Epoka University
Department of Computer Engineering

Course Name: Computer Networks
Course Code: CEN 307

Project Title:

Virtual Networking and Client-Server Architecture Application Deployment

Group Number: 1

Submitted By:

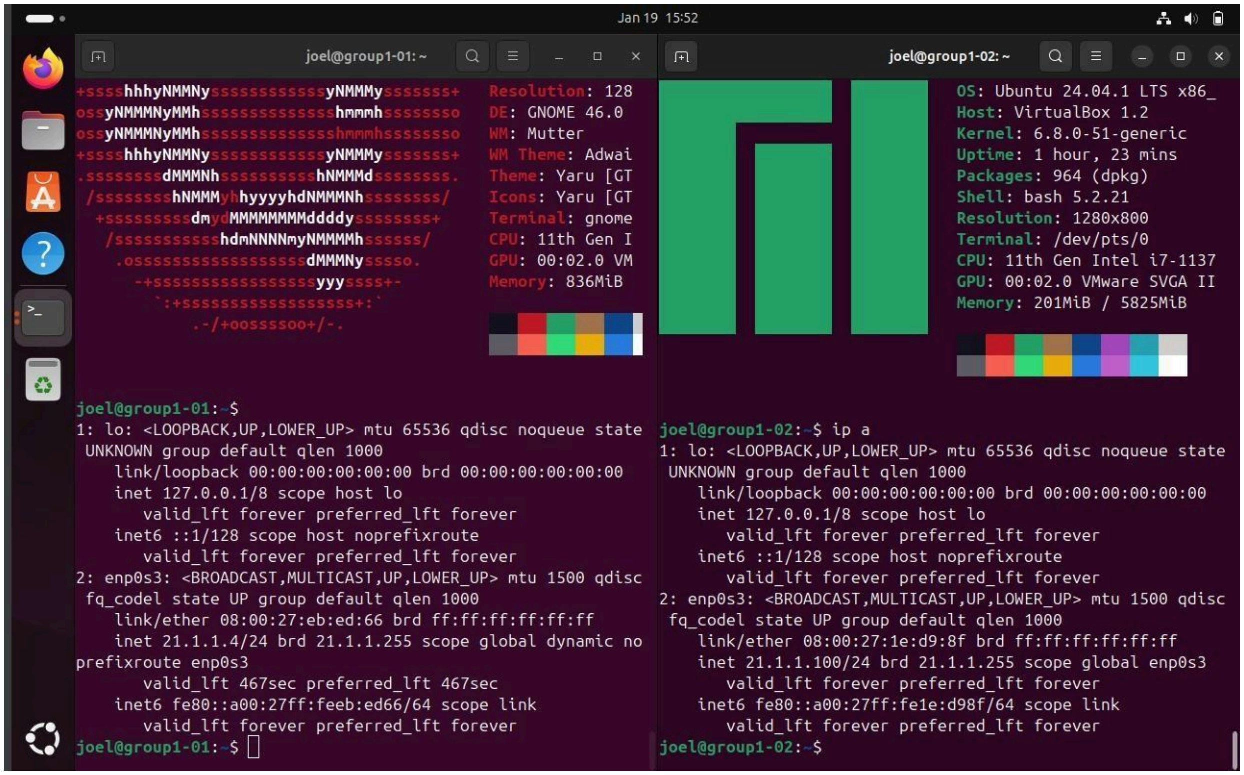
- **Student 1 Full Name:** Alvi Hysa
  Student 1 ID: 02052237
- **Student 2 Full Name:** Alesio Rajta
  Student 2 ID: 02052245
- **Student 3 Full Name:** Engjëll Abazaj
  Student 3 ID: 02052235
- **Student 4 Full Name:** Indrit Ferati
  Student 4 ID: 02052205
- **Student 5 Full Name:** Joel Bitri
  Student 5 ID: 02052259

1. *Concepts and terms explanations for Sections 1,2 and 3*

- ***NAT*** – Network Address Translation, is used to map private IP addresses in a local network to a public IP address for communication with external networks. In this use case, the Virtual NAT Network in VirtualBox enables communication between the virtual machines and external networks, such as the internet, while preserving the private IP addresses of the VMs.
- ***Subnet*** - A subnet divides a larger network into smaller segments to manage and optimize network traffic. The /24 indicates a subnet mask where the first 24 bits are used for the network identifier.
- ***2xx.xx.0/24*** – The format of the subnet address. In our case the specific address is 21.1.1.100/24. It defines a range of IP addresses for communication between devices on the network.
- ***DHCP*** – Dynamic Host Configuration Protocol, automatically assigns IP addresses and other network configurations to devices in a network. The Virtual NAT Network uses DHCP to assign IP addresses to the Ubuntu Desktop and Ubuntu Server, simplifying the network setup.
- ***DNS*** – Domain Name System, translates human-readable domain names (www.google.com) into IP addresses. DNS is configured to allow virtual machines to resolve domain names to IP addresses, facilitating internet connectivity and hostname resolution within the project.
- ***Nginx Web Server & Apache Web Server*** - These are HTTP servers used to serve web content. Nginx is known for its performance, while Apache is versatile and widely used. We installed Nginx on the Ubuntu Server to host a default web page, which is accessed from the Ubuntu Desktop Browser.
- ***OpenSSH Server*** - Allows secure remote login and file transfers using the SSH protocol. Installed on the Ubuntu Server to enable remote management from the Ubuntu Desktop using SSH.
- ***Google Chrome / Mozilla Firefox*** - These are web browsers acting as HTTP clients that send requests to web servers and display responses. We used Mozilla Firefox installed on the Ubuntu Desktop to access the web page hosted by the Nginx server on the Ubuntu Server.
- ***OpenSSH Client*** – It enables users to securely connect to an SSH server for remote management. It is pre-installed on the Ubuntu Desktop and is used to remotely access the Ubuntu Server running the OpenSSH Server.

2. *Conducted Output Tests on Sections 4 and 5*

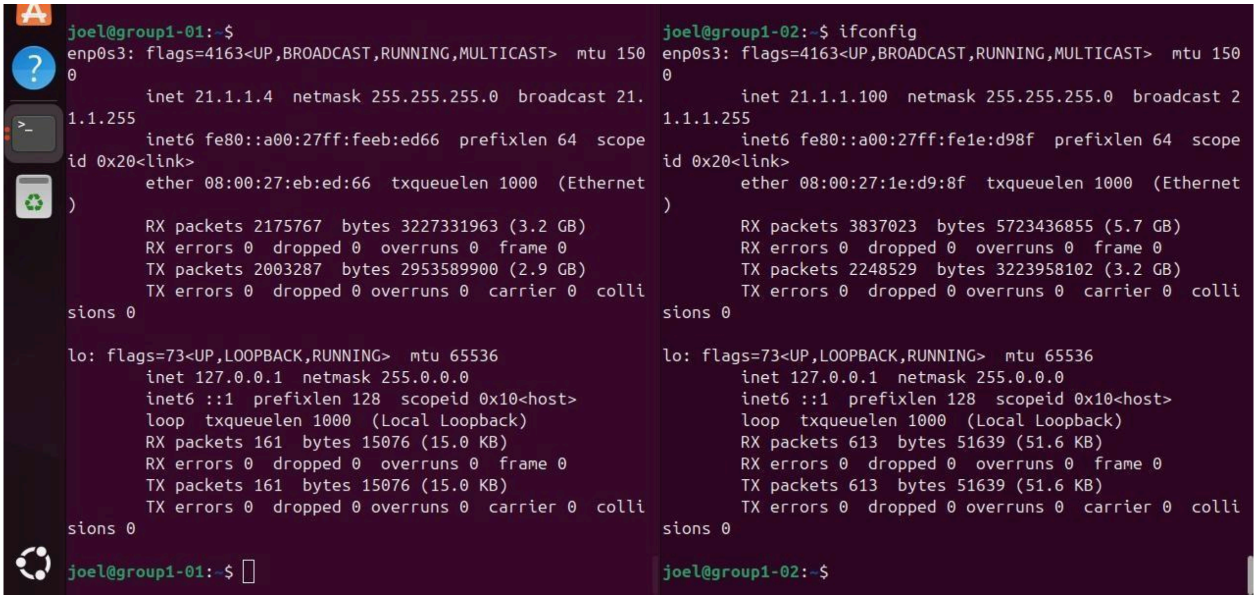
4.1 Verifying IP Addresses Assigned Statically



```
+sssshhhyNMMNyssssssssssssyNMMMysssssss+   Resolution: 128
ossyNMMMNyMMhsssssssssssssshmmmhssssssso   DE: GNOME 46.0
ossyNMMMNyMMhsssssssssssssshmmmhssssssso   WM: Mutter
+sssshhhyNMMNyssssssssssssyNMMMysssssss+   WM Theme: Adwai
.ssssssssdMMMNhsssssssssshNMMMdssssssss.   Theme: Yaru [GT
 /sssssssshNMMMyhhyyyyhdNMMMNhssssssss/    Icons: Yaru [GT
  +sssssssssdmydMMMMMMMMddddyssssssss+     Terminal: gnome
   /sssssssssssshdmNNNNmyNMMMMhssssss/     CPU: 11th Gen I
    .ossssssssssssssssssdMMMNysssso.       GPU: 00:02.0 VM
      -+sssssssssssssssssyyyssss+-         Memory: 836MiB
        `:+ssssssssssssssssss+:`
            .-/+oossssoo+/-.

joel@group1-01:~$
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
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
 fq_codel state UP group default qlen 1000
    link/ether 08:00:27:eb:ed:66 brd ff:ff:ff:ff:ff:ff
    inet 21.1.1.4/24 brd 21.1.1.255 scope global dynamic no
prefixroute enp0s3
       valid_lft 467sec preferred_lft 467sec
    inet6 fe80::a00:27ff:feeb:ed66/64 scope link
       valid_lft forever preferred_lft forever
joel@group1-01:~$
```

```
OS: Ubuntu 24.04.1 LTS x86_
Host: VirtualBox 1.2
Kernel: 6.8.0-51-generic
Uptime: 1 hour, 23 mins
Packages: 964 (dpkg)
Shell: bash 5.2.21
Resolution: 1280x800
Terminal: /dev/pts/0
CPU: 11th Gen Intel i7-1137
GPU: 00:02.0 VMware SVGA II
Memory: 201MiB / 5825MiB

joel@group1-02:~$ ip a
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
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
 fq_codel state UP group default qlen 1000
    link/ether 08:00:27:1e:d9:8f brd ff:ff:ff:ff:ff:ff
    inet 21.1.1.100/24 brd 21.1.1.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe1e:d98f/64 scope link
       valid_lft forever preferred_lft forever
joel@group1-02:~$
```

