Date: 2021-05-06 17:06 UTC

✱ All sources 5 | Internet sources 5

[0] www.bubuko.com/infodetail-2618075.html
2.4% 11 matches
⊞ 2 documents with identical matches

[3] ahmadfauzanaqil.wordpress.com/2020/06/27/fast-reverse-proxy-frp/
1.3% 10 matches

[4] ahmadfauzanaqil.wordpress.com/2020/06/27/static-vs-dinamic-routing-in-mininet/
1.9% 9 matches

[5] www.poftut.com/add-change-default-route-default-gateway-ubuntu-lnux/
0.0% 1 matches
⊞ 1 documents with identical matches

[7] linuxwiki.de/NetzwerkKonfiguration
0.0% 1 matches

**14 pages, 1018 words**

**PlagLevel: 2.4% selected / 6.8% overall**

16 matches from 8 sources, of which 8 are online sources.

**Settings**

Data policy: *Compare with web sources, Check against my documents*

Sensitivity: *Medium*

Bibliography: *Consider text*

Citation detection: *Reduce PlagLevel*

Whitelist: --

KLE Technological University

# COMPUTER NETWORKS-2

## Project on Virtual Private Network

Submitted in partial fulfillment of the requirement for the degree of

Bachelor of Engineering in

Computer Science and Engineering

Submitted By: Team A2-4

| | | |
|---|---|---|
| Apoorva Jinde | : | 01FE18BCS044 |
| Ashutosh Chauhan | : | 01FE18BCS051 |
| Ashutosh Pratap Singh | : | 01FE18BCS053 |
| Atharv Paramane | : | 01FE18BCS055 |

Under the guidance of

Ms Vijaylakshmi Mam

SCHOOL OF COMPUTER SCIENCE & ENGINEERING
HUBLI – 580 031 (India).

Academic year 2020-2021

| S.NO | Content |
|---|---|
| I | Introduction |
| | i. Problem Statement |
| | ii. Objectives |
| II | Component Requirements |
| | i. Hardware component Requirements |
| | ii. Software component Requirements |
| III | Topology |
| | i. Miniedit Visualization of Topology |
| | ii. Understanding of Topology |
| IV | Implementation |
| | i. Configuring the PPTP server |
| | ii. Configuring the PPTP clients |
| V | Screenshots |
| VI | Result Analysis |
| VII | Conclusions |
| VIII | References |
| I**X** | Drive Link |

# I. INTRODUCTION

## i. Problem Statement

"The goal is to demonstrate and design the VPN connection so that a client connected to a private network and a client connected to a public network can communicate to a server connected to a private network. Here the type of VPN connection taken into consideration is PPTP."

A VPN is a private network connection between two points over the internet. VPN stands for virtual private network, which is pretty self explanatory. It's creating a virtual dedicated connection over the internet instead of requiring a real dedicated connection for your network.

When you connect through a VPN, the data packets you send are first encrypted before the public network even sees them. The encrypted data is sent to the VPN server, which then forwards it to its destination. The VPN server ensures your anonymity because your IP address is hidden and the transmission transmits the address of the VPN server instead.

## ii. Objectives

a. Establish the connection between a client connected to private network and a server connected to a private network

b. Establish the connection between a client connected to public network and server connected to a private network

c. Provide a secure communication between the server and the clients using authentication.

