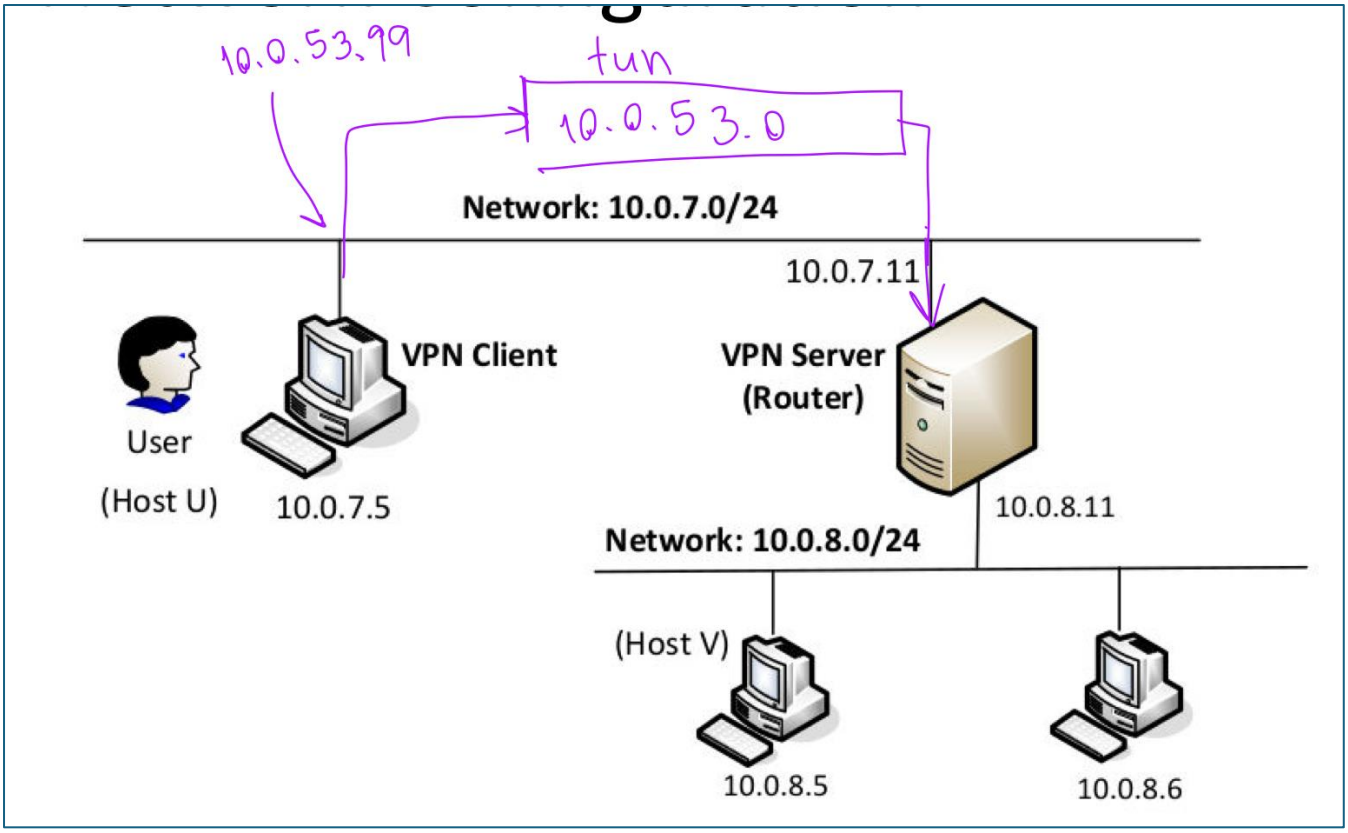
Lab 9 – VPN Tunneling

Task 1: Network Setup

ใช้ docker-compose.yml ในการทำ task

```
[04/15/25]seed@VM:~/.../volumes$ dockps
43bb387c1caa  server-router
88c230d86669  host-10.0.8.5
39eb22973d33  host-10.0.8.6
e2eaf25532bf  client-10.0.7.5
[04/15/25]seed@VM:~/.../volumes$
```



Testing:

- host ภายนอก communicate กับ VPN server ได้

```
root@e2eaf25532bf:/# ping 10.0.7.11
PING 10.0.7.11 (10.0.7.11) 56(84) bytes of data.
64 bytes from 10.0.7.11: icmp_seq=1 ttl=64 time=0.508 ms
64 bytes from 10.0.7.11: icmp_seq=2 ttl=64 time=0.068 ms
64 bytes from 10.0.7.11: icmp_seq=3 ttl=64 time=0.161 ms
64 bytes from 10.0.7.11: icmp_seq=4 ttl=64 time=0.061 ms
^C
--- 10.0.7.11 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3060ms
rtt min/avg/max/mdev = 0.061/0.199/0.508/0.182 ms
root@e2eaf25532bf:/#
```

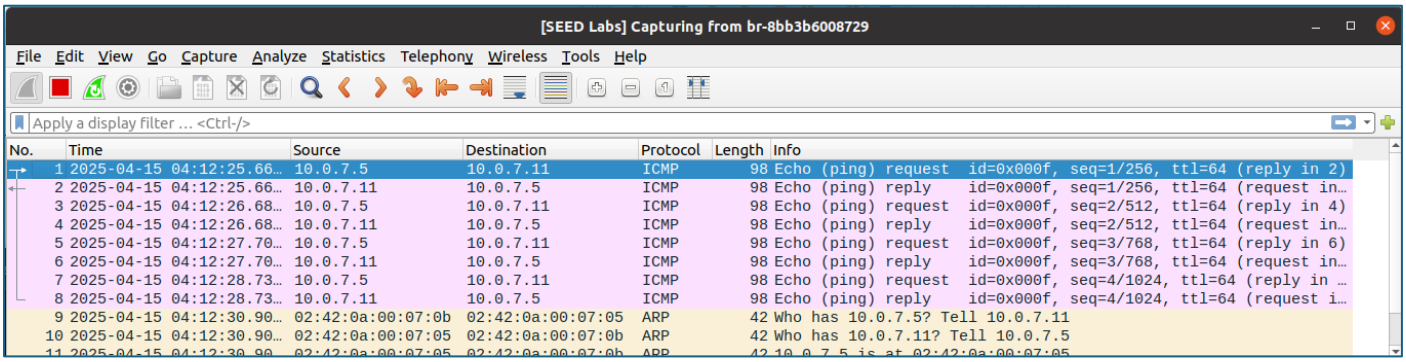
- VPN server communicate กับ host ภายในได้

```
root@43bb387c1caa:/# ping 10.0.8.5
PING 10.0.8.5 (10.0.8.5) 56(84) bytes of data.
64 bytes from 10.0.8.5: icmp_seq=1 ttl=64 time=0.195 ms
64 bytes from 10.0.8.5: icmp_seq=2 ttl=64 time=0.143 ms
64 bytes from 10.0.8.5: icmp_seq=3 ttl=64 time=0.091 ms
^C
--- 10.0.8.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2048ms
rtt min/avg/max/mdev = 0.091/0.143/0.195/0.042 ms
root@43bb387c1caa:/# ping 10.0.8.6
PING 10.0.8.6 (10.0.8.6) 56(84) bytes of data.
64 bytes from 10.0.8.6: icmp_seq=1 ttl=64 time=0.320 ms
64 bytes from 10.0.8.6: icmp_seq=2 ttl=64 time=0.096 ms
64 bytes from 10.0.8.6: icmp_seq=3 ttl=64 time=0.092 ms
^C
--- 10.0.8.6 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2048ms
rtt min/avg/max/mdev = 0.092/0.169/0.320/0.106 ms
root@43bb387c1caa:/#
```

- host ภายนอกไม่ควรจะ communicate กับ host ภายในได้

```
root@e2eaf25532bf:/# ping 10.0.8.5
PING 10.0.8.5 (10.0.8.5) 56(84) bytes of data.
^C
--- 10.0.8.5 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5106ms

root@e2eaf25532bf:/# ping 10.0.8.6
PING 10.0.8.6 (10.0.8.6) 56(84) bytes of data.
^C
--- 10.0.8.6 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5186ms

root@e2eaf25532bf:/#
```

- แสดงการตรวจจับ packet



Task 2: Create and Configure TUN Interface

Task 2.a: Name of the Interface

สร้าง TUN interface ที่ host U (10.0.7.5)