The screenshot shows the output of the *ip a* command executed on two different hosts, displaying system information and confirming the IP addresses assigned statically to both the Ubuntu Desktop and Ubuntu Server.

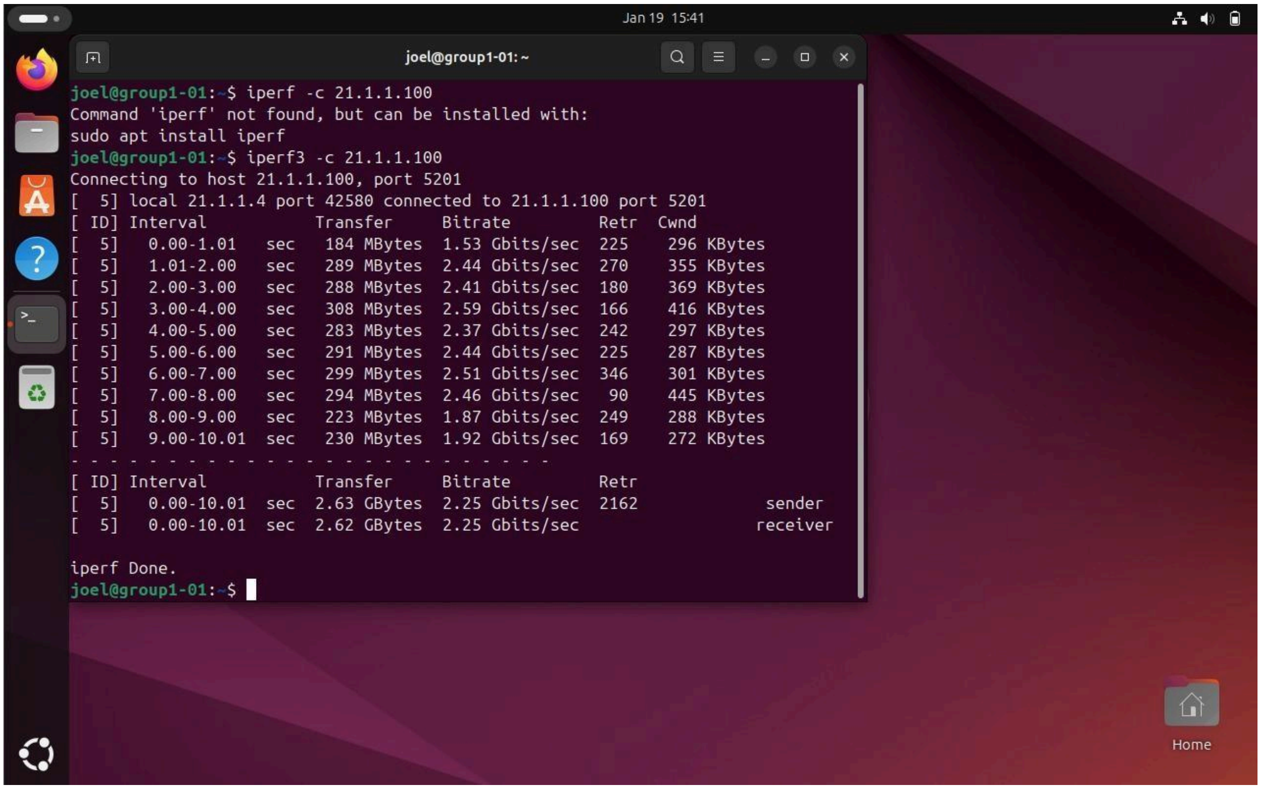


```
joel@group1-01:~$
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 150
0
        inet 21.1.1.4  netmask 255.255.255.0  broadcast 21.
1.1.255
        inet6 fe80::a00:27ff:feeb:ed66  prefixlen 64  scope
id 0x20<link>
        ether 08:00:27:eb:ed:66  txqueuelen 1000  (Ethernet
)
        RX packets 2175767  bytes 3227331963 (3.2 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2003287  bytes 2953589900 (2.9 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  colli
sions 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 161  bytes 15076 (15.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 161  bytes 15076 (15.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  colli
sions 0

joel@group1-01:~$
```

```
joel@group1-02:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 150
0
        inet 21.1.1.100  netmask 255.255.255.0  broadcast 2
1.1.1.255
        inet6 fe80::a00:27ff:fe1e:d98f  prefixlen 64  scope
id 0x20<link>
        ether 08:00:27:1e:d9:8f  txqueuelen 1000  (Ethernet
)
        RX packets 3837023  bytes 5723436855 (5.7 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2248529  bytes 3223958102 (3.2 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  colli
sions 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 613  bytes 51639 (51.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 613  bytes 51639 (51.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  colli
sions 0

joel@group1-02:~$
```

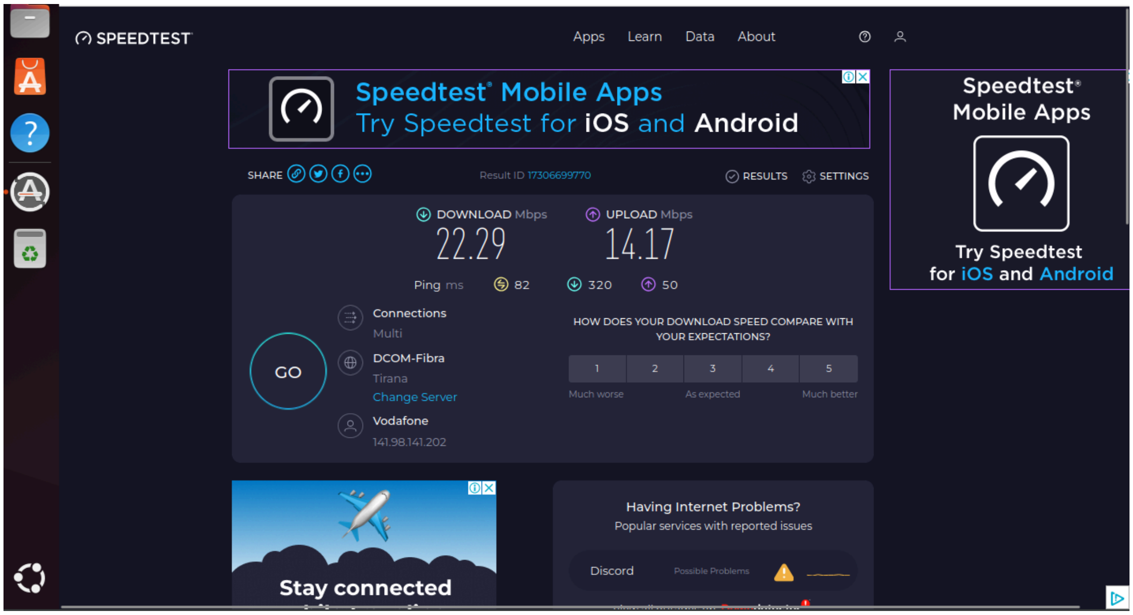
The screenshot displays the output of the *ifconfig* command executed on two different hosts. It includes detailed network interface configurations for *enp0s3* and *lo* interfaces on both hosts. For the *enp0s*3 interface, the IP addresses assigned statically are shown, confirming correct network configurations. Additionally, it includes MAC addresses, RX and TX packet statistics, and IPv6 addresses. This verifies that the IP addresses are correctly assigned statically, and the network interfaces are functioning properly on both the Ubuntu Desktop and the Ubuntu Server.

