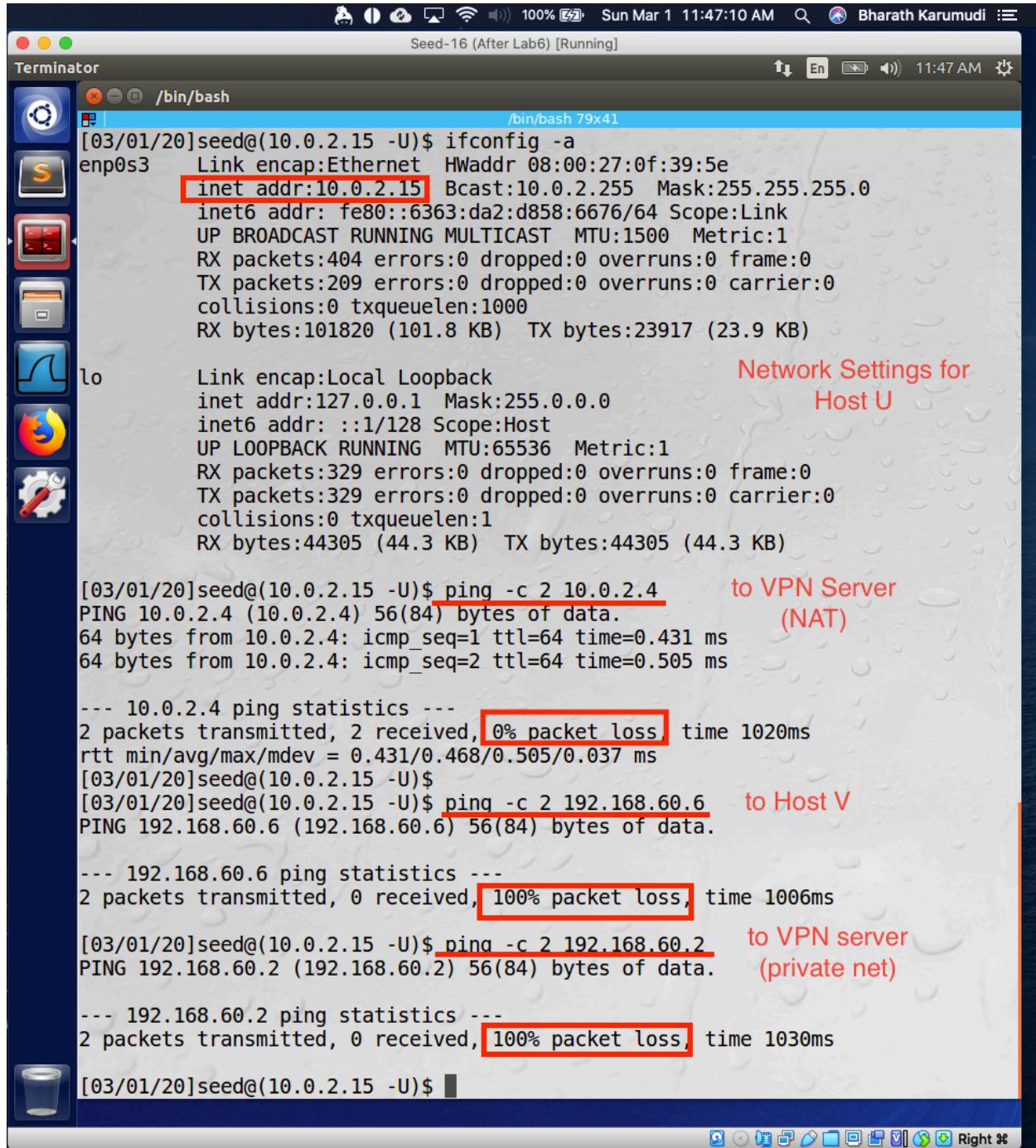


**Name:** Bharath Karumudi  
**Lab:** Virtual Private Network (VPN)

## Task 1: VM Setup



The screenshot shows a terminal window titled "Terminator" running on a Linux desktop environment. The window title bar indicates the session is "Seed-16 (After Lab6) [Running]" and the date and time are "Sun Mar 1 11:47:10 AM". The terminal window displays the following command-line output:

```
[03/01/20]seed@(10.0.2.15 -U)$ ifconfig -a
enp0s3      Link encap:Ethernet HWaddr 08:00:27:0f:39:5e
             inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0
               inet6 addr: fe80::6363:da2:d858:6676/64 Scope:Link
                  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                 RX packets:404 errors:0 dropped:0 overruns:0 frame:0
                 TX packets:209 errors:0 dropped:0 overruns:0 carrier:0
                   collisions:0 txqueuelen:1000
                  RX bytes:101820 (101.8 KB) TX bytes:23917 (23.9 KB)

lo          Link encap:Local Loopback
             inet addr:127.0.0.1 Mask:255.0.0.0
               inet6 addr: ::1/128 Scope:Host
                  UP LOOPBACK RUNNING MTU:65536 Metric:1
                 RX packets:329 errors:0 dropped:0 overruns:0 frame:0
                 TX packets:329 errors:0 dropped:0 overruns:0 carrier:0
                   collisions:0 txqueuelen:1
                  RX bytes:44305 (44.3 KB) TX bytes:44305 (44.3 KB)

[03/01/20]seed@(10.0.2.15 -U)$ ping -c 2 10.0.2.4           to VPN Server
PING 10.0.2.4 (10.0.2.4) 56(84) bytes of data.
64 bytes from 10.0.2.4: icmp_seq=1 ttl=64 time=0.431 ms
64 bytes from 10.0.2.4: icmp_seq=2 ttl=64 time=0.505 ms

--- 10.0.2.4 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1020ms
rtt min/avg/max/mdev = 0.431/0.468/0.505/0.037 ms
[03/01/20]seed@(10.0.2.15 -U)$
[03/01/20]seed@(10.0.2.15 -U)$ ping -c 2 192.168.60.6     to Host V
PING 192.168.60.6 (192.168.60.6) 56(84) bytes of data.

--- 192.168.60.6 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1006ms
[03/01/20]seed@(10.0.2.15 -U)$ ping -c 2 192.168.60.2       to VPN server
PING 192.168.60.2 (192.168.60.2) 56(84) bytes of data.

--- 192.168.60.2 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1030ms
[03/01/20]seed@(10.0.2.15 -U)$
```

Annotations in red text are present in the terminal output:

- "inet addr:10.0.2.15" is highlighted in red.
- "Network Settings for Host U" is written in red text next to the loopback interface output.
- "ping -c 2 10.0.2.4" is highlighted in red.
- "to VPN Server (NAT)" is written in red text next to the ping command.
- "0% packet loss" is highlighted in red.
- "ping -c 2 192.168.60.6" is highlighted in red.
- "to Host V" is written in red text next to the ping command.
- "100% packet loss" is highlighted in red.
- "ping -c 2 192.168.60.2" is highlighted in red.
- "to VPN server (private net)" is written in red text next to the ping command.
- "100% packet loss" is highlighted in red.

