

Linux Router

This report presents a comprehensive guide to establishing a Linux router utilizing VirtualBox alongside three Ubuntu virtual machines. The primary function of the router is to serve as a bridge facilitating communication between the client and server virtual machines. While internet connectivity might not be established within this setup, the focus remains on fostering communication among the networked machines. Alongside the detailed steps for configuration, the report emphasizes transparency by providing information on the tools utilized and their sourcing, ensuring clarity and reproducibility of the setup process. Despite the absence of direct internet access, the successful transmission of packets between the interconnected virtual machines underscores the efficacy of the configured network environment.

Step 1: Setting up Virtual Machines in VirtualBox:

1. Download, Install VirtualBox and Ubuntu Server:

-Download the VirtualBox installer from the official website:

Source: <https://www.virtualbox.org/wiki/Downloads>

-Download Ubuntu Server from the official website:

Source: <https://ubuntu.com/download/server>

-Follow the installation instructions for your operating system.

2. Create Virtual Machines:

-Open VirtualBox and click on "New" to create a new virtual machine.

-Choose a name for the virtual machine (e.g., "Router").

-Select "Linux" as the type and "Ubuntu (64-bit)" as the version.

-Allocate RAM and create a virtual hard disk.

-Repeat the process to create two more virtual machines for the client and server.

3. Configure Network Adapters:

-Select each virtual machine, then go to "Settings" > "Network".

-For the router VM:

i). Adapter 1: Select "NAT" to provide internet access.

ii). Adapter 2: Choose "Internal Network" and name it (e.g., "internal_network").

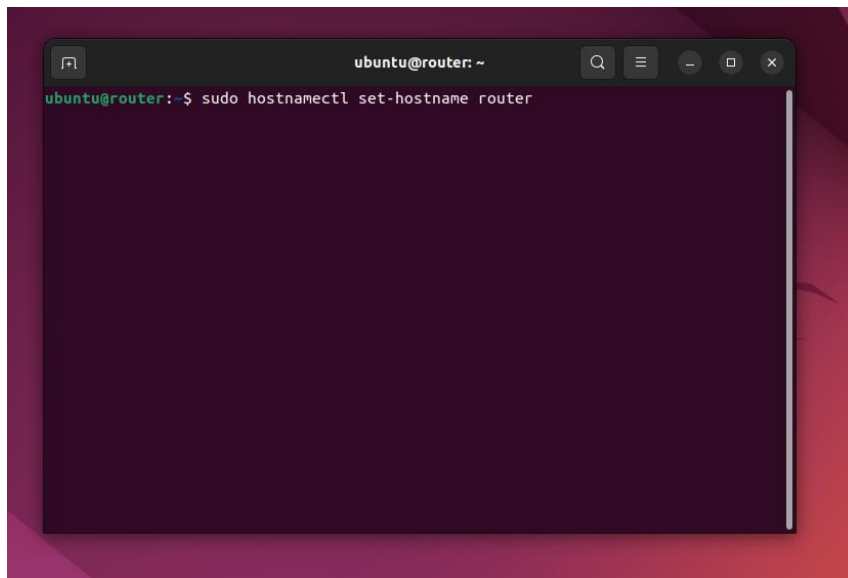
-For the client and server VMs:

Adapter 1: Choose "Internal Network" and select the same network name as the router's second adapter ("internal_network").

Step 2: Installation and Configuration of Router:

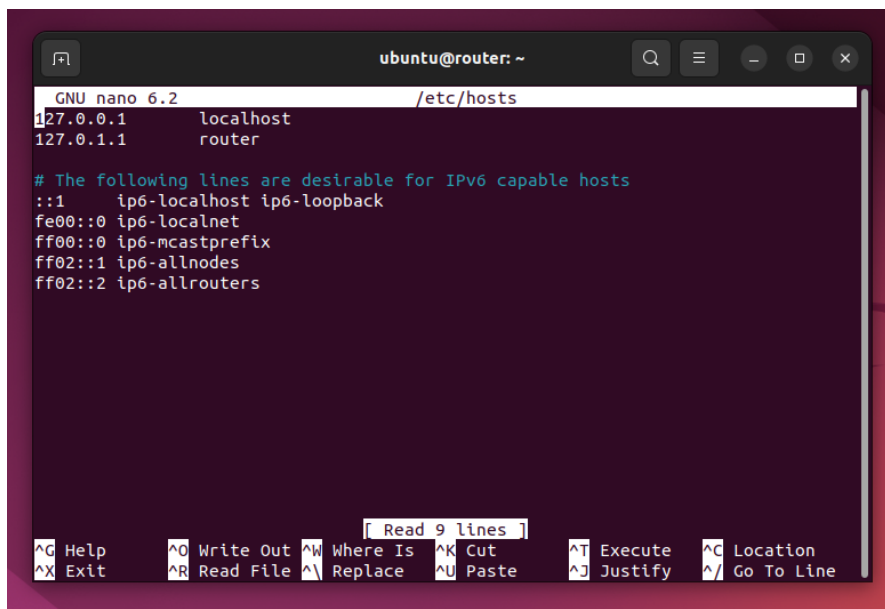
-The first step in setting up our router is to change the hostname of our machine for easy identification, in our case, "router". This step is optional.

Using this command: **sudo hostnamectl set-hostname router**

A terminal window titled 'ubuntu@router: ~' with search, menu, and window control icons. The command 'sudo hostnamectl set-hostname router' has been entered and executed, resulting in a blank prompt.

```
ubuntu@router:~$ sudo hostnamectl set-hostname router
```

Additionally, edit the hosts file using the following command: **sudo nano /etc/hosts**, this will launch a text editor. Rename the second line 127.0.0.1 to router. After making this change, save the file and exit the editor. Finally, restart your machine so that the new name can take effect.

A terminal window titled 'ubuntu@router: ~' showing the nano 6.2 text editor editing the file '/etc/hosts'. The second line '127.0.0.1' has been changed to 'router'. The bottom of the screen shows nano editor shortcuts.

```
GNU nano 6.2 /etc/hosts
1 127.0.0.1    localhost
2 127.0.1.1    router

# The following lines are desirable for IPv6 capable hosts
::1        ip6-localhost ip6-loopback
fe00::0    ip6-localnet
ff00::0    ip6-mcastprefix
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
```

[Read 9 lines]

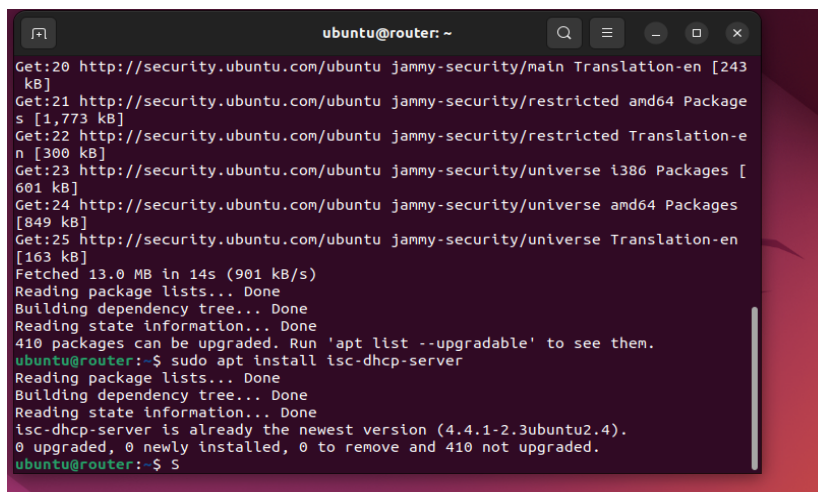
^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location
^X Exit	^R Read File	^_ Replace	^U Paste	^J Justify	^_ Go To Line

Configuring DHCP Server in our router machine

Dynamic Host Configuration Protocol (DHCP): enables centralized IP address management within a network. When machines are added to a network, they send DHCP requests seeking configuration details such as IP address, subnet mask, gateway, and DNS server from any available DHCP server. This automation simplifies network setup and administration, ensuring devices can seamlessly connect and communicate within the network infrastructure.