Open ▾ tun.py ~/Documents/Lab9-VPN/volumes

```python
#!/usr/bin/env python3

import fcntl
import struct
import os
import time
from scapy.all import *

TUNSETIFF = 0x400454ca
IFF_TUN   = 0x0001
IFF_TAP   = 0x0002
IFF_NO_PI = 0x1000

# Create the tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'tun%d', IFF_TUN | IFF_NO_PI)
ifname_bytes  = fcntl.ioctl(tun, TUNSETIFF, ifr)

# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))

while True:
   time.sleep(10)

```

```
root@e2eaf25532bf:/volumes# ls
tap     tun2.py        tun_client_select.py  tun_server.py   tun_server_select.py
tun.py  tun_client.py  tun_read.py           tun_server2.py
root@e2eaf25532bf:/volumes# chmod a+x tun.py
root@e2eaf25532bf:/volumes# tun.py
Interface Name: tun0
```

เมื่อใช้คำสั่ง ip address จะเห็น interface: tun0 ตามที่รันโค้ดไว้

```
root@e2eaf25532bf:/# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: tun0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 500
    link/none
77: eth0@if78: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:00:07:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.0.7.5/24 brd 10.0.7.255 scope global eth0
       valid_lft forever preferred_lft forever
root@e2eaf25532bf:/#
```

Task 2.b: Set up the TUN Interface

ใส่ IP ให้ tun0 และสั่ง up interface ที่ host U จะเห็นว่า tun0 มี ip แล้ว และ state กลายเป็น <UP>

```
root@e2eaf25532bf:/# ip addr add 10.0.53.99/24 dev tun0
root@e2eaf25532bf:/# ip link set dev tun0 up
root@e2eaf25532bf:/# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen
 500
    link/none
    inet 10.0.53.99/24 scope global tun0
       valid_lft forever preferred_lft forever
77: eth0@if78: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:00:07:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.0.7.5/24 brd 10.0.7.255 scope global eth0
       valid_lft forever preferred_lft forever
```

Task 2.c: Read from the TUN Interface

แก้ไขโค้ด: เติมบรรทัดที่ 23-24 เพื่อ assign IP และสั่ง interface up และแก้ไข while loop ให้รออ่าน packet ที่ผ่าน tun0

```
23 os.system("ip addr add 10.0.53.99/24 dev {}".format(ifname))
24 os.system("ip link set dev {} up".format(ifname))
25
26 while True:
27     # Get a packet from the tun interface
28     packet = os.read(tun, 2048)
29     if packet:
30         ip = IP(packet)
31         print(ip.summary())
```

```
root@e2eaf25532bf:/volumes# tun.py
Interface Name: tun0
```

```
root@e2eaf25532bf:/# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen
 500
    link/none
    inet 10.0.53.99/24 scope global tun0
       valid_lft forever preferred_lft forever
77: eth0@if78: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:00:07:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.0.7.5/24 brd 10.0.7.255 scope global eth0
       valid_lft forever preferred_lft forever
```

- ทดลอง ping 10.0.53.0/24 จาก host U แล้วตรวจสอบ packet ที่ tun.py จับได้: จะเห็นว่า tun.py สามารถดักจับได้

```
root@e2eaf25532bf:/volumes# tun.py
Interface Name: tun0
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
```

```
root@e2eaf25532bf:/volumes# ping 10.0.53.1
PING 10.0.53.1 (10.0.53.1) 56(84) bytes of data.
```

- ทดลอง ping 10.0.8.0/24 จาก host U แล้วตรวจสอบ packet ที่ tun.py จับได้: จะเห็นว่า tun.py ดักจับไม่ได้ เนื่องจาก tun0 ยังมีเพียงฝั่ง host U (10.0.7.5)

```
root@e2eaf25532bf:/volumes# tun.py
Interface Name: tun0
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
```

```
root@e2eaf25532bf:/volumes# ping 10.0.8.5
PING 10.0.8.5 (10.0.8.5) 56(84) bytes of data.
```

Task 2.d: Write to the TUN Interface

แก้ไขโค้ดให้เป็นไปดังนี้

- เมื่อได้ packet ที่ tun0 ถ้าเป็น ICMP echo request ให้สร้าง ICMP echo reply ส่งกลับไปผ่าน tun0

tun2.py

```
25
26 while True:
27     # Get a packet from the tun interface
28     packet = os.read(tun, 2048)
29     if packet:
30         pkt = IP(packet)
31         print(pkt.summary())
32
33         # Send out a spoof packet using the tun interface
34         if ICMP in pkt:
35             newip = IP(src=pkt[IP].dst, dst=pkt[IP].src, ihl=pkt[IP].ihl)
36             newip.ttl = 99
37             newicmp   = ICMP(type=0, id=pkt[ICMP].id, seq=pkt[ICMP].seq)
38             if pkt.haslayer(Raw):
39                 data = pkt[Raw].load
40                 newpkt = newip/newicmp/data
41             else:
42                 newpkt = newip/newicmp
43
44             os.write(tun, bytes(newpkt))
```

```
root@e2eaf25532bf:/volumes# ping 10.0.53.1
PING 10.0.53.1 (10.0.53.1) 56(84) bytes of data.
64 bytes from 10.0.53.1: icmp_seq=1 ttl=99 time=16.0 ms
64 bytes from 10.0.53.1: icmp_seq=2 ttl=99 time=1.76 ms
64 bytes from 10.0.53.1: icmp_seq=3 ttl=99 time=1.59 ms
64 bytes from 10.0.53.1: icmp_seq=4 ttl=99 time=1.56 ms
64 bytes from 10.0.53.1: icmp_seq=5 ttl=99 time=1.15 ms
64 bytes from 10.0.53.1: icmp_seq=6 ttl=99 time=1.53 ms
```

```
root@e2eaf25532bf:/volumes# tun2.py
Interface Name: tun0
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.53.1 echo-request 0 / Raw
```

จะเห็นว่า ping มี echo reply ตอบกลับ

- เมื่อได้ packet ที่ tun0 ให้ส่ง data อะไรก็ได้ไปยัง tun0

```
25
26 while True:
27     # Get a packet from the tun interface
28     packet = os.read(tun, 2048)
29     if packet:
30         os.write(tun, bytes("testtest", encoding='utf8'))
```

```
root@e2eaf25532bf:/volumes# tun.py
Interface Name: tun0
Traceback (most recent call last):
  File "./tun.py", line 30, in <module>
    os.write(tun, bytes("testtest", encoding='utf8'))
OSError: [Errno 22] Invalid argument
root@e2eaf25532bf:/volumes#
root@e2eaf25532bf:/volumes# ping 10.0.53.1
PING 10.0.53.1 (10.0.53.1) 56(84) bytes of data.
```

หลุดออกจาก tun.py เพราะเกิด error

Task 3: Send the IP Packet to VPN Server Through a Tunnel

- ทำการรับ IP packet ที่เข้า tun0 มาเป็น payload ของ UDP packet เป็นการทำ IP tunneling

ฝั่ง Server (router)

Open ▾ | tun_server.py ~/Documents/Lab9-VPN/volumes | Sav

```
1 #!/usr/bin/env python3
2
3 from scapy.all import *
4
5 IP_A = "0.0.0.0"
6 PORT = 9090
7
8 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
9 sock.bind((IP_A, PORT))
10
11 while True:
12     data, (ip, port) = sock.recvfrom(2048)
13     print("{}:{} --> {}:{}".format(ip, port, IP_A, PORT))
14     pkt = IP(data)
15     print("   Inside: {} --> {}".format(pkt.src, pkt.dst))
```

```
root@43bb387c1caa:/volumes# tun_server.py
```

ฝั่ง Client (host U)

Open | tun_client.py ~/Documents/Lab9-VPN/volumes | Save

tun_server.py | tun_client.py

```python
#!/usr/bin/env python3

import fcntl
import struct
import os
import time
from scapy.all import *

TUNSETIFF = 0x400454ca
IFF_TUN   = 0x0001
IFF_TAP   = 0x0002
IFF_NO_PI = 0x1000

# Create the tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'tun%d', IFF_TUN | IFF_NO_PI)
ifname_bytes  = fcntl.ioctl(tun, TUNSETIFF, ifr)

# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))

# Configure the interface
os.system("ip addr add 10.0.53.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

# Set up routing
os.system("ip route add 10.0.8.0/24 dev {}".format(ifname))

# Create UDP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

while True:
   # Get a packet from the tun interface
   packet = os.read(tun, 2048)
   if packet:
      pkt = IP(packet)
      print(pkt.summary())

      # Send the packet via the tunnel
      sock.sendto(packet, ("10.0.7.11", 9090))

