**COMPUTER NETWORK SECURITY**

**LAB-9**

**VPN Tunneling**

NAME: VISHWAS M

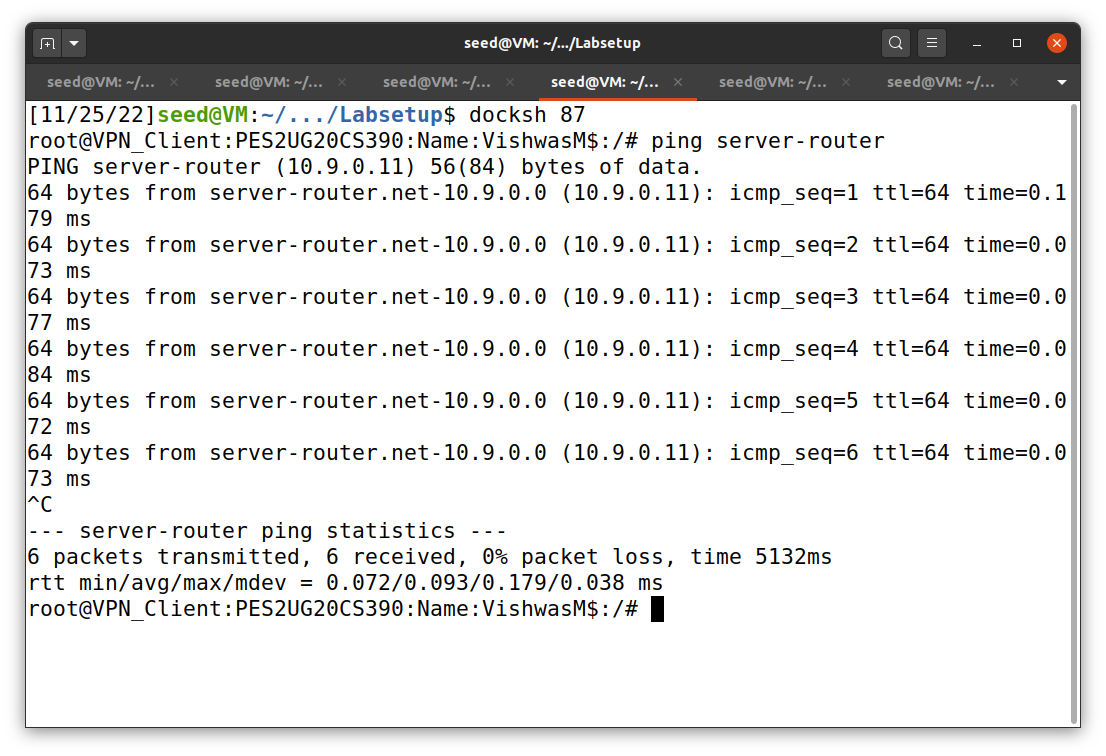
SRN: PES2UG20CS390

SEC: F

DATE:26/11/2022

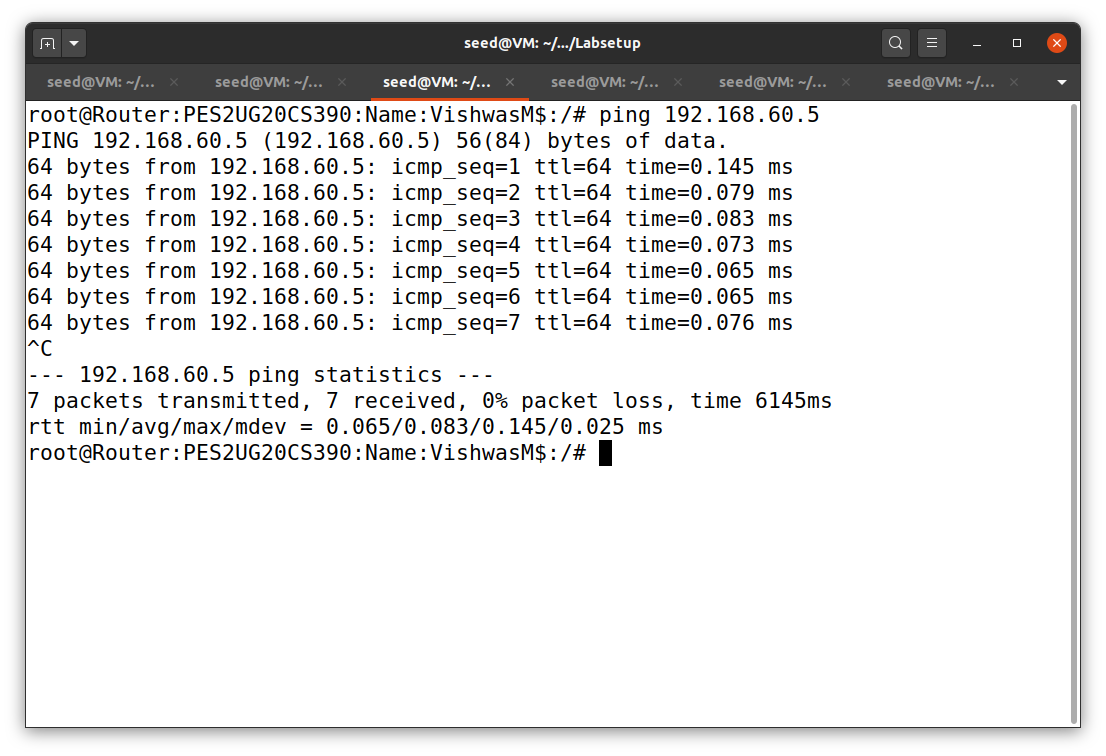
Task1: Network Security

Host U - 10.9.0.5 can communicate with VPN Server (server-router)



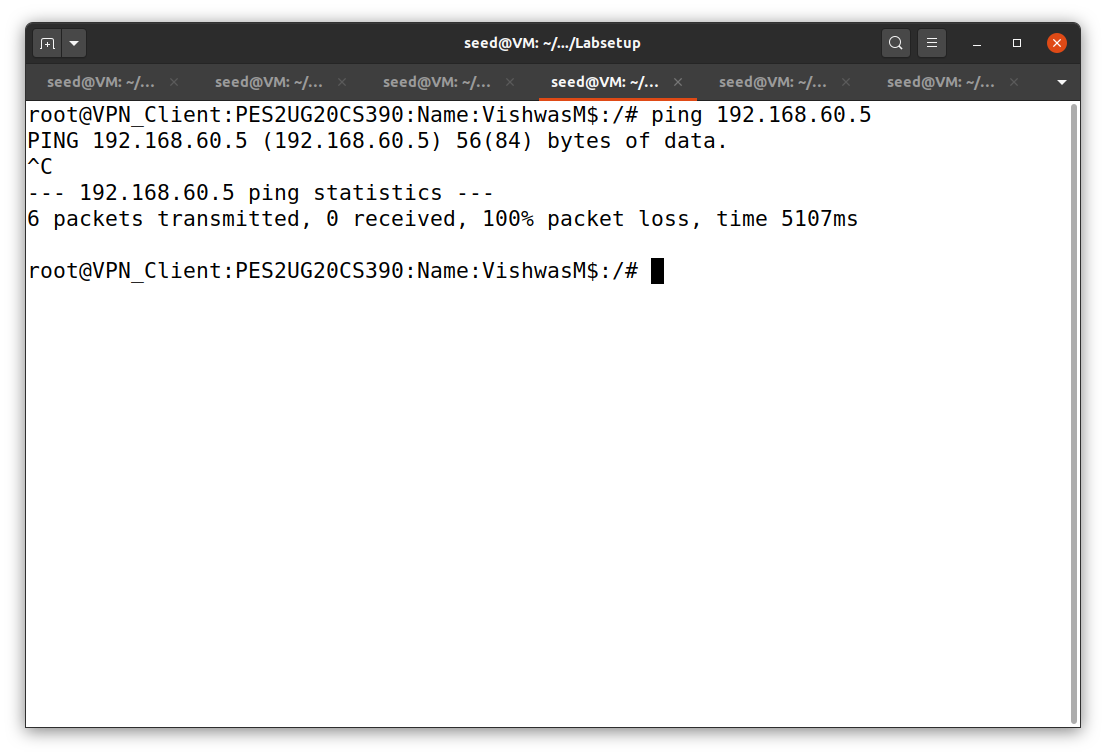
VPN Server (server-router) can communicate with Host V (host-192.168.60.5)

On server-router



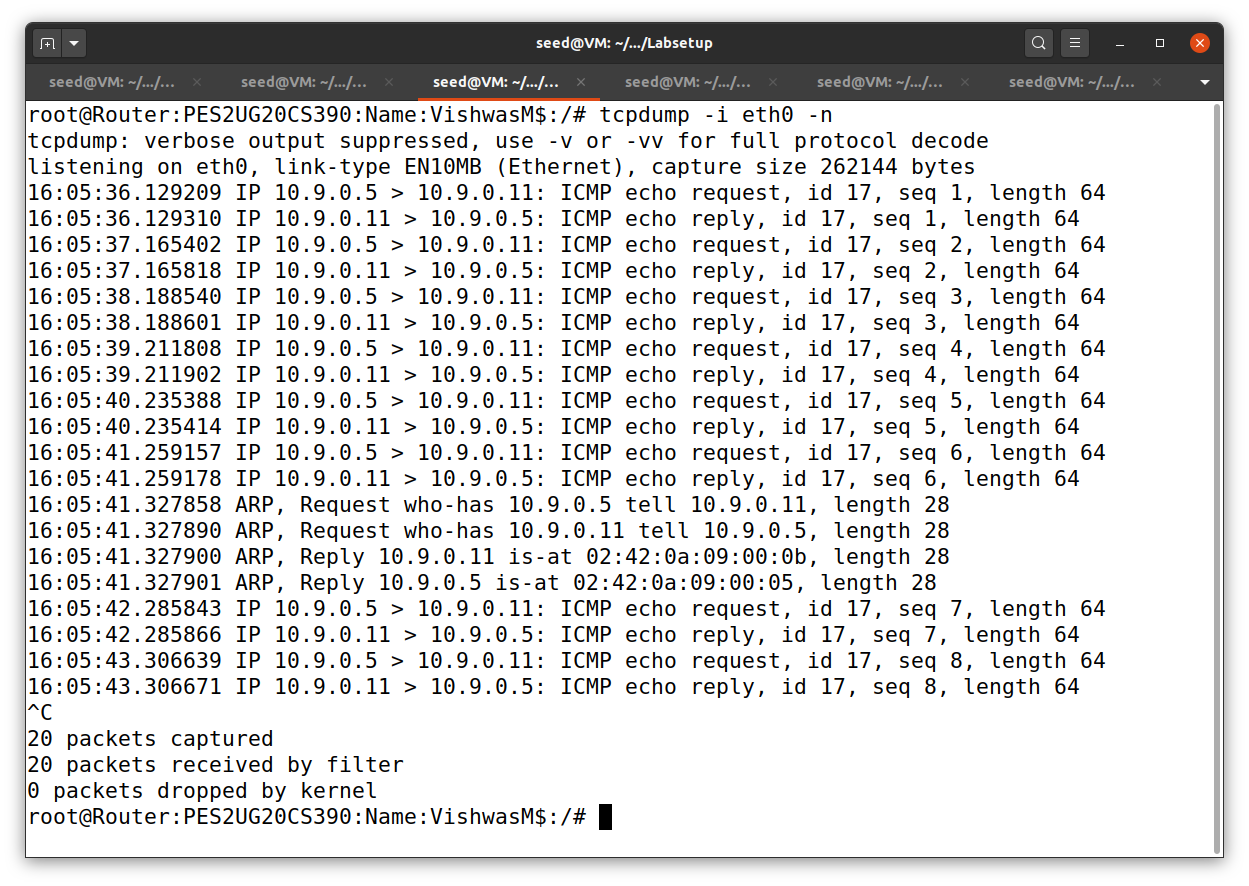
Host U (Client - 10.9.0.5) should not be able to communicate with Host V (host 192.168.60.5)