## II. COMPONENTS RE**Q**UIREMENTS

### i. Hardware component requirements

a. RAM : 2 GB

b. Hard Disk : 20 GB

### ii. Software component requirements

a. Operating system: Ubuntu 16.04

b. Mininet

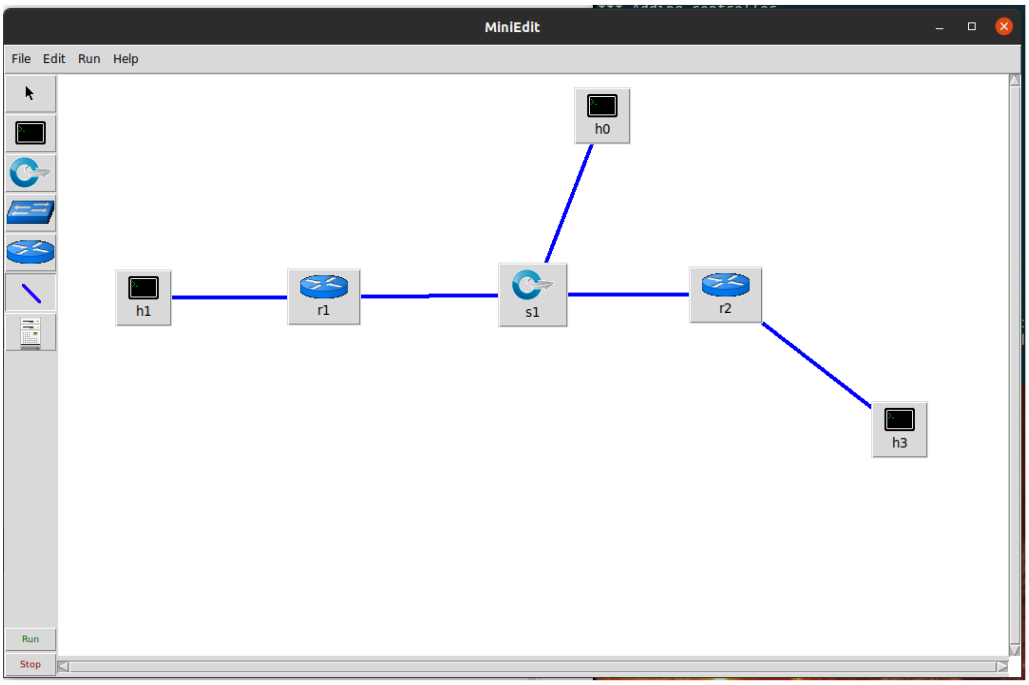
c. Mininet Dependencies + Core files

d. PPTP server installation

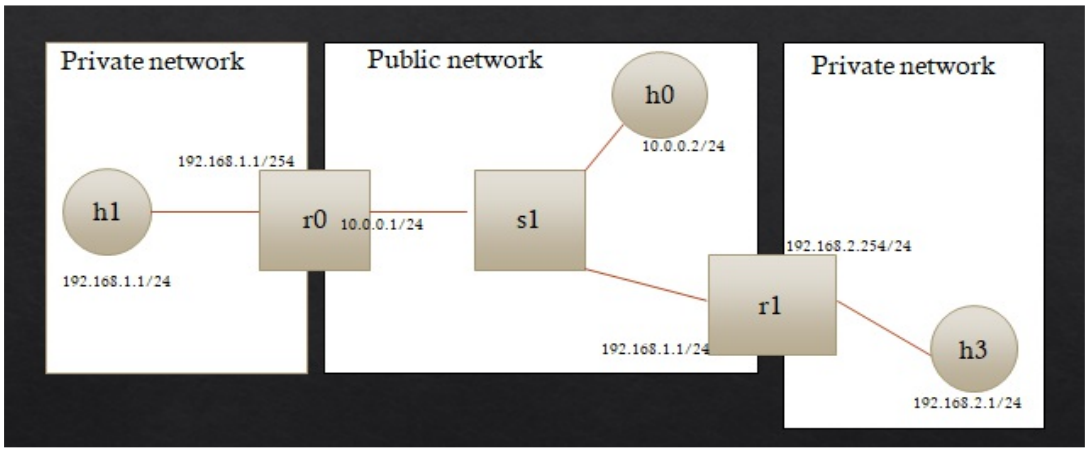
e. PPTP client installation

## III. TOPOLOGY

### i. Miniedit Visualization of Topology




### ii. Understanding of Topology



We have created a topology consisting of two private networks and one public network. There are a total of two routers, one switch and three hosts. IP addresses are assigned according to the network.