```

```
root@e2eaf25532bf:/volumes# tun_client.py
Interface Name: tun0
```

Testing: ทดลอง ping จาก host U (10.0.7.5) ไปยัง host V (10.0.8.0/24): จะเห็นว่ามี packet เข้า tun0 แต่ยังไม่มีการตอบรับจากปลายทาง เนื่องจาก tunnel ยังไม่อนุญาต forwarding

```
root@43bb387c1caa:/volumes# ls
tap     tun2.py        tun_client_select.py
tun.py  tun_client.py  tun_read.py
root@43bb387c1caa:/volumes# tun_server.py
10.0.7.5:38806 --> 0.0.0.0:9090
   Inside: 10.0.53.99 --> 10.0.8.5
10.0.7.5:38806 --> 0.0.0.0:9090
   Inside: 10.0.53.99 --> 10.0.8.5
10.0.7.5:38806 --> 0.0.0.0:9090
   Inside: 10.0.53.99 --> 10.0.8.5
10.0.7.5:38806 --> 0.0.0.0:9090
   Inside: 10.0.53.99 --> 10.0.8.5
10.0.7.5:38806 --> 0.0.0.0:9090
   Inside: 10.0.53.99 --> 10.0.8.5
```

```
root@e2eaf25532bf:/volumes# tun_client.py
Interface Name: tun0
IP / ICMP 10.0.53.99 > 10.0.8.5 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.8.5 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.8.5 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.8.5 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.8.5 echo-request 0 / Raw

root@e2eaf25532bf:/volumes# ping 10.0.8.5
PING 10.0.8.5 (10.0.8.5) 56(84) bytes of data.
^C
--- 10.0.8.5 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4096ms

root@e2eaf25532bf:/volumes#
```

Task 4: Set Up the VPN Server

- ปรับโค้ด tun_server.py ให้มีคุณสมบัติดังต่อไปนี้

    - สร้าง tun0, assign IP, up interface
    - ดักจับ packet ที่เข้ามา tun0 ได้
    - เขียน packet ส่งกลับผ่าน tun0 ได้

Open | tun_server2.py ~/Documents/Lab9-VPN/volumes | Save

tun_server.py | tun_client.py | tun_server2.py

```
#!/usr/bin/python3

import fcntl
import struct
import os
from scapy.all import *

IP_A = "0.0.0.0"
PORT = 9090

TUNSETIFF = 0x400454ca
IFF_TUN   = 0x0001
IFF_TAP   = 0x0002
IFF_NO_PI = 0x1000

# Create a tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'tun%d', IFF_TUN | IFF_NO_PI)
ifname_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)
ifname  = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))

# Set up the tun interface
os.system("ip addr add 10.0.53.1/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((IP_A, PORT))

while True:
    data, (ip, port) = sock.recvfrom(2048)
    pkt = IP(data)
    print("{}:{} --> {}:{}".format(ip, port, IP_A, PORT))
    print("   Inside: {} --> {}".format(pkt.src, pkt.dst))
    os.write(tun, data)
```

ก่อนรันโค้ดต้องทำการเปิด ipv4.ip_forward ที่ server ก่อน

```
root@43bb387c1caa:/volumes# sysctl -a |grep ipv4.ip_forward
net.ipv4.ip_forward = 1
net.ipv4.ip_forward_update_priority = 1
net.ipv4.ip_forward_use_pmtu = 0
root@43bb387c1caa:/volumes#
```

```
root@43bb387c1caa:/volumes# tun_server2.py
Interface Name: tun0
10.0.7.5:46588 --> 0.0.0.0:9090
   Inside: 10.0.53.99 --> 10.0.8.5
10.0.7.5:46588 --> 0.0.0.0:9090
   Inside: 10.0.53.99 --> 10.0.8.5
10.0.7.5:46588 --> 0.0.0.0:9090
   Inside: 10.0.53.99 --> 10.0.8.5
```

```
root@88c230d86669:/# tcpdump -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
16:04:40.021328 IP 10.0.53.99 > 88c230d86669: ICMP echo request, id 163, seq 1, length 64
16:04:40.022218 IP 88c230d86669 > 10.0.53.99: ICMP echo reply, id 163, seq 1, length 64
16:04:40.027131 IP 88c230d86669.44792 > 10.0.2.3.domain: 46533+ PTR? 99.53.0.10.in-addr.arpa. (4
```

```
    packet = os.read(tun, 2048)
KeyboardInterrupt

root@e2eaf25532bf:/volumes# tun_client.py
Interface Name: tun0
IP / ICMP 10.0.53.99 > 10.0.8.5 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.8.5 echo-request 0 / Raw
IP / ICMP 10.0.53.99 > 10.0.8.5 echo-request 0 / Raw
```

```
root@e2eaf25532bf:/volumes# ping 10.0.8.5
PING 10.0.8.5 (10.0.8.5) 56(84) bytes of data.
^C
--- 10.0.8.5 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2055ms

root@e2eaf25532bf:/volumes#
```

เมื่อทดลอง ping จะเห็นว่าฝั่ง host V (terminal สีม่วง) ได้รับ packet ICMP echo request มีการตอบ ICMP echo reply กลับไปยัง tun0 (10.0.53.99) แต่ไปไม่ถึงต้นทางที่ ping เนื่องจาก tunnel ยังเป็น one direction

Task 5: Handling Traffic in Both Directions

- ปรับโค้ดให้สามารถติดต่อกันได้ทั้งสองทาง

tun_server_select.py

```python
#!/usr/bin/python3

#import select
import fcntl
import struct
import os
from scapy.all import *

IP_A = "0.0.0.0"
PORT = 9090

TUNSETIFF = 0x400454ca
IFF_TUN   = 0x0001
IFF_TAP   = 0x0002
IFF_NO_PI = 0x1000

# Create a tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'tun%d', IFF_TUN | IFF_NO_PI)
ifname_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)
ifname  = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))

# Set up the tun interface and routing
os.system("ip addr add 10.0.53.1/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((IP_A, PORT))

# We need to initialize ip and port (their values do not matter)
ip   = '0.0.0.0'
port = 10000

fds = [sock, tun]
while True:
  # this will block until at least one socket is ready
  ready, _, _ = select.select(fds, [], [])

  for fd in ready:
    if fd is sock:
       data, (ip, port) = sock.recvfrom(2048)
       pkt = IP(data)
       print("From socket <==: {} --> {}".format(pkt.src, pkt.dst))
       os.write(tun, data)

    if fd is tun:
       packet = os.read(tun, 2048)
       pkt = IP(packet)
       print("From tun    ==>: {} --> {}".format(pkt.src, pkt.dst))
       sock.sendto(packet, (ip, port))
```

tun_client_select.py

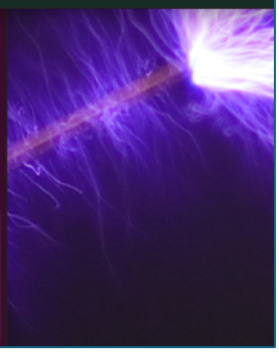
```python
#!/usr/bin/python3

import fcntl
import struct
import os
from scapy.all import *

TUNSETIFF = 0x400454ca
IFF_TUN   = 0x0001
IFF_TAP   = 0x0002
IFF_NO_PI = 0x1000

# Create a tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'tun%d', IFF_TUN | IFF_NO_PI)
ifname_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)
ifname  = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))

# Set up the tun interface and routing
os.system("ip addr add 10.0.53.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

# Set up routing
os.system("ip route add 10.0.8.0/24 dev {}".format(ifname))

# Create UDP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

fds = [sock, tun]
while True:
  # this will block until at least one socket is ready
  ready, _, _ = select.select(fds, [], [])

  for fd in ready:
    if fd is sock:
       data, (ip, port) = sock.recvfrom(2048)
       pkt = IP(data)
       print("From socket <==: {} --> {}".format(pkt.src, pkt.dst))
       os.write(tun, data)

    if fd is tun:
       packet = os.read(tun, 2048)
       pkt = IP(packet)
       print("From tun    ==>: {} --> {}".format(pkt.src, pkt.dst))
       sock.sendto(packet, ('10.0.7.11', 9090))
```