Fig1: Network Connections and Validations of Host U (VPN Client)

Network  
Connections  
of VPN Server

```
[03/01/20]seed@(10.0.2.4 192.168.60.2 -VPN)$ ifconfig -a
enp0s3      Link encap:Ethernet HWaddr 08:00:27:46:8d:61
             inet addr:10.0.2.4 Bcast:10.0.2.255 Mask:255.255.255.0
                          inet6 addr: fe80::e3db:2332%enp0s3/64 Scope:Link
                           UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                           RX packets:45 errors:0 dropped:0 overruns:0 frame:0
                           TX packets:206 errors:0 dropped:0 overruns:0 carrier:0
                           collisions:0 txqueuelen:1000
                           RX bytes:6898 (6.8 KB) TX bytes:19936 (19.9 KB)

enp0s8      Link encap:Ethernet HWaddr 08:00:27:8c:fd:2b
             inet addr:192.168.60.2 Bcast:192.168.60.255 Mask:255.255.255.0
                          inet6 addr: fe80::ede5:c6e5%enp0s8/64 Scope:Link
                           UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                           RX packets:275 errors:0 dropped:0 overruns:0 frame:0
                           TX packets:175 errors:0 dropped:0 overruns:0 carrier:0
                           collisions:0 txqueuelen:1000
                           RX bytes:24347 (24.3 KB) TX bytes:29109 (29.1 KB)

lo          Link encap:Local Loopback
             inet addr:127.0.0.1 Mask:255.0.0.0
                          inet6 addr: ::1/128 Scope:Host
                           UP LOOPBACK RUNNING MTU:65536 Metric:1
                           RX packets:198 errors:0 dropped:0 overruns:0 frame:0
                           TX packets:198 errors:0 dropped:0 overruns:0 carrier:0
                           collisions:0 txqueuelen:1
                           RX bytes:32480 (32.4 KB) TX bytes:32480 (32.4 KB)

[03/01/20]seed@(10.0.2.4 192.168.60.2 -VPN)$ ping -c 2 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=0.402 ms      to Host U
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=0.551 ms

--- 10.0.2.15 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1029ms
rtt min/avg/max/mdev = 0.402/0.476/0.551/0.077 ms
[03/01/20]seed@(10.0.2.4 192.168.60.2 -VPN)$
[03/01/20]seed@(10.0.2.4 192.168.60.2 -VPN)$ ping -c 2 192.168.60.6
PING 192.168.60.6 (192.168.60.6) 56(84) bytes of data.
64 bytes from 192.168.60.6: icmp_seq=1 ttl=64 time=0.743 ms      to Host V
64 bytes from 192.168.60.6: icmp_seq=2 ttl=64 time=0.596 ms

--- 192.168.60.6 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1020ms
rtt min/avg/max/mdev = 0.596/0.669/0.743/0.077 ms
[03/01/20]seed@(10.0.2.4 192.168.60.2 -VPN)$
```

Fig2: Network Connections and Validations of VPN Server

[03/01/20]seed@(192.168.60.6 -V)\$ clear

[03/01/20]seed@(192.168.60.6 -V)\$ ifconfig -a

enp0s3 Link encap:Ethernet HWaddr 08:00:27:da:46:86  
inet addr: 192.168.60.6 Bcast:192.168.60.255 Mask:255.255.255.0  
inet6 addr: fe80::e4d3:f92d:d3d9:7c41/64 Scope:Link  
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
RX packets:10 errors:0 dropped:0 overruns:0 frame:0  
TX packets:151 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:1000  
RX bytes:1451 (1.4 KB) TX bytes:14404 (14.4 KB)

lo Link encap:Local Loopback  
inet addr:127.0.0.1 Mask:255.0.0.0  
inet6 addr: ::1/128 Scope:Host  
UP LOOPBACK RUNNING MTU:65536 Metric:1  
RX packets:192 errors:0 dropped:0 overruns:0 frame:0  
TX packets:192 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:1  
RX bytes:23424 (23.4 KB) TX bytes:23424 (23.4 KB)

[03/01/20]seed@(192.168.60.6 -V)\$ ping -c 2 10.0.2.15 to Host U

PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.  
From 192.168.60.6 icmp\_seq=1 Destination Host Unreachable  
From 192.168.60.6 icmp\_seq=2 Destination Host Unreachable

--- 10.0.2.15 ping statistics ---  
2 packets transmitted, 0 received, +2 errors, 100% packet loss, time 1010ms pipe 2

[03/01/20]seed@(192.168.60.6 -V)\$ ping -c 2 10.0.2.4 to VPN server (NAT)

PING 10.0.2.4 (10.0.2.4) 56(84) bytes of data.  
From 192.168.60.6 icmp\_seq=1 Destination Host Unreachable  
From 192.168.60.6 icmp\_seq=2 Destination Host Unreachable

--- 10.0.2.4 ping statistics ---  
2 packets transmitted, 0 received, +2 errors, 100% packet loss, time 1031ms pipe 2

[03/01/20]seed@(192.168.60.6 -V)\$ ping -c 2 192.168.60.2 to VPN Server (private net)

PING 192.168.60.2 (192.168.60.2) 56(84) bytes of data.  
64 bytes from 192.168.60.2: icmp\_seq=1 ttl=64 time=0.334 ms  
64 bytes from 192.168.60.2: icmp\_seq=2 ttl=64 time=0.515 ms

--- 192.168.60.2 ping statistics ---  
2 packets transmitted, 2 received, 0% packet loss, time 1026ms  
rtt min/avg/max/mdev = 0.334/0.424/0.515/0.092 ms

[03/01/20]seed@(192.168.60.6 -V)\$

Fig3: Network Connections and Validations of Host V (Private network host)

```
[03/01/20]seed@(10.0.2.4 192.168.60.2 -VPN)$ route -n
Kernel IP routing table
Destination      Gateway         Genmask        Flags Metric Ref  Use Iface
0.0.0.0          10.0.2.1       0.0.0.0       UG    100   0    0 enp0s3
0.0.0.0          192.168.60.1  0.0.0.0       UG    101   0    0 enp0s8
10.0.2.0          0.0.0.0        255.255.255.0 U     100   0    0 enp0s3
169.254.0.0      0.0.0.0        255.255.0.0   U     1000  0    0 enp0s3
192.168.60.0     0.0.0.0        255.255.255.0 U     100   0    0 enp0s8
[03/01/20]seed@(10.0.2.4 192.168.60.2 -VPN)$
```

Fig 4: Routing table of VPN server

**Observation:** Set up the network connections for VPN server, VPN Client (Host U) and Private network host (Host V). After the setup, the Host U is able to communicate with VPN server on NAT (simulated Internet) but not to Host V. VPN server is able to communicate with both Host U and Host V via corresponding networks. Host V is able to communicate with VPN server via internal network but not to Host U. So, the only way that Host U and Host V will be communicated is via VPN server.

