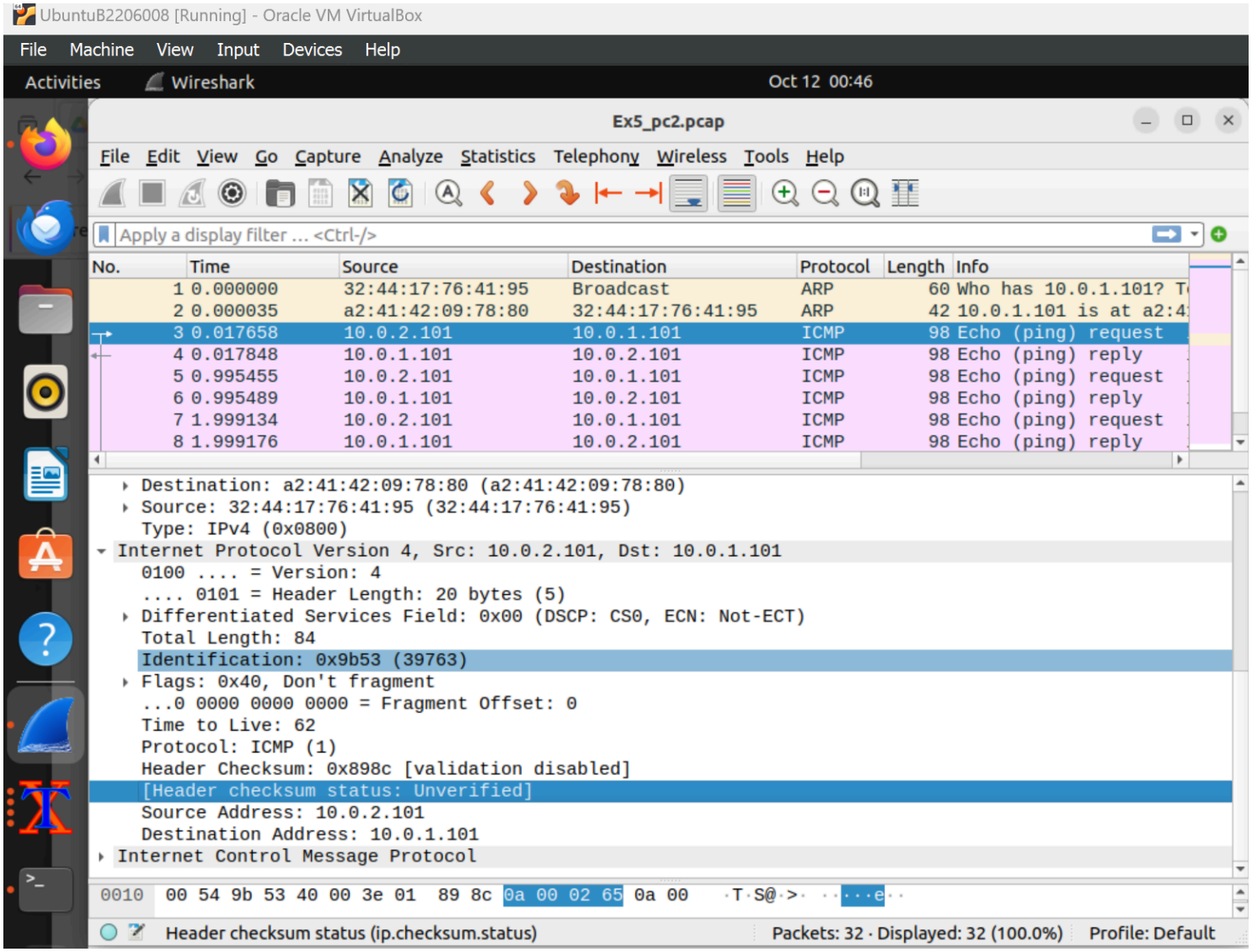
=> Time to Live: 1 bytes



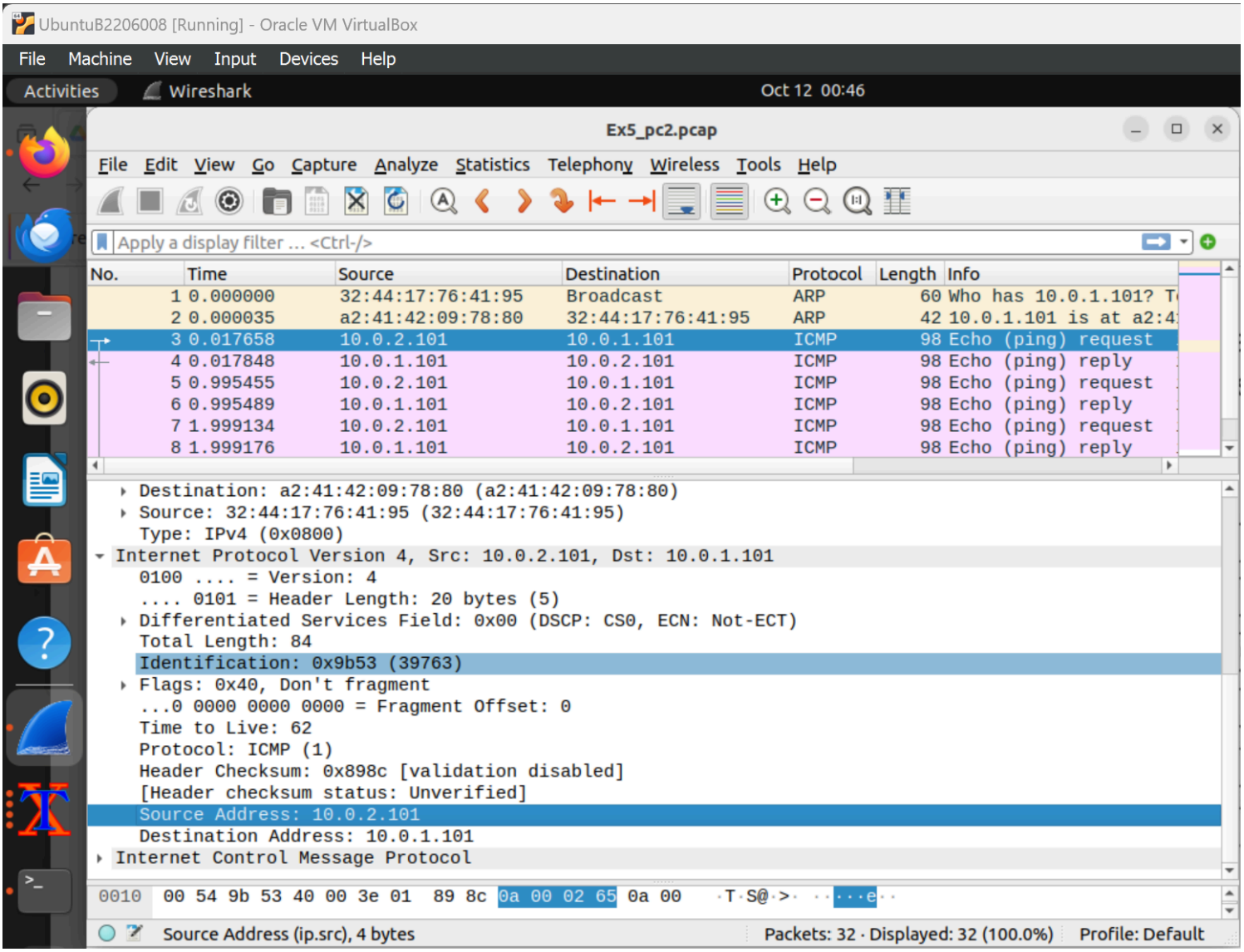