On Client 10.9.0.5

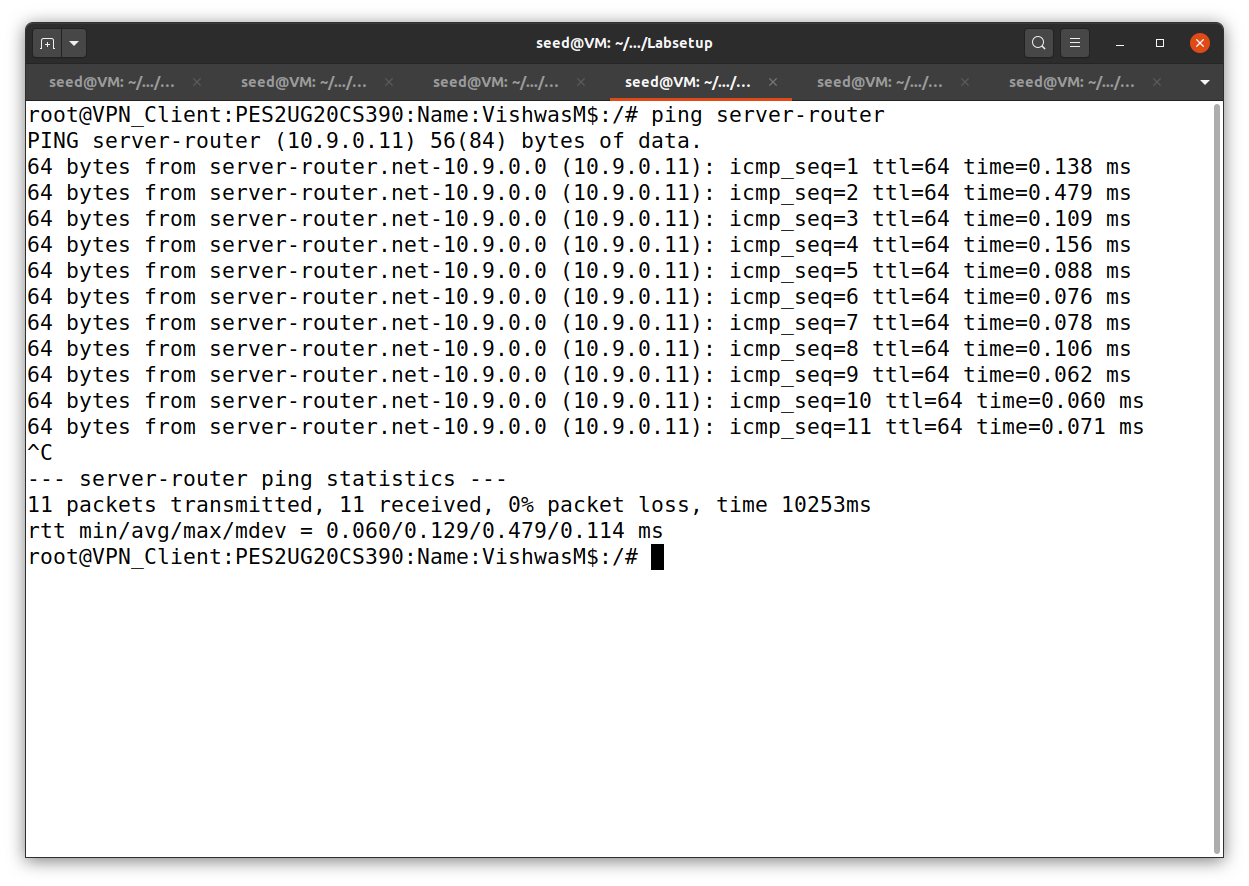


Run tcpdump on the router, and sniff the traffic on each of the networks. Show that you can capture packets.

On server-router run –

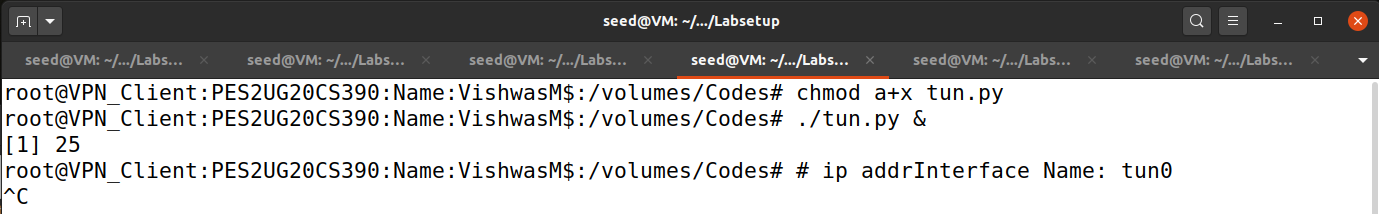


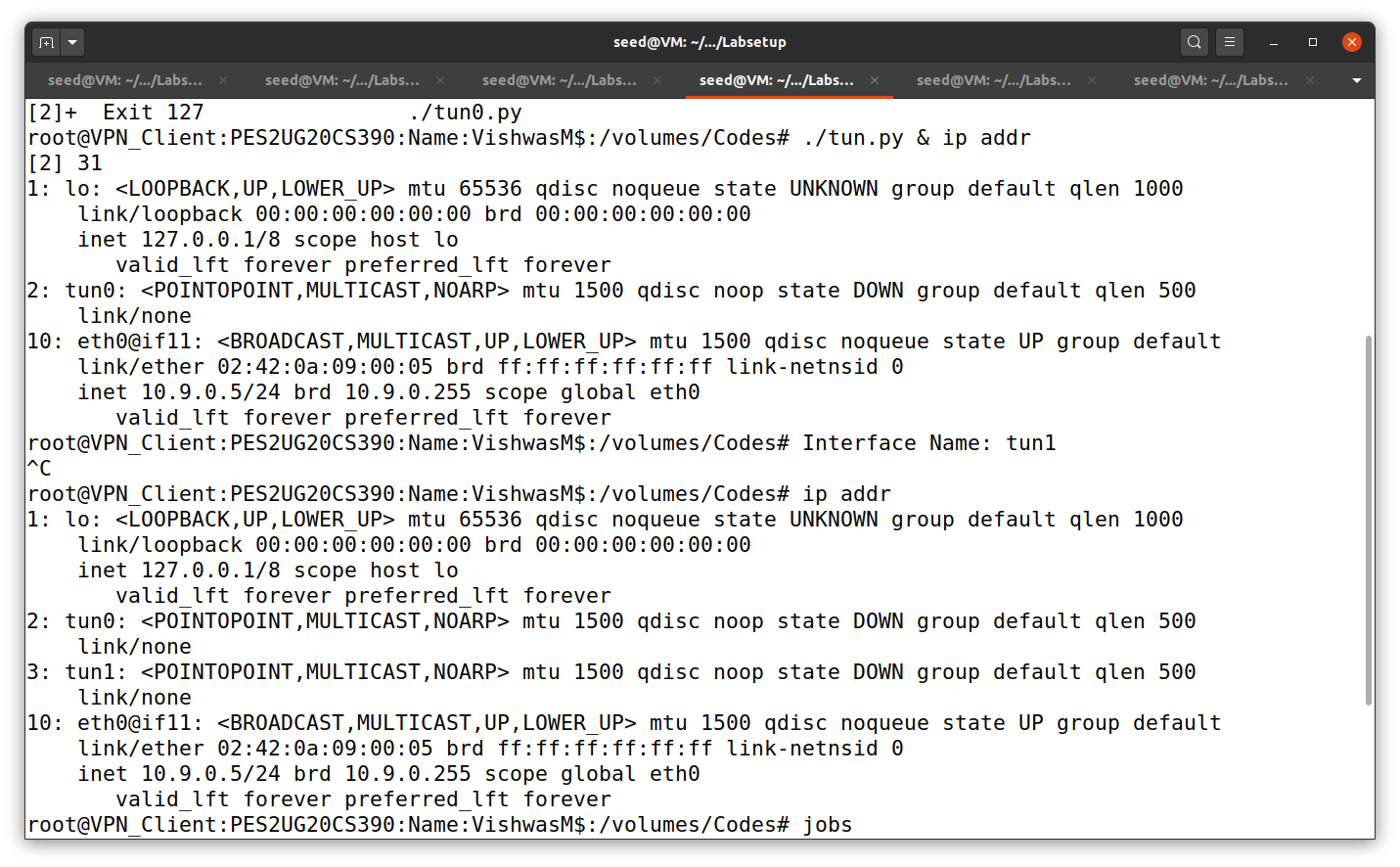
On Client - 10.9.0.5:



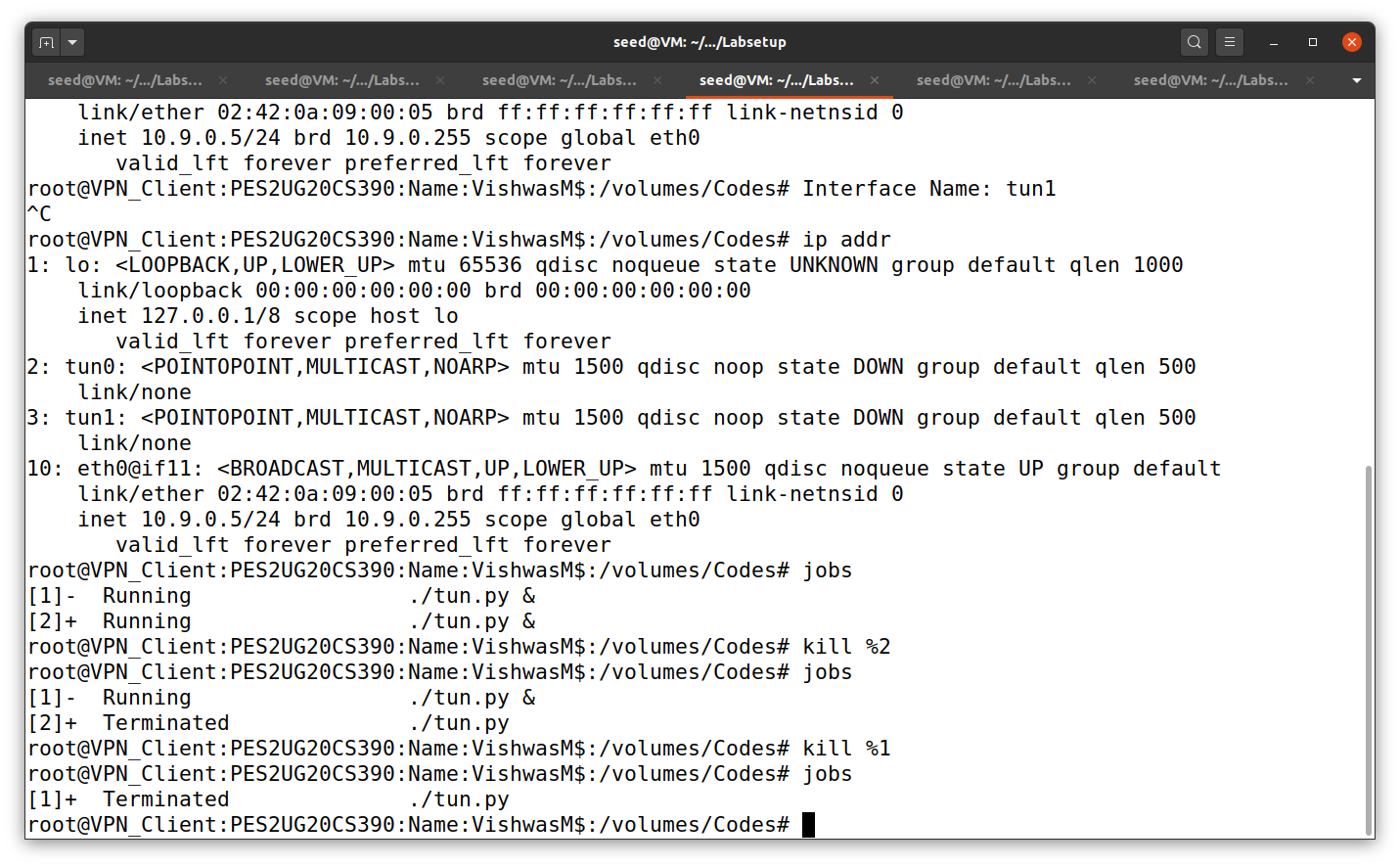
Task2: Create and Configure TUN Interface