Our topology extracted in form of python file.

```
#!/usr/bin/env python
from mininet.cli import CLI
from mininet.net import Mininet
from mininet.link import Link,TCLink,Intf
if '__main__' == __name__:
  net = Mininet(link=TCLink)
  h1 = net.addHost('h1',ip="192.168.1.1/24")
  h2 = net.addHost('h2',ip="10.0.0.3/24")
  h3 = net.addHost('h3',ip="192.168.2.1/24")
  s0 = net.addHost('s0')
  r0 = net.addHost('r0')
  r1 = net.addHost('r1')
  net.addLink(h1, r0)
  net.addLink(r0, s0)
  net.addLink(s0, h2)
  net.addLink(s0, r1)
  net.addLink(r1, h3)
  net.build()
  r0.cmd("echo 1   /proc/sys/net/ipv4/ip_forward")
  r1.cmd("echo 1   /proc/sys/net/ipv4/ip_forward")
  s0.cmd("brctl addbr br0")
  s0.cmd("brctl addif br0 s0-eth0")
  s0.cmd("brctl addif br0 s0-eth1")
  s0.cmd("brctl addif br0 s0-eth2")
  s0.cmd("brctl addif br0 s0-eth3")
  s0.cmd("ifconfig br0 up")
  r0.cmd("ip addr add 192.168.1.254/24 brd + dev r0-eth0")
  r0.cmd("ip addr add 10.0.0.1/24 brd + dev r0-eth1")
  h1.cmd("ip addr add 192.168.1.1/24 brd + dev h1-eth0")
  h1.cmd("ip route add default via 192.168.1.254")
  r0.cmd("iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o r0-eth1 -j MASQUERADE")
  #h1.cmd("python -m SimpleHTTPServer 80")
  h2.cmd("ip addr add 10.0.0.3/24 brd + dev h2-eth0")
  r1.cmd("ip addr add 10.0.0.2/24 brd + dev r1-eth0")
  r1.cmd("ip addr add 192.168.2.254/24 brd + dev r1-eth1")
  r1.cmd("iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -o r1-eth0 -j MASQUERADE")
  r1.cmd("modprobe ip_nat_pptp")
  h3.cmd("ip addr add 192.168.2.1/24 brd + dev h3-eth0")
  h3.cmd("ip route add default via 192.168.2.254")

  CLI(net)
  net.stop()
```

## IV. IMPLEMENTATION

### i. Configuring the PPTP server:

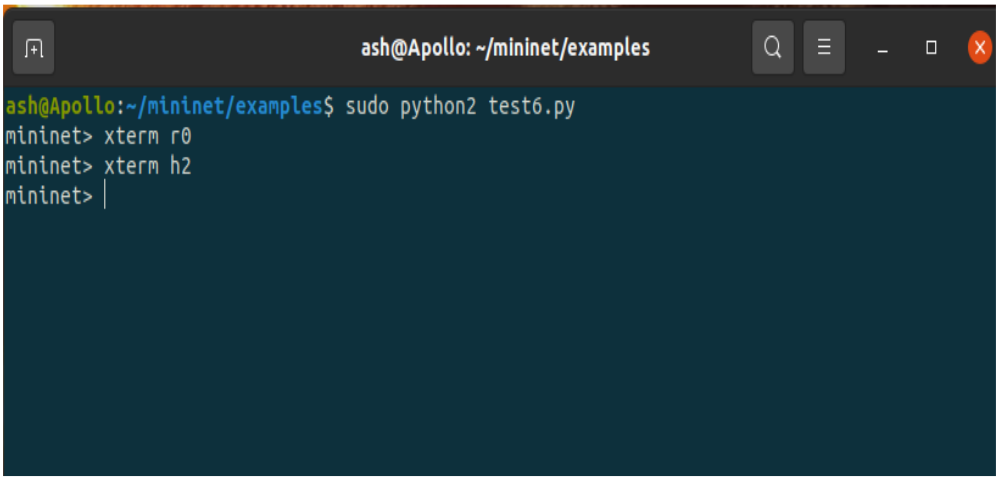
1. We have to run the constructed topology.
2. From the mininet of that topology we open two xterms r0 and h2.
3. In r0 we try to configure the ip address of the server .
4. Now we will set up an account with username and password.
5. Now start the pptpd server using the required commands.
6. Now we need to set up the PPTP client.