4.2 Measuring throughput



```
joel@group1-01:~$ iperf -c 21.1.1.100
Command 'iperf' not found, but can be installed with:
sudo apt install iperf
joel@group1-01:~$ iperf3 -c 21.1.1.100
Connecting to host 21.1.1.100, port 5201
[  5] local 21.1.1.4 port 42580 connected to 21.1.1.100 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.01   sec   184 MBytes  1.53 Gbits/sec  225    296 KBytes
[  5]   1.01-2.00   sec   289 MBytes  2.44 Gbits/sec  270    355 KBytes
[  5]   2.00-3.00   sec   288 MBytes  2.41 Gbits/sec  180    369 KBytes
[  5]   3.00-4.00   sec   308 MBytes  2.59 Gbits/sec  166    416 KBytes
[  5]   4.00-5.00   sec   283 MBytes  2.37 Gbits/sec  242    297 KBytes
[  5]   5.00-6.00   sec   291 MBytes  2.44 Gbits/sec  225    287 KBytes
[  5]   6.00-7.00   sec   299 MBytes  2.51 Gbits/sec  346    301 KBytes
[  5]   7.00-8.00   sec   294 MBytes  2.46 Gbits/sec   90    445 KBytes
[  5]   8.00-9.00   sec   223 MBytes  1.87 Gbits/sec  249    288 KBytes
[  5]   9.00-10.01  sec   230 MBytes  1.92 Gbits/sec  169    272 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.01  sec  2.63 GBytes  2.25 Gbits/sec  2162             sender
[  5]   0.00-10.01  sec  2.62 GBytes  2.25 Gbits/sec                  receiver

iperf Done.
joel@group1-01:~$ 
```

The screenshot shows the terminal window on the Ubuntu Desktop where the *iperf3* command was used to test the network bandwidth between the Ubuntu Desktop and the Ubuntu Server. The command *iperf3 -c 21.1.1.100* initiates the test, connecting to the host at IP address *21.1.1.100* on port 5201. The results show the transfer rate and bandwidth performance over a 10 second interval. This test verifies the local network connectivity between the end hosts by measuring the network's bandwidth and performance.



Result ID 17306699770

DOWNLOAD Mbps 22.29

UPLOAD Mbps 14.17

Ping ms 82 320 50

Connections Multi

DCOM-Fibra Tirana

Vodafone 141.98.141.202

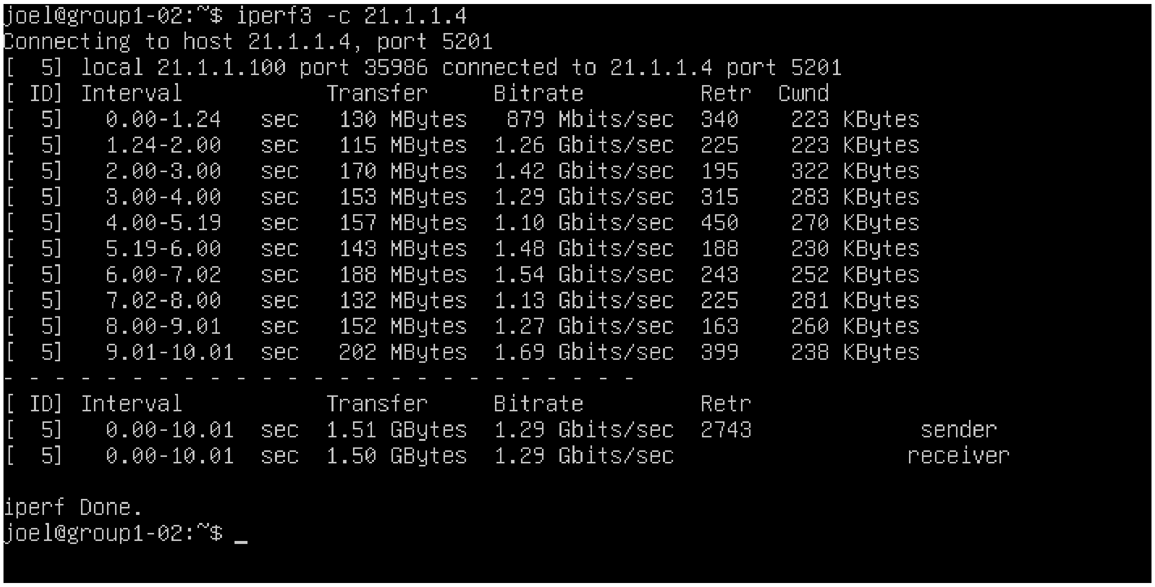
HOW DOES YOUR DOWNLOAD SPEED COMPARE WITH YOUR EXPECTATIONS?

1	2	3	4	5
Much worse		As expected		Much better

Having Internet Problems?
Popular services with reported issues

Discord Possible Problems

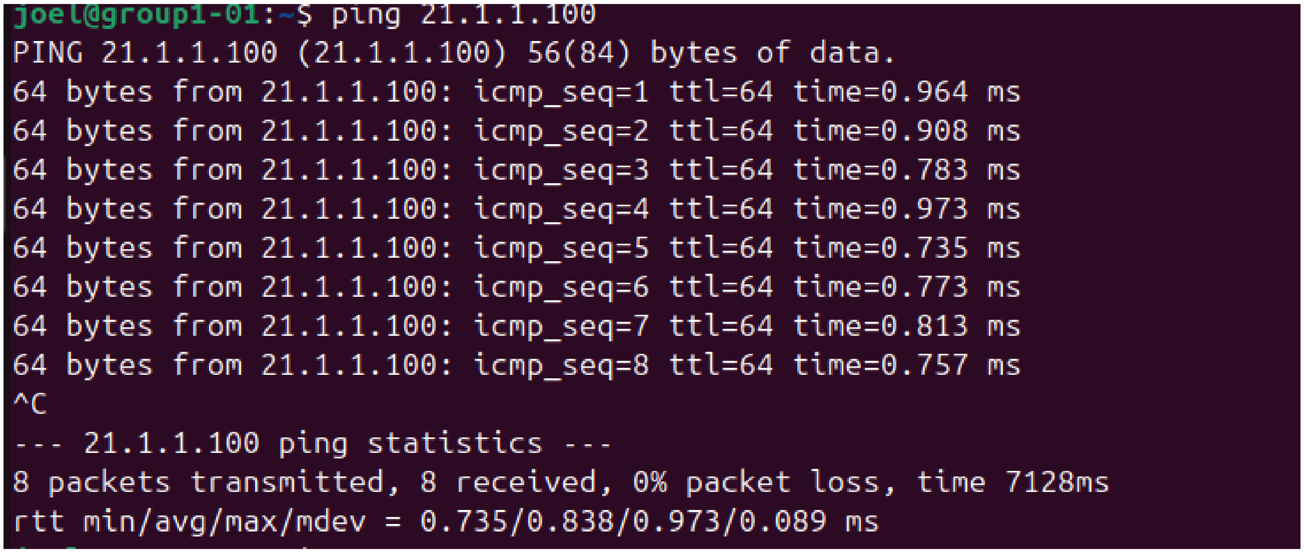
Measuring throughput from the server



```
joel@group1-02:~$ iperf3 -c 21.1.1.4
Connecting to host 21.1.1.4, port 5201
[  5] local 21.1.1.100 port 35986 connected to 21.1.1.4 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.24   sec   130 MBytes   879 Mbits/sec  340    223 KBytes
[  5]   1.24-2.00   sec   115 MBytes  1.26 Gbits/sec  225    223 KBytes
[  5]   2.00-3.00   sec   170 MBytes  1.42 Gbits/sec  195    322 KBytes
[  5]   3.00-4.00   sec   153 MBytes  1.29 Gbits/sec  315    283 KBytes
[  5]   4.00-5.19   sec   157 MBytes  1.10 Gbits/sec  450    270 KBytes
[  5]   5.19-6.00   sec   143 MBytes  1.48 Gbits/sec  188    230 KBytes
[  5]   6.00-7.02   sec   188 MBytes  1.54 Gbits/sec  243    252 KBytes
[  5]   7.02-8.00   sec   132 MBytes  1.13 Gbits/sec  225    281 KBytes
[  5]   8.00-9.01   sec   152 MBytes  1.27 Gbits/sec  163    260 KBytes
[  5]   9.01-10.01  sec   202 MBytes  1.69 Gbits/sec  399    238 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.01  sec  1.51 GBytes  1.29 Gbits/sec  2743             sender
[  5]   0.00-10.01  sec  1.50 GBytes  1.29 Gbits/sec                  receiver

iperf Done.
joel@group1-02:~$ _
```