Task 2.a: Name of the Interface

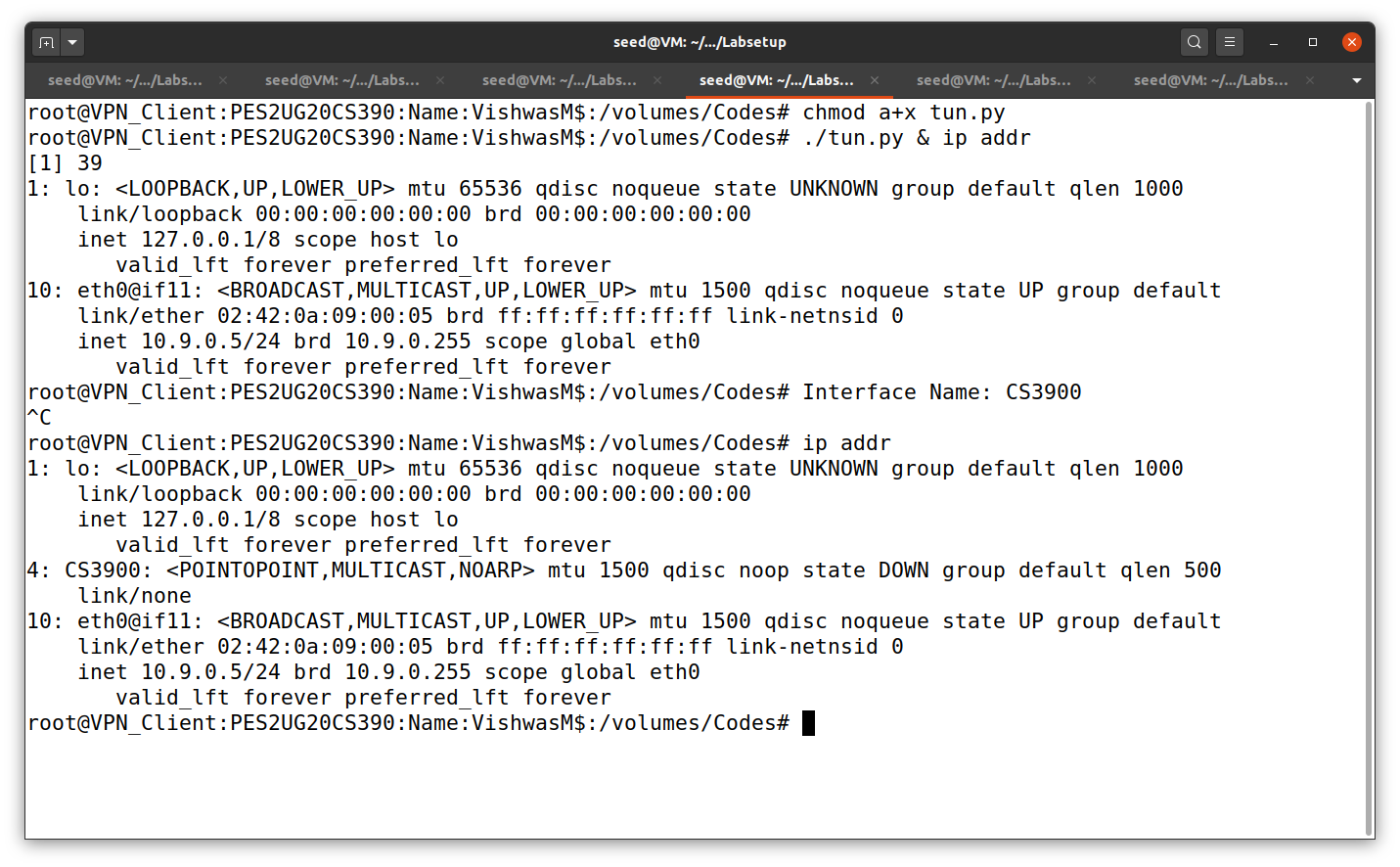




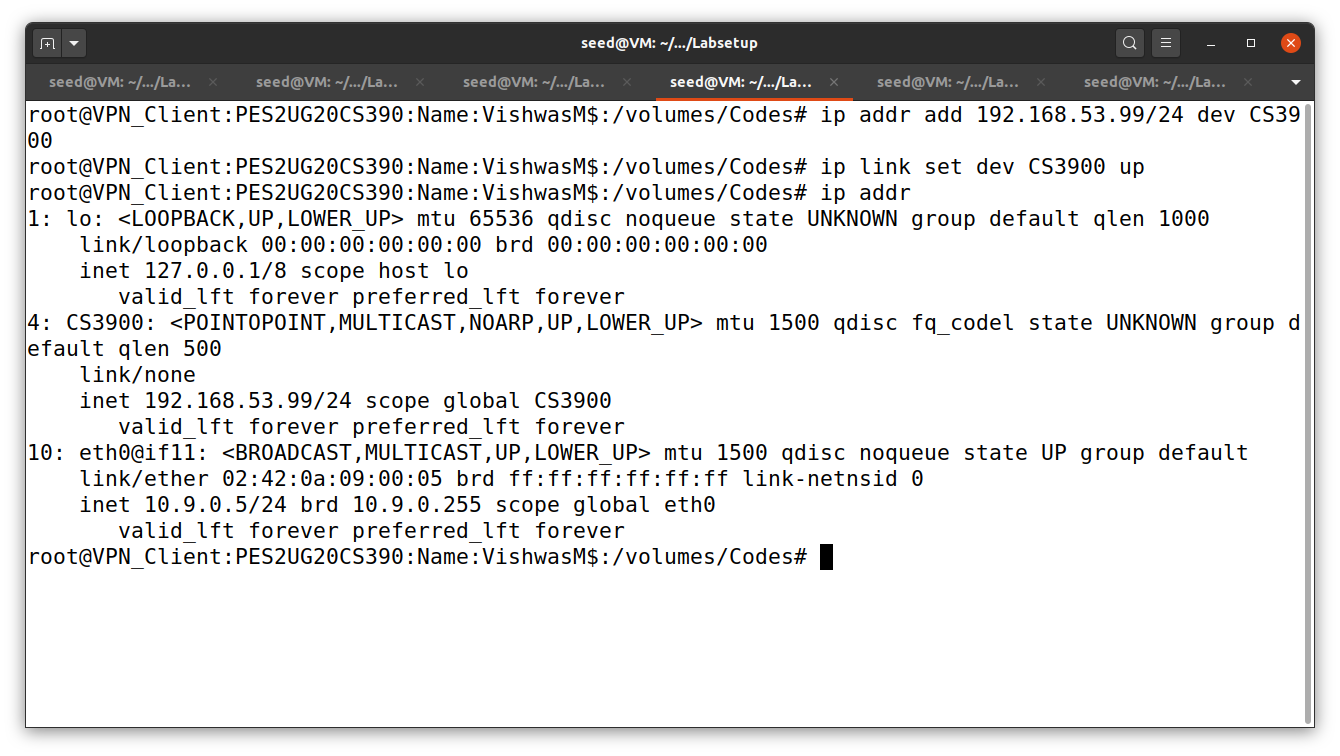
We use the cmd Jobs to check the TUN interface and we use kill command to kill the tunnel process.



Here I am changing the name of the TUN interface as ‘CS390’.



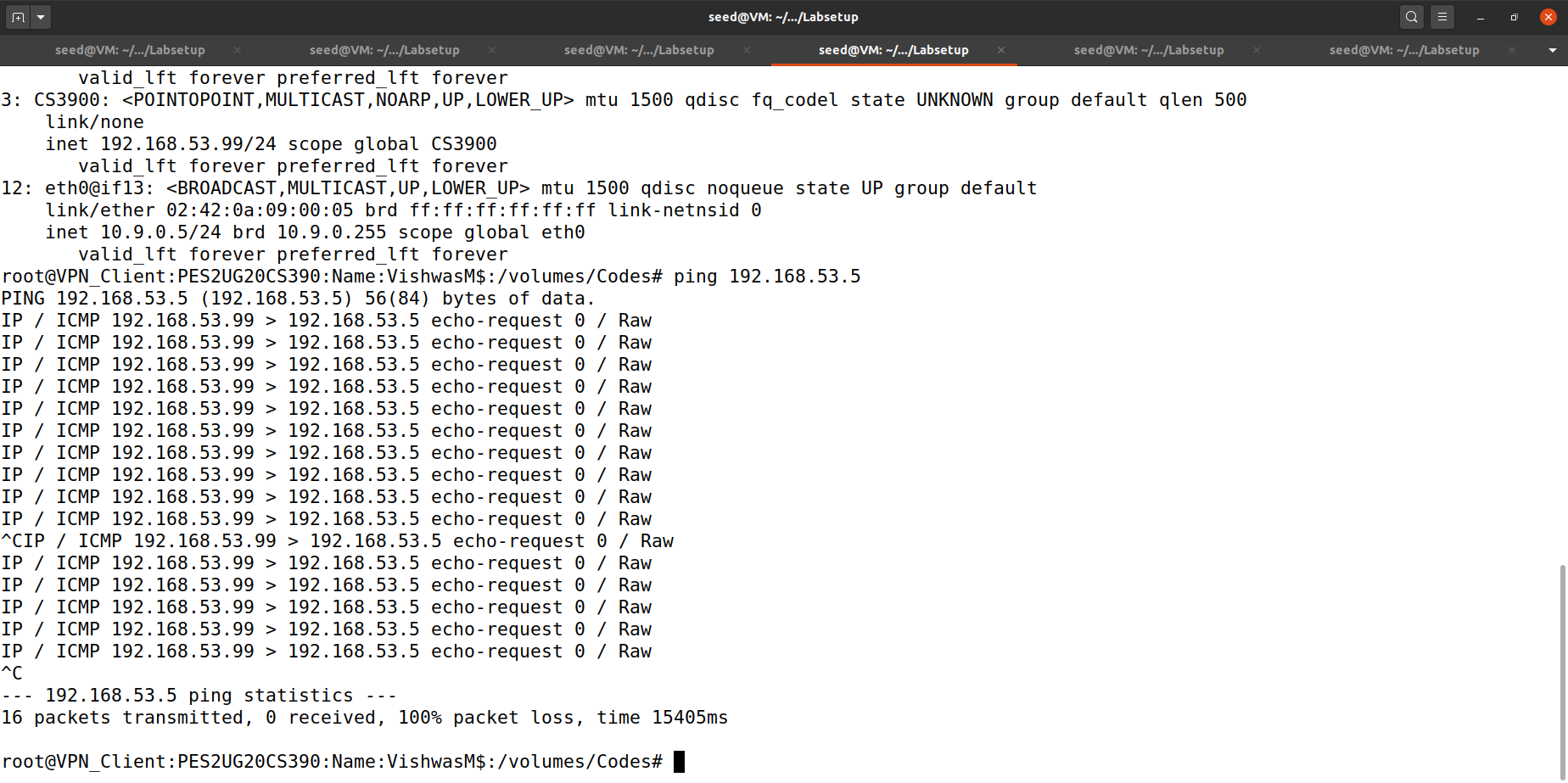
Task 2.b: Set up the TUN interface



First, we assigned an IP address to the TUN interface. Then we need to bring up the interface because it is still in down state.

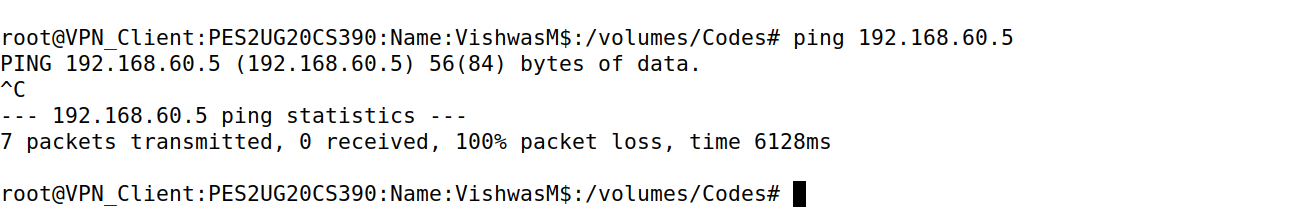
Task 2.c: Read from the TUN interface

When we ping 192.168.53.5:



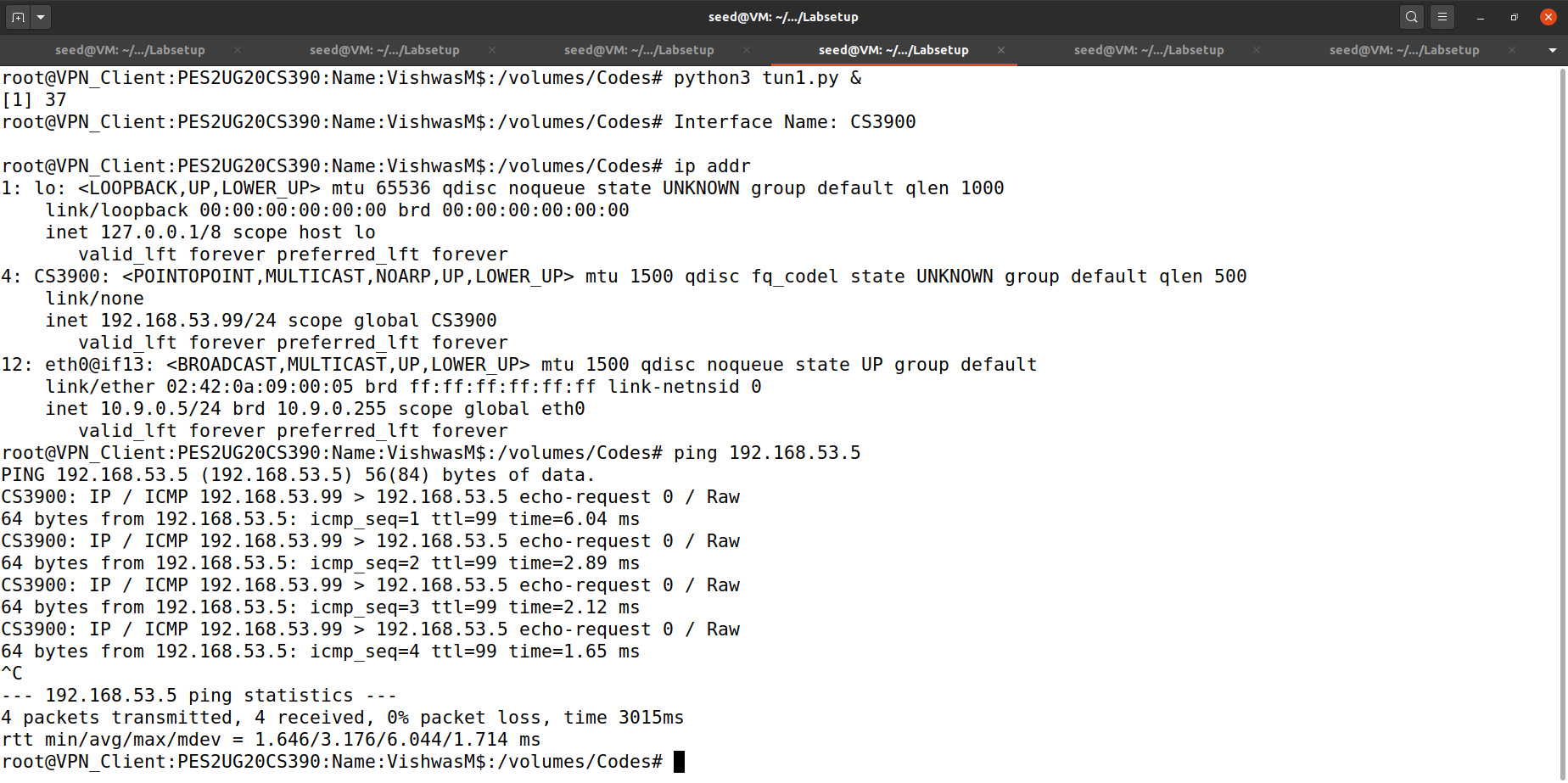
First, we had to change the code in tun.py. Then we had to run the updated python file. Then we are trying to read all the packets coming to TUN interface

When we ping 192.168.60.5:



We don’t get any output when we ping this ip addr because it is still not receiving any packets from the TUN interface.