Testing: ทดลอง ping จาก host V ไป host U

```
root@43bb387c1caa:/volumes# tun_server_select.py
Interface Name: tun0
From socket <==: 10.0.7.5 --> 10.0.8.5
From socket <==: 10.0.7.5 --> 10.0.8.5
From socket <==: 10.0.7.5 --> 10.0.8.5
From socket <==: 10.0.7.5 --> 10.0.8.5
From socket <==: 10.0.7.5 --> 10.0.8.5
```

```
root@e2eaf25532bf:/volumes# tun_client_select.py
Interface Name: tun0
From tun    ==>: 10.0.7.5 --> 10.0.8.5
From tun    ==>: 10.0.7.5 --> 10.0.8.5
From tun    ==>: 10.0.7.5 --> 10.0.8.5
From tun    ==>: 10.0.7.5 --> 10.0.8.5
From tun    ==>: 10.0.7.5 --> 10.0.8.5
```

```
root@88c230d86669:/# ping 10.0.7.5
PING 10.0.7.5 (10.0.7.5) 56(84) bytes of data.
64 bytes from 10.0.7.5: icmp_seq=1 ttl=63 time=3.11 ms
64 bytes from 10.0.7.5: icmp_seq=2 ttl=63 time=1.71 ms
64 bytes from 10.0.7.5: icmp_seq=3 ttl=63 time=1.40 ms
64 bytes from 10.0.7.5: icmp_seq=4 ttl=63 time=1.49 ms
64 bytes from 10.0.7.5: icmp_seq=5 ttl=63 time=1.95 ms
^C
--- 10.0.7.5 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 1.397/1.930/3.107/0.618 ms
root@88c230d86669:/#
```



No.	Time	Source	Destination	Protocol	Length	Info
1	2025-04-16 01:1…	10.0.8.5	10.0.7.5	ICMP	98	Echo (ping) request id=0x0022, seq=1/256, ttl=64 (reply in 2)
2	2025-04-16 01:1…	10.0.7.5	10.0.8.5	ICMP	98	Echo (ping) reply id=0x0022, seq=1/256, ttl=63 (request in 1)
3	2025-04-16 01:1…	10.0.8.5	10.0.7.5	ICMP	98	Echo (ping) request id=0x0022, seq=2/512, ttl=64 (reply in 4)
4	2025-04-16 01:1…	10.0.7.5	10.0.8.5	ICMP	98	Echo (ping) reply id=0x0022, seq=2/512, ttl=63 (request in 3)
5	2025-04-16 01:1…	10.0.8.5	10.0.7.5	ICMP	98	Echo (ping) request id=0x0022, seq=3/768, ttl=64 (reply in 6)
6	2025-04-16 01:1…	10.0.7.5	10.0.8.5	ICMP	98	Echo (ping) reply id=0x0022, seq=3/768, ttl=63 (request in 5)
7	2025-04-16 01:1…	10.0.8.5	10.0.7.5	ICMP	98	Echo (ping) request id=0x0022, seq=4/1024, ttl=64 (reply in 8)
8	2025-04-16 01:1…	10.0.7.5	10.0.8.5	ICMP	98	Echo (ping) reply id=0x0022, seq=4/1024, ttl=63 (request in 7)
9	2025-04-16 01:1…	10.0.8.5	10.0.7.5	ICMP	98	Echo (ping) request id=0x0022, seq=5/1280, ttl=64 (reply in 10)
10	2025-04-16 01:1…	10.0.7.5	10.0.8.5	ICMP	98	Echo (ping) reply id=0x0022, seq=5/1280, ttl=63 (request in 9)

- ทดลอง telnet จาก host U ไป host V

```
root@43bb387c1caa:/volumes# tun_server_select.py
Interface Name: tun0
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.53.99 --> 10.0.8.5
```

```
root@e2eaf25532bf:/volumes# tun_client_select.py
Interface Name: tun0
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.53.99 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.53.99
From socket <==: 10.0.8.5 --> 10.0.53.99
From tun    ==>: 10.0.53.99 --> 10.0.8.5
```

```
root@e2eaf25532bf:/volumes# telnet 10.0.8.5
Trying 10.0.8.5...
Connected to 10.0.8.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
88c230d86669 login:
```

No.	Time	Source	Destination	Protocol	Length	Info
31	2025-04-16 01:1…	02:42:0a:00:08:05	02:42:0a:00:08:0b	ARP	42	10.0.8.5 is at 02:42:0a:00:08:05
32	2025-04-16 01:1…	10.0.8.5	10.0.53.99	TELNET	78	Telnet Data ...
33	2025-04-16 01:1…	10.0.53.99	10.0.8.5	TCP	66	43908 → 23 [ACK] Seq=849814445 Ack=1642750666 Win=64256 Len=0 TSv…
34	2025-04-16 01:1…	10.0.8.5	10.0.53.99	TELNET	81	Telnet Data ...
35	2025-04-16 01:1…	10.0.53.99	10.0.8.5	TELNET	69	Telnet Data ...
36	2025-04-16 01:1…	10.0.8.5	10.0.53.99	TCP	66	23 → 43908 [ACK] Seq=1642750681 Ack=849814448 Win=65152 Len=0 TSv…
37	2025-04-16 01:1…	10.0.53.99	10.0.8.5	TCP	66	43908 → 23 [ACK] Seq=849814448 Ack=1642750681 Win=64256 Len=0 TSv…
38	2025-04-16 01:1…	10.0.8.5	10.0.53.99	TELNET	84	Telnet Data ...
39	2025-04-16 01:1…	10.0.53.99	10.0.8.5	TELNET	75	Telnet Data ...
40	2025-04-16 01:1…	10.0.8.5	10.0.53.99	TCP	66	23 → 43908 [ACK] Seq=1642750699 Ack=849814457 Win=65152 Len=0 TSv…
41	2025-04-16 01:1…	10.0.53.99	10.0.8.5	TCP	66	43908 → 23 [ACK] Seq=849814457 Ack=1642750699 Win=64256 Len=0 TSv…
42	2025-04-16 01:1…	10.0.53.99	10.0.8.5	TELNET	100	Telnet Data ...
43	2025-04-16 01:1…	10.0.8.5	10.0.53.99	TCP	66	23 → 43908 [ACK] Seq=1642750699 Ack=849814491 Win=65152 Len=0 TSv…