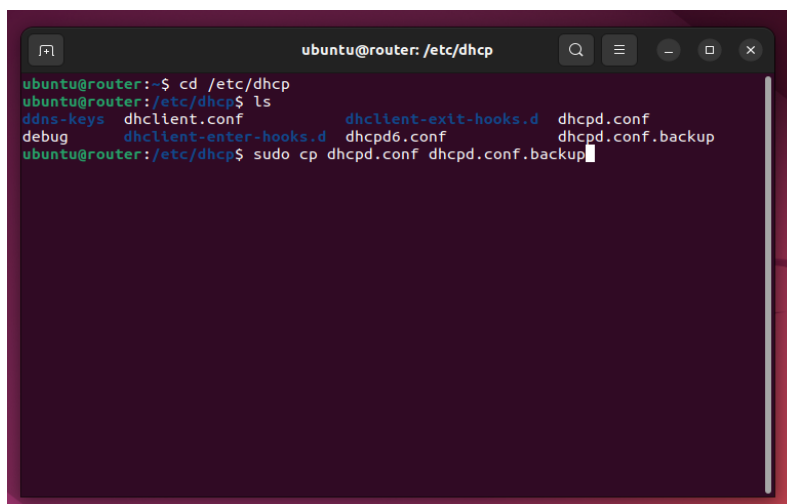
We need to update packages and install DHCP server: To set up a DHCP server, first, update your package list using: `sudo apt update`

Then, install the ISC DHCP server package with: `sudo apt install isc-dhcp-server`



```
ubuntu@router: ~  
Get:20 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [243 kB]  
Get:21 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Package s [1,773 kB]  
Get:22 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-e n [300 kB]  
Get:23 http://security.ubuntu.com/ubuntu jammy-security/universe i386 Packages [ 601 kB]  
Get:24 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [849 kB]  
Get:25 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [163 kB]  
Fetched 13.0 MB in 14s (901 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
410 packages can be upgraded. Run 'apt list --upgradable' to see them.  
ubuntu@router:~$ sudo apt install isc-dhcp-server  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
isc-dhcp-server is already the newest version (4.4.1-2.3ubuntu2.4).  
0 upgraded, 0 newly installed, 0 to remove and 410 not upgraded.  
ubuntu@router:~$ S
```

After successfully installing dhcp-server we need to make few changes in the dhcpd.conf file which we can find by navigating to this folder **/etc/dhcp** using this command `cd /etc/dhcp` and copy **dhcpd.conf** file to a backup file using this command `sudo cp dhcpd.conf dhcpd.conf.backup` in case we make errors we can quickly look at backup file for reference as we are goi to edit **dhcpd.conf**.



```
ubuntu@router: /etc/dhcp  
ubuntu@router:~$ cd /etc/dhcp  
ubuntu@router:/etc/dhcp$ ls  
ddns-keys  dhcpclient.conf          dhcpclient-exit-hooks.d  dhcpd.conf  
debug      dhcpclient-enter-hooks.d  dhcpd6.conf              dhcpd.conf.backup  
ubuntu@router:/etc/dhcp$ sudo cp dhcpd.conf dhcpd.conf.backup
```

Modify /etc/dhcp/dhcpd.conf to configure the network information you want to serve:

```
ddns-update-style none;

option domain-name-servers 8.8.8.8, 8.8.4.4;

default-lease-time 600;

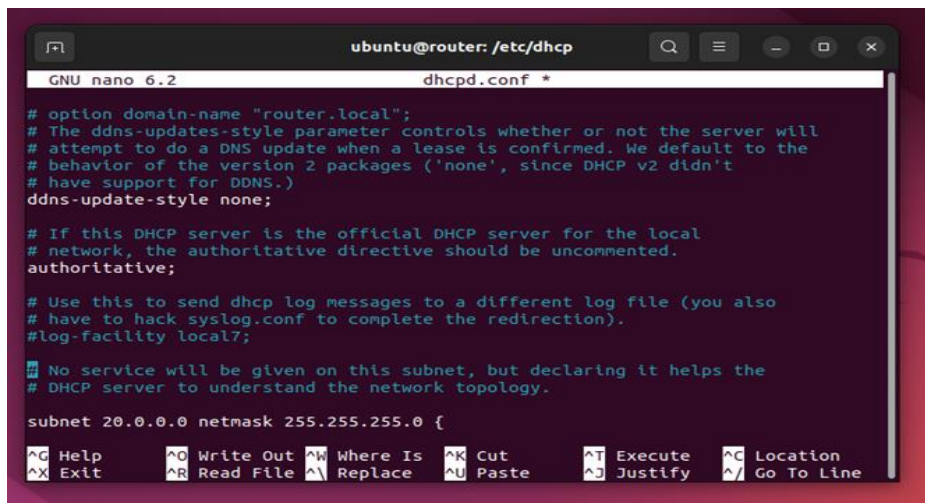
max-lease-time 7200;

authoritative;

subnet 20.0.0.0 netmask 255.255.255.0 {

    range 20.0.0.5 20.0.0.10;

    option routers 20.0.0.1;}
```



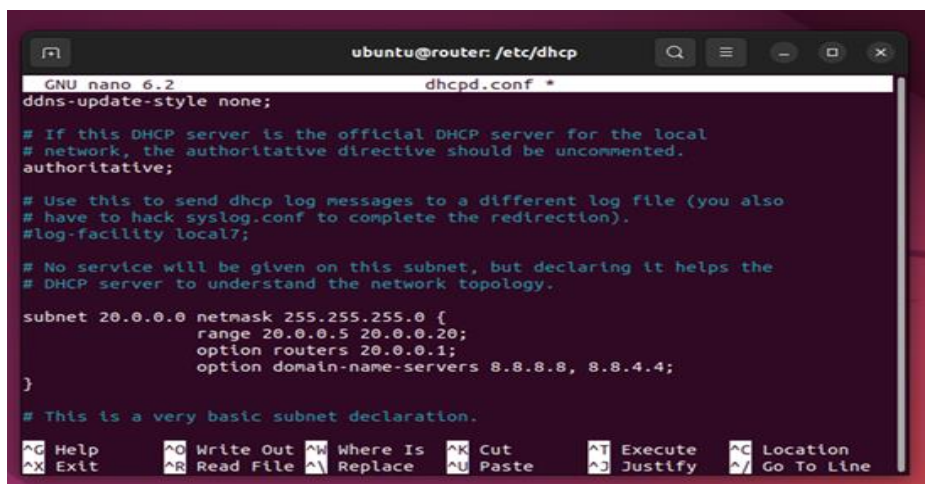
```
ubuntu@router: /etc/dhcp
GNU nano 6.2      dhcpd.conf *
# option domain-name "router.local";
# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style none;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
#log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

subnet 20.0.0.0 netmask 255.255.255.0 {
```



```
ubuntu@router: /etc/dhcp
GNU nano 6.2      dhcpd.conf *
ddns-update-style none;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
#log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

subnet 20.0.0.0 netmask 255.255.255.0 {
    range 20.0.0.5 20.0.0.20;
    option routers 20.0.0.1;
    option domain-name-servers 8.8.8.8, 8.8.4.4;
}

# This is a very basic subnet declaration.
```

In this configuration:

ddns-update-style none; : Disables Dynamic DNS updates.

option domain-name-servers 8.8.8.8, 8.8.4.4; Specifies the DNS servers as Google's public DNS servers.

default-lease-time 600; Sets the default lease time for IP addresses to 600 seconds (10 minutes).