The screenshot shows the output of a *ping* command directed at IP address 21.1.1.4 to verify if the target host is reachable and to measure the response time. Additionally, it also shows the *iperf3* command on the Ubuntu Server

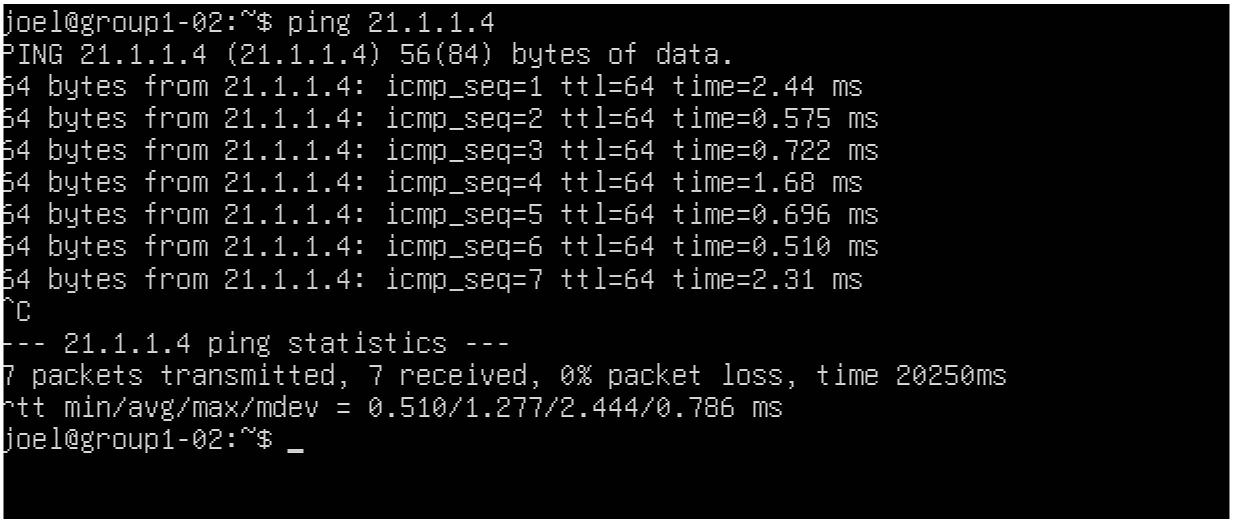
4.3 Using ping command to measure RTT



```
joel@group1-01:~$ ping 21.1.1.100
PING 21.1.1.100 (21.1.1.100) 56(84) bytes of data.
64 bytes from 21.1.1.100: icmp_seq=1 ttl=64 time=0.964 ms
64 bytes from 21.1.1.100: icmp_seq=2 ttl=64 time=0.908 ms
64 bytes from 21.1.1.100: icmp_seq=3 ttl=64 time=0.783 ms
64 bytes from 21.1.1.100: icmp_seq=4 ttl=64 time=0.973 ms
64 bytes from 21.1.1.100: icmp_seq=5 ttl=64 time=0.735 ms
64 bytes from 21.1.1.100: icmp_seq=6 ttl=64 time=0.773 ms
64 bytes from 21.1.1.100: icmp_seq=7 ttl=64 time=0.813 ms
64 bytes from 21.1.1.100: icmp_seq=8 ttl=64 time=0.757 ms
^C
--- 21.1.1.100 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7128ms
rtt min/avg/max/mdev = 0.735/0.838/0.973/0.089 ms
```

The screenshot shows the results of a **ping** command directed at the server from the desktop, measuring the **round-trip time** for packets sent to the public end host. The results indicate successful external network connectivity by displaying response times for each packet.

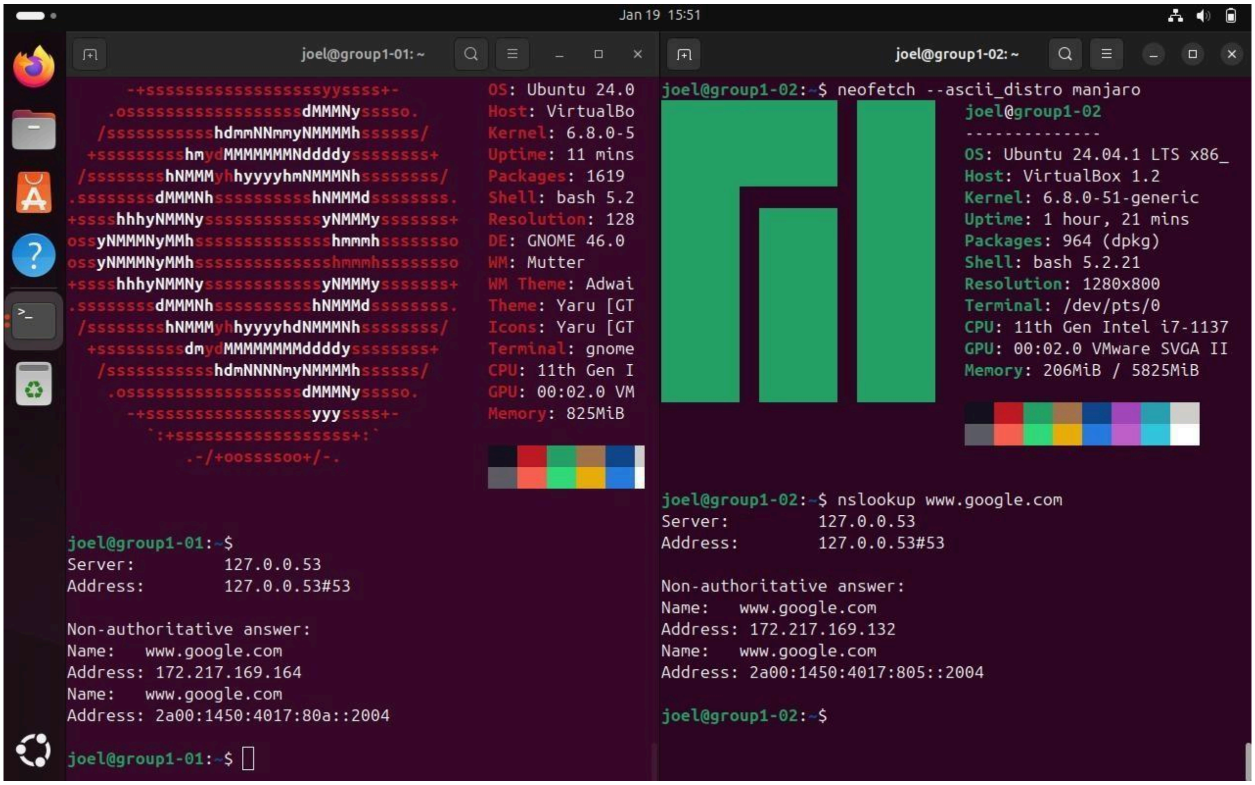
Server



```
joel@group1-02:~$ ping 21.1.1.4
PING 21.1.1.4 (21.1.1.4) 56(84) bytes of data.
64 bytes from 21.1.1.4: icmp_seq=1 ttl=64 time=2.44 ms
64 bytes from 21.1.1.4: icmp_seq=2 ttl=64 time=0.575 ms
64 bytes from 21.1.1.4: icmp_seq=3 ttl=64 time=0.722 ms
64 bytes from 21.1.1.4: icmp_seq=4 ttl=64 time=1.68 ms
64 bytes from 21.1.1.4: icmp_seq=5 ttl=64 time=0.696 ms
64 bytes from 21.1.1.4: icmp_seq=6 ttl=64 time=0.510 ms
64 bytes from 21.1.1.4: icmp_seq=7 ttl=64 time=2.31 ms
^C
--- 21.1.1.4 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 20250ms
rtt min/avg/max/mdev = 0.510/1.277/2.444/0.786 ms
joel@group1-02:~$ _
```

The server view of the same *ping* command but this time from the server to the desktop.