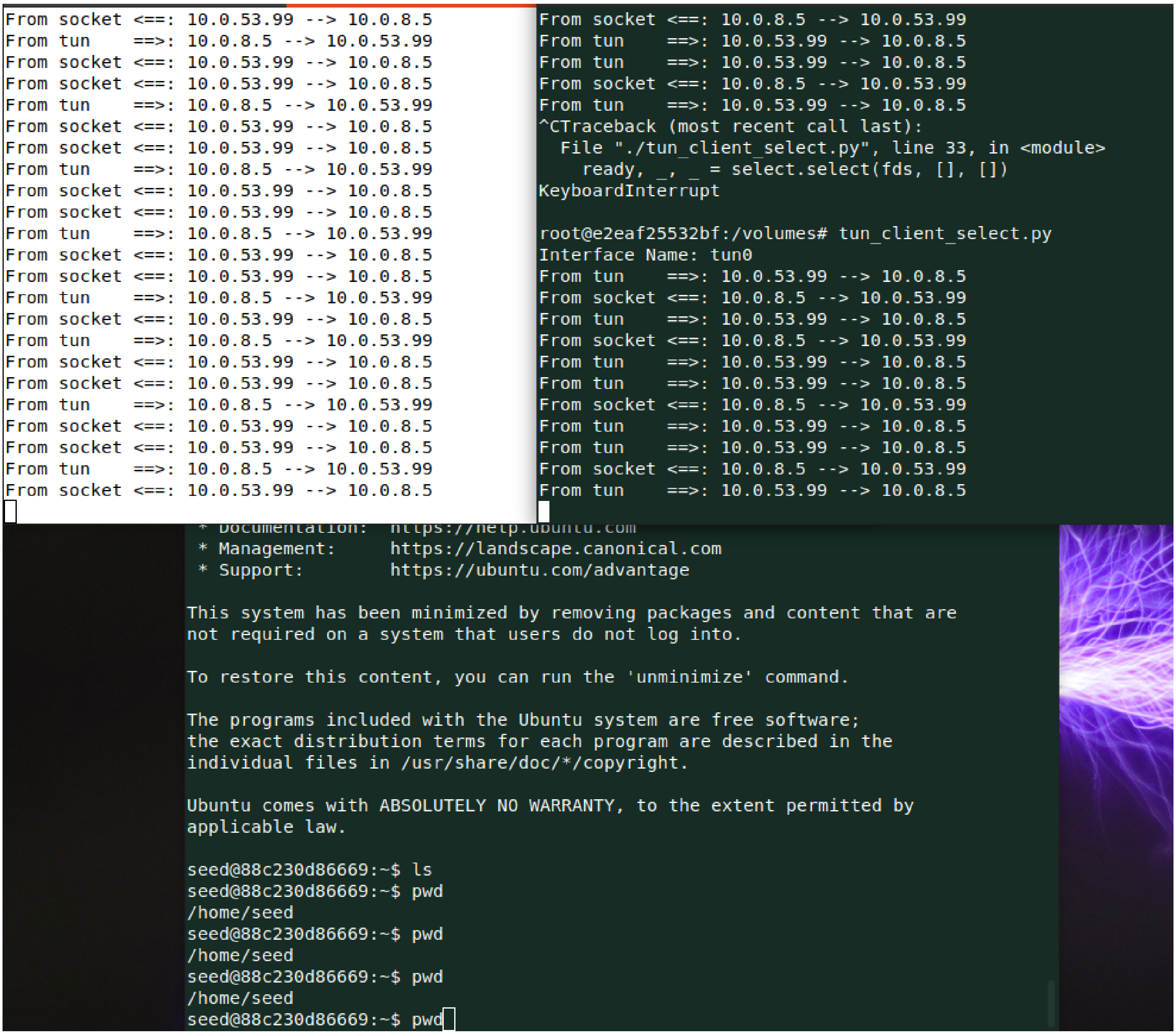
Task 6: Tunnel-Breaking Experiment

- ทดลองหยุด tun server หรือ tun client ฝั่งใดฝั่งหนึ่ง แล้วเกิดอะไรขึ้น

    - host U ที่ telnet host V อยู่จะไม่สามารถพิมพ์ข้อความได้

- เมื่อเชื่อม connection ใหม่อีกครั้ง เกิดอะไรขึ้น

    - หากเชื่อม connection ทันเวลา telnet จะยังไม่หลุด และสามารถพิมพ์ข้อความได้ปกติ



Task 7: Routing Experiment on Host V

- ลบ default route บน host V เพิ่ม route ให้ผ่าน router (VPN server) แน่นอน

- ก่อนลบ default route

```
root@88c230d86669:/# ip route
default via 10.0.8.11 dev eth0
10.0.8.0/24 dev eth0 proto kernel scope link src 10.0.8.5
root@88c230d86669:/#
```

```
root@43bb387c1caa:/volumes# tun_server_select.py
Interface Name: tun0
From socket <==: 10.0.7.5 --> 10.0.8.5
From socket <==: 10.0.7.5 --> 10.0.8.5
From socket <==: 10.0.7.5 --> 10.0.8.5
From socket <==: 10.0.7.5 --> 10.0.8.5
```

```
root@e2eaf25532bf:/volumes# tun_client_select.py
Interface Name: tun0
From tun    ==>: 10.0.7.5 --> 10.0.8.5
From tun    ==>: 10.0.7.5 --> 10.0.8.5
From tun    ==>: 10.0.7.5 --> 10.0.8.5
From tun    ==>: 10.0.7.5 --> 10.0.8.5
```

```
root@88c230d86669:/# ping 10.0.7.5
PING 10.0.7.5 (10.0.7.5) 56(84) bytes of data.
64 bytes from 10.0.7.5: icmp_seq=1 ttl=63 time=1.63 ms
64 bytes from 10.0.7.5: icmp_seq=2 ttl=63 time=2.10 ms
64 bytes from 10.0.7.5: icmp_seq=3 ttl=63 time=1.87 ms
64 bytes from 10.0.7.5: icmp_seq=4 ttl=63 time=1.37 ms
^C
--- 10.0.7.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3032ms
rtt min/avg/max/mdev = 1.373/1.744/2.101/0.271 ms
root@88c230d86669:/#
```

- หลังลบ default route

```
root@88c230d86669:/# ip route del default
root@88c230d86669:/# ip route
10.0.8.0/24 dev eth0 proto kernel scope link src 10.0.8.5
root@88c230d86669:/# ip route add 10.0.7.0/24 via 10.0.8.11
root@88c230d86669:/# ip route
10.0.7.0/24 via 10.0.8.11 dev eth0
10.0.8.0/24 dev eth0 proto kernel scope link src 10.0.8.5
root@88c230d86669:/#
```

```
root@43bb387c1caa:/volumes# tun_server_select.py
Interface Name: tun0
From socket <==: 10.0.7.5 --> 10.0.8.5
From socket <==: 10.0.7.5 --> 10.0.8.5
From socket <==: 10.0.7.5 --> 10.0.8.5
From socket <==: 10.0.7.5 --> 10.0.8.5
```

```
root@e2eaf25532bf:/volumes# tun_client_select.py
Interface Name: tun0
From tun    ==>: 10.0.7.5 --> 10.0.8.5
From tun    ==>: 10.0.7.5 --> 10.0.8.5
From tun    ==>: 10.0.7.5 --> 10.0.8.5
From tun    ==>: 10.0.7.5 --> 10.0.8.5
```

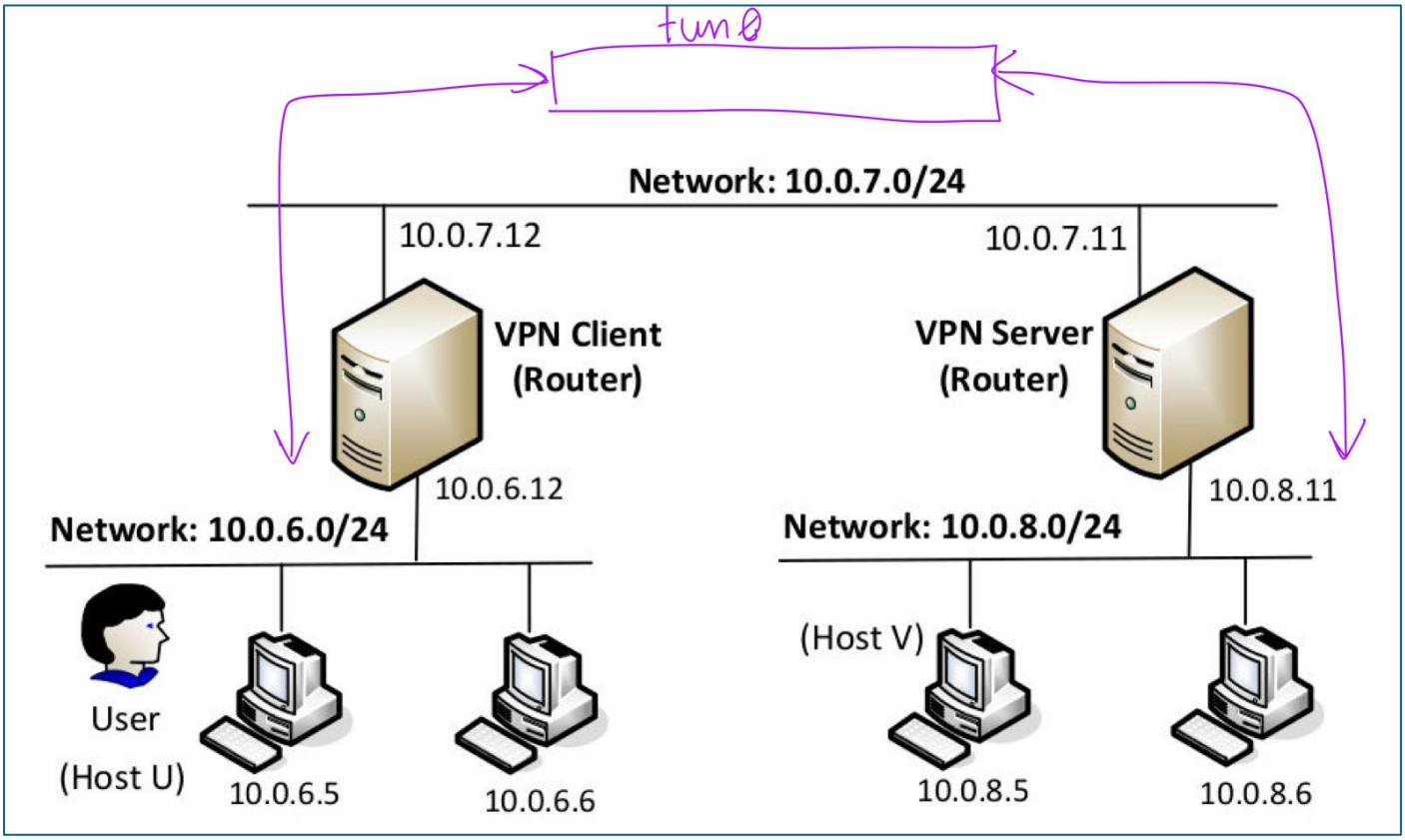
```
root@88c230d86669:/# ping 10.0.7.5
PING 10.0.7.5 (10.0.7.5) 56(84) bytes of data.
64 bytes from 10.0.7.5: icmp_seq=1 ttl=63 time=1.76 ms
64 bytes from 10.0.7.5: icmp_seq=2 ttl=63 time=2.33 ms
64 bytes from 10.0.7.5: icmp_seq=3 ttl=63 time=3.71 ms
64 bytes from 10.0.7.5: icmp_seq=4 ttl=63 time=2.51 ms
^C
--- 10.0.7.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3011ms
rtt min/avg/max/mdev = 1.762/2.577/3.712/0.710 ms
root@88c230d86669:/#
```