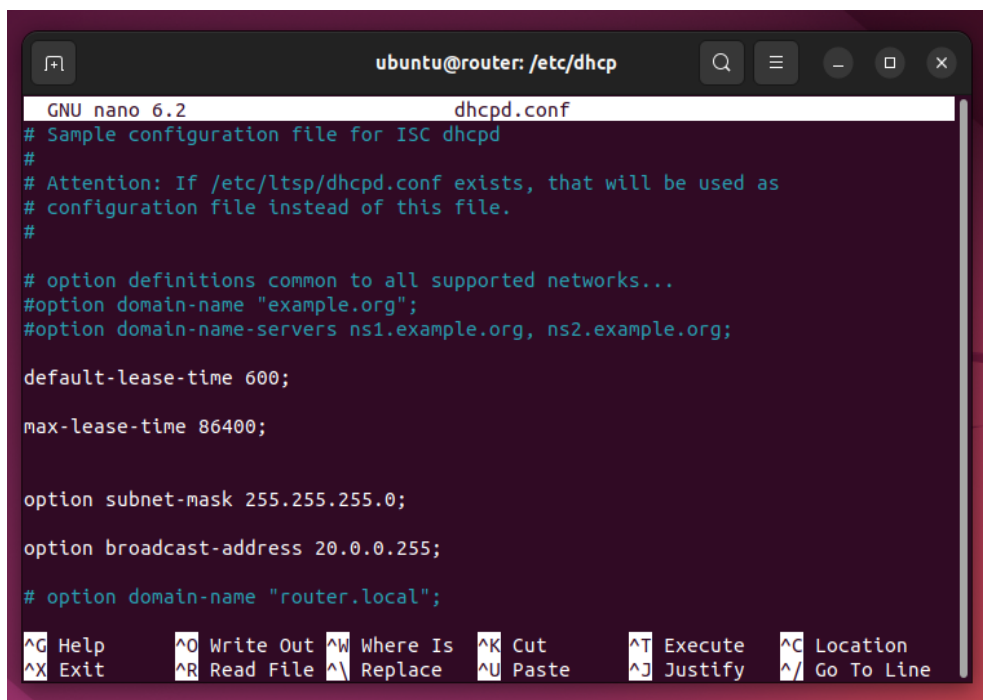
max-lease-time 7200; Sets the maximum lease time for IP addresses to 7200 seconds (2 hours).

authoritative; Declares this DHCP server as authoritative for the specified subnet.

ubnet 20.0.0.0 netmask 255.255.255.0 { ... }: Defines the subnet and its associated configuration.

range 20.0.0.5 20.0.0.10; Specifies the range of IP addresses to be assigned by the DHCP server.

option routers 20.0.0.1; Specifies the default gateway/router for clients on this subnet.



```
ubuntu@router: /etc/dhcp
GNU nano 6.2          dhcpd.conf
# Sample configuration file for ISC dhcpd
#
# Attention: If /etc/ltsp/dhcpd.conf exists, that will be used as
# configuration file instead of this file.
#
# option definitions common to all supported networks...
#option domain-name "example.org";
#option domain-name-servers ns1.example.org, ns2.example.org;

default-lease-time 600;

max-lease-time 86400;

option subnet-mask 255.255.255.0;

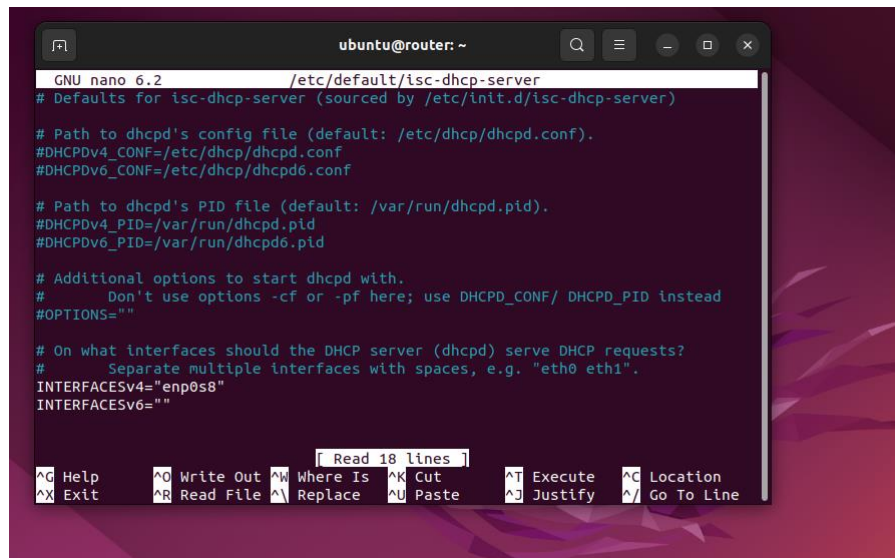
option broadcast-address 20.0.0.255;

# option domain-name "router.local";

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify  ^_ Go To Line
```

Modify `/etc/default/isc-dhcp-server` to add the interface **enp0s8** which you should serve requests on: # `sudo nano /etc/default/isc-dhcp-server`