4.4 Performing DNS Request/Response

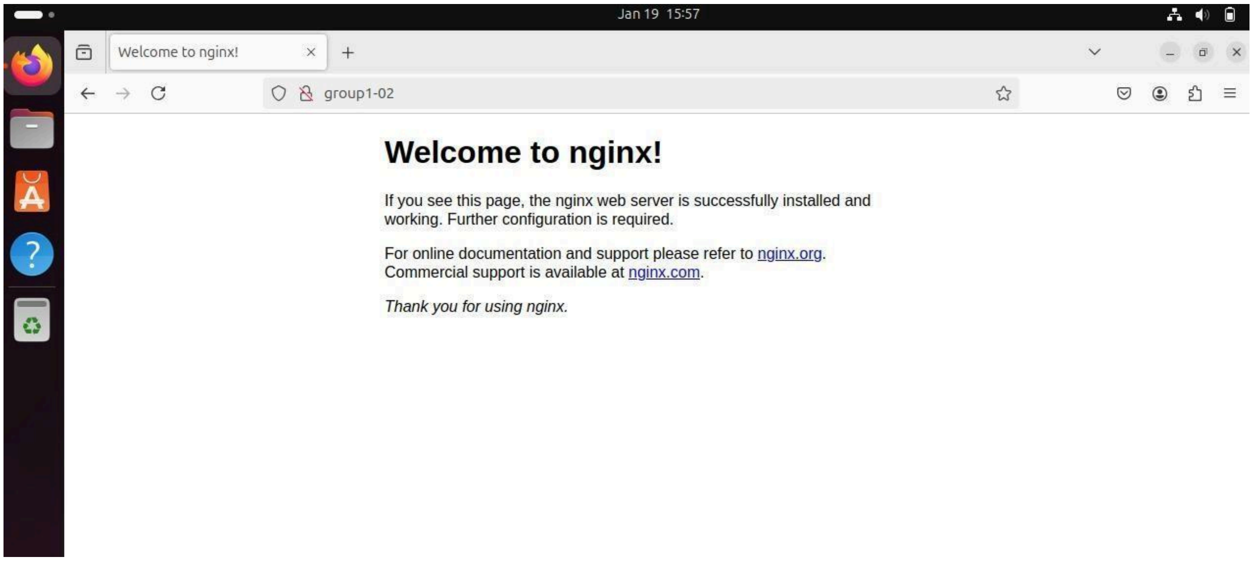


```
      -+ssssssssssssssssssyyssss+-       OS: Ubuntu 24.0
    .ossssssssssssssssssdMMMNysssso.     Host: VirtualBo
   /ssssssssssshdmmNNmmyNMMMMhssssss/    Kernel: 6.8.0-5
  +ssssssssshmydMMMMMMMNddddyssssssss+   Uptime: 11 mins
 /sssssssshNMMMyhhyyyyhmNMMMNhssssssss/  Packages: 1619
.ssssssssdMMMNhsssssssssshNMMMdssssssss. Shell: bash 5.2
+sssshhhyNMMNyssssssssssssyNMMMysssssss+ Resolution: 128
ossyNMMMNyMMhsssssssssssssshmmmhssssssso DE: GNOME 46.0
ossyNMMMNyMMhsssssssssssssshmmmhssssssso WM: Mutter
+sssshhhyNMMNyssssssssssssyNMMMysssssss+ WM Theme: Adwai
.ssssssssdMMMNhsssssssssshNMMMdssssssss. Theme: Yaru [GT
 /sssssssshNMMMyhhyyyyhdNMMMNhssssssss/  Icons: Yaru [GT
  +sssssssssdmydMMMMMMMMddddyssssssss+   Terminal: gnome
   /ssssssssssshdmNNNNmyNMMMMhssssss/    CPU: 11th Gen I
    .ossssssssssssssssssdMMMNysssso.     GPU: 00:02.0 VM
      -+sssssssssssssssssyyyssss+-       Memory: 825MiB
        `:+ssssssssssssssssss+:`
            .-/+oossssoo+/-.


joel@group1-01:~$
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.169.164
Name:   www.google.com
Address: 2a00:1450:4017:80a::2004

joel@group1-01:~$
```

```
joel@group1-02:~$ neofetch --ascii_distro manjaro
                               joel@group1-02
                               --------------
                               OS: Ubuntu 24.04.1 LTS x86_
                               Host: VirtualBox 1.2
                               Kernel: 6.8.0-51-generic
                               Uptime: 1 hour, 21 mins
                               Packages: 964 (dpkg)
                               Shell: bash 5.2.21
                               Resolution: 1280x800
                               Terminal: /dev/pts/0
                               CPU: 11th Gen Intel i7-1137
                               GPU: 00:02.0 VMware SVGA II
                               Memory: 206MiB / 5825MiB


joel@group1-02:~$ nslookup www.google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.169.132
Name:   www.google.com
Address: 2a00:1450:4017:805::2004

joel@group1-02:~$
```

The screenshot shows the results of the *nslookup* command executed on two different hosts. The command queries the DNS for the IP address corresponding to the hostname *www.google.com*, verifying that DNS is accessible and responds correctly to the requests from both local end hosts.

5.1 Accessing Nginx Web Page



Welcome to nginx!

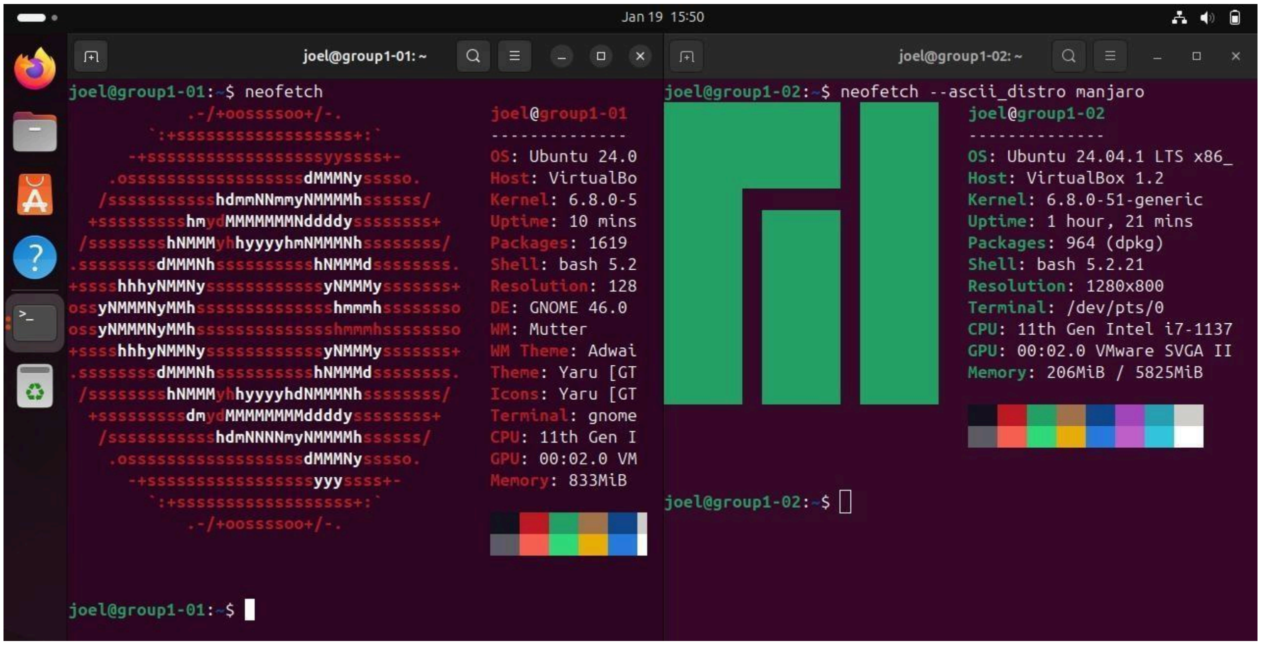
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

The screenshot shows the default welcome page for the nginx web server accessed from a web browser. This confirms that nginx is successfully installed and operational on the Ubuntu Server.