Task 8: VPN Between Private Networks

- เปลี่ยน container เป็น docker-compose2.yml

```
[04/16/25]seed@VM:~/.../Lab9-VPN$ docker-compose -f docker-compose2.yml up
Creating network "net-10.0.6.0" with the default driver
Creating network "net-10.0.7.0" with the default driver
Creating network "net-10.0.8.0" with the default driver
Creating host-10.0.8.5           ... done
Creating host-10.0.6.6           ... done
Creating client-10.0.7.12        ... done
Creating host-10.0.8.6           ... done
Creating server-router-10.0.7.11 ... done
Creating host-10.0.6.5           ... done
Attaching to host-10.0.6.6, host-10.0.8.6, server-router-10.0.7.11, host-10.0.6.
5, host-10.0.8.5, client-10.0.7.12
host-10.0.6.5 |  * Starting internet superserver inetd                 [ OK ]
host-10.0.6.6 |  * Starting internet superserver inetd                 [ OK ]
host-10.0.8.6 |  * Starting internet superserver inetd                 [ OK ]
host-10.0.8.5 |  * Starting internet superserver inetd                 [ OK ]
```

```
[04/16/25]seed@VM:~/.../Lab9-VPN$ dockps
8cb71cd9daf0  host-10.0.8.6
9b6d27499474  host-10.0.6.5
a5d2bbe68b2c  server-router-10.0.7.11
626d90b62665  host-10.0.6.6
7956441634cc  client-10.0.7.12
adb3648bdbf7  host-10.0.8.5
[04/16/25]seed@VM:~/.../Lab9-VPN$
```



Testing: ทดสอบ ping แล้วตรวจสอบว่า packet วิ่งผ่าน tunnel หรือไม่

- ปรับแก้ ip บนโค้ด tun_server_select (+เติมบรรทัดที่ 29) และ tun_client_select

tun_server_select รันที่ host: server-router-10.0.7.11

```
23
24 # Set up the tun interface and routing
25 os.system("ip addr add 10.0.8.11/24 dev {}".format(ifname))
26 os.system("ip link set dev {} up".format(ifname))
27
28 # Set up routing
29 os.system("ip route add 10.0.0.0/16 dev {} via 10.0.8.11".format(ifname))
30
```

```
root@a5d2bbe68b2c:/volumes# tun_server_select2.py
Interface Name: tun0
```

tun_client_select รันที่ host: client-10.0.7.12

```
19
20 # Set up the tun interface and routing
21 os.system("ip addr add 10.0.6.12/24 dev {}".format(ifname))
22 os.system("ip link set dev {} up".format(ifname))
23
24 # Set up routing
25 os.system("ip route add 10.0.0.0/16 dev {} via 10.0.6.12".format(ifname))
26
```

```
root@7956441634cc:/volumes# tun_client_select2.py
Interface Name: tun0
```

host U แก้ไข ip route

```
root@9b6d27499474:/# ip route
default via 10.0.6.12 dev eth0
10.0.6.0/24 dev eth0 proto kernel scope link src 10.0.6.5
root@9b6d27499474:/# ip route del default
root@9b6d27499474:/# ip route add 10.0.0.0/24 via 10.0.6.12
root@9b6d27499474:/# ip route
10.0.0.0/24 via 10.0.6.12 dev eth0
10.0.6.0/24 dev eth0 proto kernel scope link src 10.0.6.5
root@9b6d27499474:/#
root@9b6d27499474:/# ip route add 10.0.0.0/16 via 10.0.6.12
root@9b6d27499474:/# ip route
10.0.0.0/24 via 10.0.6.12 dev eth0
10.0.0.0/16 via 10.0.6.12 dev eth0
10.0.6.0/24 dev eth0 proto kernel scope link src 10.0.6.5
root@9b6d27499474:/#
```

host V แก้ไข ip route

```
root@adb3648bdbf7:/# ip route
default via 10.0.8.11 dev eth0
10.0.8.0/24 dev eth0 proto kernel scope link src 10.0.8.5
root@adb3648bdbf7:/# ip route del default
root@adb3648bdbf7:/# ip route add 10.0.0.0/24 via 10.0.8.11
root@adb3648bdbf7:/# ip route
10.0.0.0/24 via 10.0.8.11 dev eth0
10.0.8.0/24 dev eth0 proto kernel scope link src 10.0.8.5
root@adb3648bdbf7:/#
root@adb3648bdbf7:/# ip route add 10.0.0.0/16 via 10.0.8.11
root@adb3648bdbf7:/# ip route
10.0.0.0/24 via 10.0.8.11 dev eth0
10.0.0.0/16 via 10.0.8.11 dev eth0
10.0.8.0/24 dev eth0 proto kernel scope link src 10.0.8.5
root@adb3648bdbf7:/#
```

ทดสอบ ping host U -> host V: ที่ Wireshark จะเห็นเป็น packet UDP

```
root@7956441634cc:/volumes# tun_client_select2.py
Interface Name: tun0
From tun    ==>: 10.0.6.5 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.6.5
From tun    ==>: 10.0.6.5 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.6.5
From tun    ==>: 10.0.6.5 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.6.5
From tun    ==>: 10.0.6.5 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.6.5
```

```
root@a5d2bbe68b2c:/volumes# tun_server_select2.py
Interface Name: tun0
From socket <==: 10.0.6.5 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.6.5
From socket <==: 10.0.6.5 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.6.5
From socket <==: 10.0.6.5 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.6.5
From socket <==: 10.0.6.5 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.6.5
```

```
root@9b6d27499474:/# ping 10.0.8.5
PING 10.0.8.5 (10.0.8.5) 56(84) bytes of data.
64 bytes from 10.0.8.5: icmp_seq=1 ttl=62 time=4.91 ms
64 bytes from 10.0.8.5: icmp_seq=2 ttl=62 time=2.43 ms
64 bytes from 10.0.8.5: icmp_seq=3 ttl=62 time=2.38 ms
64 bytes from 10.0.8.5: icmp_seq=4 ttl=62 time=1.63 ms
^C
--- 10.0.8.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3014ms
rtt min/avg/max/mdev = 1.629/2.838/4.911/1.238 ms
root@9b6d27499474:/#
```

No.	Time	Source	Destination	Protocol	Length	Info
1	2025-04-16 03:16:22.57…	10.0.7.12	10.0.7.11	UDP	126	44614 → 9090 Len=84
2	2025-04-16 03:16:22.57…	10.0.7.11	10.0.7.12	UDP	126	9090 → 44614 Len=84
3	2025-04-16 03:16:23.58…	10.0.7.12	10.0.7.11	UDP	126	44614 → 9090 Len=84
4	2025-04-16 03:16:23.58…	10.0.7.11	10.0.7.12	UDP	126	9090 → 44614 Len=84
5	2025-04-16 03:16:24.58…	10.0.7.12	10.0.7.11	UDP	126	44614 → 9090 Len=84
6	2025-04-16 03:16:24.58…	10.0.7.11	10.0.7.12	UDP	126	9090 → 44614 Len=84
7	2025-04-16 03:16:25.58…	10.0.7.12	10.0.7.11	UDP	126	44614 → 9090 Len=84
8	2025-04-16 03:16:25.58…	10.0.7.11	10.0.7.12	UDP	126	9090 → 44614 Len=84