=> Protocol: 1 byte



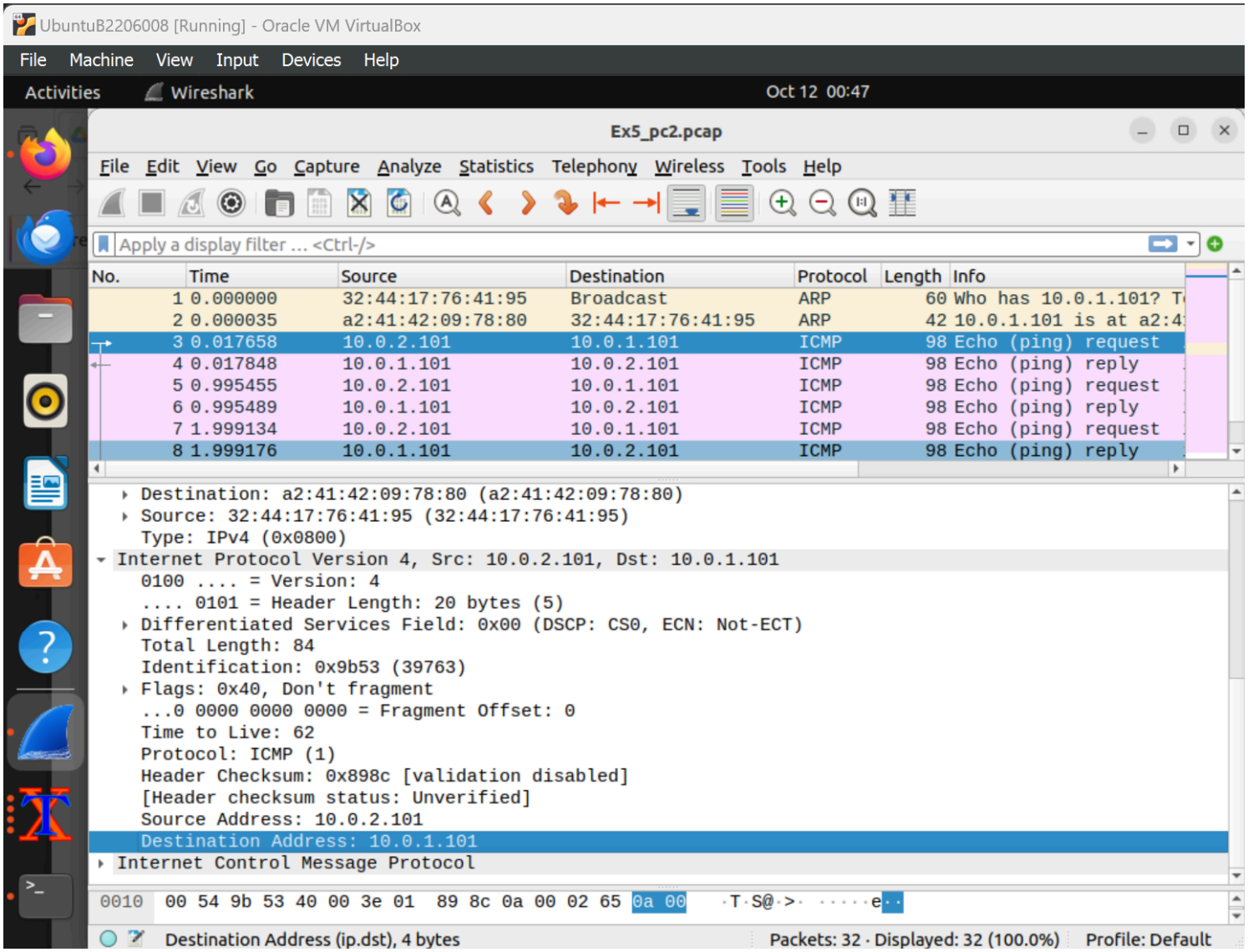
=>Header Checksum: 2 bytes