5.2 OpenSSH



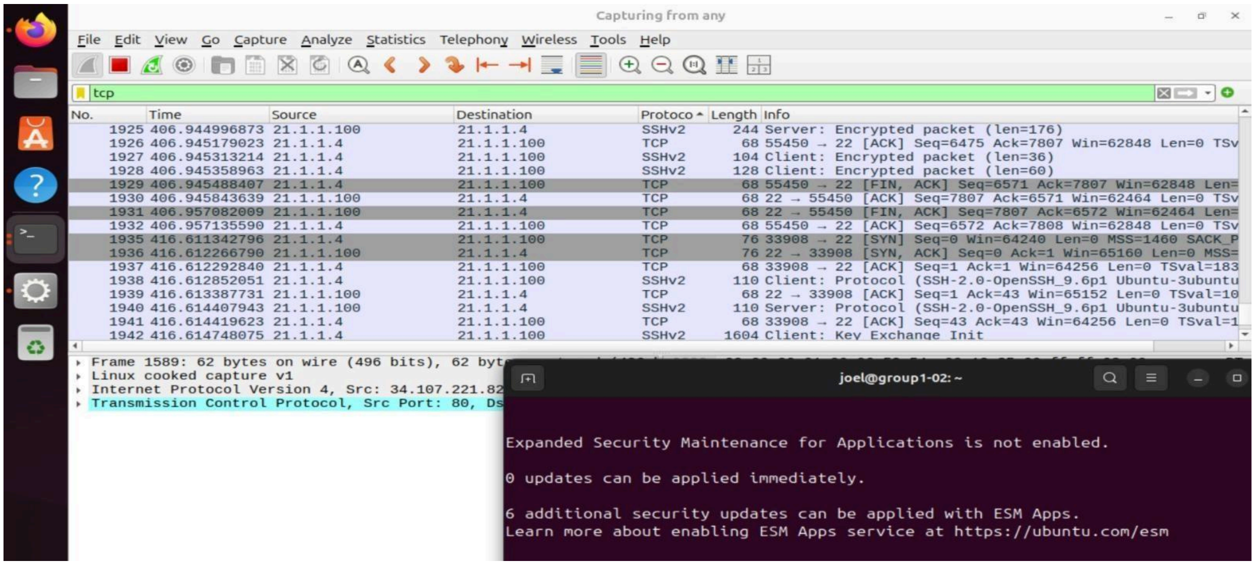
```
joel@group1-01:~$ neofetch
            .-/+oossssoo+/-.               joel@group1-01
        `:+ssssssssssssssssss+:`           --------------
      -+ssssssssssssssssssyyssss+-         OS: Ubuntu 24.0
    .ossssssssssssssssssdMMMNysssso.       Host: VirtualBo
   /ssssssssssshdmmNNmmyNMMMMhssssss/      Kernel: 6.8.0-5
  +ssssssssshmydMMMMMMMNddddyssssssss+     Uptime: 10 mins
 /sssssssshNMMMyhhyyyyhmNMMMNhssssssss/    Packages: 1619
.ssssssssdMMMNhsssssssssshNMMMdssssssss.   Shell: bash 5.2
+sssshhhyNMMNyssssssssssssyNMMMysssssss+   Resolution: 128
ossyNMMMNyMMhsssssssssssssshmmmhssssssso   DE: GNOME 46.0
ossyNMMMNyMMhsssssssssssssshmmmhssssssso   WM: Mutter
+sssshhhyNMMNyssssssssssssyNMMMysssssss+   WM Theme: Adwai
.ssssssssdMMMNhsssssssssshNMMMdssssssss.   Theme: Yaru [GT
 /sssssssshNMMMyhhyyyyhdNMMMNhssssssss/    Icons: Yaru [GT
  +sssssssssdmydMMMMMMMMddddyssssssss+     Terminal: gnome
   /ssssssssssshdmNNNNmyNMMMMhssssss/      CPU: 11th Gen I
    .ossssssssssssssssssdMMMNysssso.       GPU: 00:02.0 VM
      -+sssssssssssssssssyyyssss+-         Memory: 833MiB
        `:+ssssssssssssssssss+:`
            .-/+oossssoo+/-.


joel@group1-01:~$
```

```
joel@group1-02:~$ neofetch --ascii_distro manjaro
                    joel@group1-02
                    --------------
                    OS: Ubuntu 24.04.1 LTS x86_
                    Host: VirtualBox 1.2
                    Kernel: 6.8.0-51-generic
                    Uptime: 1 hour, 21 mins
                    Packages: 964 (dpkg)
                    Shell: bash 5.2.21
                    Resolution: 1280x800
                    Terminal: /dev/pts/0
                    CPU: 11th Gen Intel i7-1137
                    GPU: 00:02.0 VMware SVGA II
                    Memory: 206MiB / 5825MiB


joel@group1-02:~$
```

The screenshot mainly shows the connection between the Desktop and the Server through OpenSSH. It also shows the output of the *neofetch* command executed on two different hosts.