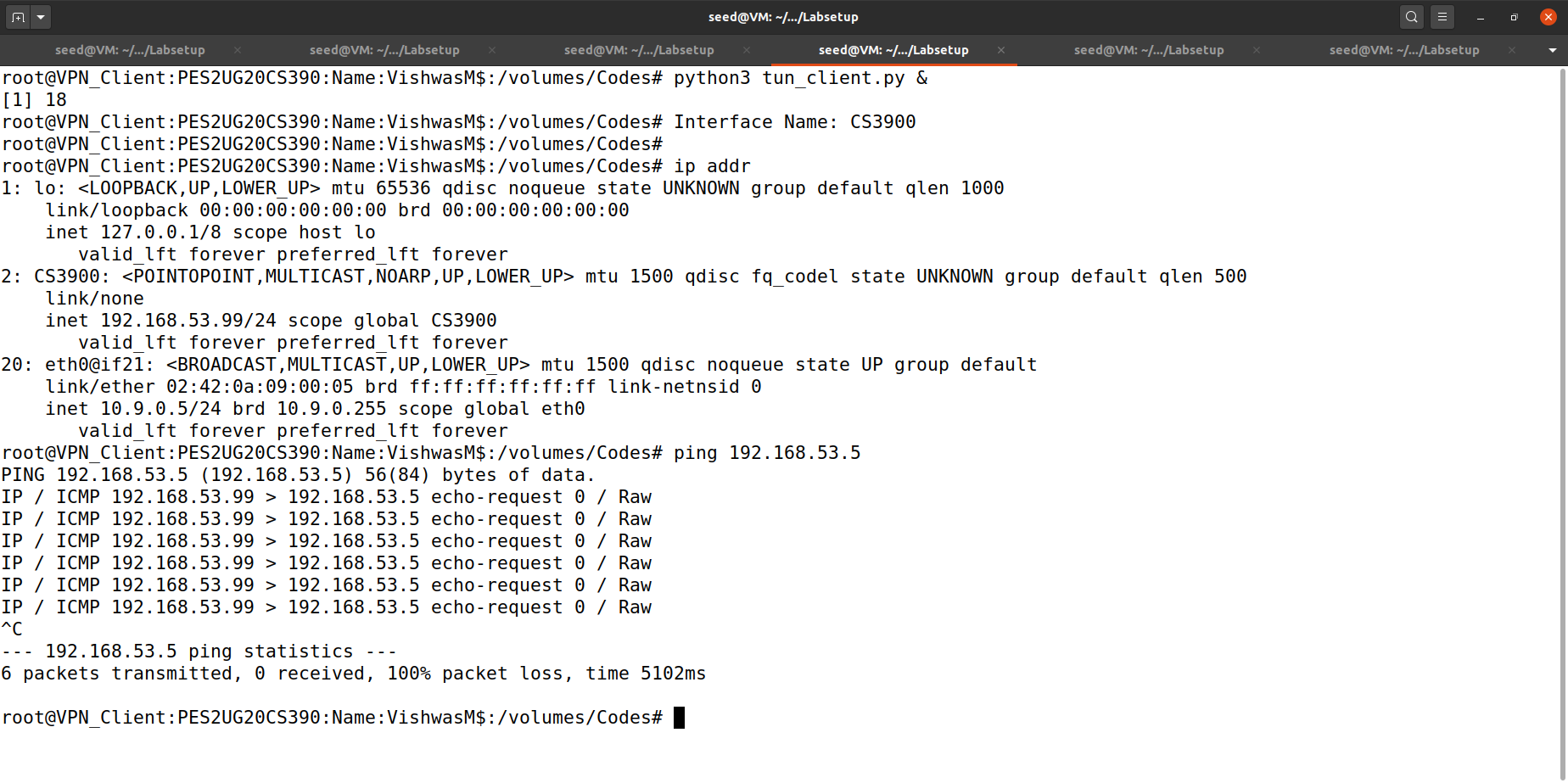
Task 2.d: Write to the TUN interface

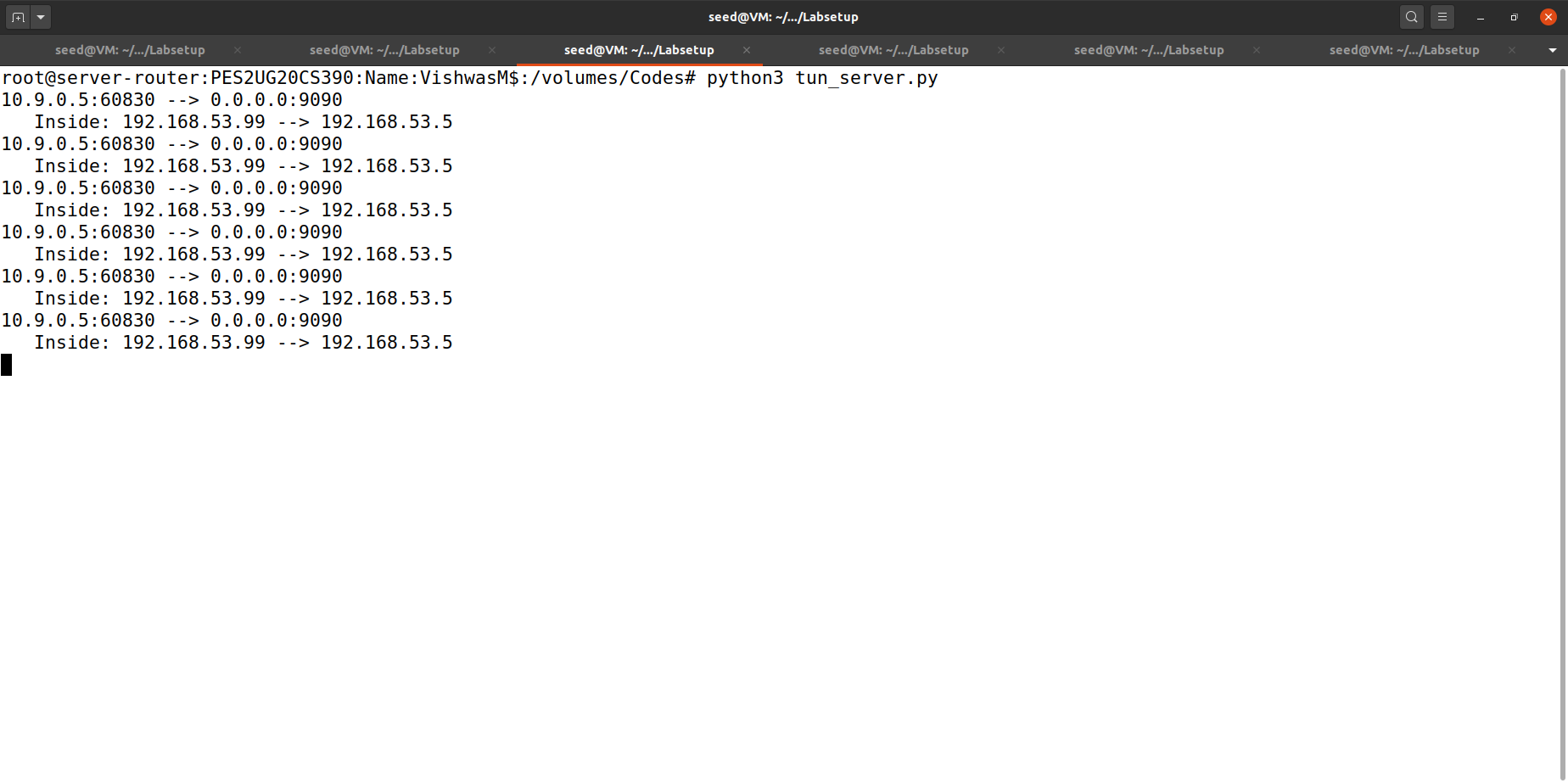


As we can see here both request and reply ICMP packets are seen which means that TUN interface is accepting as well as receiving the packets.

Task 3: Send the IP packet to VPN Server through

a Tunnel



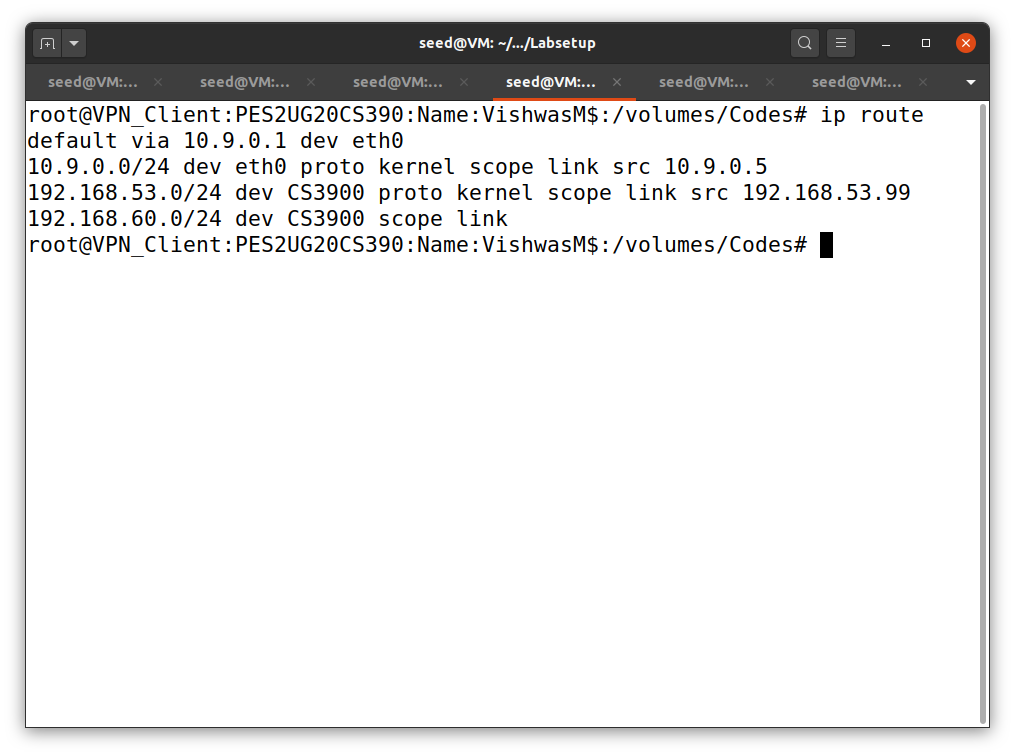
We can see that the packets are 100% lost in the above screenshot.

In the server we can see how the packets are moving inside the tunnel as we can see in the above screenshot.



Here we are pinging to 192.168.60.5.