Interfaces can be located on this command # `ifconfig`



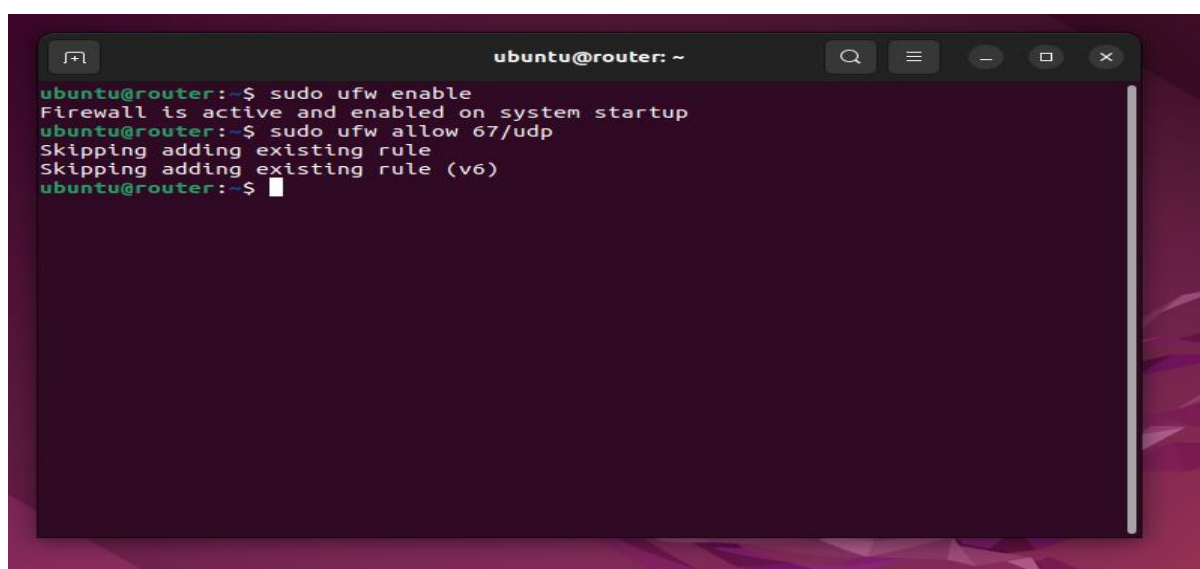
```
ubuntu@router: ~  
GNU nano 6.2 /etc/default/isc-dhcp-server  
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)  
  
# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).  
#DHCPDV4_CONF=/etc/dhcp/dhcpd.conf  
#DHCPDV6_CONF=/etc/dhcp/dhcpd6.conf  
  
# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).  
#DHCPDV4_PID=/var/run/dhcpd.pid  
#DHCPDV6_PID=/var/run/dhcpd6.pid  
  
# Additional options to start dhcpd with.  
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead  
#OPTIONS=""  
  
# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?  
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".  
INTERFACESv4="enp0s8"  
INTERFACESv6=""  
  
Read 18 lines  
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

To set up a firewall using iptables, you can follow these steps:

-Enable the firewall using this command: `sudo ufw enable`

-Allow port 67 UDP, which will be used by DHCP, using this command: `sudo ufw allow 67/udp`

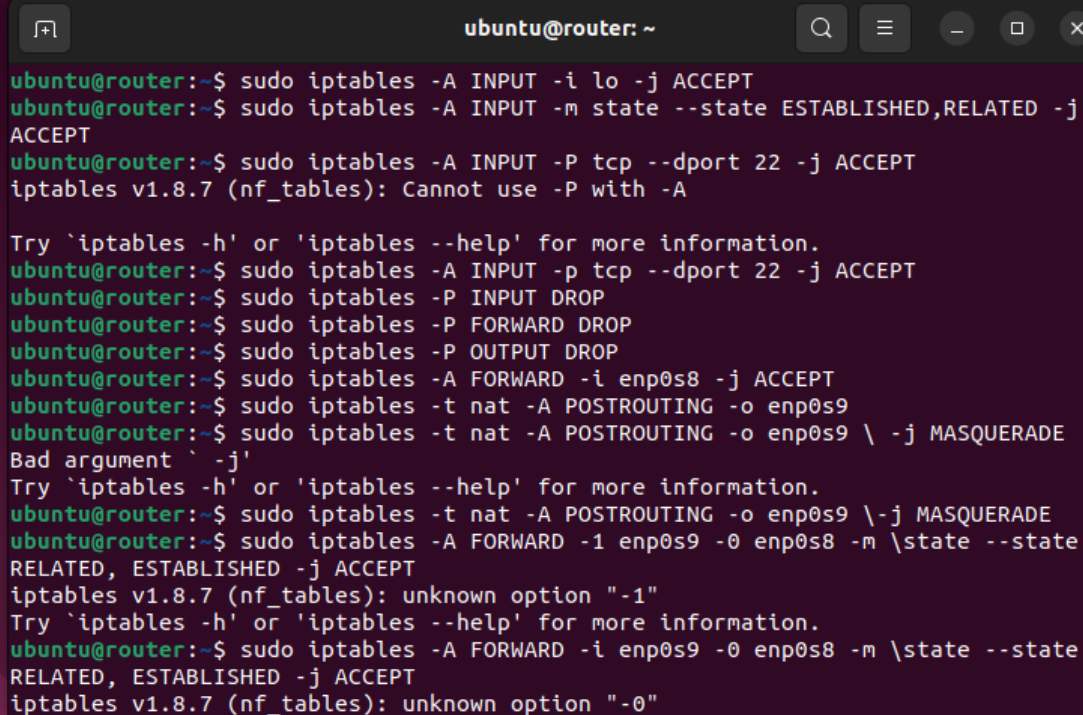
These commands will activate the firewall and permit incoming UDP traffic on port 67, which is essential for DHCP functionality.



```
ubuntu@router: ~  
ubuntu@router:~$ sudo ufw enable  
Firewall is active and enabled on system startup  
ubuntu@router:~$ sudo ufw allow 67/udp  
Skipping adding existing rule  
Skipping adding existing rule (v6)  
ubuntu@router:~$
```

A properly configured firewall should be configured in a default deny configuration with specific allows (Whitelist) for what you want to accept.

```
# sudo iptables -A INPUT -i lo -j ACCEPT
# sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
# sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
# sudo iptables -P INPUT DROP
# sudo iptables -P FORWARD DROP
# sudo iptables -P OUTPUT ACCEPT
# sudo iptables -A FORWARD -i eth0 -j ACCEPT
# sudo iptables -t nat -A POSTROUTING -o eth2 -j MASQUERADE
# sudo iptables -A FORWARD -i eth2 -o eth0 -m \state --state RELATED,ESTABLISHED -j ACCEPT
# sudo iptables -A FORWARD -i eth0 -j ACCEPT
```

A terminal window titled 'ubuntu@router: ~' showing a series of iptables commands and their outputs. The commands are: 1. 'sudo iptables -A INPUT -i lo -j ACCEPT' (successful). 2. 'sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT' (successful). 3. 'sudo iptables -A INPUT -P tcp --dport 22 -j ACCEPT' (successful). 4. 'iptables v1.8.7 (nf_tables): Cannot use -P with -A' (error). 5. 'Try `iptables -h` or `iptables --help` for more information.' (help message). 6. 'sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT' (successful). 7. 'sudo iptables -P INPUT DROP' (successful). 8. 'sudo iptables -P FORWARD DROP' (successful). 9. 'sudo iptables -P OUTPUT DROP' (successful). 10. 'sudo iptables -A FORWARD -i enp0s8 -j ACCEPT' (successful). 11. 'sudo iptables -t nat -A POSTROUTING -o enp0s9' (successful). 12. 'sudo iptables -t nat -A POSTROUTING -o enp0s9 \ -j MASQUERADE' (error: 'Bad argument -j'). 13. 'Try `iptables -h` or `iptables --help` for more information.' (help message). 14. 'sudo iptables -t nat -A POSTROUTING -o enp0s9 -j MASQUERADE' (successful). 15. 'sudo iptables -A FORWARD -i enp0s9 -o enp0s8 -m \state --state RELATED, ESTABLISHED -j ACCEPT' (error: 'iptables v1.8.7 (nf_tables): unknown option "-1"'). 16. 'Try `iptables -h` or `iptables --help` for more information.' (help message). 17. 'sudo iptables -A FORWARD -i enp0s9 -o enp0s8 -m \state --state RELATED, ESTABLISHED -j ACCEPT' (error: 'iptables v1.8.7 (nf_tables): unknown option "-0"').

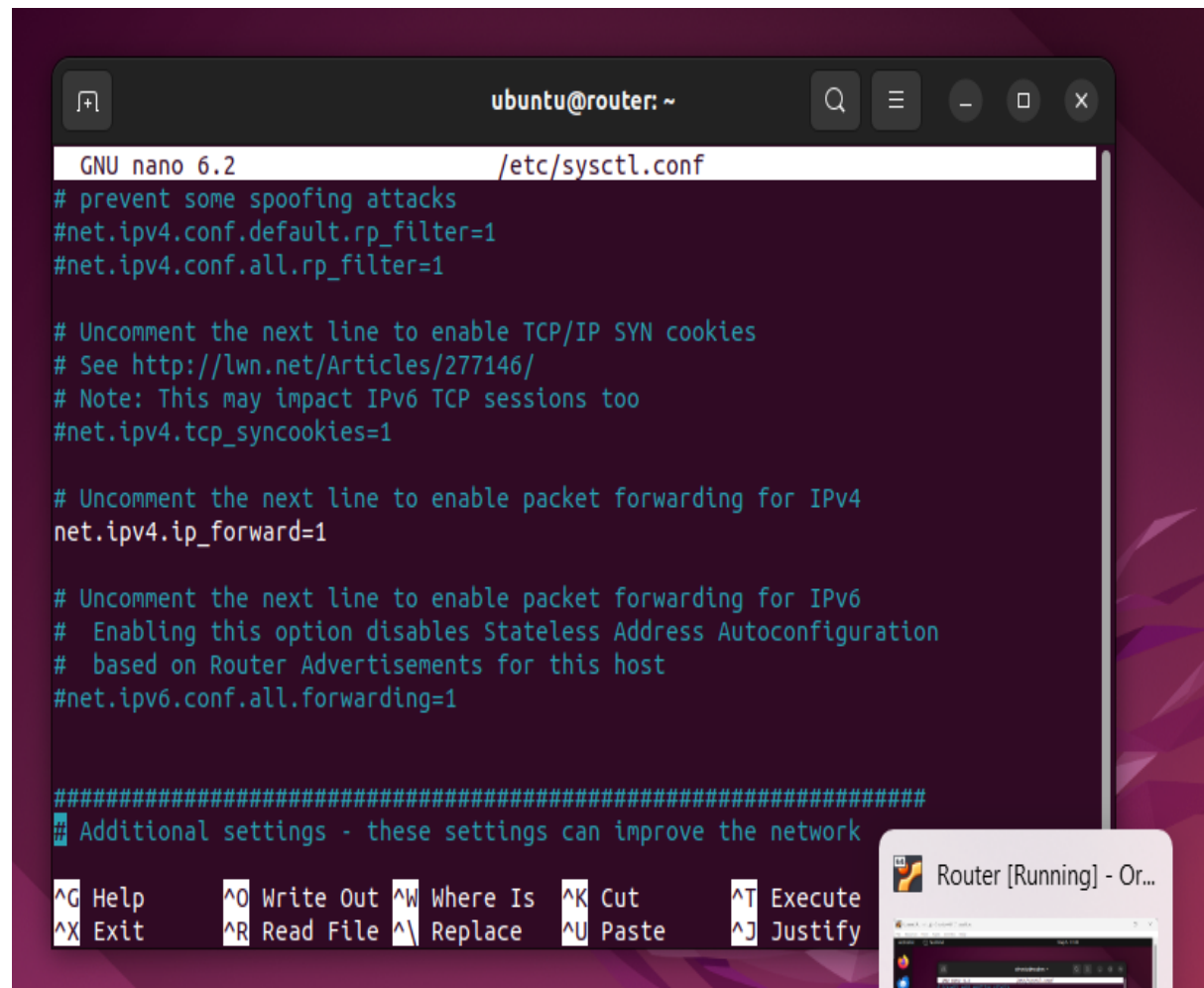
```
ubuntu@router:~$ sudo iptables -A INPUT -i lo -j ACCEPT
ubuntu@router:~$ sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
ubuntu@router:~$ sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables v1.8.7 (nf_tables): Cannot use -P with -A