### ii. Configuring the PPTP clients

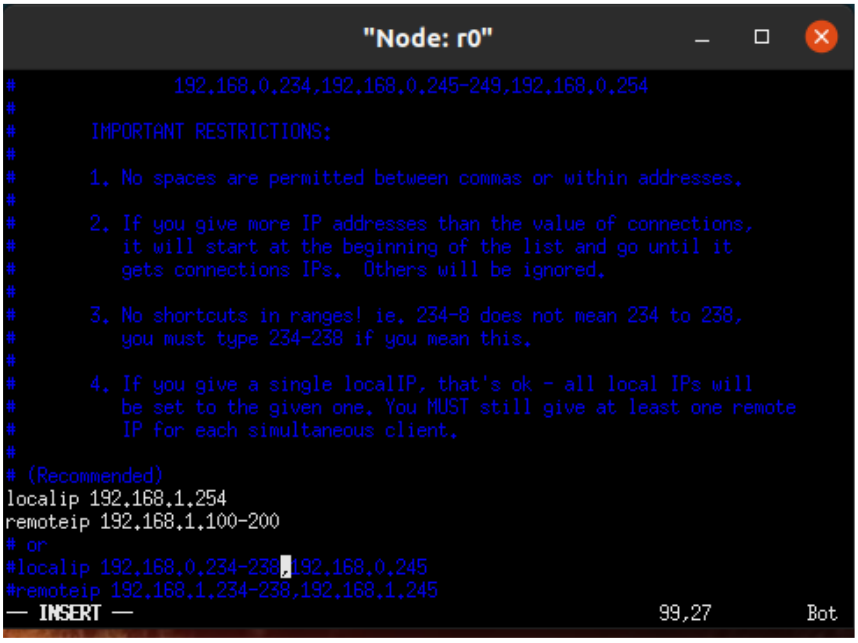
1. We have to create a VPN on the client side to connect to the PPTP server we created.
2. After that the VPN will try to connect to the server and we will get the notification of the VPN connected.
3. Set the routing table for host h2.
4. Start the http server at h3. And h3 connects to h1. (h3 ping h1).

## V. SCREENSHOTS

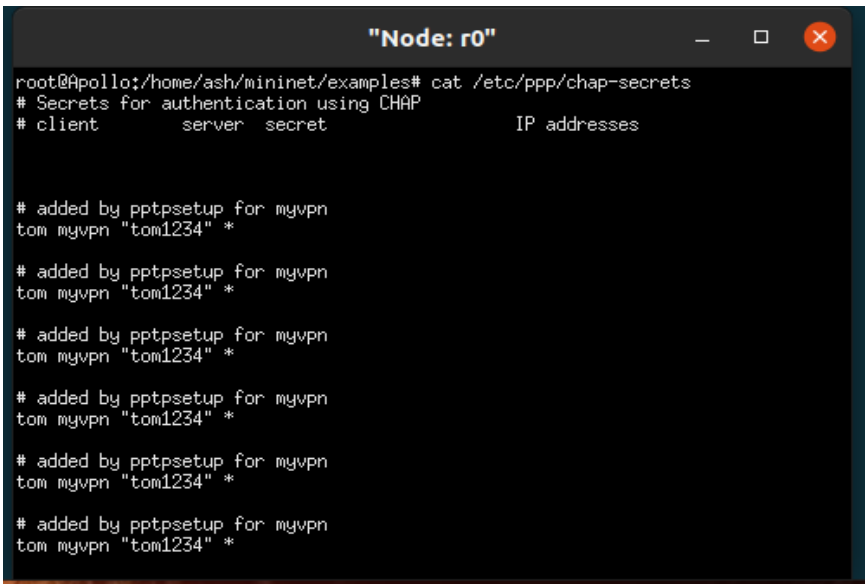
Step 1: Run the python file containing the topology created. And open the xterms r0 and h2 where r0 will be acting as the pptp server.




Step 2: Open the vim editor in r0 and set up the pptp server ip address.