**Explanation:**

1. Created a NAT network of 10.0.2.0/24
2. Create a private network (privnet) of 192.168.60.0/24
3. The Host U is placed in NAT network and assigned IP as 10.0.2.15 (Fig1).
4. The Host V is place in private network and assigned IP as 192.168.60.6 (Fig 3).
5. The VPN server is placed on both the networks by having two network adapters and allocated the IP as 10.0.2.4 (on NAT) and 192.168.60.2 (on privnet) as shown in Fig 2 and routing table is as shown in Fig 4.

## Task 2: Creating a VPN Tunnel using TUN/TAP

### Step1: Run VPN Server

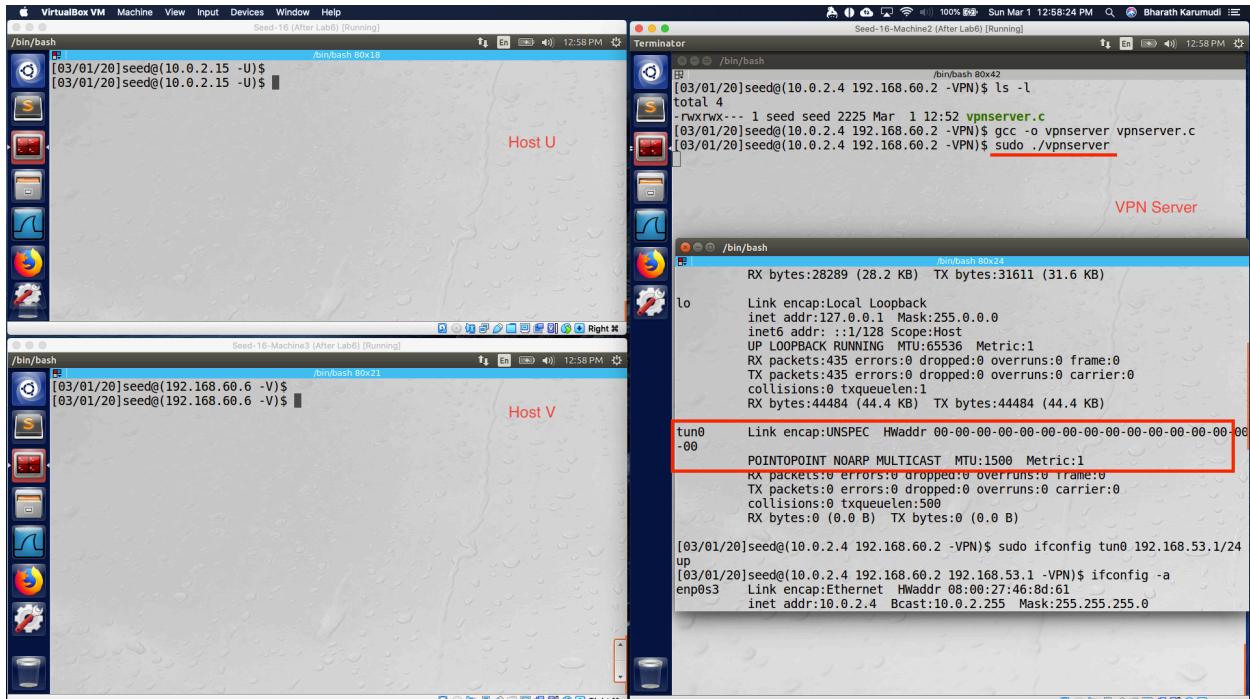


Fig1: Creating the TUN interface (tun0) on VPN server

```
#include <fcntl.h>
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <arpa/inet.h>
#include <linux/if.h>
#include <linux/if_tun.h>
#include <sys/ioctl.h>

#define PORT_NUMBER 55555
#define BUFF_SIZE 2000

struct sockaddr_in peerAddr;
int tunfd;
struct ifreq ifr;
memset(&ifr, 0, sizeof(ifr));
ifr.ifr_flags = IFF_TUN | IFF_NO_PI;
tunfd = open("/dev/net/tun", O_RDWR);
ioctl(tunfd, TUNSETIFF, &ifr);
return tunfd;
}

int initUDPServer() {
    int sockfd;
    struct sockaddr_in server;
    char buff[100];
    memset(&server, 0, sizeof(server));
    server.sin_family = AF_INET;
    server.sin_addr.s_addr = htonl(INADDR_ANY);
    server.sin_port = htons(PORT_NUMBER);

    sockfd = socket(AF_INET, SOCK_DGRAM, 0);
    bind(sockfd, (struct sockaddr*)&server, sizeof(server));
    // Wait for the VPN client to "connect".
    bzero(buff, 100);
    int peerAddrLen = sizeof(struct sockaddr_in);
    int len = recvfrom(sockfd, buff, 100, 0,
```

Fig1a: vpnserv.c program

Seed-16-Machine2 (After Lab6) [Running]

Terminator

```
[03/01/20]seed@(10.0.2.4 192.168.60.2 -VPN)$ ls -l
total 4
-rwxrwx--- 1 seed seed 2225 Mar 1 12:52 vpnserver.c
[03/01/20]seed@(10.0.2.4 192.168.60.2 -VPN)$ gcc -o vpnserver vpnserver.c
[03/01/20]seed@(10.0.2.4 192.168.60.2 -VPN)$ sudo ./vpnserver
```

/bin/bash

```
lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
              UP LOOPBACK RUNNING MTU:65536 Metric:1
              RX packets:470 errors:0 dropped:0 overruns:0 frame:0
              TX packets:470 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:1
              RX bytes:47883 (47.8 KB) TX bytes:47883 (47.8 KB)

tun0    Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
        inet addr:192.168.53.1 P-t-P:192.168.53.1 Mask:255.255.255.0
        inet6 addr: fe80::f14e:9904%tun0/64 Scope:Link
              UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1500 Metric:1
              RX packets:0 errors:0 dropped:0 overruns:0 frame:0
              TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:500
              RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

VPN Server

```
[03/01/20]seed@(10.0.2.4 192.168.60.2 192.168.53.1 -VPN)$ sudo sysctl net.ipv4.ip_forward=1
net.ipv4.ip forward = 1
[03/01/20]seed@(10.0.2.4 192.168.60.2 192.168.53.1 -VPN)$
```

```
[03/01/20]seed@(10.0.2.4 192.168.60.2 192.168.53.1 -VPN)$ route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref  Use Iface
0.0.0.0          10.0.2.1       0.0.0.0        UG   100    0    0 enp0s3
0.0.0.0          192.168.60.1   0.0.0.0        UG   101    0    0 enp0s8
10.0.2.0          0.0.0.0        255.255.255.0  U     100    0    0 enp0s3
169.254.0.0       0.0.0.0        255.255.0.0    U     1000   0    0 enp0s3
192.168.53.0     0.0.0.0        255.255.255.0  U     0      0    0 tun0
192.168.60.0     0.0.0.0        255.255.255.0  U     100    0    0 enp0s8
```

Fig2: Assigned IP to tun0 and VPN server routing table.

**Observation:** When compiled and executed the vpnserver.c program, the TUN interface tun0 was created and assigned the IP address as 192.168.53.1 for tun0 and made it up. Also made the host as gateway.

**Explanation:** The `vpnserver.c` was executed with sudo to create the `tun0` interface as shown in Fig 1 and using the `ifconfig` assigned the `192.168.53.1` IP address to the `tun0` interface and using `sysctl` made the VPN host as gateway so it forward the packets (Fig2).