Try `iptables -h` or `iptables --help` for more information.
ubuntu@router:~$ sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
ubuntu@router:~$ sudo iptables -P INPUT DROP
ubuntu@router:~$ sudo iptables -P FORWARD DROP
ubuntu@router:~$ sudo iptables -P OUTPUT DROP
ubuntu@router:~$ sudo iptables -A FORWARD -i enp0s8 -j ACCEPT
ubuntu@router:~$ sudo iptables -t nat -A POSTROUTING -o enp0s9
ubuntu@router:~$ sudo iptables -t nat -A POSTROUTING -o enp0s9 \ -j MASQUERADE
Bad argument -j
Try `iptables -h` or `iptables --help` for more information.
ubuntu@router:~$ sudo iptables -t nat -A POSTROUTING -o enp0s9 -j MASQUERADE
ubuntu@router:~$ sudo iptables -A FORWARD -i enp0s9 -o enp0s8 -m \state --state RELATED, ESTABLISHED -j ACCEPT
iptables v1.8.7 (nf_tables): unknown option "-1"
Try `iptables -h` or `iptables --help` for more information.
ubuntu@router:~$ sudo iptables -A FORWARD -i enp0s9 -o enp0s8 -m \state --state RELATED, ESTABLISHED -j ACCEPT
iptables v1.8.7 (nf_tables): unknown option "-0"
```


IP Forwarding using `net.ipv4.ip_forward`

We need to edit the `/etc/sysctl.conf` file using `# sudo nano /etc/sysctl.conf`, to make sure the new setting survives a reboot.

so we need to uncomment the line `# net.ipv4.ip_forward = 0` and change its value from `net.ipv4.ip_forward = 0` to `net.ipv4.ip_forward = 1`



The screenshot shows a terminal window titled 'ubuntu@router: ~' with a search bar and window controls. The nano editor is open to the file `/etc/sysctl.conf`. The content of the file is as follows:

```
GNU nano 6.2 /etc/sysctl.conf
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
# Enabling this option disables Stateless Address Autoconfiguration
# based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1

#####
# Additional settings - these settings can improve the network
```

At the bottom of the terminal, there is a table of nano editor shortcuts:

<code>^G</code> Help	<code>^O</code> Write Out	<code>^W</code> Where Is	<code>^K</code> Cut	<code>^T</code> Execute
<code>^X</code> Exit	<code>^R</code> Read File	<code>^\\</code> Replace	<code>^U</code> Paste	<code>^J</code> Justify

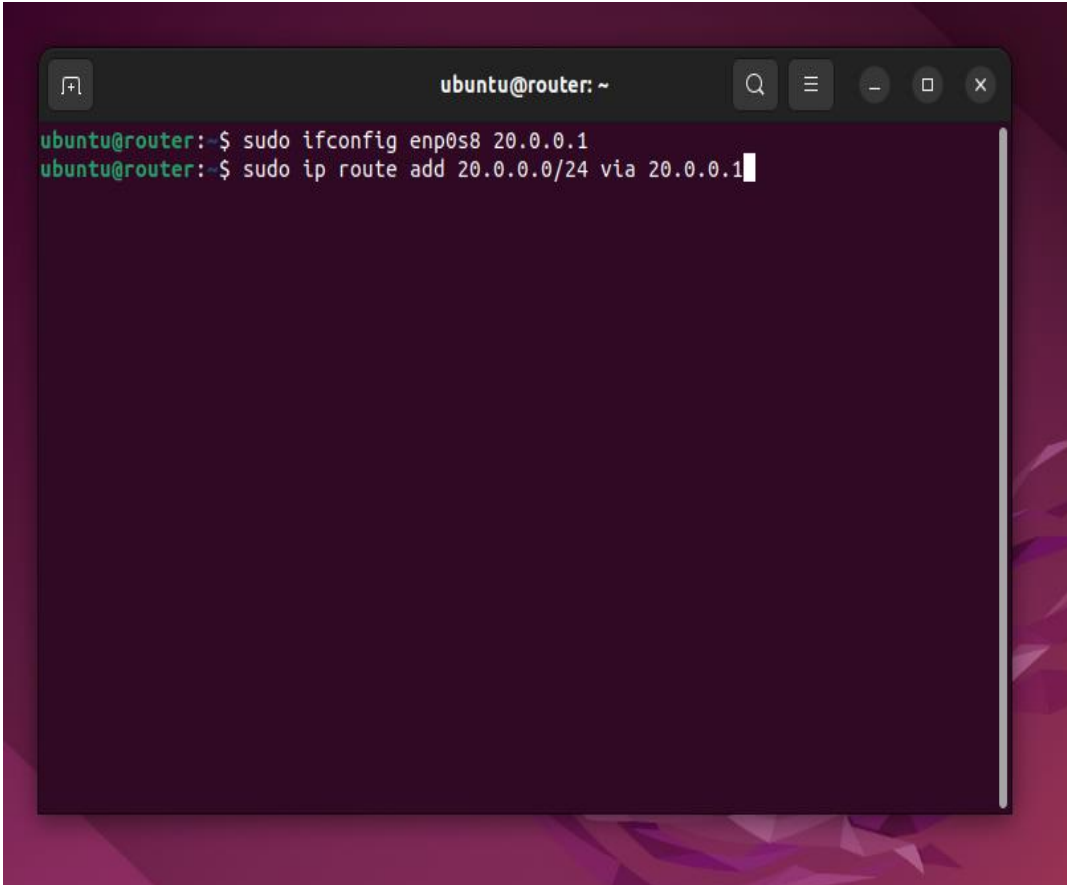
In the bottom right corner, there is a small window titled 'Router [Running] - Or...' showing a network diagram.

Give the interface **enp0s8 IP address** that matches the router option defined in the dhcpd.conf file and add the route via the IP address given to the interface **enp0s8** using commands.