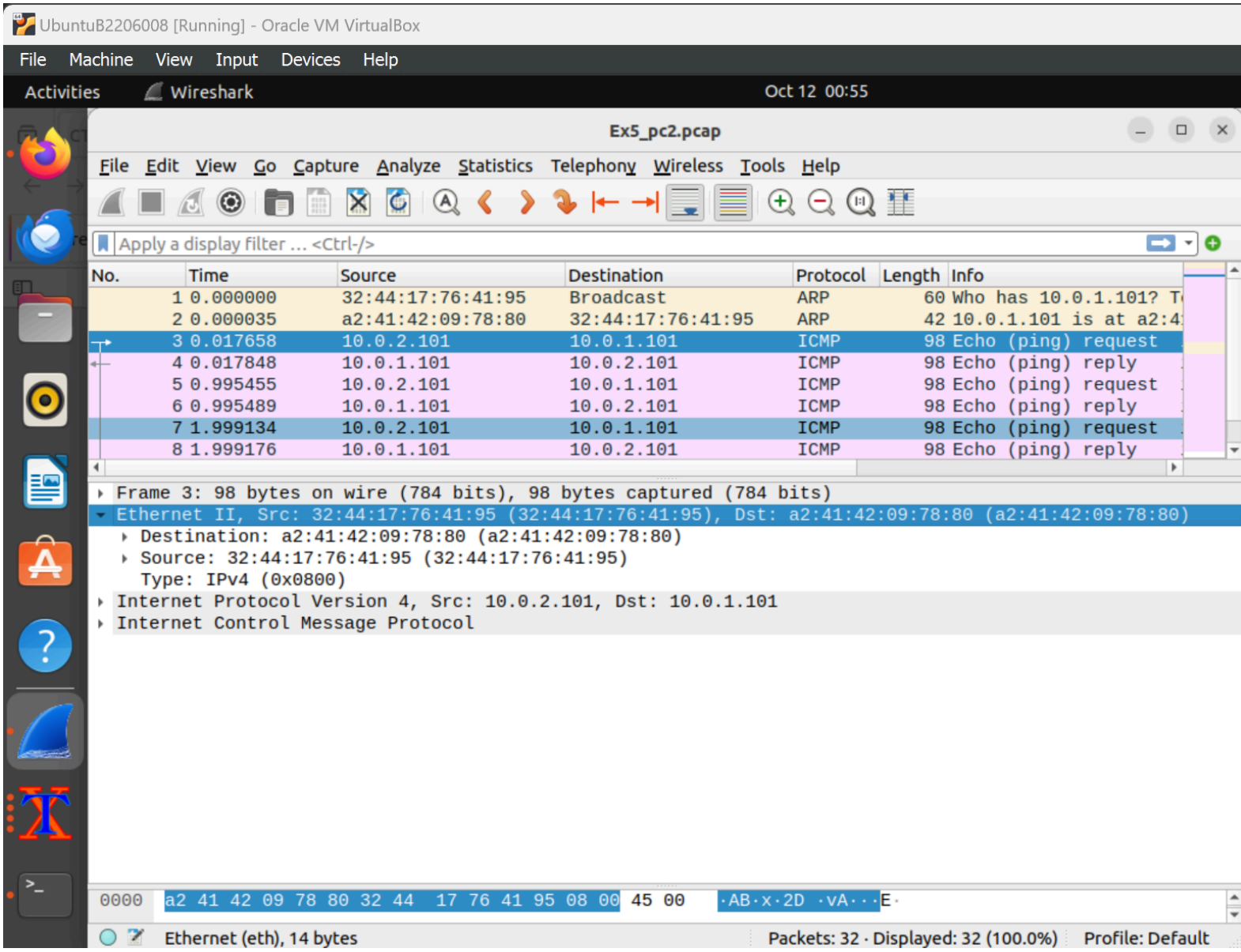
=> Header checksum status: 0 byte



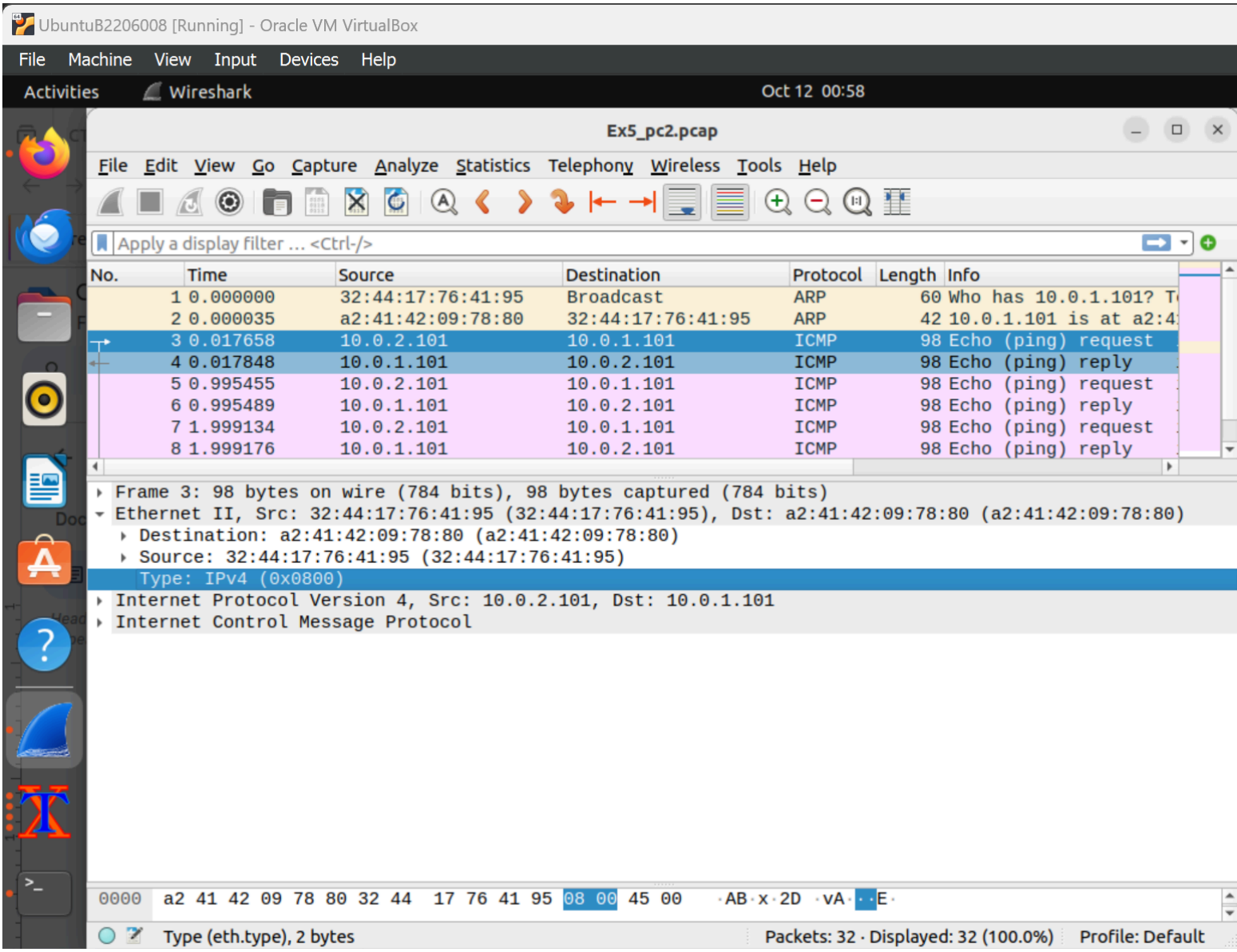
=> Source Address: 4 bytes



=> Destination Address: 4 bytes



- **MAC address of source and destination host?**
  Destination: a2:41:42:09:78:80
  Source: 32:44:17:76:41:95



- **What is the Type value? -** 0x0800