```
sudo ifconfig tun0 192.168.53.1/24 up
```

## **Step2: Run VPN Client**

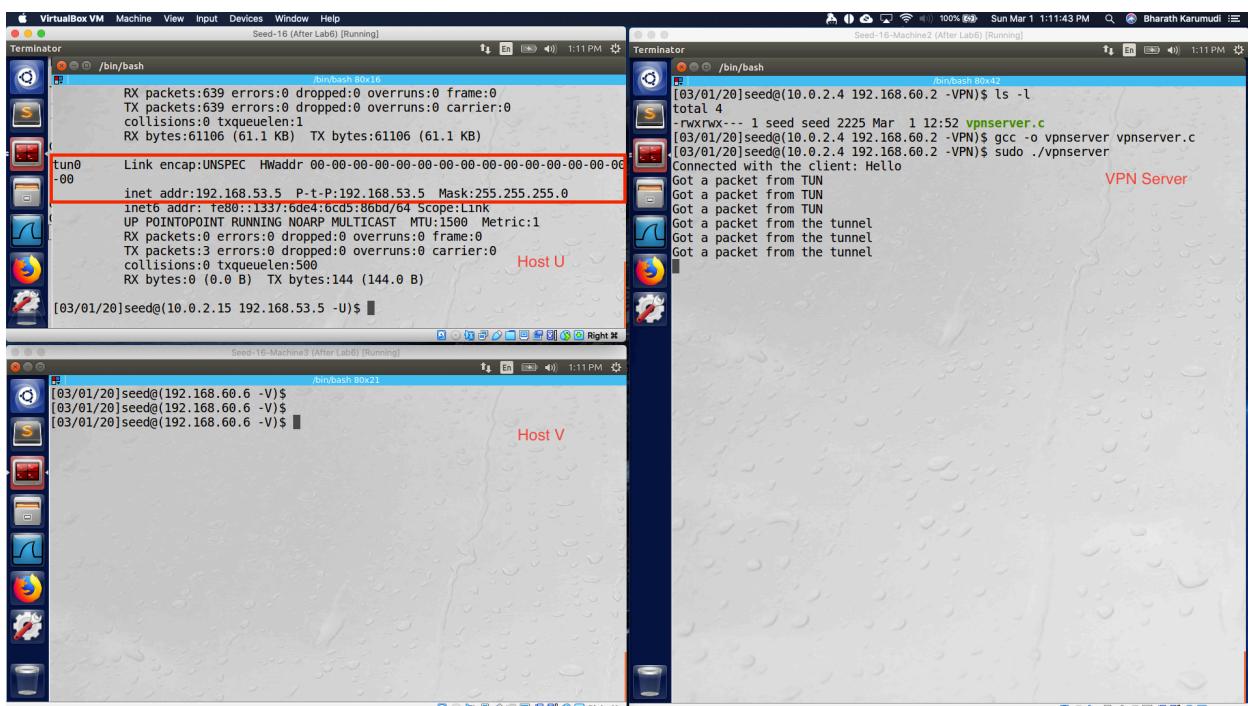


Fig3: Established TUN interface on VPN Client and is up and running

```

1 #include <fcntl.h>
2 #include <stdio.h>
3 #include <unistd.h>
4 #include <string.h>
5 #include <arpa/inet.h>
6 #include <linux/if.h>
7 #include <linux/if_tun.h>
8 #include <sys/ioctl.h>
9
10 #define BUFF_SIZE 2000
11 #define PORT_NUMBER 5555
12 #define SERVER_IP "10.0.2.4"
13 struct sockaddr_in peerAddr;
14
15 int createTunDevice() {
16     int tunfd;
17     struct ifreq ifr;
18     memset(&ifr, 0, sizeof(ifr));
19
20     ifr.ifr_flags = IFF_TUN | IFF_NO_PI;
21
22     tunfd = open("/dev/net/tun", O_RDWR);
23     ioctl(tunfd, TUNSETIFF, &ifr);
24
25     return tunfd;
26 }
27
28 int connectToUDPServer(){
29     int sockfd;
30     char hello="Hello";
31
32     memset(&peerAddr, 0, sizeof(peerAddr));
33     peerAddr.sin_family = AF_INET;
34     peerAddr.sin_port = htons(PORT_NUMBER);
35     peerAddr.sin_addr.s_addr = inet_addr(SERVER_IP);
36
37     sockfd = socket(AF_INET, SOCK_DGRAM, 0);
38
39     // Send a hello message to "connect" with the VPN server
40     sendto(sockfd, hello, strlen(hello), 0,
41             (struct sockaddr *) &peerAddr, sizeof(peerAddr));
42
43     return sockfd;
44 }

```

Fig 3a: vpncclient.c program

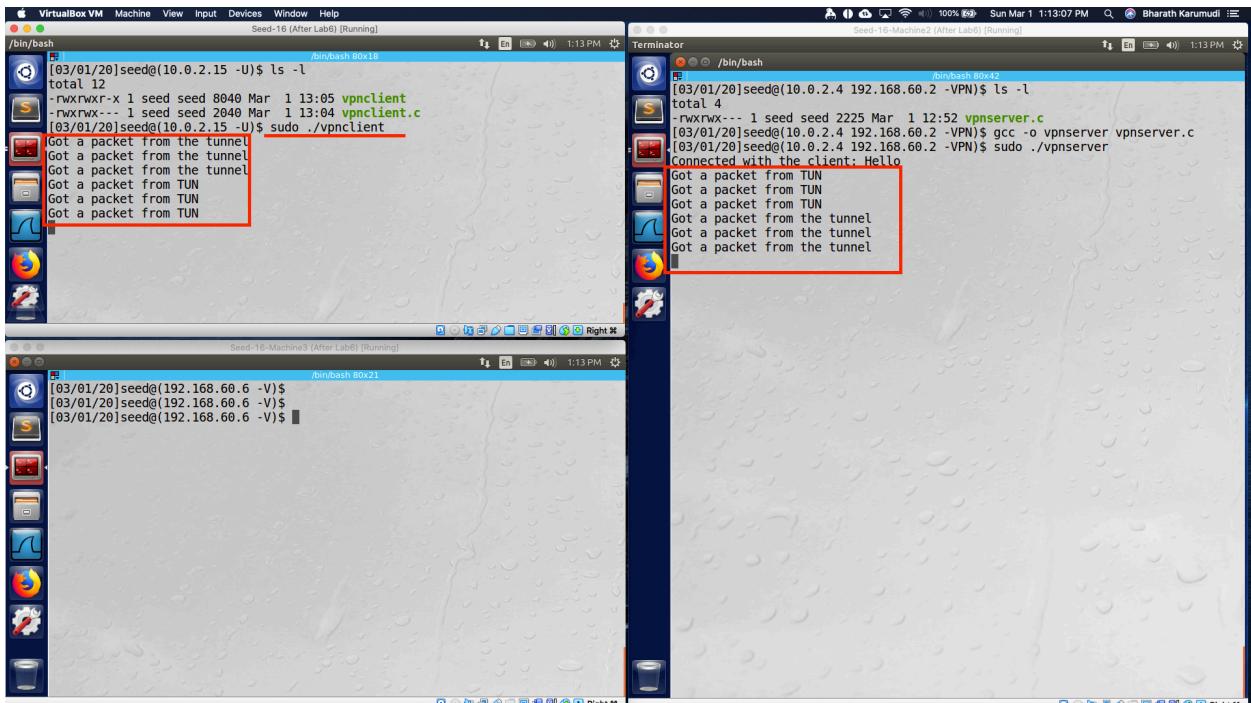


Fig4: Running the VPNClient program

**Observation:** Executed the VPNCClient.c and created a tun0 interface on Host U but down and no IP allocated. Made the interface up and allocated the IP to interface as 192.168.53.5 and the client connected to VPN server.

**Explanation:** When executed the VPNClient.c on the client (Host U) first the TUN interface was created as tun0 but no IP allocated and also down (Fig 3). So, using the ifconfig allocated the IP as 192.168.53.5 and made the interface up `sudo ifconfig tun0 192.168.53.5/24 up` and the client connected to VPN server (Fig 4).

### Step 3: Set Up Routing on Client and Server VMs

The screenshot shows two terminal windows side-by-side. The left window is titled "Seed-16 (After Lab6) [Running]" and the right window is titled "Seed-16-4Machine2 (After Lab6) [Running]". Both windows show terminal sessions with red boxes highlighting specific route entries.

**Host U : VPN Client (Left Window):**