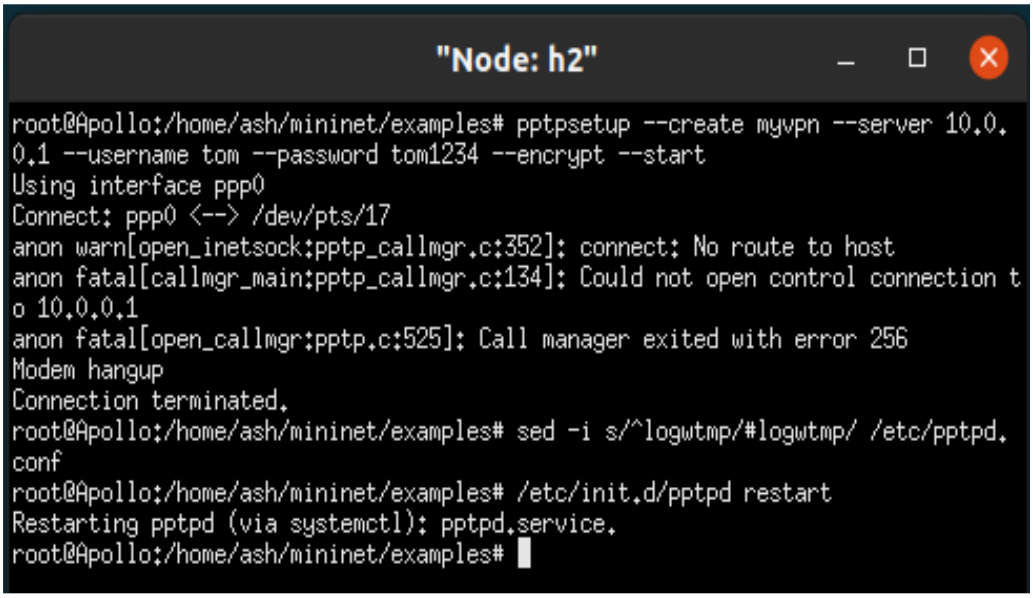
Step 3: Setup an account with username and password.



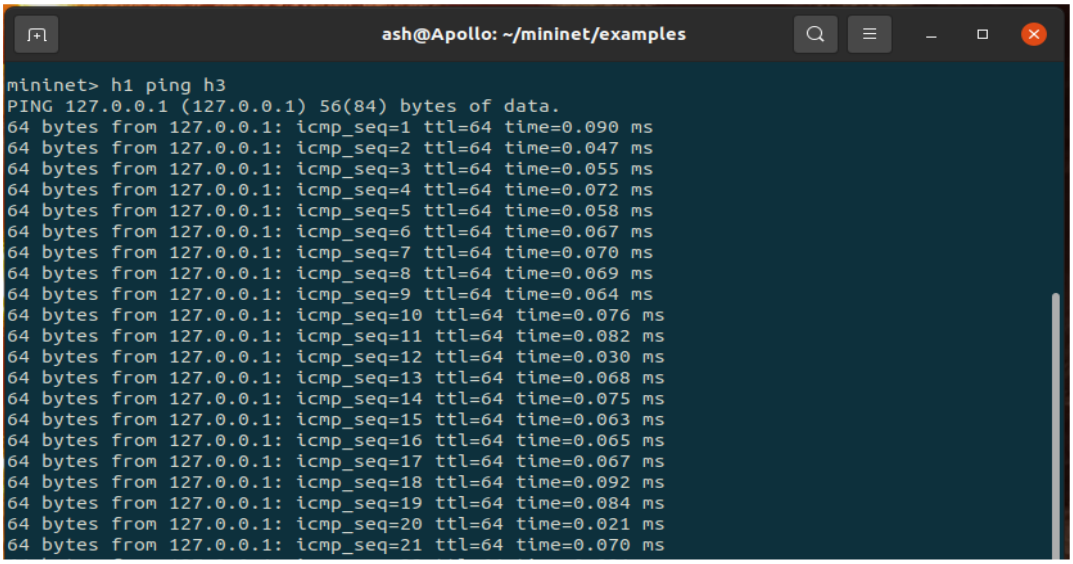

Step 4: Start the PPTP server in r0 using the following commands shown.