ทดสอบ ping host V -> host U

```
root@7956441634cc:/volumes# tun_client_select2.py
Interface Name: tun0
From tun    ==>: 10.0.6.5 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.6.5
From tun    ==>: 10.0.6.5 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.6.5
From tun    ==>: 10.0.6.5 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.6.5
From tun    ==>: 10.0.6.5 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.6.5
From socket <==: 10.0.8.5 --> 10.0.6.5
From tun    ==>: 10.0.6.5 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.6.5
From tun    ==>: 10.0.6.5 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.6.5
From tun    ==>: 10.0.6.5 --> 10.0.8.5
From socket <==: 10.0.8.5 --> 10.0.6.5
From tun    ==>: 10.0.6.5 --> 10.0.8.5
```

```
root@a5d2bbe68b2c:/volumes# tun_server_select2.py
Interface Name: tun0
From socket <==: 10.0.6.5 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.6.5
From socket <==: 10.0.6.5 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.6.5
From socket <==: 10.0.6.5 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.6.5
From socket <==: 10.0.6.5 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.6.5
From tun    ==>: 10.0.8.5 --> 10.0.6.5
From socket <==: 10.0.6.5 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.6.5
From socket <==: 10.0.6.5 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.6.5
From socket <==: 10.0.6.5 --> 10.0.8.5
From tun    ==>: 10.0.8.5 --> 10.0.6.5
From socket <==: 10.0.6.5 --> 10.0.8.5
```

```
root@9b6d27499474:/# ping 10.0.8.5
PING 10.0.8.5 (10.0.8.5) 56(84) bytes of data.
64 bytes from 10.0.8.5: icmp_seq=1 ttl=62 time=4.91 ms
64 bytes from 10.0.8.5: icmp_seq=2 ttl=62 time=2.43 ms
64 bytes from 10.0.8.5: icmp_seq=3 ttl=62 time=2.38 ms
64 bytes from 10.0.8.5: icmp_seq=4 ttl=62 time=1.63 ms
^C
--- 10.0.8.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3014ms
rtt min/avg/max/mdev = 1.629/2.838/4.911/1.238 ms
root@9b6d27499474:/#
```

```
root@adb3648bdbf7:/# ping 10.0.6.5
PING 10.0.6.5 (10.0.6.5) 56(84) bytes of data.
64 bytes from 10.0.6.5: icmp_seq=1 ttl=62 time=2.45 ms
64 bytes from 10.0.6.5: icmp_seq=2 ttl=62 time=2.52 ms
64 bytes from 10.0.6.5: icmp_seq=3 ttl=62 time=2.62 ms
64 bytes from 10.0.6.5: icmp_seq=4 ttl=62 time=2.22 ms
^C
--- 10.0.6.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3008ms
rtt min/avg/max/mdev = 2.224/2.454/2.619/0.145 ms
root@adb3648bdbf7:/#
```

10	2025-04-16 03:16:27.60…	02:42:0a:00:07:0c	02:42:0a:00:07:0b	ARP	42	Who has 10.0.7.11? Tell 10.0.7.12
11	2025-04-16 03:16:27.60…	02:42:0a:00:07:0c	02:42:0a:00:07:0b	ARP	42	10.0.7.12 is at 02:42:0a:00:07:0c
12	2025-04-16 03:16:27.60…	02:42:0a:00:07:0b	02:42:0a:00:07:0c	ARP	42	10.0.7.11 is at 02:42:0a:00:07:0b
13	2025-04-16 03:18:56.81…	10.0.7.11	10.0.7.12	UDP	126	9090 → 44614 Len=84
14	2025-04-16 03:18:56.81…	10.0.7.12	10.0.7.11	UDP	126	44614 → 9090 Len=84
15	2025-04-16 03:18:57.81…	10.0.7.11	10.0.7.12	UDP	126	9090 → 44614 Len=84
16	2025-04-16 03:18:57.81…	10.0.7.12	10.0.7.11	UDP	126	44614 → 9090 Len=84
17	2025-04-16 03:18:58.81…	10.0.7.11	10.0.7.12	UDP	126	9090 → 44614 Len=84
18	2025-04-16 03:18:58.81…	10.0.7.12	10.0.7.11	UDP	126	44614 → 9090 Len=84
19	2025-04-16 03:18:59.82…	10.0.7.11	10.0.7.12	UDP	126	9090 → 44614 Len=84
20	2025-04-16 03:18:59.82…	10.0.7.12	10.0.7.11	UDP	126	44614 → 9090 Len=84
21	2025-04-16 03:19:01.97…	02:42:0a:00:07:0b	02:42:0a:00:07:0c	ARP	42	Who has 10.0.7.12? Tell 10.0.7.11
22	2025-04-16 03:19:01.97…	02:42:0a:00:07:0c	02:42:0a:00:07:0b	ARP	42	Who has 10.0.7.11? Tell 10.0.7.12
23	2025-04-16 03:19:01.97…	02:42:0a:00:07:0c	02:42:0a:00:07:0b	ARP	42	10.0.7.12 is at 02:42:0a:00:07:0c
24	2025-04-16 03:19:01.97…	02:42:0a:00:07:0b	02:42:0a:00:07:0c	ARP	42	10.0.7.11 is at 02:42:0a:00:07:0b

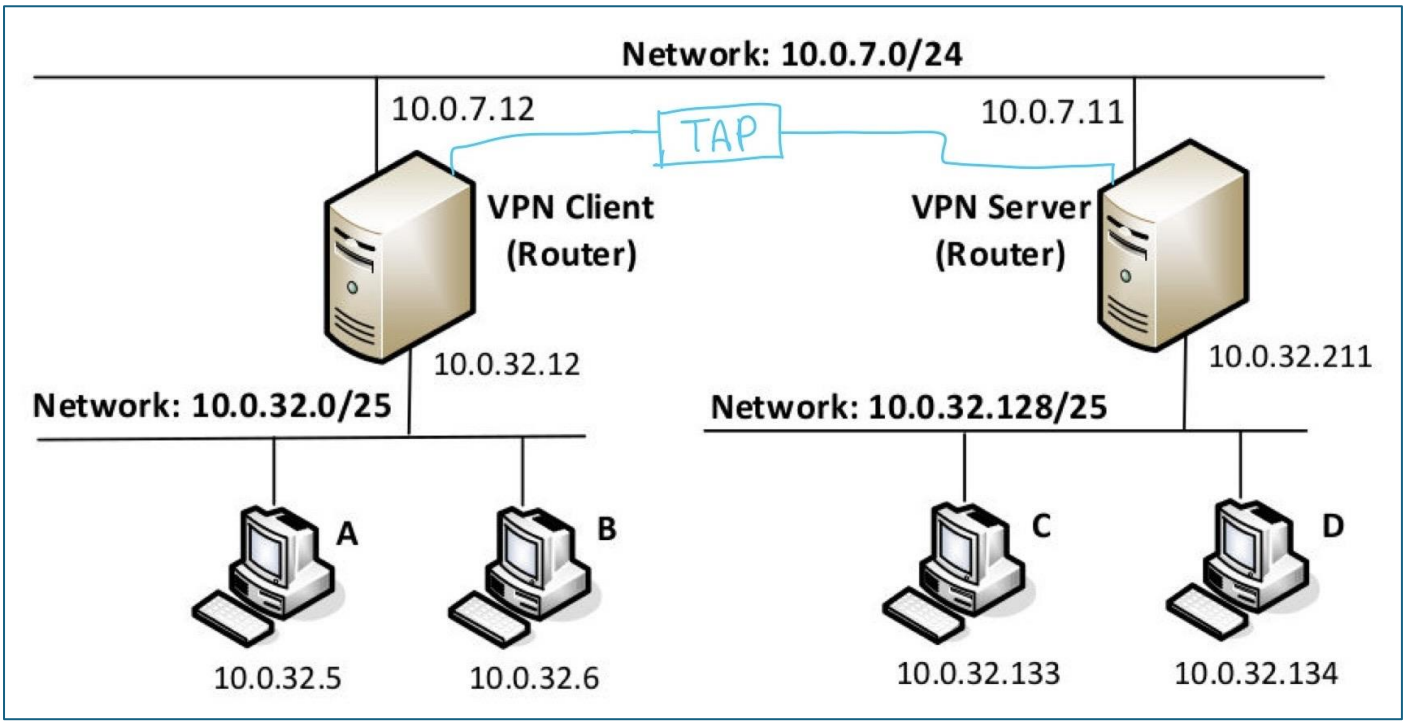
Task 9: Experiment with the TAP Interface