```
[03/01/20]seed@(10.0.2.15 192.168.53.5)$ route -n
Kernel IP routing table
Destination     Gateway      Genmask      Flags Metric Ref  Use Iface
0.0.0.0         10.0.2.1    0.0.0.0      UG        100    0      0 enp0s3
10.0.2.0        0.0.0.0     255.255.255.0 U          100    0      0 enp0s3
169.254.0.0     0.0.0.0     255.255.0.0   U          1000   0      0 enp0s3
192.168.53.0    0.0.0.0     255.255.255.0 U          0       0      0 tun0
[03/01/20]seed@(10.0.2.15 192.168.53.5)$
[03/01/20]seed@(10.0.2.15 192.168.53.5)$ sudo route add -net 192.168.60.0/24 tun0
[03/01/20]seed@(10.0.2.15 192.168.53.5)$ route -n
Kernel IP routing table
Destination     Gateway      Genmask      Flags Metric Ref  Use Iface
0.0.0.0         10.0.2.1    0.0.0.0      UG        100    0      0 enp0s3
10.0.2.0        0.0.0.0     255.255.255.0 U          100    0      0 enp0s3
169.254.0.0     0.0.0.0     255.255.0.0   U          1000   0      0 enp0s3
192.168.53.0    0.0.0.0     255.255.255.0 U          0       0      0 tun0
192.168.60.0    0.0.0.0     255.255.255.0 U          0       0      0 tun0
[03/01/20]seed@(10.0.2.15 192.168.53.5)$
[03/01/20]seed@(10.0.2.15 192.168.53.5)$
```

**VPN Server (Right Window):**