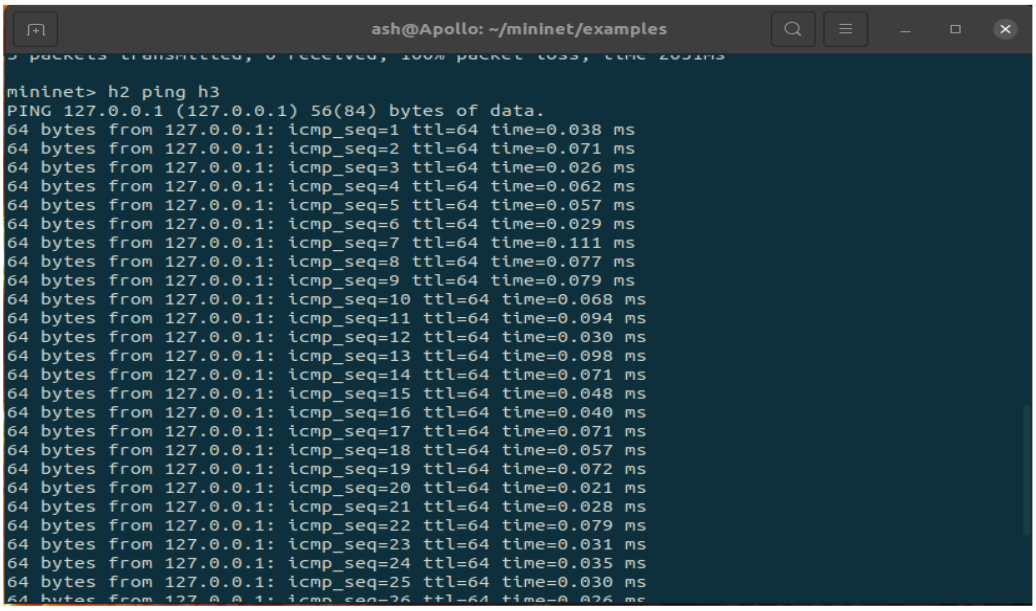
/usr/sbin/pptpd -d -f -l 10.0.0.1

Step 5:We have to create a VPN on the client side to connect to the PPTP server we created. At the first attempt we get an error of connection terminated.We need to fix the error using the commands in r0:
sed -i s/^logwtmp/#logwtmp/  /etc/pptpd.conf
Then restart the pptp server again.



Step 6: Now the VPN has been connected, we can try pinging the hosts from private to private or private to public network.
h1 from private network1 and h3 from private network2.