- เปลี่ยน container เป็น docker-compose3.yml

- ทดลองใช้ TAN interface (MAC layer) แทน TUN interface (IP layer)

```
[04/16/25]seed@VM:~/.../Lab9-VPN$ docker-compose -f docker-compose3.yml up
Creating network "net-private-1" with the default driver
Creating network "net-10.0.7.0" with the default driver
Creating network "net-private-2" with the default driver
Creating vpn-server-10.0.7.11 ... done
Creating host-D-10.0.32.134   ... done
Creating host-B-10.0.32.6     ... done
Creating vpn-client-10.0.7.12 ... done
Creating host-A-10.0.32.5     ... done
Creating host-C-10.0.32.133   ... done
Attaching to host-D-10.0.32.134, host-B-10.0.32.6, vpn-server-10.0.7.11, host-A-
10.0.32.5, host-C-10.0.32.133, vpn-client-10.0.7.12
host-A-10.0.32.5 |  * Starting internet superserver inetd                 [ OK ]
host-B-10.0.32.6 |  * Starting internet superserver inetd                 [ OK ]
host-C-10.0.32.133 |  * Starting internet superserver inetd               [ OK ]
host-D-10.0.32.134 |  * Starting internet superserver inetd               [ OK ]
```

```
[04/16/25]seed@VM:~/.../Lab9-VPN$ dockps
0c99584f45fb  host-A-10.0.32.5
c04cc3d0574c  host-C-10.0.32.133
5ec1931fd7ad  vpn-client-10.0.7.12
5282271fdd4f  host-B-10.0.32.6
7caf08e04319  host-D-10.0.32.134
5f737e2b8492  vpn-server-10.0.7.11
[04/16/25]seed@VM:~/.../Lab9-VPN$
```



volumes/tap/tap_server.py รันที่ vpn-server-10.0.7.11

```
root@5f737e2b8492:/volumes/tap# tap_server.py
Interface Name: tap0
From tap    ==>: 02:42:0a:00:20:d3 --> 01:00:5e:00:00:16
                 IP: 0.0.0.0 --> 224.0.0.22
From tap    ==>: 02:42:0a:00:20:d3 --> 01:00:5e:00:00:16
                 IP: 0.0.0.0 --> 224.0.0.22
From socket <==: 02:42:0a:00:20:0c --> 01:00:5e:00:00:16
                 IP 0.0.0.0 --> 224.0.0.22
From socket <==: 02:42:0a:00:20:0c --> 01:00:5e:00:00:16
                 IP 0.0.0.0 --> 224.0.0.22
```

volumes/tap/tap_client.py รันที่ vpn-client-10.0.7.12

```
root@5ec1931fd7ad:/volumes/tap# tap_client.py
Interface Name: tap0
From tap    ==>: 02:42:0a:00:20:0c --> 01:00:5e:00:00:16
                 IP: 0.0.0.0 --> 224.0.0.22
From tap    ==>: 02:42:0a:00:20:0c --> 01:00:5e:00:00:16
                 IP: 0.0.0.0 --> 224.0.0.22
```

ทดสอบ ping host A -> host C ที่ Wireshark จะเห็นเป็น packet UDP

```
From tap    ==>: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP: 10.0.32.5 --> 10.0.32.133
From socket <==: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP 10.0.32.133 --> 10.0.32.5
From tap    ==>: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP: 10.0.32.5 --> 10.0.32.133
From socket <==: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP 10.0.32.133 --> 10.0.32.5
From tap    ==>: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP: 10.0.32.5 --> 10.0.32.133
From socket <==: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP 10.0.32.133 --> 10.0.32.5
From tap    ==>: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP: 10.0.32.5 --> 10.0.32.133
From socket <==: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP 10.0.32.133 --> 10.0.32.5
From socket <==: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 ARP
From tap    ==>: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 ARP
```

```
root@0c99584f45fb:/# ping 10.0.32.133
PING 10.0.32.133 (10.0.32.133) 56(84) bytes of data.
64 bytes from 10.0.32.133: icmp_seq=1 ttl=64 time=5.18 ms
64 bytes from 10.0.32.133: icmp_seq=2 ttl=64 time=2.98 ms
64 bytes from 10.0.32.133: icmp_seq=3 ttl=64 time=3.09 ms
64 bytes from 10.0.32.133: icmp_seq=4 ttl=64 time=2.88 ms
^C
--- 10.0.32.133 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 2.879/3.532/5.177/0.952 ms
root@0c99584f45fb:/#
```

```
From socket <==: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP 10.0.32.5 --> 10.0.32.133
From tap    ==>: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP: 10.0.32.133 --> 10.0.32.5
From socket <==: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP 10.0.32.5 --> 10.0.32.133
From tap    ==>: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP: 10.0.32.133 --> 10.0.32.5
From socket <==: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP 10.0.32.5 --> 10.0.32.133
From tap    ==>: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP: 10.0.32.133 --> 10.0.32.5
From socket <==: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP 10.0.32.5 --> 10.0.32.133
From tap    ==>: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP: 10.0.32.133 --> 10.0.32.5
From tap    ==>: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 ARP
From socket <==: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 ARP
```

No.	Time	Source	Destination	Protocol	Lengt
1	2025-04-16 04:53:32.88…	10.0.7.11	10.0.7.12	UDP	11
2	2025-04-16 04:53:35.27…	10.0.7.12	10.0.7.11	UDP	8
3	2025-04-16 04:53:35.27…	10.0.7.11	10.0.7.12	UDP	8
4	2025-04-16 04:53:35.27…	10.0.7.12	10.0.7.11	UDP	14
5	2025-04-16 04:53:35.27…	10.0.7.11	10.0.7.12	UDP	14
6	2025-04-16 04:53:36.27…	10.0.7.12	10.0.7.11	UDP	14
7	2025-04-16 04:53:36.27…	10.0.7.11	10.0.7.12	UDP	14
8	2025-04-16 04:53:37.27…	10.0.7.12	10.0.7.11	UDP	14
9	2025-04-16 04:53:37.27…	10.0.7.11	10.0.7.12	UDP	14
10	2025-04-16 04:53:38.00…	02:42:0a:00:07:0b	02:42:0a:00:07:0c	ARP	4

ทดสอบ ping host C -> host A

```
From socket <==: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP 10.0.32.133 --> 10.0.32.5
From tap    ==>: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP: 10.0.32.5 --> 10.0.32.133
From socket <==: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP 10.0.32.133 --> 10.0.32.5
From tap    ==>: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP: 10.0.32.5 --> 10.0.32.133
From socket <==: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP 10.0.32.133 --> 10.0.32.5
From tap    ==>: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP: 10.0.32.5 --> 10.0.32.133
From socket <==: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 ARP
From tap    ==>: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 ARP
From tap    ==>: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 ARP
From socket <==: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 ARP
```

24	2025-04-16 05:05:41.39…	10.0.7.11	10.0.7.12	UDP
25	2025-04-16 05:05:41.39…	10.0.7.12	10.0.7.11	UDP
26	2025-04-16 05:05:43.50…	02:42:0a:00:07:0c	02:42:0a:00:07:0b	ARP
27	2025-04-16 05:05:43.50…	02:42:0a:00:07:0b	02:42:0a:00:07:0c	ARP
28	2025-04-16 05:05:43.50…	02:42:0a:00:07:0b	02:42:0a:00:07:0c	ARP
29	2025-04-16 05:05:43.50…	10.0.7.11	10.0.7.12	UDP
30	2025-04-16 05:05:43.50…	02:42:0a:00:07:0c	02:42:0a:00:07:0b	ARP
31	2025-04-16 05:05:43.51…	10.0.7.12	10.0.7.11	UDP
32	2025-04-16 05:05:43.51…	10.0.7.12	10.0.7.11	UDP
33	2025-04-16 05:05:43.51…	10.0.7.11	10.0.7.12	UDP

```
From tap    ==>: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP: 10.0.32.133 --> 10.0.32.5
From socket <==: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP 10.0.32.5 --> 10.0.32.133
From tap    ==>: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP: 10.0.32.133 --> 10.0.32.5
From socket <==: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP 10.0.32.5 --> 10.0.32.133
From tap    ==>: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 IP: 10.0.32.133 --> 10.0.32.5
From socket <==: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 IP 10.0.32.5 --> 10.0.32.133
From tap    ==>: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 ARP
From socket <==: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 ARP
From socket <==: 02:42:0a:00:20:05 --> 02:42:0a:00:20:85
                 ARP
From tap    ==>: 02:42:0a:00:20:85 --> 02:42:0a:00:20:05
                 ARP
```

```
root@c04cc3d0574c:/# ping 10.0.32.5
PING 10.0.32.5 (10.0.32.5) 56(84) bytes of data.
64 bytes from 10.0.32.5: icmp_seq=1 ttl=64 time=4.96 ms
64 bytes from 10.0.32.5: icmp_seq=2 ttl=64 time=3.12 ms
64 bytes from 10.0.32.5: icmp_seq=3 ttl=64 time=2.51 ms
64 bytes from 10.0.32.5: icmp_seq=4 ttl=64 time=2.86 ms
^C
--- 10.0.32.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 2.505/3.360/4.955/0.946 ms
root@c04cc3d0574c:/#
```