```
sudo ifconfig enp0s8
```

```
sudo ip route add 20.0.0.0/24 via 20.0.0.1
```

Alternatively, we can add static address to interface enp0s8 to save us a lot of time to run these commands each time the router reboots.

A terminal window titled 'ubuntu@router: ~' with standard window controls. It shows two commands being executed: 'sudo ifconfig enp0s8 20.0.0.1' and 'sudo ip route add 20.0.0.0/24 via 20.0.0.1'. The second command is currently being typed, with a cursor at the end of the line.

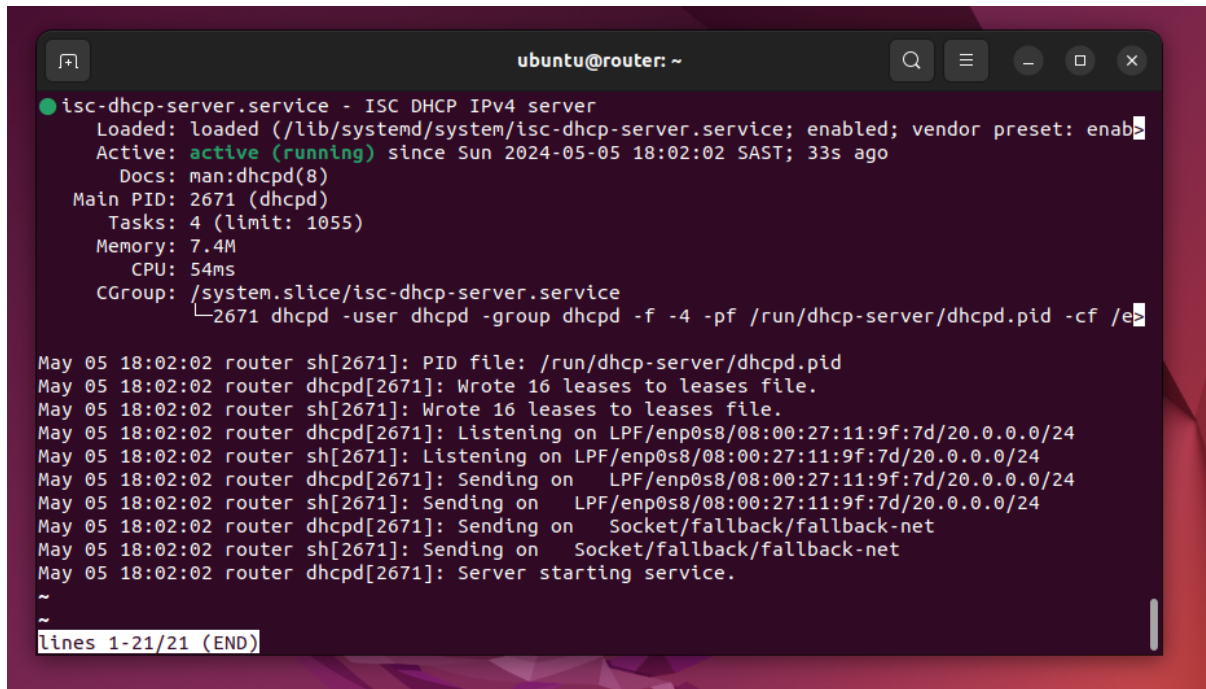
```
ubuntu@router:~$ sudo ifconfig enp0s8 20.0.0.1
ubuntu@router:~$ sudo ip route add 20.0.0.0/24 via 20.0.0.1
```

-Restart NetworkManager and DHCP server, execute the following commands:

```
sudo systemctl restart NetworkManager
```

```
sudo systemctl restart isc-dhcp-server
```

These commands will restart both services, ensuring that any changes made to the network configuration or DHCP server settings take effect.

A terminal window titled 'ubuntu@router: ~' displays the status of the 'isc-dhcp-server.service'. The service is shown as 'active (running)' since Sun 2024-05-05 18:02:02 SAST, 33s ago. It is managed by 'man:dhcpd(8)', has a main PID of 2671, 4 tasks, 7.4M memory, and 54ms CPU. The CGroup is '/system.slice/isc-dhcp-server.service'. Below this, a series of log messages show the DHCP server starting, writing leases, listening on the network interface 'LPF/enp0s8/08:00:27:11:9f:7d/20.0.0.0/24', and sending responses. The terminal output ends with '~' and 'lines 1-21/21 (END)'.

```
ubuntu@router: ~  
● isc-dhcp-server.service - ISC DHCP IPv4 server  
   Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vendor preset: enab  
   Active: active (running) since Sun 2024-05-05 18:02:02 SAST; 33s ago  
     Docs: man:dhcpd(8)  
    Main PID: 2671 (dhcpd)  
      Tasks: 4 (limit: 1055)  
     Memory: 7.4M  
        CPU: 54ms  
    CGroup: /system.slice/isc-dhcp-server.service  
            └─2671 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/dhcpd.pid -cf /e  
  
May 05 18:02:02 router sh[2671]: PID file: /run/dhcp-server/dhcpd.pid  
May 05 18:02:02 router dhcpd[2671]: Wrote 16 leases to leases file.  
May 05 18:02:02 router sh[2671]: Wrote 16 leases to leases file.  
May 05 18:02:02 router dhcpd[2671]: Listening on LPF/enp0s8/08:00:27:11:9f:7d/20.0.0.0/24  
May 05 18:02:02 router sh[2671]: Listening on LPF/enp0s8/08:00:27:11:9f:7d/20.0.0.0/24  
May 05 18:02:02 router dhcpd[2671]: Sending on LPF/enp0s8/08:00:27:11:9f:7d/20.0.0.0/24  
May 05 18:02:02 router sh[2671]: Sending on LPF/enp0s8/08:00:27:11:9f:7d/20.0.0.0/24  
May 05 18:02:02 router dhcpd[2671]: Sending on Socket/fallback/fallback-net  
May 05 18:02:02 router sh[2671]: Sending on Socket/fallback/fallback-net  
May 05 18:02:02 router dhcpd[2671]: Server starting service.  
~  
~  
lines 1-21/21 (END)
```

-To check if the ISC DHCP server is active, you can use the following command:

```
Sudo systemctl status isc-dhcp-server
```

Executing this command will provide information about the current status of the ISC DHCP server service, including whether it is active (running) or inactive (stopped). It also displays additional details such as the service's process ID (PID), memory usage, and recent logs.

-Now that the DHCP server is up and running, we can proceed to deploy client and server machines for testing our DHCP configuration settings and packet transmission. It's imperative to connect the client machines to the same network as the router. The DHCP server has successfully allocated IP addresses 20.0.0.5 and 20.0.0.6 to the client machines, demonstrating its functionality as intended.

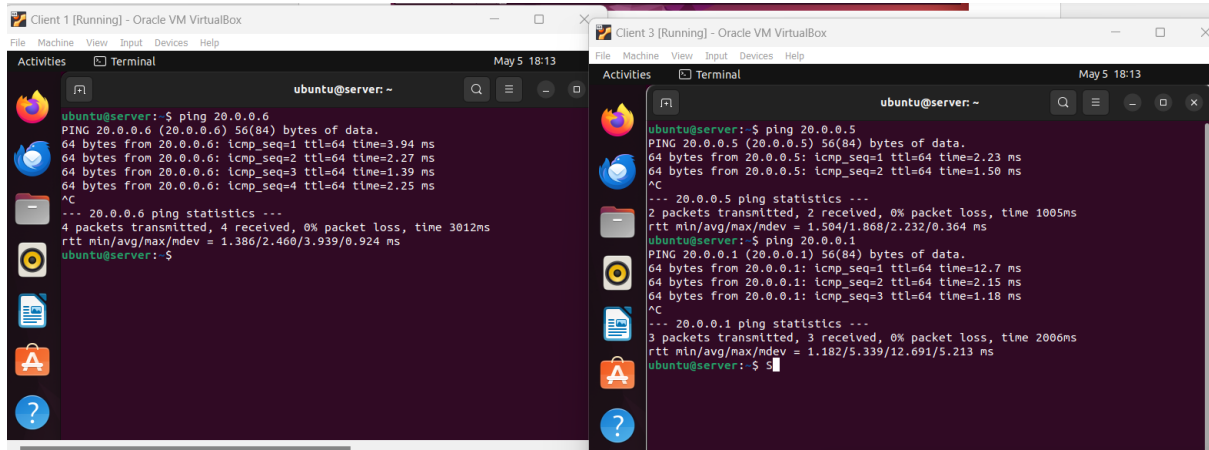
```
ubuntu@router: ~  
lines 1-21/21 (END)  
ubuntu@router:~$ sudo systemctl status isc-dhcp-server  
● isc-dhcp-server.service - ISC DHCP IPv4 server  
   Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vendor preset: enabled)  
   Active: active (running) since Sun 2024-05-05 18:02:02 SAST; 6min ago  
     Docs: man:dhcpcd(8)  
    Main PID: 2671 (dhcpcd)  
      Tasks: 4 (limit: 1055)  
    Memory: 7.4M  
       CPU: 110ms  
    CGroup: /system.slice/isc-dhcp-server.service  
            └─2671 dhcpcd -user dhcpcd -group dhcpcd -f -4 -pf /run/dhcp-server/dhcpcd.pid -cf /etc/dhc  
May 05 18:02:02 router sh[2671]: Sending on   LPF/enp0s8/08:00:27:11:9f:7d/20.0.0.0/24  
May 05 18:02:02 router dhcpcd[2671]: Sending on   Socket/fallback/fallback-net  
May 05 18:02:02 router sh[2671]: Sending on   Socket/fallback/fallback-net  
May 05 18:02:02 router dhcpcd[2671]: Server starting service.  
May 05 18:03:16 router dhcpcd[2671]: DHCPREQUEST for 192.168.8.100 from d4:e6:b7:a6:0c:c1 via enp0s8:  
May 05 18:03:16 router dhcpcd[2671]: DHCPNAK on 192.168.8.100 to d4:e6:b7:a6:0c:c1 via enp0s8  
May 05 18:07:24 router dhcpcd[2671]: DHCPREQUEST for 20.0.0.5 from 08:00:27:ad:0f:13 via enp0s8  
May 05 18:07:24 router dhcpcd[2671]: DHCPACK on 20.0.0.5 to 08:00:27:ad:0f:13 (server) via enp0s8  
May 05 18:07:29 router dhcpcd[2671]: DHCPREQUEST for 20.0.0.6 from 08:00:27:fd:76:57 via enp0s8  
May 05 18:07:29 router dhcpcd[2671]: DHCPACK on 20.0.0.6 to 08:00:27:fd:76:57 (server) via enp0s8  
lines 1-21/21 (END)
```

```
Client 1 [Running] - Oracle VM VirtualBox  
File Machine View Input Devices Help  
Activities  
ubuntu@server: ~  
May 5 18:09  
ubuntu@server:~$ ifconfig  
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255  
    inet6 fe80::ccc3:f78:3b85:61be prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:2e:6b:e0 txqueuelen 1000 (Ethernet)  
    RX packets 1078 bytes 1342608 (1.3 MB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 561 bytes 49341 (49.3 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 20.0.0.5 netmask 255.255.255.0 broadcast 20.0.0.255  
    inet6 fe80::ab51:2c59:b70c:1312 prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:ad:0f:13 txqueuelen 1000 (Ethernet)  
    RX packets 241 bytes 22127 (22.1 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 291 bytes 27944 (27.9 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 138 bytes 12272 (12.2 KB)  
  