```
[03/01/20]seed@(10.0.2.4 192.168.60.2 192.168.53.1 -VPN)$ route -n
Kernel IP routing table
Destination     Gateway      Genmask      Flags Metric Ref  Use Iface
0.0.0.0         10.0.2.1    0.0.0.0      UG        100    0      0 enp0s3
0.0.0.0         192.168.60.1 0.0.0.0      UG        101    0      0 enp0s8
10.0.2.0        0.0.0.0     255.255.255.0 U          100    0      0 enp0s3
169.254.0.0     0.0.0.0     255.255.0.0   U          1000   0      0 enp0s3
192.168.53.0    0.0.0.0     255.255.255.0 U          0       0      0 tun0
192.168.60.0    0.0.0.0     255.255.255.0 U          100    0      0 enp0s8
[03/01/20]seed@(10.0.2.4 192.168.60.2 192.168.53.1 -VPN)$
```

Fig5: Routing setup on VPN Client (Left) and VPN Server (Right)

**Observation:** The tun0 route was added by default on client for 192.168.53.0/24 net but to communicate with the private network (192.168.60.0/24) the route was not there, so added the route. On the VPN server the entry was added automatically on enp0s3 interface.

### Explanation:

On Host U added the route 192.168.60.0/24 on tun0 using:

`sudo route add -net 192.168.60.0/24 tun0` – this added the entry to the route table and any packet that has a destination of this network will be routed to tun0 network interface, which is the tunnel between Host U and VPN Server. Once the VPN server receives it, it will forward the packet on enp0s3 interface where the private network is.

#### Step 4: Set Up Routing on Host V

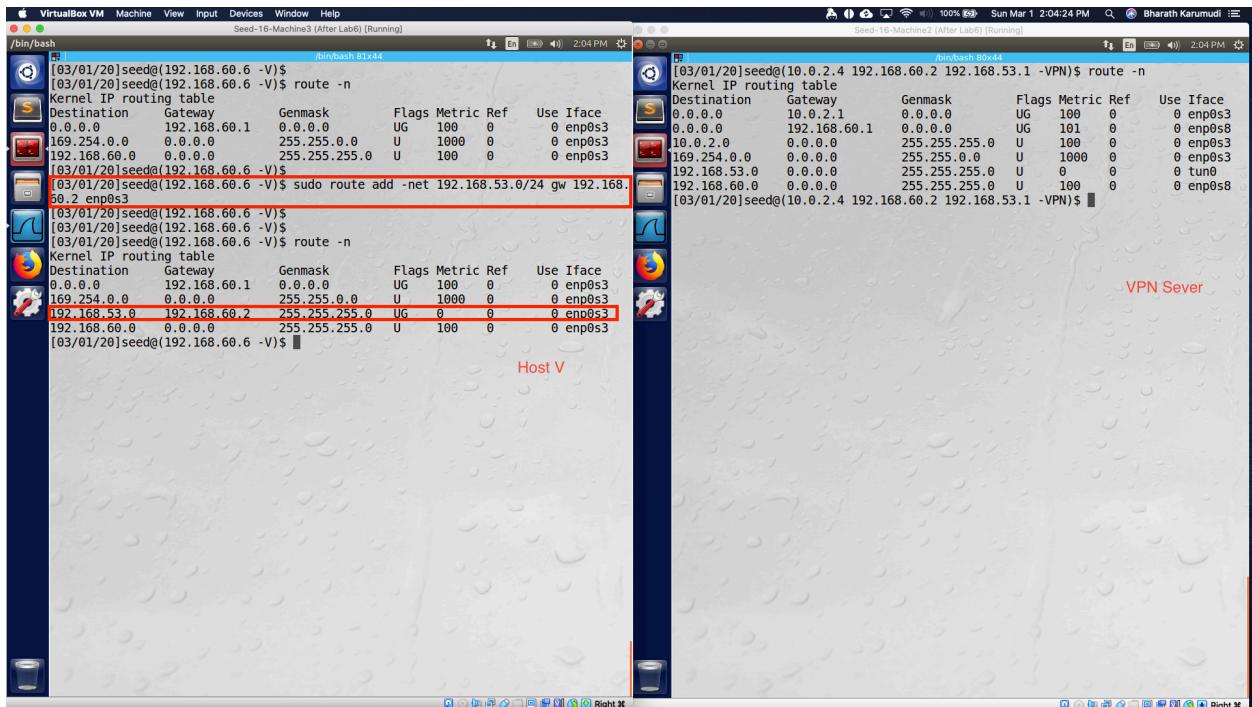