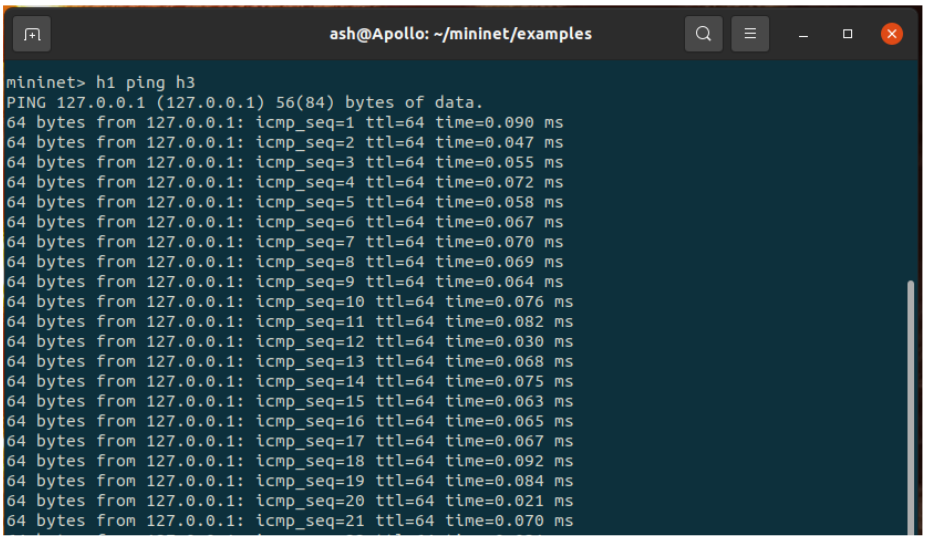


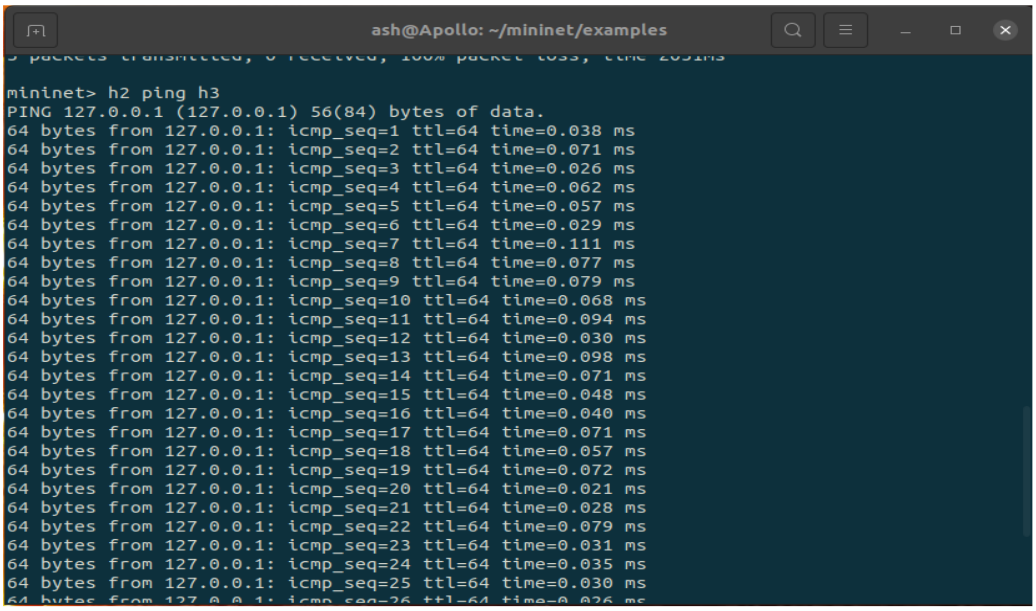



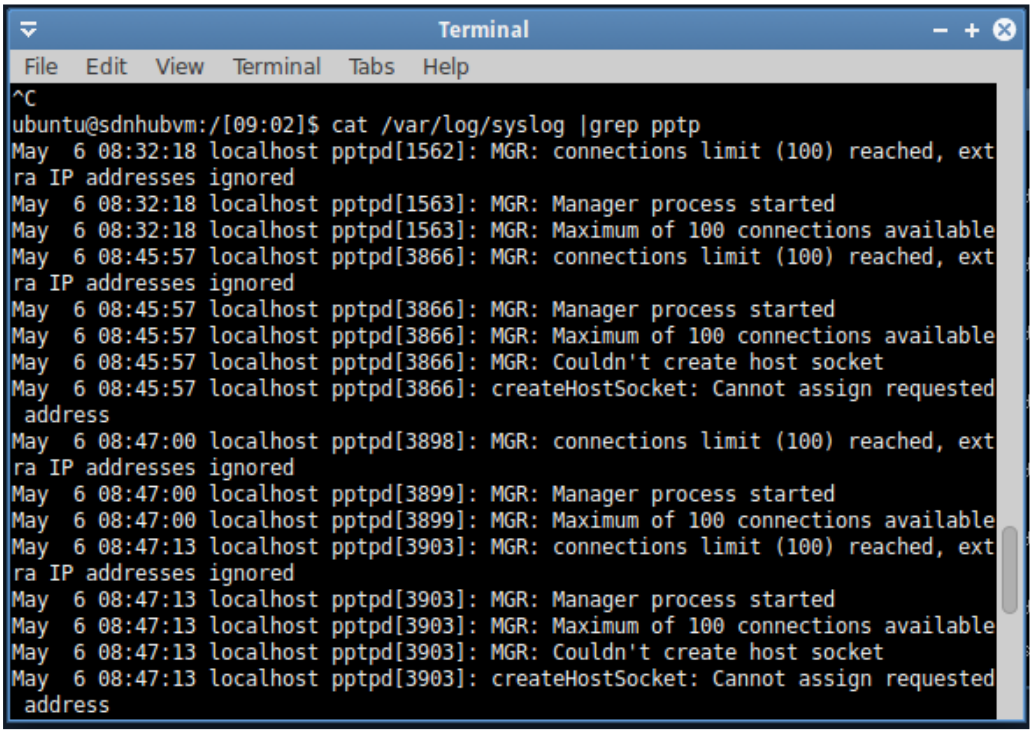

We see that the hosts are communicating and that the VPN server is working.

## VI. RESULT ANALYSIS

We see that the hosts are able to ping each other.






## VII. CONCLUSION

We are able to extend the private network using a public network such as the internet. As there was no secure connection between the private and public networks but when we created VPN, hosts from different private and public networks were able to communicate securely. The hosts from the private and public networks were not able to communicate before, but connecting to the VPN the hosts are able to communicate securely.

## VIII. REFERENCES

1. §http://blog.fens.me/vpn-pptp-client-ubuntu/

2. §https://bugs.launchpad.net/ubuntu/+source/pptpd/+bug/1451419

## IX. Drive link:

https://drive.google.com/drive/folders/1JVzFkGADk_Y2CqCjv0ilJuBl0E43NuDA?usp=sharing