Client 3 [Running] - Oracle VM VirtualBox  
File Machine View Input Devices Help  
Activities  
Terminal  
ubuntu@server: ~  
May 5 18:09  
ubuntu@server:~$ ifconfig  
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255  
    inet6 fe80::ccc3:f78:3b85:61be prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:2e:6b:e0 txqueuelen 1000 (Ethernet)  
    RX packets 467 bytes 545882 (545.8 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 334 bytes 34580 (34.5 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 20.0.0.6 netmask 255.255.255.0 broadcast 20.0.0.255  
    inet6 fe80::7d91:dc0e:35c:987c prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:fd:76:57 txqueuelen 1000 (Ethernet)  
    RX packets 192 bytes 16404 (16.4 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 275 bytes 29563 (29.5 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 146 bytes 13880 (13.8 KB)
```

We can test the communication between the clients by executing this command:

Ping 20.0.0.6 for client 2 and Ping 20.0.0.5 for client 1

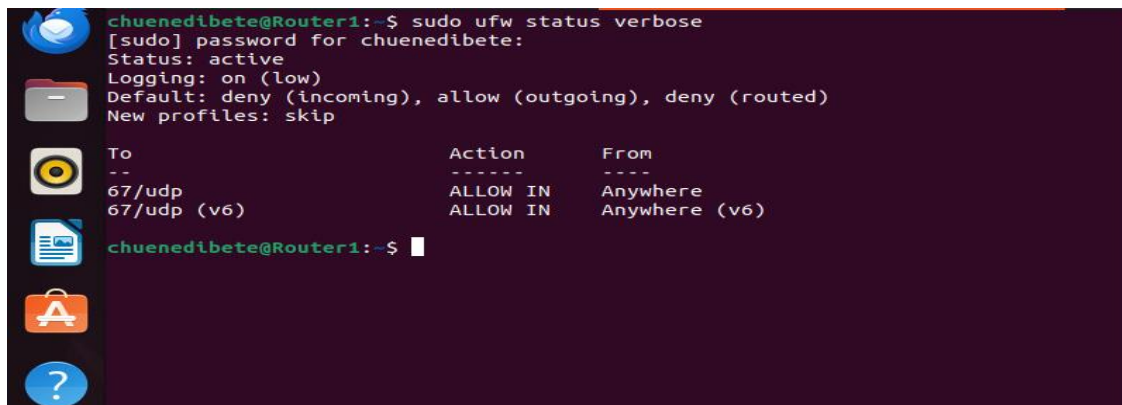
clients can communicate with each other and router.



The image shows two terminal windows from Oracle VM VirtualBox. The left window, titled 'Client 1 [Running] - Oracle VM VirtualBox', shows a terminal session on 'ubuntu@server: ~' where the command 'ping 20.0.0.6' is executed. The output shows four successful pings with times around 3.94ms, 2.27ms, 1.39ms, and 2.25ms, followed by statistics: 4 packets transmitted, 4 received, 0% packet loss, time 3012ms, and rtt values ranging from 1.386 to 3.939ms. The right window, titled 'Client 3 [Running] - Oracle VM VirtualBox', shows a terminal session on 'ubuntu@server: ~' where the command 'ping 20.0.0.5' is executed. The output shows three successful pings with times around 2.23ms, 1.50ms, and 1.18ms, followed by statistics: 3 packets transmitted, 3 received, 0% packet loss, time 2006ms, and rtt values ranging from 1.182 to 5.339ms.

To check the status of the firewall and view detailed information about its rules, including whether it's active or inactive, use the command: `sudo ufw status verbose`.

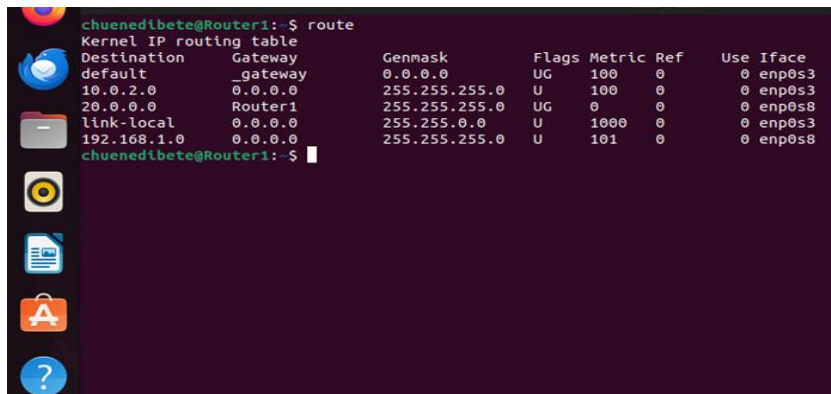
The firewall is active according to this output:



The image shows a terminal window titled 'chuenedibete@Router1:~\$ sudo ufw status verbose'. The output indicates that the firewall is active. It shows logging is on (low), the default policy is deny for incoming and outgoing traffic, and new profiles are skipped. Below this, a table of rules is displayed:

To	Action	From
67/udp	ALLOW IN	Anywhere
67/udp (v6)	ALLOW IN	Anywhere (v6)

To check the routing table, we use this command: `route`.



The image shows a terminal window titled 'chuenedibete@Router1:~\$ route'. The output displays the kernel IP routing table with columns for Destination, Gateway, Genmask, Flags, Metric, Ref, Use, and Iface. The table shows routes for the default gateway (0.0.0.0), 10.0.2.0 (via 0.0.0.0), 20.0.0.0 (via Router1), link-local (0.0.0.0), and 192.168.1.0 (via 0.0.0.0).

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
default	gateway	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
20.0.0.0	Router1	255.255.255.0	UG	0	0	0	enp0s8
link-local	0.0.0.0	255.255.0.0	U	1000	0	0	enp0s3
192.168.1.0	0.0.0.0	255.255.255.0	U	101	0	0	enp0s8

Challenges Faced

The initial stages of resource management presented considerable difficulties because it was difficult to manage host machine resources like CPU memory and storage to guarantee sufficient performance for every virtual machine particularly when running multiple VMs at once. Furthermore, the computers slowness made these issues worse making resource allocation and optimization tasks even more difficult. To ensure that there are no connectivity problems network interfaces and routing tables must be carefully configured. In order to successfully enable IP forwarding initial routing errors must be troubleshooted.

It was difficult to make sure that the network interfaces on the router and the client/server virtual machines were configured correctly especially when there were several adapters and different network kinds involved. It was difficult to set up firewall rules on the router to permit or prohibit types of traffic especially when taking both incoming and outgoing traffic into account. It was difficult to configure the routers DHCP (Dynamic Host Configuration Protocol) to assign IP addresses to client devices automatically lease times and IP address ranges needed to be carefully considered. Specifically, when integrating with external DNS servers it was difficult to configure DNS (Domain Name System) resolution on the router to translate domain names to IP addresses and vice versa. It took careful planning and configuration to ensure network security including encrypting sensitive data and guarding against different kinds of cyber threats. Several settings and configurations needed to be adjusted in order to maximize throughput and minimize latency while optimizing the routers overall performance.

During setup and configuration there were difficulties dealing with incompatibilities between various software versions hardware configurations and virtualization platforms (like VirtualBox). It was very difficult and time-consuming to fill in the knowledge and documentation gaps regarding networking principles Ubuntu Server administration and VirtualBox configuration.

Lessons Learned

In the process of setting up the router, I learned that having a deeper understanding of IP addressing and routing is crucial. Troubleshooting skills became invaluable as I encountered various configuration issues that needed prompt resolution. Documenting every step taken, including commands executed and configurations made, proved essential for future reference and troubleshooting. I also realized the importance of implementing version control systems like Git for configuration files and regularly backing up virtual machine snapshots to prevent data loss. Managing time effectively was a challenge, especially when balancing tasks like planning, implementation, testing, and troubleshooting.

Collaboration with other members and seeking assistance from peers provided valuable insights and support throughout the project. Embracing a growth mindset and continuously learning about the latest technologies and best practices were essential for success. Ethical considerations, such as privacy, security, and data protection, were always at the forefront of my mind during the configuration process.

I had to ensure responsible and professional conduct in every aspect of the project. Being resourceful and adaptable in solving problems and overcoming challenges was crucial, especially when encountering unexpected issues. seeking feedback from peers and reflecting on both successes and failures helped me grow both personally and professionally. It allowed me to learn from my experiences and continuously improve my skills. Overall, the process of creating the router taught me valuable lessons in networking, troubleshooting, time management, collaboration, continuous learning, ethical considerations, and resilience.

Recommendations for Improvement

Based on the experience gained during this project, I suggest several improvements for future router installation and configuration efforts. First, developing a standardized configuration process can ensure consistency and maintain a high level of security and performance across the organization. This entails establishing a well-defined, step-by-step approach to router configuration. Secondly, investing in comprehensive training for network administrators on router configuration best practices is crucial. Hands-on experience with various router models and features enhances their skills and confidence in managing network infrastructure effectively.

Addressing resource limitations proactively is another key aspect. This involves optimizing configurations, minimizing resource-intensive processes, and considering hardware upgrades or replacements where feasible to improve overall system performance and efficiency. Implementing collaborative learning platforms or online forums where students can share experiences, ask questions, and collaborate on problem-solving fosters peer-to-peer learning and creates a supportive learning community. Integrating troubleshooting resources directly into documentation, such as troubleshooting guides, flowcharts, and common error solutions, empowers students to diagnose and resolve issues independently, promoting self-directed learning. Improving the quality and accessibility of documentation by incorporating multimedia elements like video tutorials, interactive guides, and annotated screenshots caters to diverse learning styles and facilitates understanding for students with varying levels of expertise.

Expanding the scope of router installation and configuration projects to include a broader range of scenarios and challenges, such as multi-site deployments or integration with cloud services, exposes students to real-world complexities and prepares them for diverse networking environments. And establishing feedback mechanisms, such as surveys or feedback forms, to gather input from students about their experiences with router installation and configuration projects, helps identify areas for improvement and refine future project iterations. This continuous feedback loop ensures that the learning process remains dynamic and responsive to student needs.