3. *Network Performance Measures and Wireshark Outputs for Section 7*

Three-way Handshake

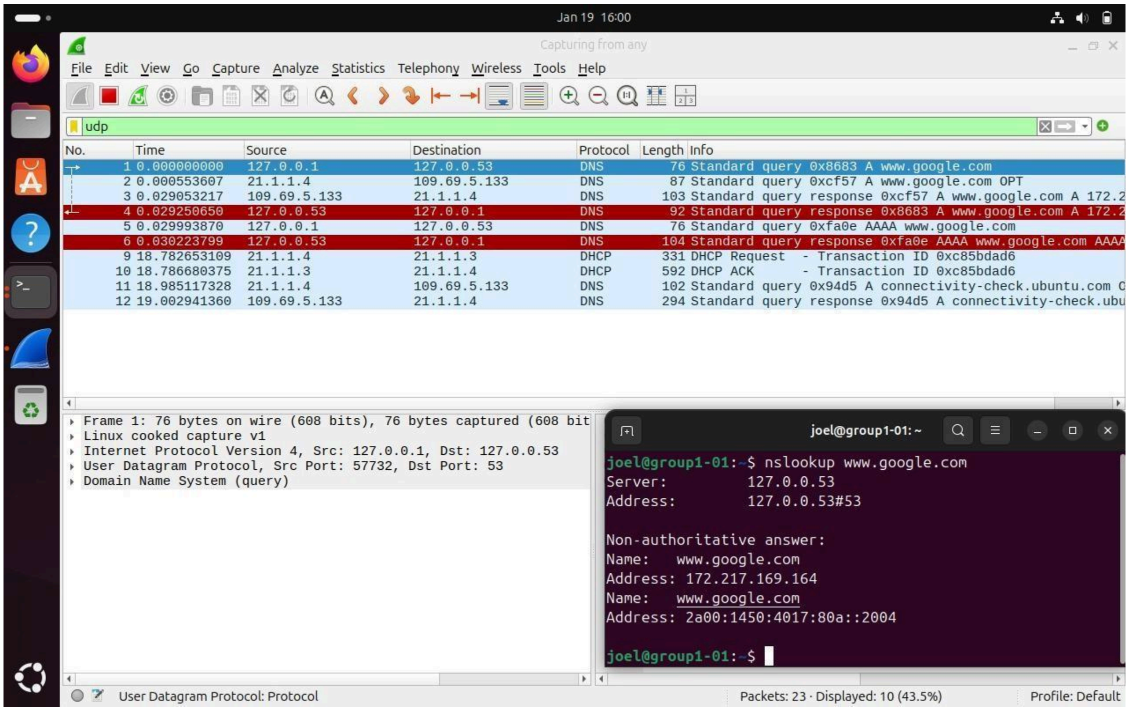

Capturing from any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
joel@group1-02: ~

```
tcp
No.   Time         Source       Destination  Protoco Length Info
 1925 406.944996873 21.1.1.100  21.1.1.4     SSHv2    244 Server: Encrypted packet (len=176)
 1926 406.945179023 21.1.1.4    21.1.1.100   TCP       68 55450 → 22 [ACK] Seq=6475 Ack=7807 Win=62848 Len=0 TSv
 1927 406.945313214 21.1.1.4    21.1.1.100   SSHv2    104 Client: Encrypted packet (len=36)
 1928 406.945358963 21.1.1.4    21.1.1.100   SSHv2    128 Client: Encrypted packet (len=60)
 1929 406.945488407 21.1.1.4    21.1.1.100   TCP       68 55450 → 22 [FIN, ACK] Seq=6571 Ack=7807 Win=62848 Len=
 1930 406.945843639 21.1.1.100  21.1.1.4     TCP       68 22 → 55450 [ACK] Seq=7807 Ack=6571 Win=62464 Len=0 TSv
 1931 406.957082009 21.1.1.100  21.1.1.4     TCP       68 22 → 55450 [FIN, ACK] Seq=7807 Ack=6572 Win=62464 Len=
 1932 406.957135590 21.1.1.4    21.1.1.100   TCP       68 55450 → 22 [ACK] Seq=6572 Ack=7808 Win=62848 Len=0 TSv
 1935 416.611342796 21.1.1.4    21.1.1.100   TCP       76 33908 → 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_P
 1936 416.612266790 21.1.1.100  21.1.1.4     TCP       76 22 → 33908 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=
 1937 416.612292840 21.1.1.4    21.1.1.100   TCP       68 33908 → 22 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=183
 1938 416.612852051 21.1.1.4    21.1.1.100   SSHv2    110 Client: Protocol (SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu
 1939 416.613387731 21.1.1.100  21.1.1.4     TCP       68 22 → 33908 [ACK] Seq=1 Ack=43 Win=65152 Len=0 TSval=10
 1940 416.614407943 21.1.1.100  21.1.1.4     SSHv2    110 Server: Protocol (SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu
 1941 416.614419623 21.1.1.4    21.1.1.100   TCP       68 33908 → 22 [ACK] Seq=43 Ack=43 Win=64256 Len=0 TSval=1
 1942 416.614748075 21.1.1.4    21.1.1.100   SSHv2   1604 Client: Key Exchange Init

▸ Frame 1589: 62 bytes on wire (496 bits), 62 byt
▸ Linux cooked capture v1
▸ Internet Protocol Version 4, Src: 34.107.221.82
▸ Transmission Control Protocol, Src Port: 80, Ds
```

```
Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

6 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm
```

The screenshot shows the Wireshark network packet analyzer capturing the TCP three-way handshake packets between two hosts. The captured packets include SYN, SYN-ACK, and ACK flags, which are essential for establishing a TCP connection. This process ensures reliable communication between the Ubuntu Desktop and the Ubuntu Server.

DNS Request/Response Packets

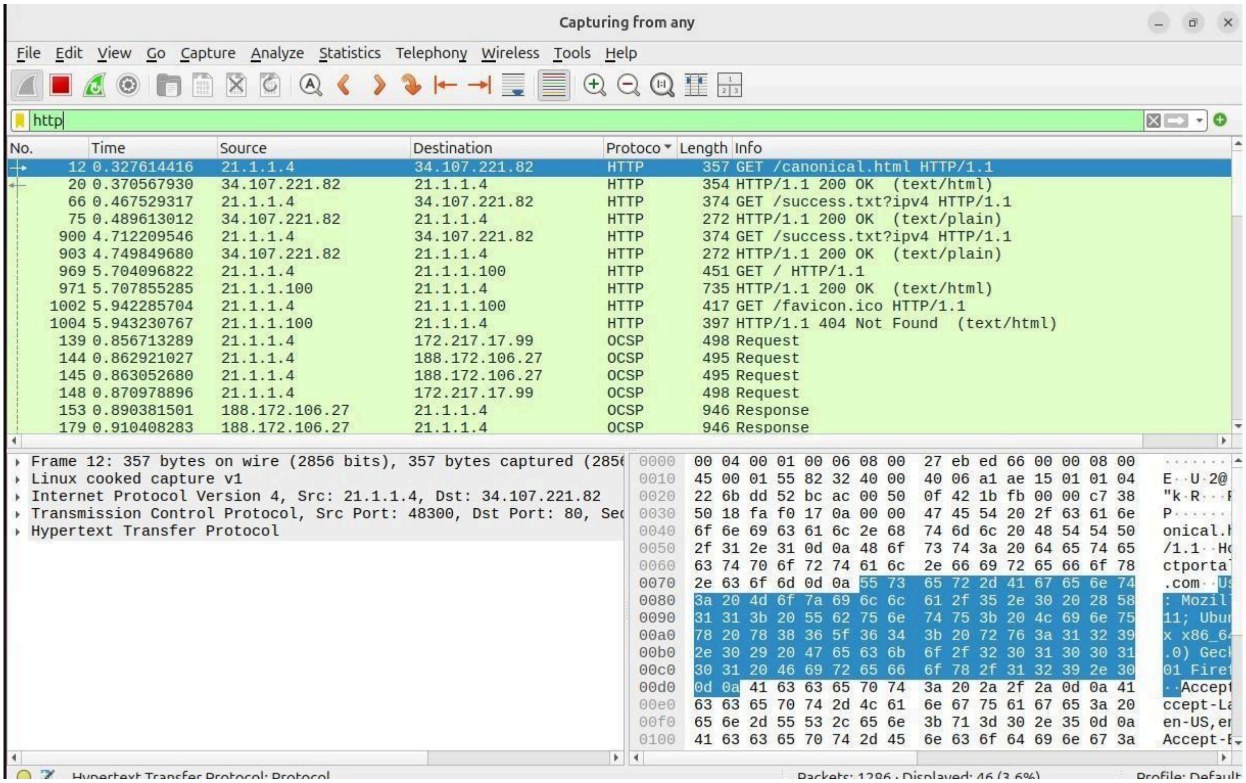

Jan 19 16:00
Capturing from any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
joel@group1-01: ~

```
udp
No.  Time         Source        Destination   Protocol Length Info
   1 0.000000000  127.0.0.1     127.0.0.53    DNS       76 Standard query 0x8683 A www.google.com
   2 0.000553607  21.1.1.4      109.69.5.133  DNS       87 Standard query 0xcf57 A www.google.com OPT
   3 0.029053217  109.69.5.133  21.1.1.4      DNS      103 Standard query response 0xcf57 A www.google.com A 172.2
   4 0.029250650  127.0.0.53    127.0.0.1     DNS       92 Standard query response 0x8683 A www.google.com A 172.2
   5 0.029993870  127.0.0.1     127.0.0.53    DNS       76 Standard query 0xfa0e AAAA www.google.com
   6 0.030223799  127.0.0.53    127.0.0.1     DNS      104 Standard query response 0xfa0e AAAA www.google.com AAAA
   9 18.782653109 21.1.1.4      21.1.1.3      DHCP     331 DHCP Request  - Transaction ID 0xc85bdad6
  10 18.786680375 21.1.1.3      21.1.1.4      DHCP     592 DHCP ACK      - Transaction ID 0xc85bdad6
  11 18.985117328 21.1.1.4      109.69.5.133  DNS      102 Standard query 0x94d5 A connectivity-check.ubuntu.com C
  12 19.002941360 109.69.5.133  21.1.1.4      DNS      294 Standard query response 0x94d5 A connectivity-check.ubu

▸ Frame 1: 76 bytes on wire (608 bits), 76 bytes captured (608 bit
▸ Linux cooked capture v1
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.53
▸ User Datagram Protocol, Src Port: 57732, Dst Port: 53
▸ Domain Name System (query)

User Datagram Protocol: Protocol        Packets: 23 · Displayed: 10 (43.5%)        Profile: Default
```

```
joel@group1-01:~$ nslookup www.google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.169.164
Name:   www.google.com
Address: 2a00:1450:4017:80a::2004

joel@group1-01:~$
```

The screenshot shows the use of Wireshark to capture DNS request and response packets. It demonstrates the process of resolving the domain name *www.google.com* to its associated IP addresses by displaying the DNS queries and responses. The terminal output confirms the execution of the *nslookup www.google.com* command, and Wireshark captures the corresponding DNS traffic between the local host and the DNS server.