Fig6: Routing setup on Private network host (Host V)

**Observation:** There were no routes on Host V to respond to Host U, so added a route on Host V.

**Explanation:** When the packet arrives from Host U to Host V via VPN server, the Host V should respond in the same private channel. To achieve that we need a route to be added on the Host V and with: `sudo route add -net 192.168.53.0/24 gw 192.169.50.2 enp0s3` a new route was added (Fig 6). With the above command, any reply packet that has a destination IP as 192.168.53.0/24 will be sent to enp0s3 interface and uses the VPN server 192.169.50.2 as gateway. So, the packet from Host V will go to Host U via VPN server.

## Step 5: Test the VPN Tunnel:

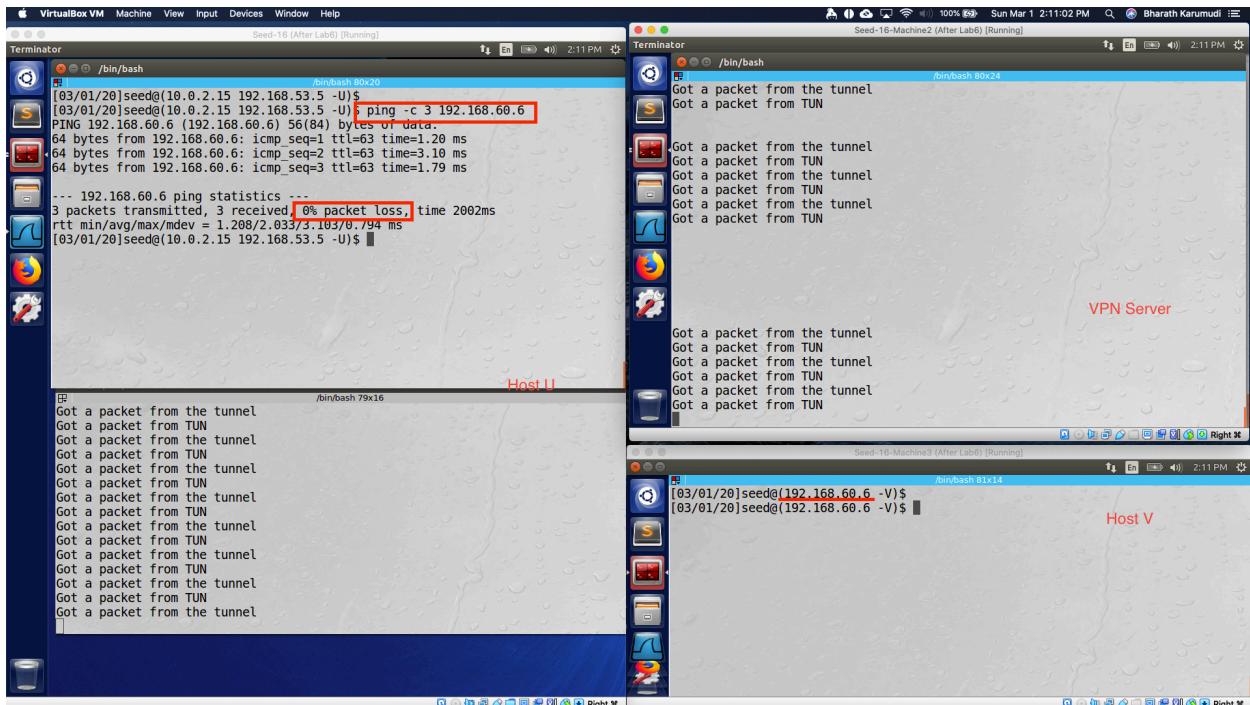


Fig7: Ping test from Host U to Host V

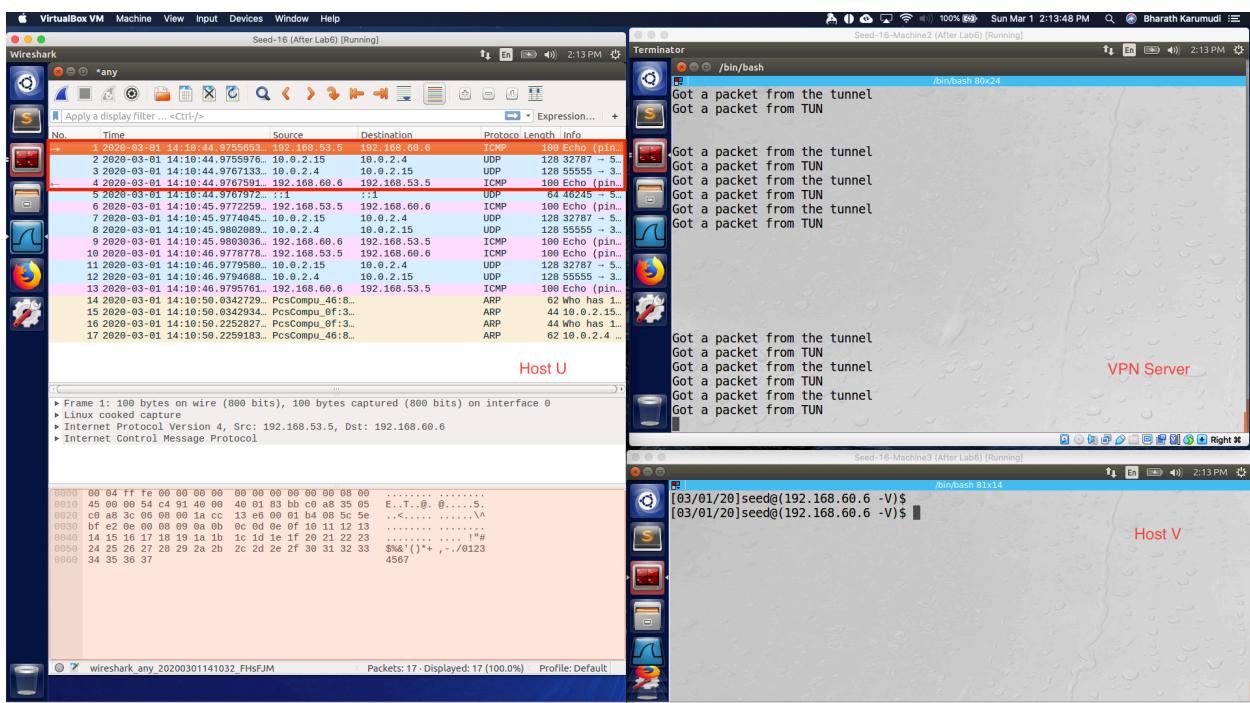


Fig8: Wireshark observation on Host U for ICMP traffic

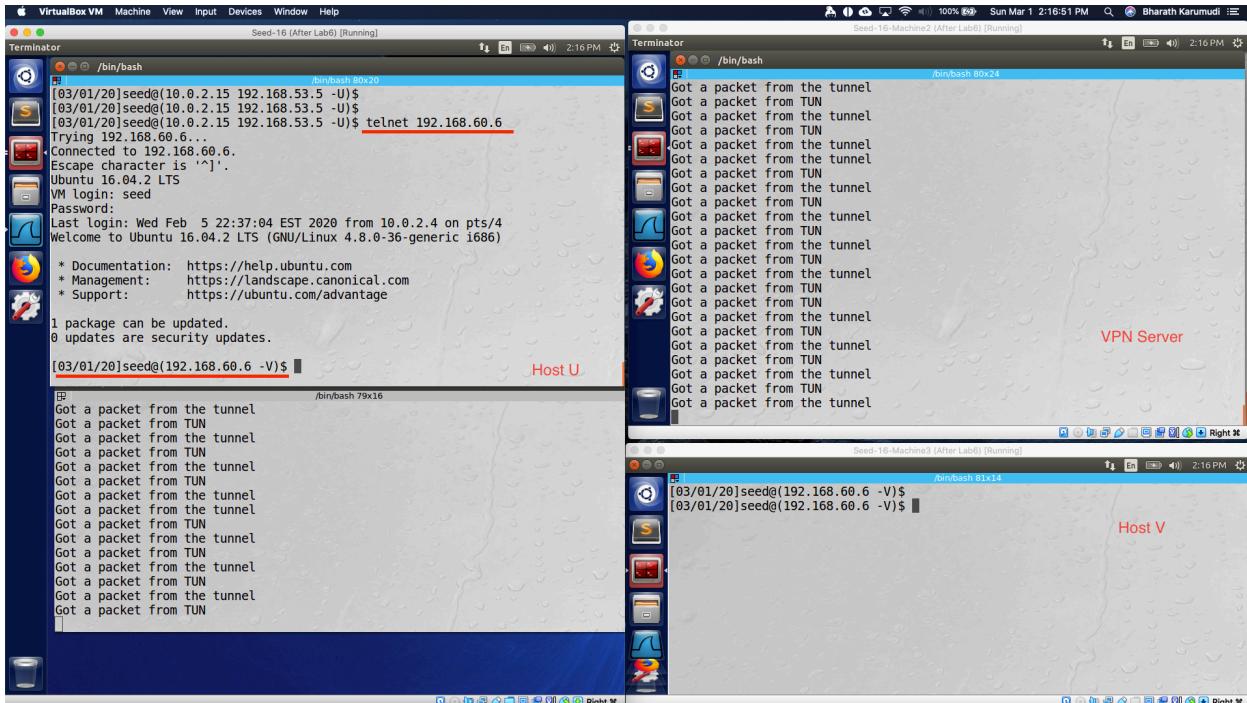


Fig9: Telnet test from Host U to Host V

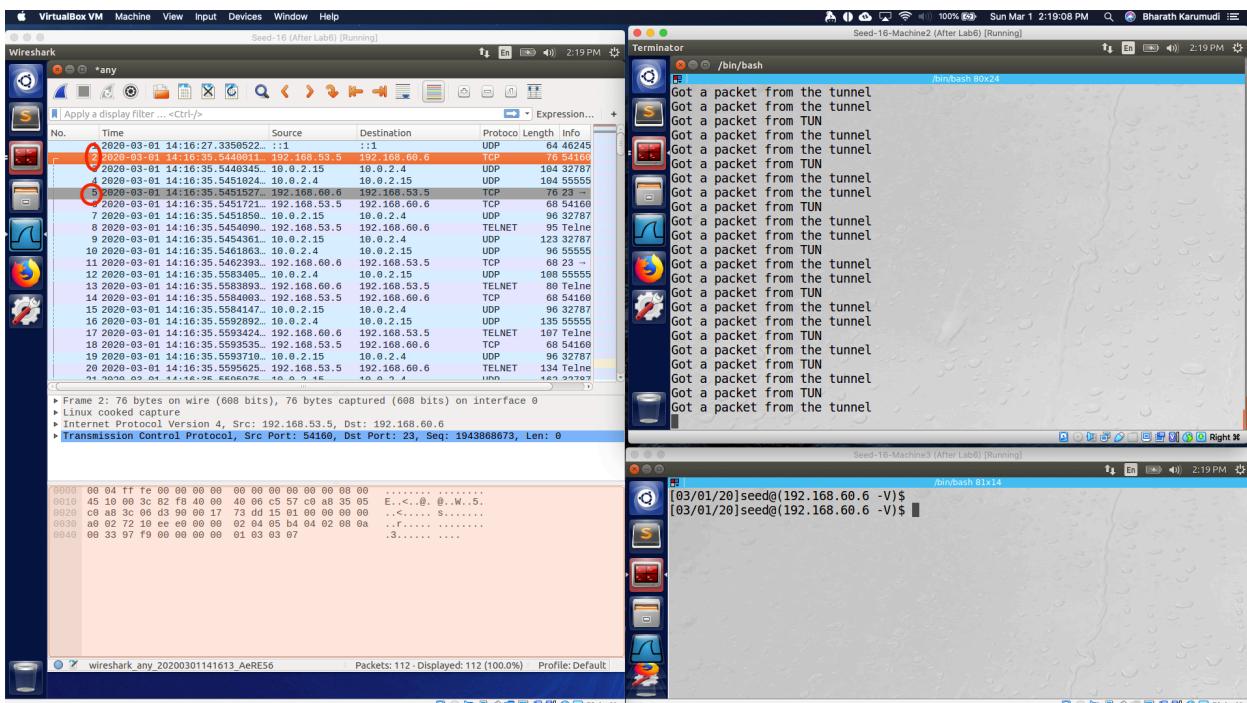


Fig10: Wireshark observation for telnet connection

**Observation:** Once all the network setup was done, from Host U sent a ping to Host V and received the ping response. Also connected to Host V over telnet from Host U.

### Explanation:

1. Sent a ping from Host U to Host V and received all the packets (Fig 7).
2. Observed the traffic on Wireshark (Fig 8) and can see the ICMP traffic went through tunnel and the UDP went on regular network (NAT in this case) – these are physical connection between VPN Client and VPN server.