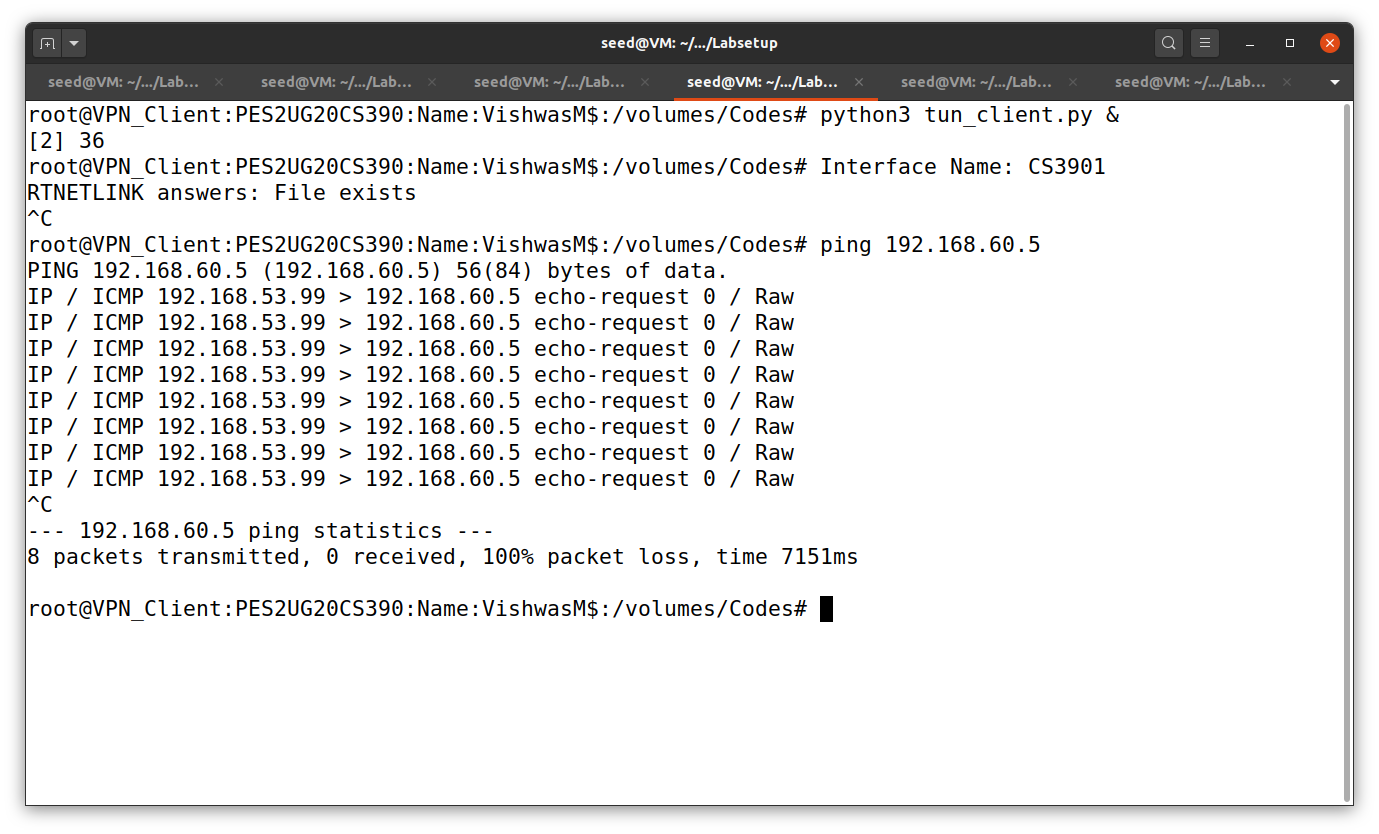




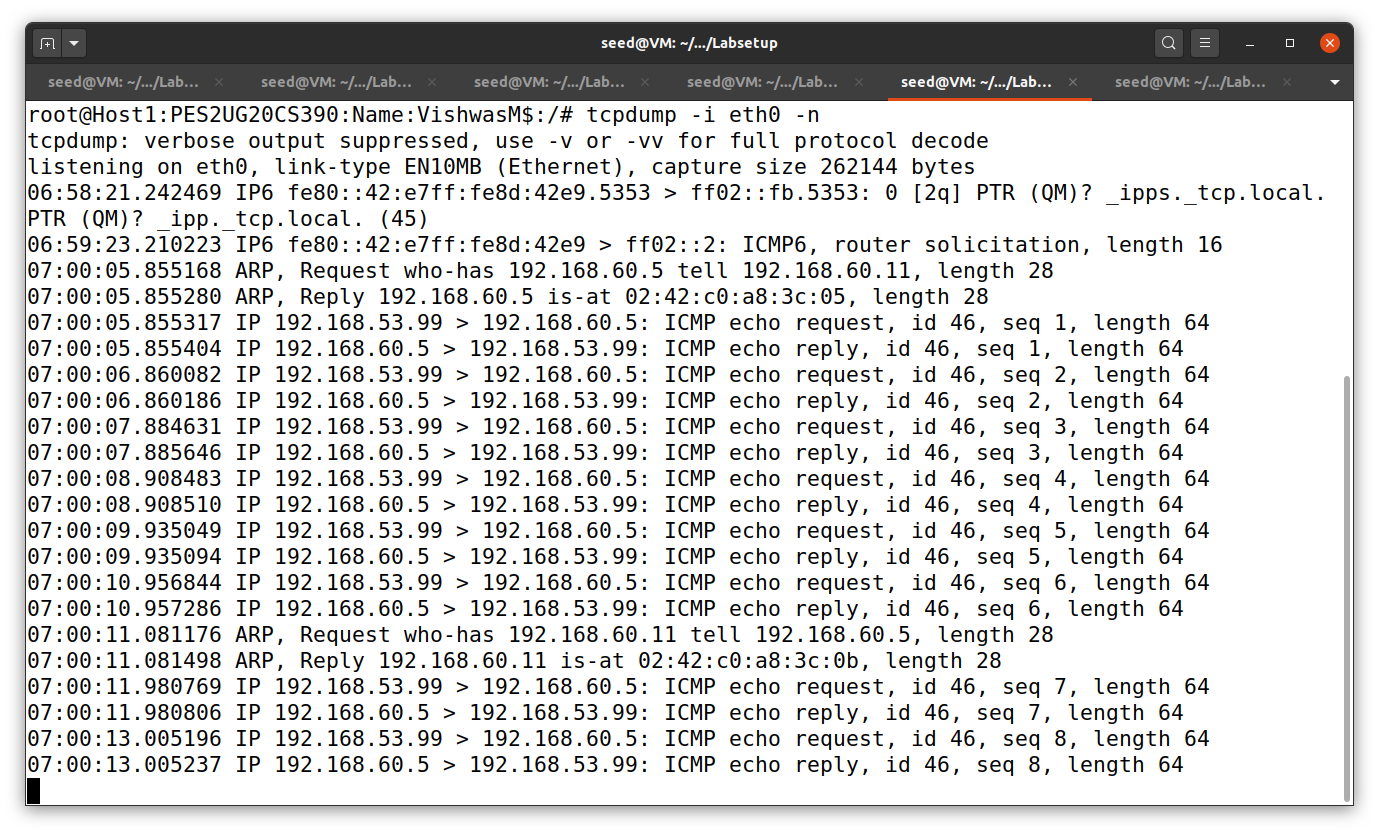
We can see the IP routing routing table with the help of the above command mentioned in the above screenshot.

Task 4: Set up the VPN server

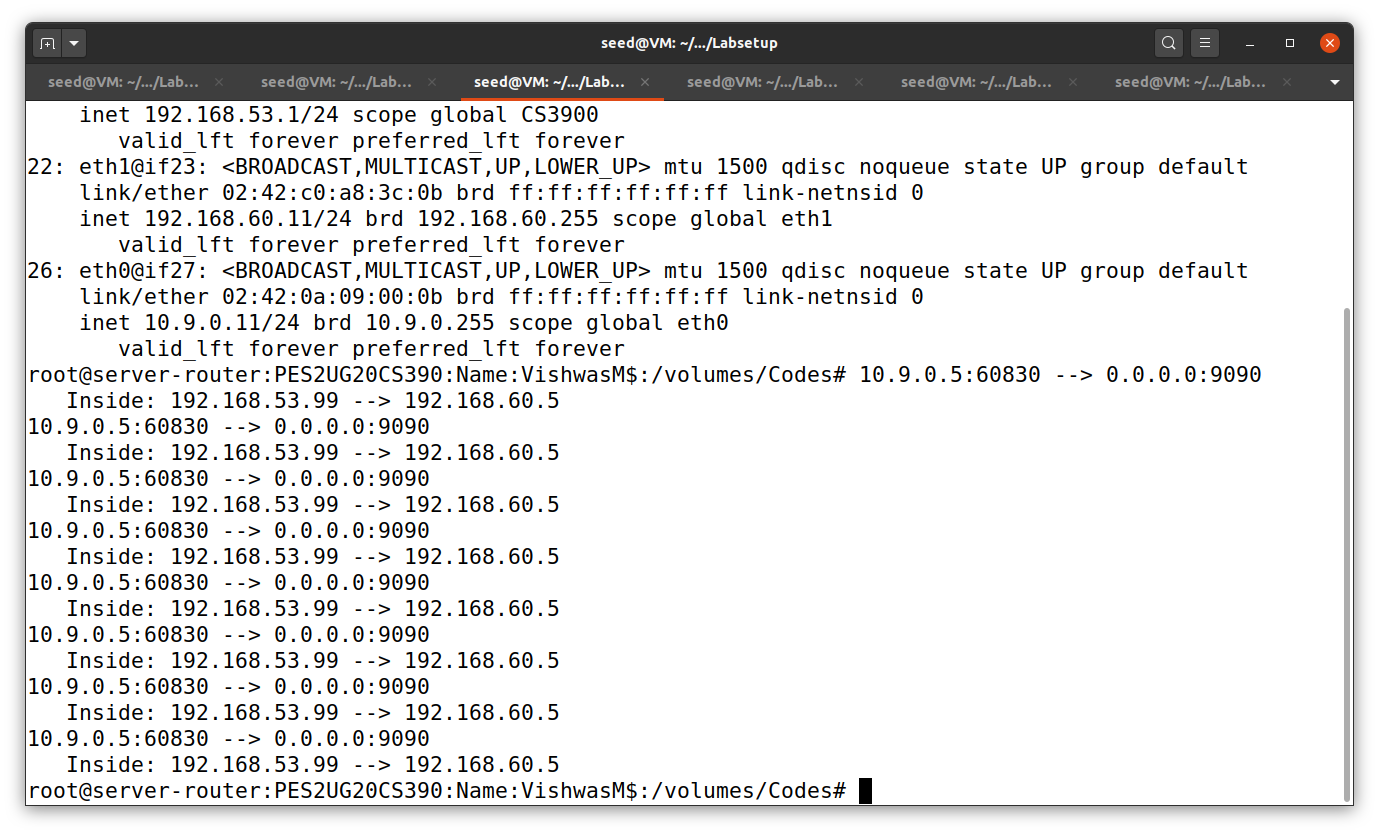
At Client:



At Host1(192.168.60.5):

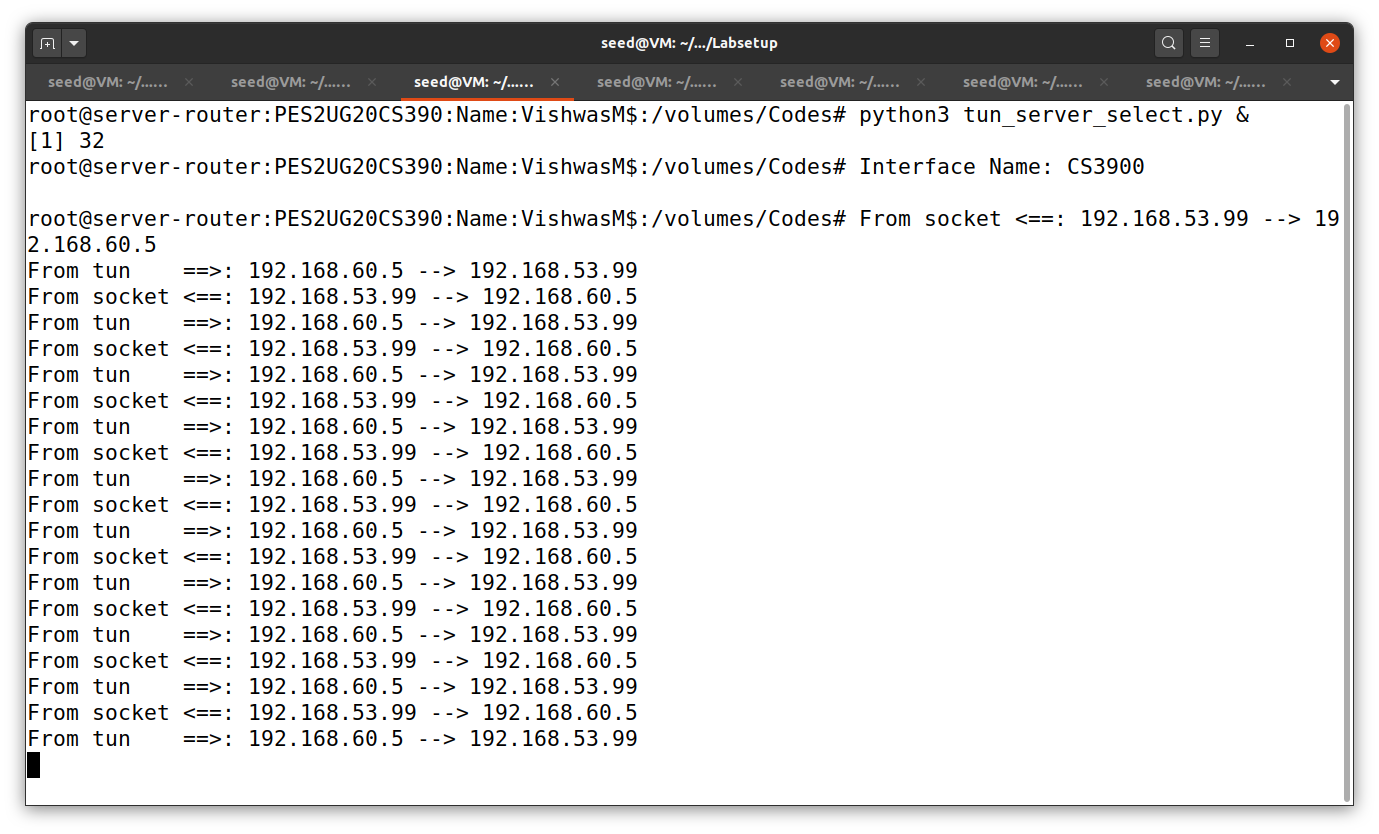


At Serve-Router:

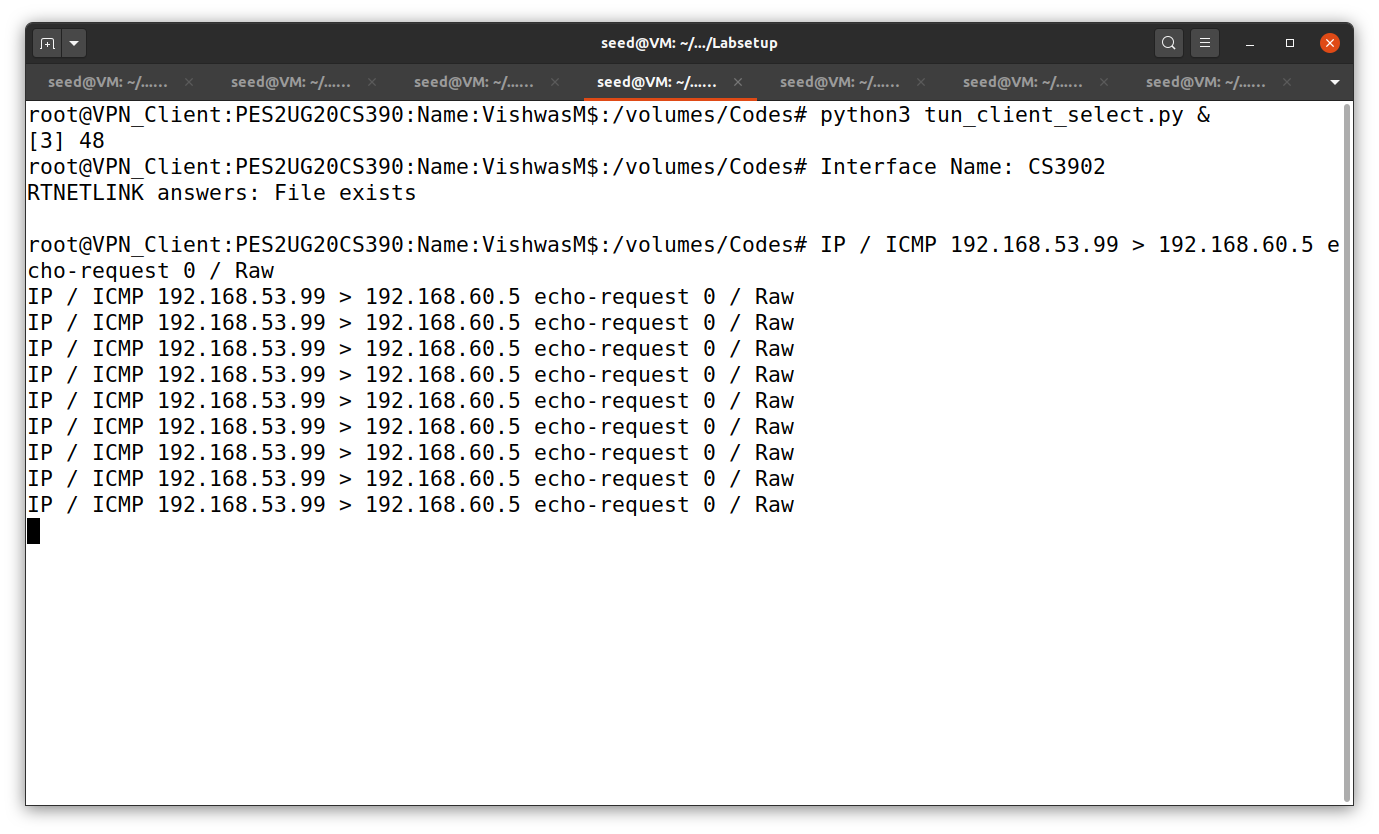


Task 5: Handling Traffic in Both Directions

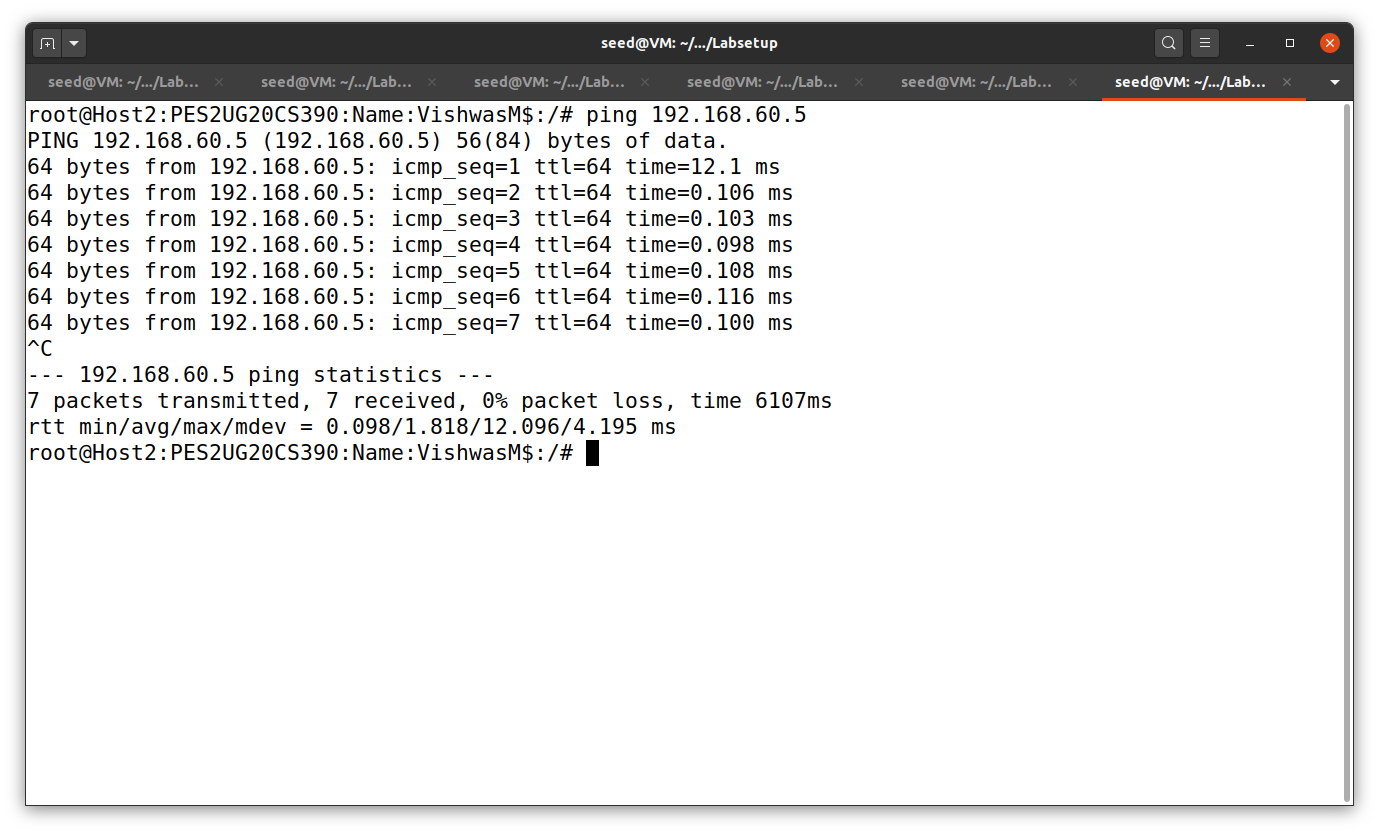
In server-router:



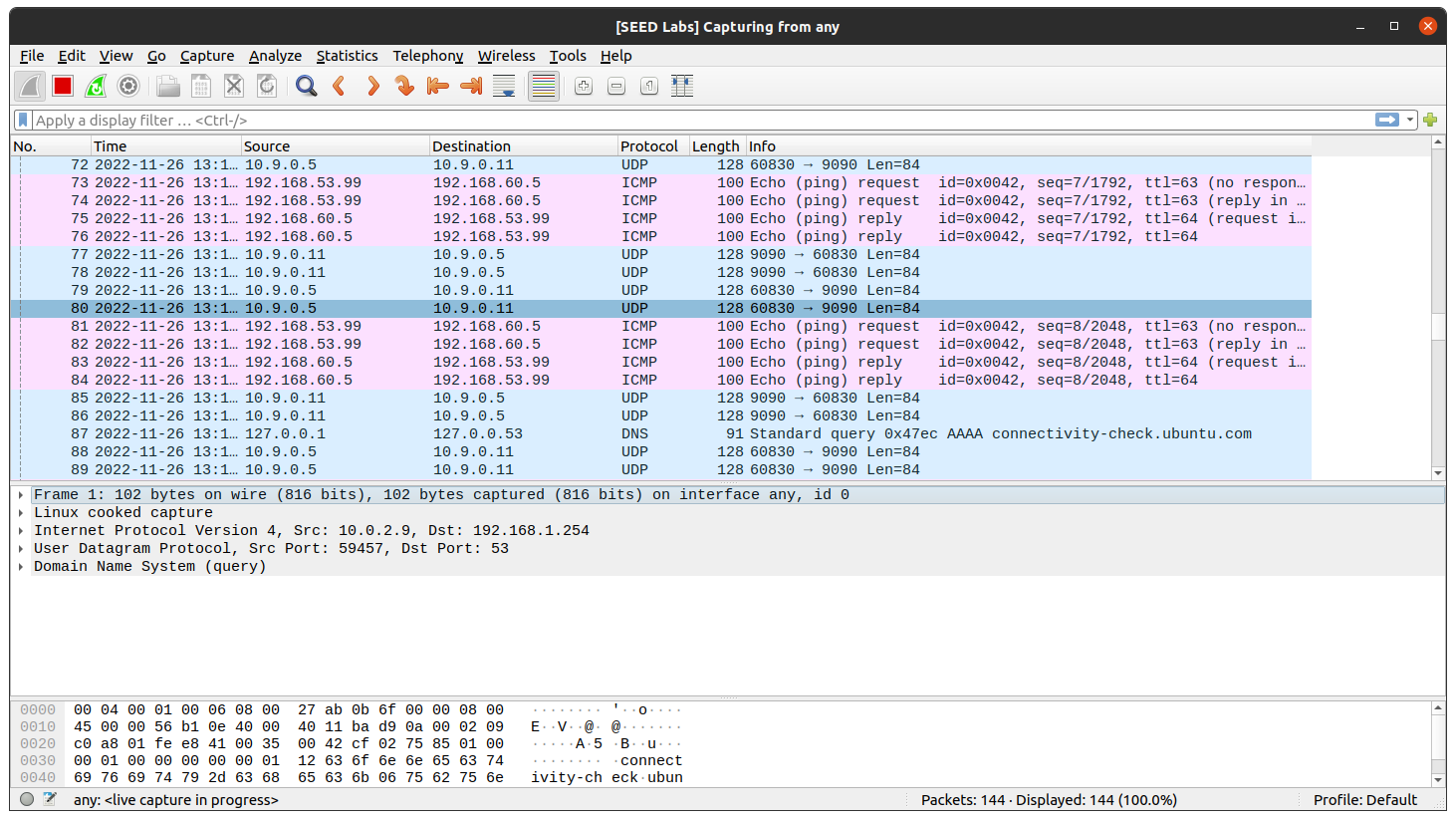
In VPN\_Client:



In another VPN\_Client server:

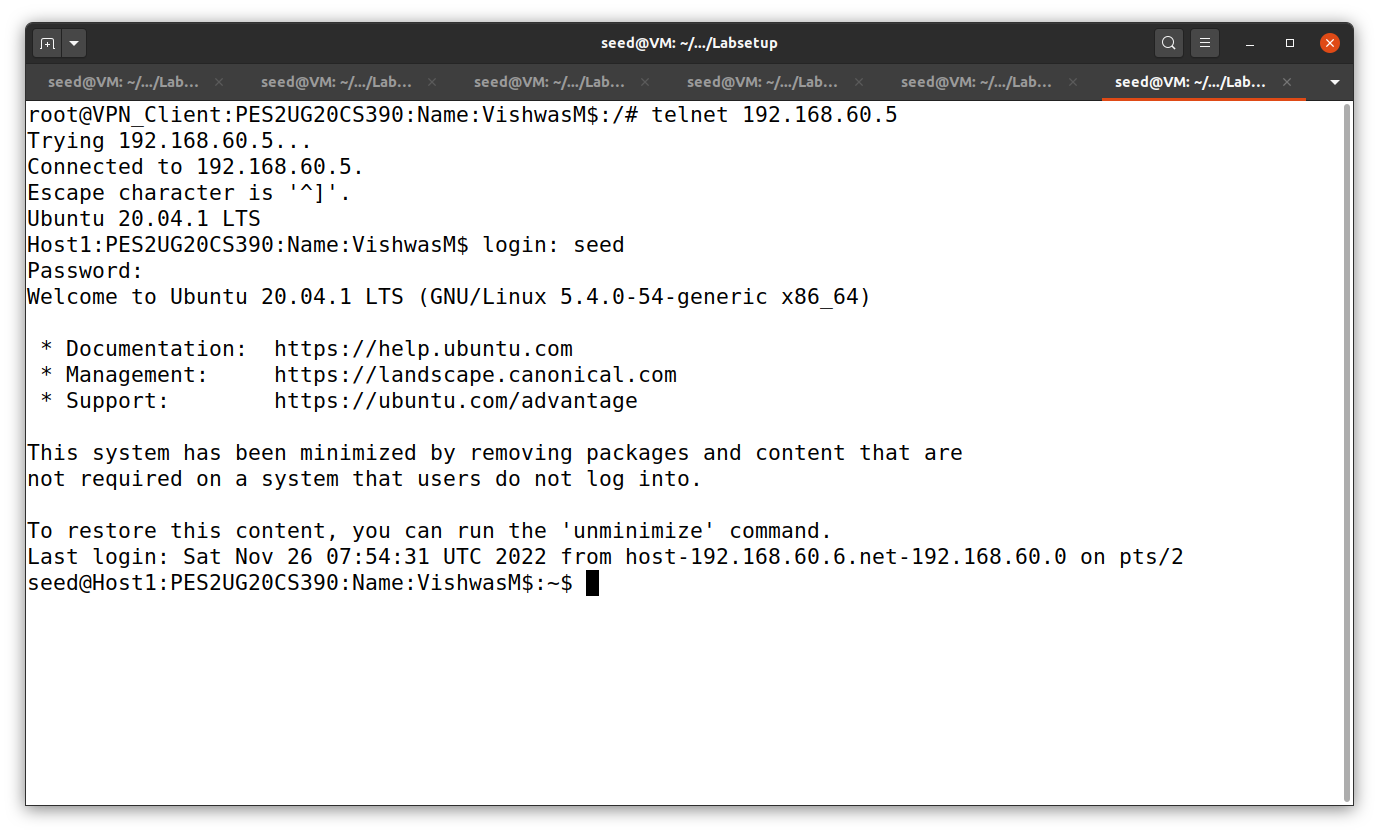


Wireshark capture:

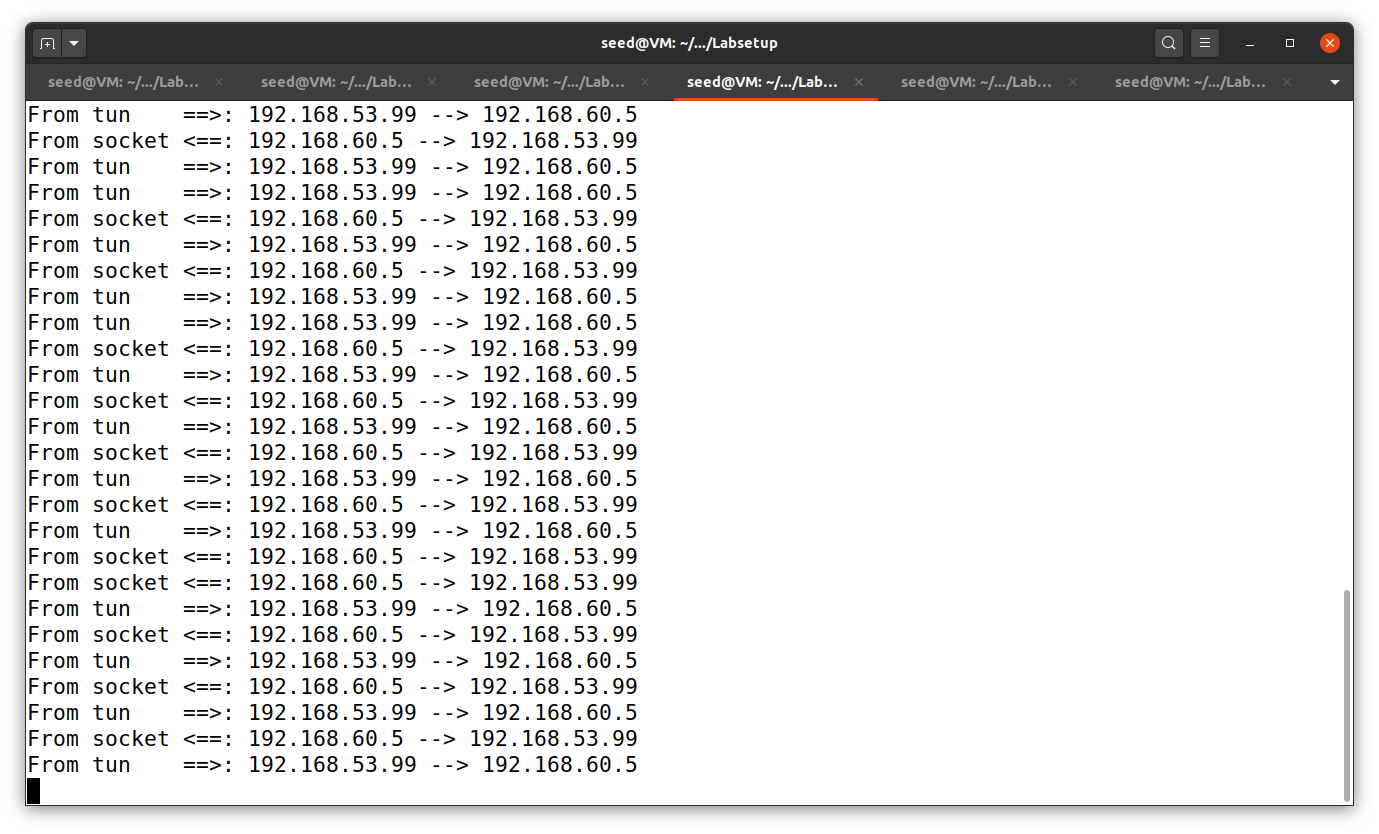


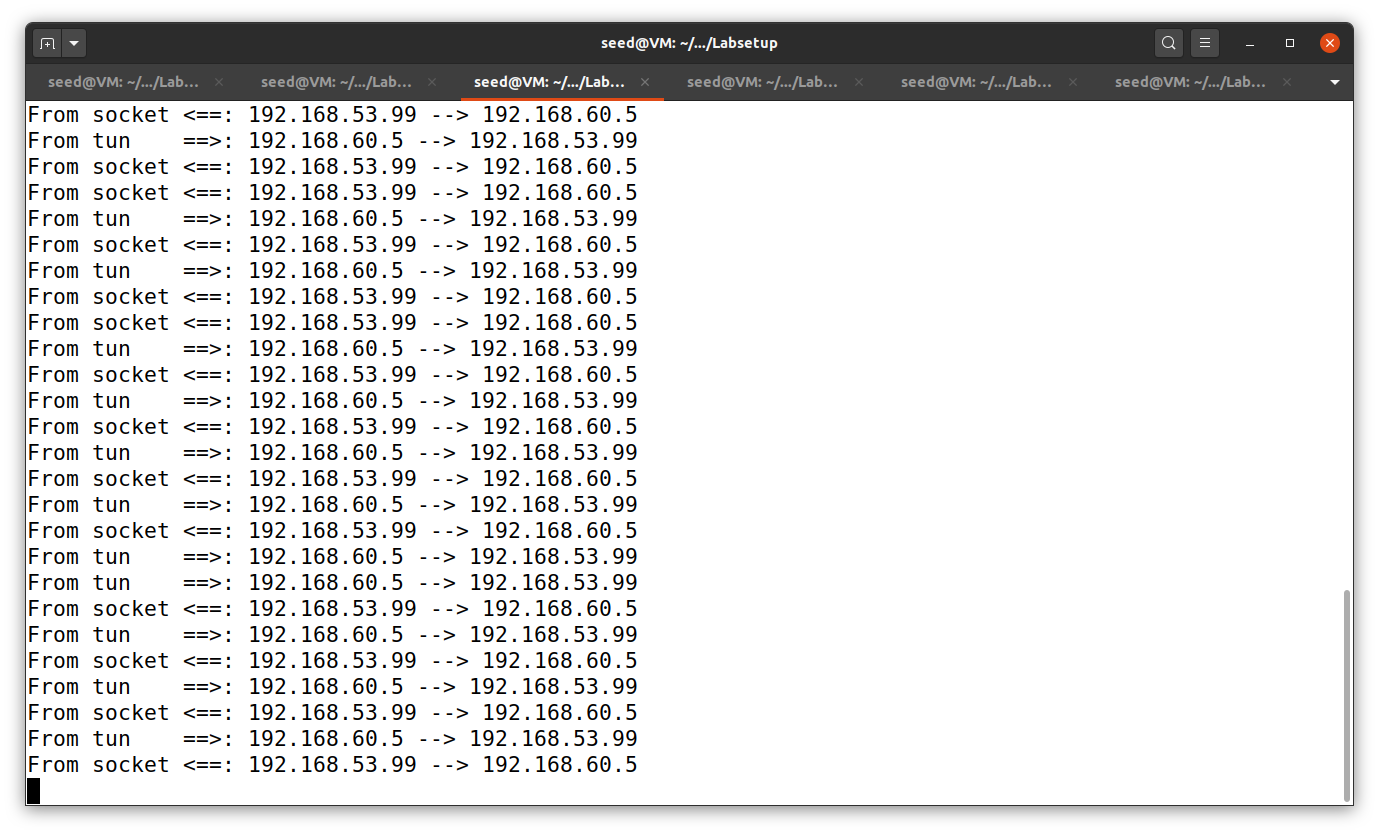
When we do a telnet connection:

In another VPN\_Client where we establish telnet connection:

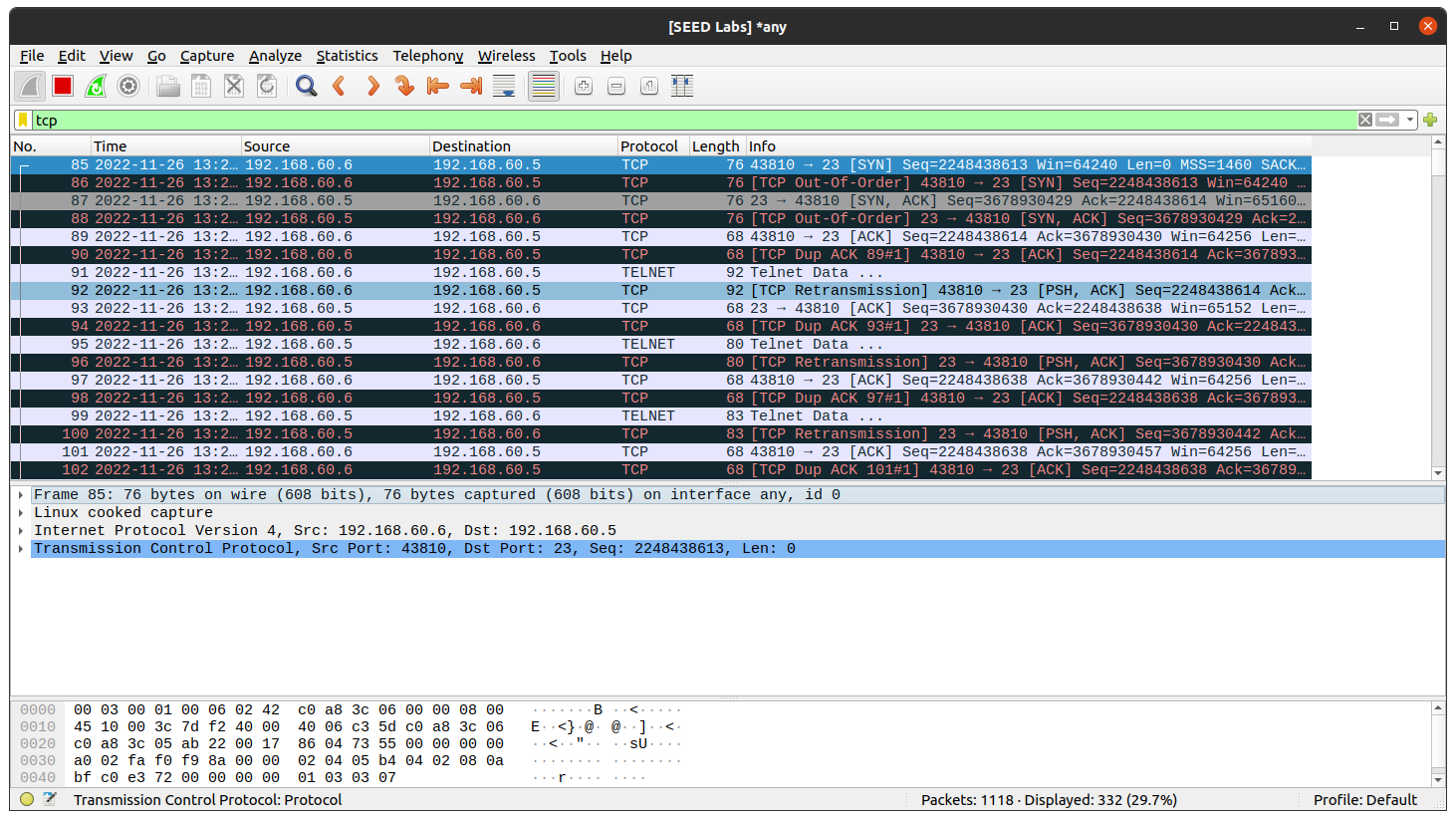


VPN\_Client:



Server- router :  


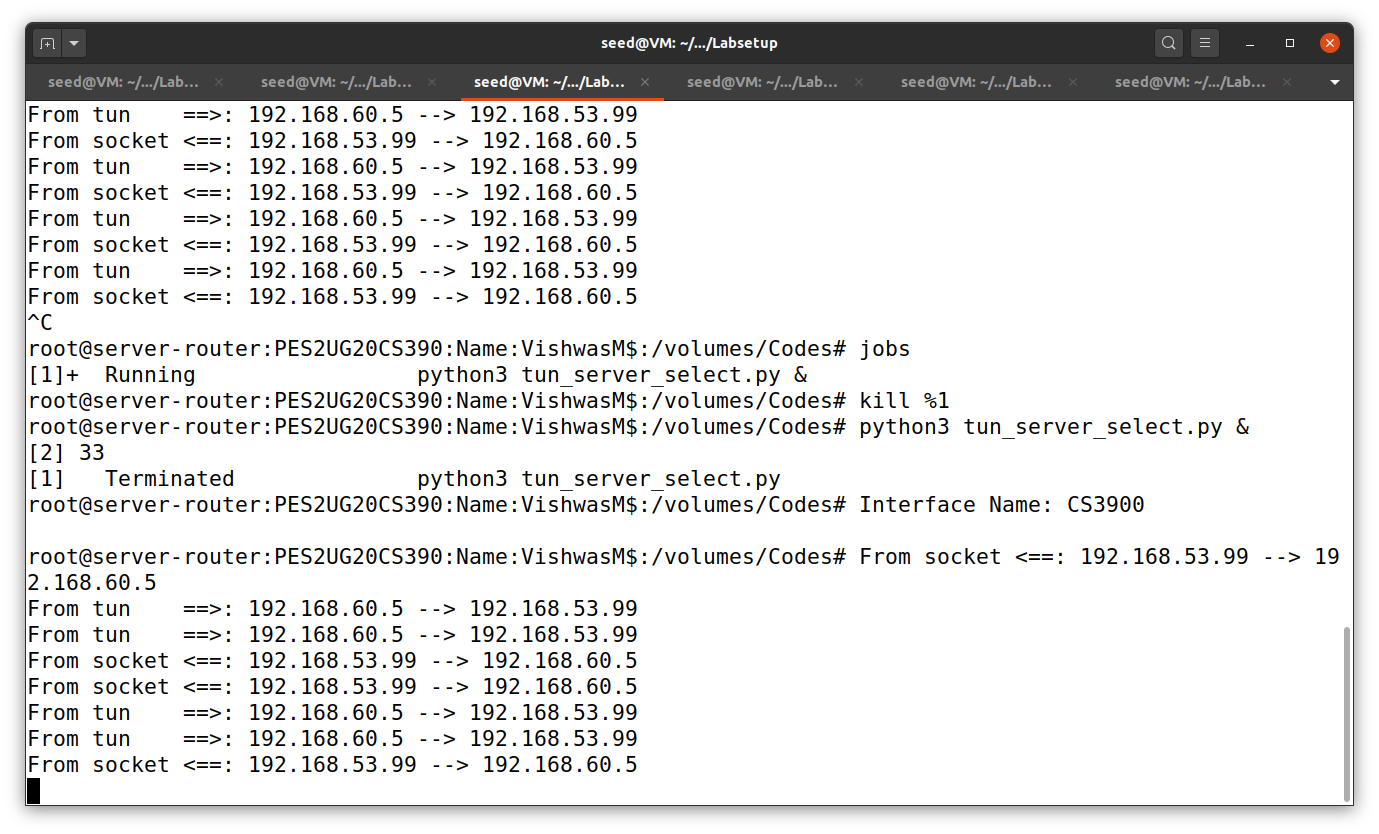
Wireshark Capture:



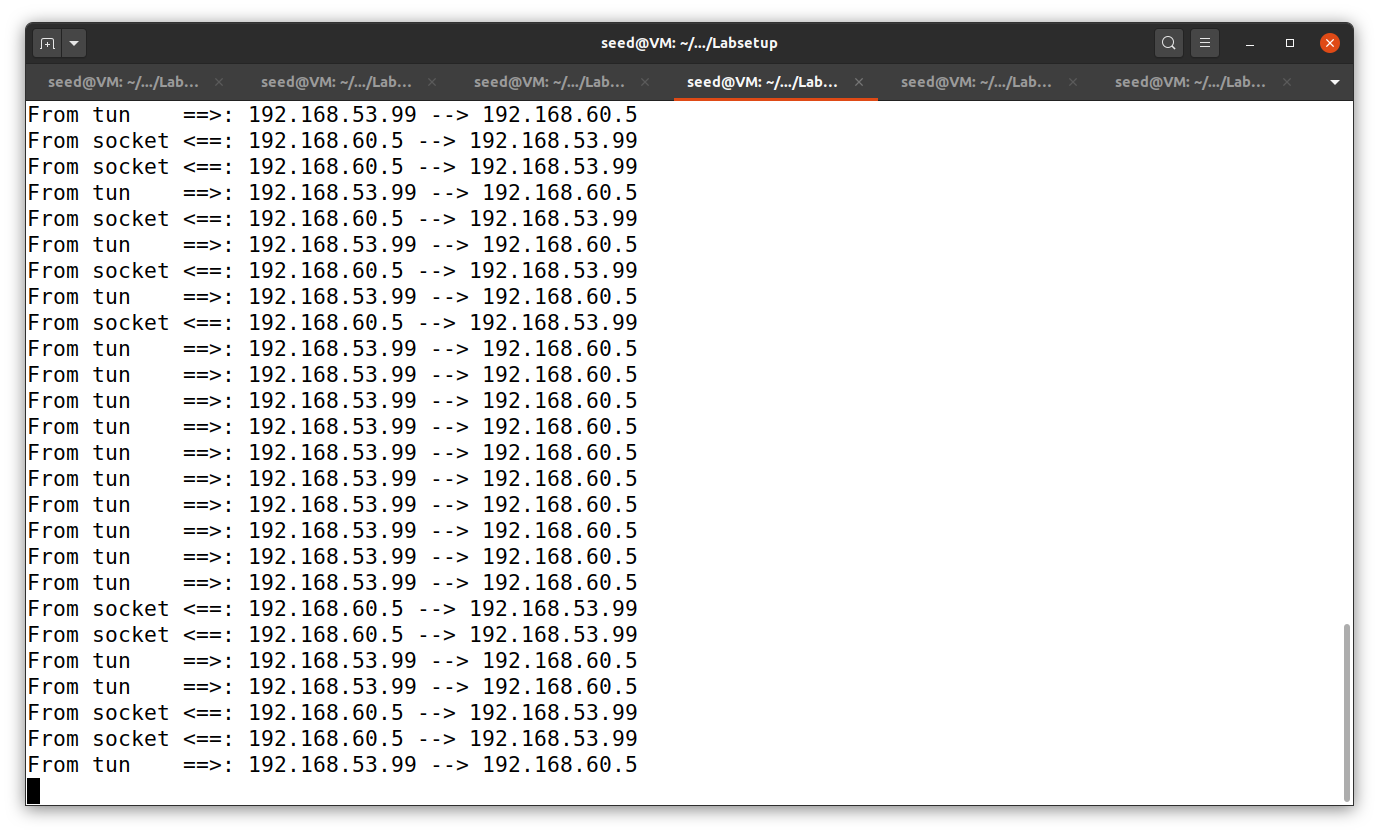
Task 6: Tunnel Breaking Experiment

Here we are breaking the Tunnel connection by pressing ctrl+C. When we try and press something in the client it will work as the screen gets stuck.

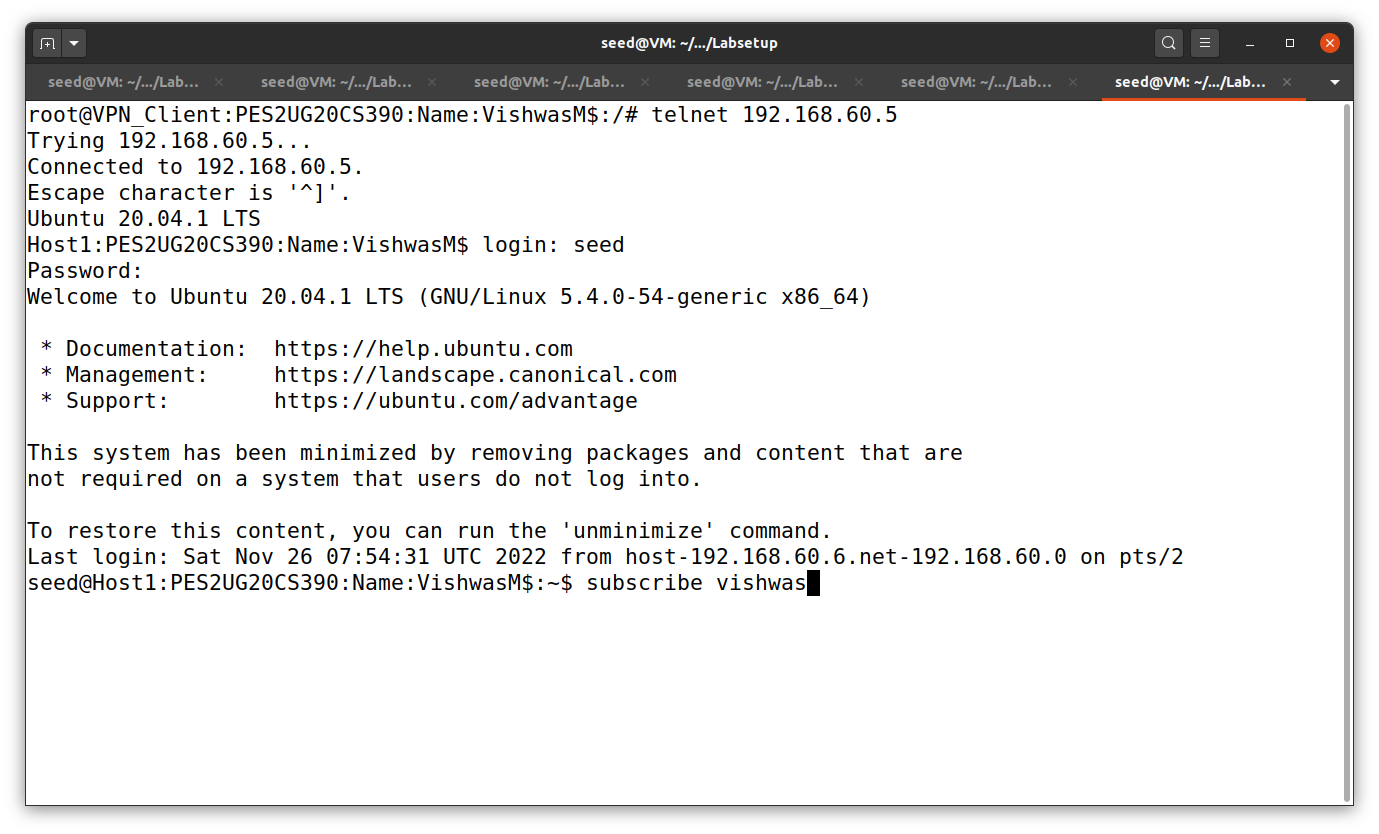
When we connect the tunnel again the word that we typed in terminal gets printed on the terminal which indicates that the tunnel is back and the connection is successful.



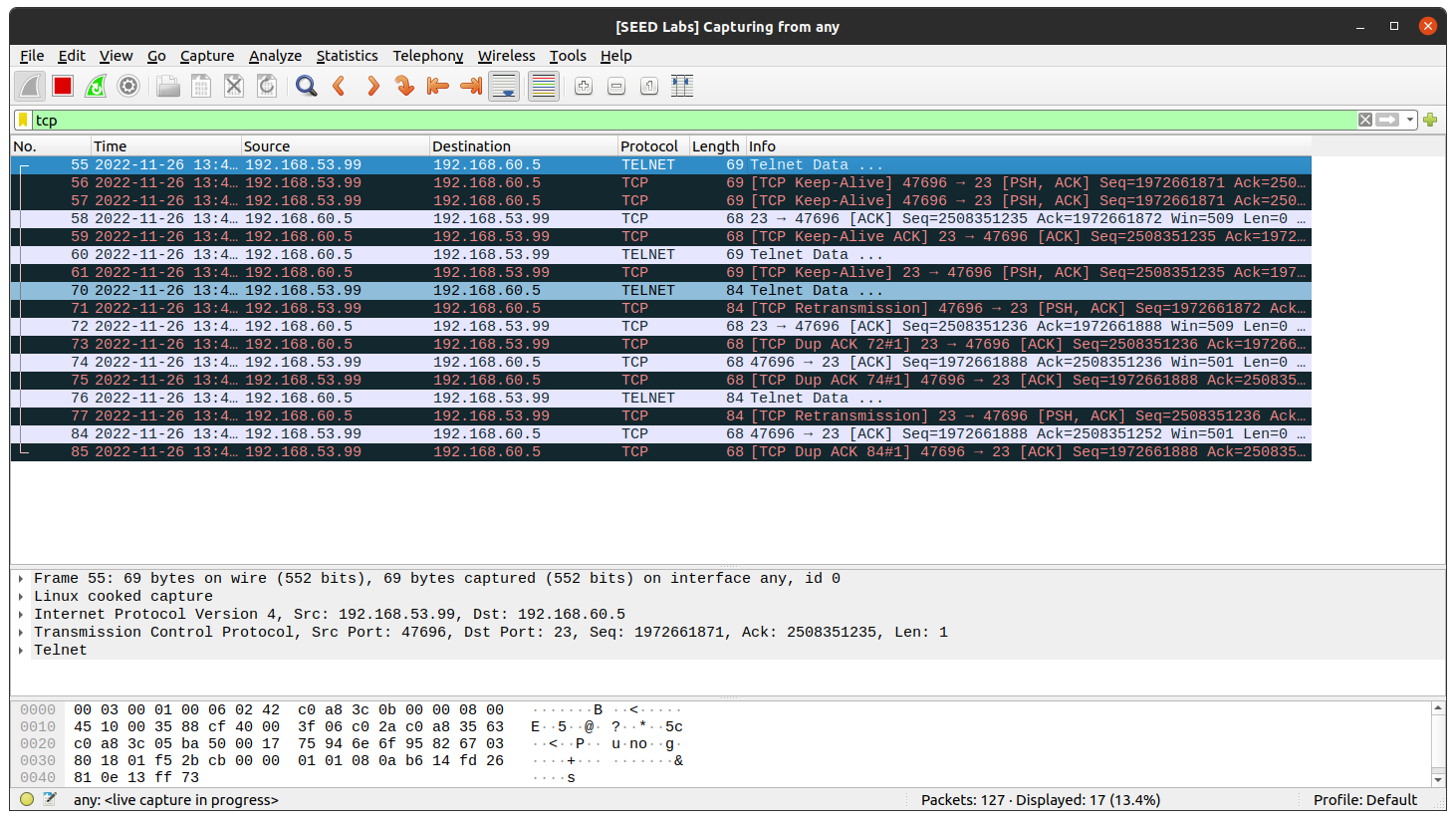
VPN\_Client:



Server-Router:



Wireshark Capture:



Here we can observe that the connection which we just broke and connected in clearly depicted here.