3. Connected from Host U to Host V over the telnet (Fig 9).
4. The network traffic was observed over the Wireshark (Fig 9), all the telnet and TCP went through tunnel and the physical connection went from non-tunnel.

### Step 5: Tunnel-Breaking Test.

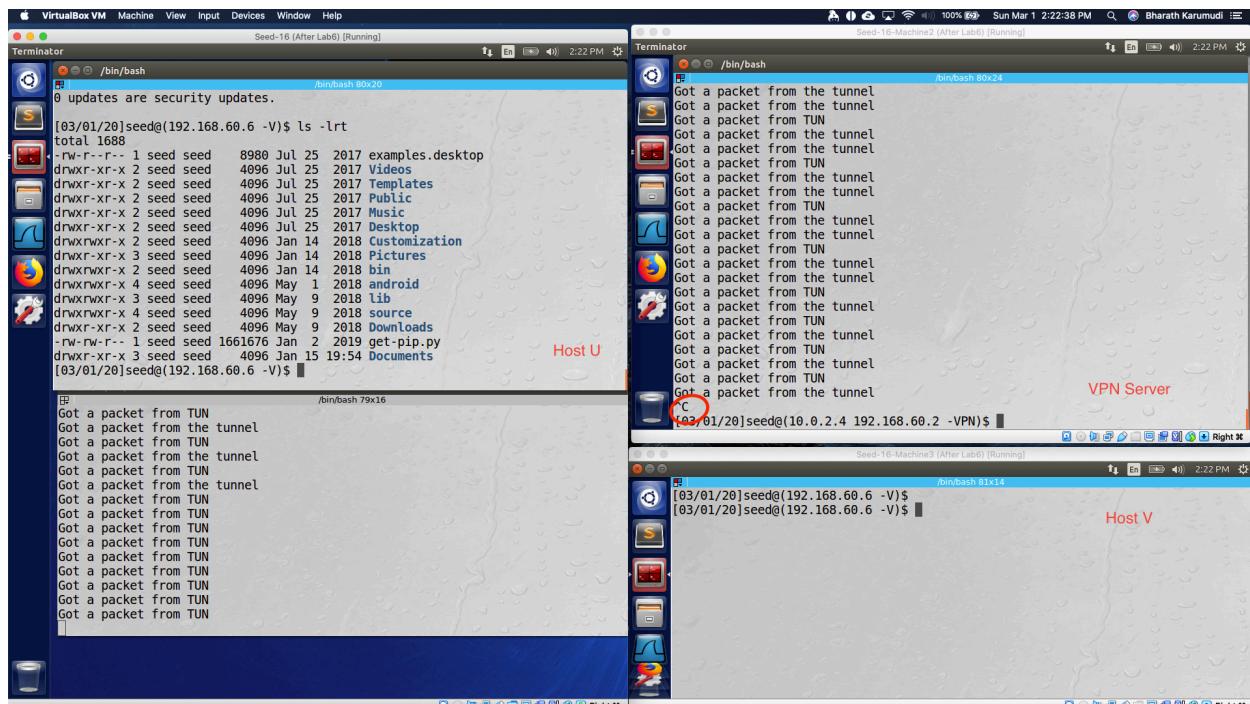


Fig11: Terminated the tunnel

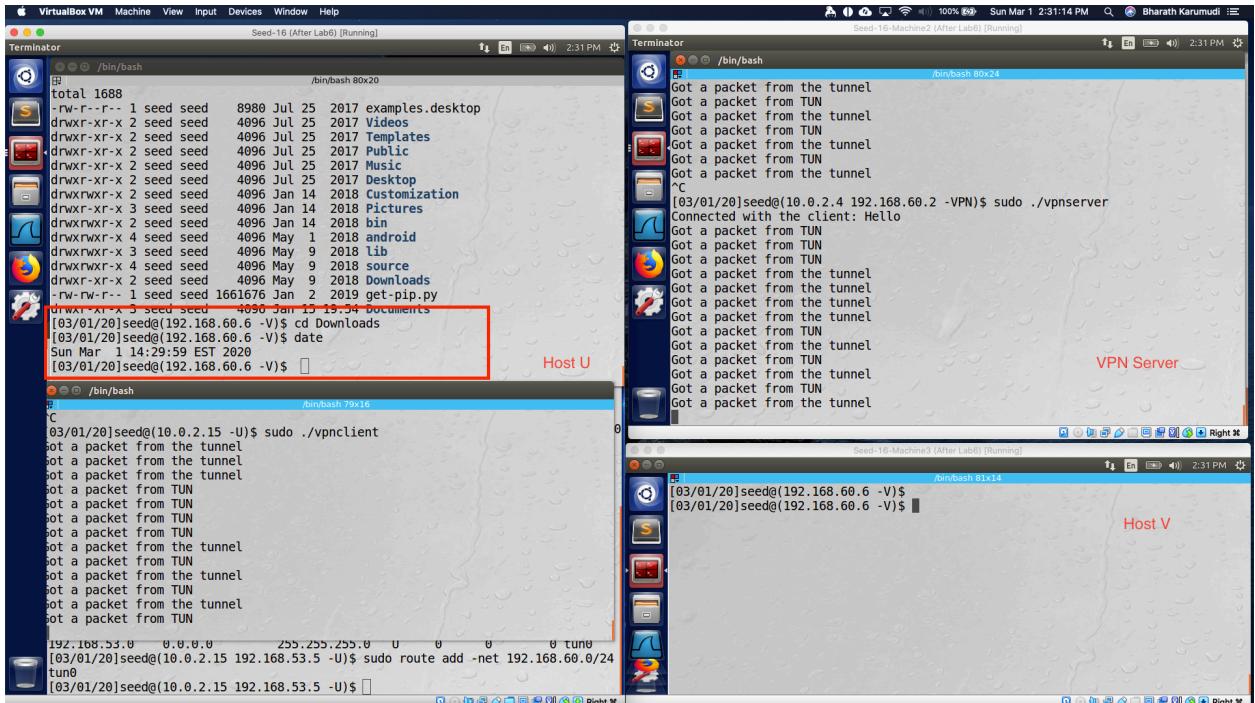


Fig12: Re-established the tunnel

**Observation:** Established a telnet connection from Host U to Host V and by terminating the tunnel, the telnet connection was just not responding anything we type but it was not broken. After re-establishing the tunnel between VPN Client and VPN server all the packets that are in buffer were sent out and got the response.

### Explanation:

Established a telnet from Host U to Host V by having a VPN connection (tunnel) between Host U and VPN server (Fig 11). Interrupted the tunnel but the **telnet** window on Host U is still **active** and issued some commands (cd Downloads and date) but nothing was visible on the terminal. Once re-establishing the tunnel between Host U and VPN server (have to add the routing entries again on client and bring the tunnel up on both client VPN server) the commands that were typed earlier were sent to Host V and got the response – meaning the typed commands are in the buffer and once the connection established those packets are sent out through the tunnel.