HTTP Request/Response Packets


Capturing from any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http

```
No.   Time         Source          Destination     Protoco Length Info
   12 0.327614416  21.1.1.4        34.107.221.82   HTTP       357 GET /canonical.html HTTP/1.1
   20 0.370567930  34.107.221.82   21.1.1.4        HTTP       354 HTTP/1.1 200 OK  (text/html)
   66 0.467529317  21.1.1.4        34.107.221.82   HTTP       374 GET /success.txt?ipv4 HTTP/1.1
   75 0.489613012  34.107.221.82   21.1.1.4        HTTP       272 HTTP/1.1 200 OK  (text/plain)
  900 4.712209546  21.1.1.4        34.107.221.82   HTTP       374 GET /success.txt?ipv4 HTTP/1.1
  903 4.749849680  34.107.221.82   21.1.1.4        HTTP       272 HTTP/1.1 200 OK  (text/plain)
  969 5.704096822  21.1.1.4        21.1.1.100      HTTP       451 GET / HTTP/1.1
  971 5.707855285  21.1.1.100      21.1.1.4        HTTP       735 HTTP/1.1 200 OK  (text/html)
 1002 5.942285704  21.1.1.4        21.1.1.100      HTTP       417 GET /favicon.ico HTTP/1.1
 1004 5.943230767  21.1.1.100      21.1.1.4        HTTP       397 HTTP/1.1 404 Not Found  (text/html)
  139 0.856713289  21.1.1.4        172.217.17.99   OCSP       498 Request
  144 0.862921027  21.1.1.4        188.172.106.27  OCSP       495 Request
  145 0.863052680  21.1.1.4        188.172.106.27  OCSP       495 Request
  148 0.870978896  21.1.1.4        172.217.17.99   OCSP       498 Request
  153 0.890381501  188.172.106.27  21.1.1.4        OCSP       946 Response
  179 0.910408283  188.172.106.27  21.1.1.4        OCSP       946 Response
```

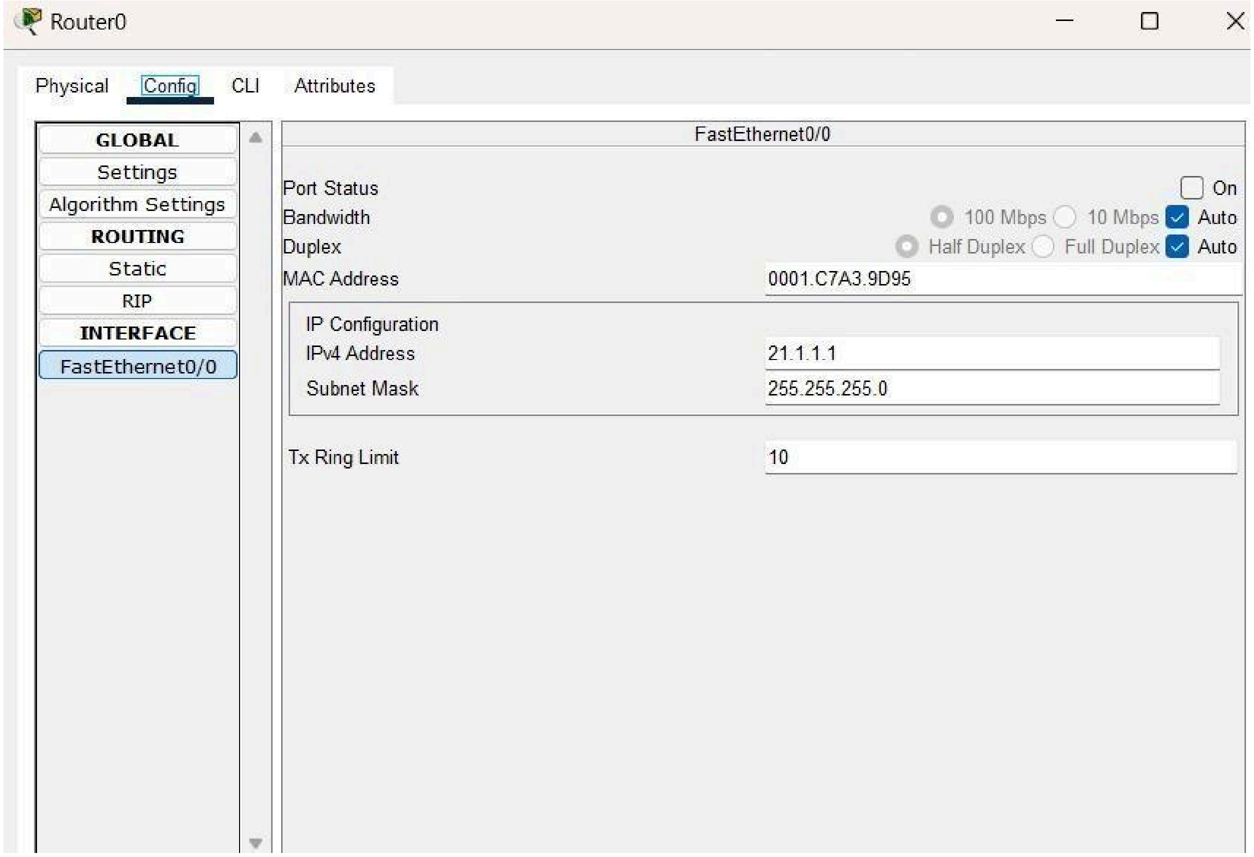
```
▸ Frame 12: 357 bytes on wire (2856 bits), 357 bytes captured (2856
▸ Linux cooked capture v1
▸ Internet Protocol Version 4, Src: 21.1.1.4, Dst: 34.107.221.82
▸ Transmission Control Protocol, Src Port: 48300, Dst Port: 80, Sec
▸ Hypertext Transfer Protocol
```

```
0000  00 04 00 01 00 06 08 00  27 eb ed 66 00 00 08 00   ········
0010  45 00 01 55 82 32 40 00  40 06 a1 ae 15 01 01 04   E··U·2@
0020  22 6b dd 52 bc ac 00 50  0f 42 1b fb 00 00 c7 38   "k·R···F
0030  50 18 fa f0 17 0a 00 00  47 45 54 20 2f 63 61 6e   P·······
0040  6f 6e 69 63 61 6c 2e 68  74 6d 6c 20 48 54 54 50   onical.h
0050  2f 31 2e 31 0d 0a 48 6f  73 74 3a 20 64 65 74 65   /1.1··Ho
0060  63 74 70 6f 72 74 61 6c  2e 66 69 72 65 66 6f 78   ctporta
0070  2e 63 6f 6d 0d 0a 55 73  65 72 2d 41 67 65 6e 74   .com··Us
0080  3a 20 4d 6f 7a 69 6c 6c  61 2f 35 2e 30 20 28 58   : Mozil
0090  31 31 3b 20 55 62 75 6e  74 75 3b 20 4c 69 6e 75   11; Ubun
00a0  78 20 78 38 36 5f 36 34  3b 20 72 76 3a 31 32 39   x x86_64
00b0  2e 30 29 20 47 65 63 6b  6f 2f 32 30 31 30 30 31   .0) Geck
00c0  30 31 20 46 69 72 65 66  6f 78 2f 31 32 39 2e 30   01 Firef
00d0  0d 0a 41 63 63 65 70 74  3a 20 2a 2f 2a 0d 0a 41   ··Accept
00e0  63 63 65 70 74 2d 4c 61  6e 67 75 61 67 65 3a 20   ccept-La
00f0  65 6e 2d 55 53 2c 65 6e  3b 71 3d 30 2e 35 0d 0a   en-US,en
0100  41 63 63 65 70 74 2d 45  6e 63 6f 64 69 6e 67 3a   Accept-E
```

Hypertext Transfer Protocol: Protocol   Packets: 1286 · Displayed: 46 (3.6%)   Profile: Default

The screenshot shows Wireshark capturing HTTP traffic. It displays packets exchanged between a web browser on the Ubuntu Desktop and an HTTP server. The selected packet is an HTTP GET request for a specific resource. The detailed view includes protocol headers and data, demonstrating how the packet is structured and transmitted over the network.

4. *Cisco Packet Tracer*



FastEthernet0/0

Port Status □ On

Bandwidth ○ 100 Mbps ○ 10 Mbps ☑ Auto

Duplex ○ Half Duplex ○ Full Duplex ☑ Auto

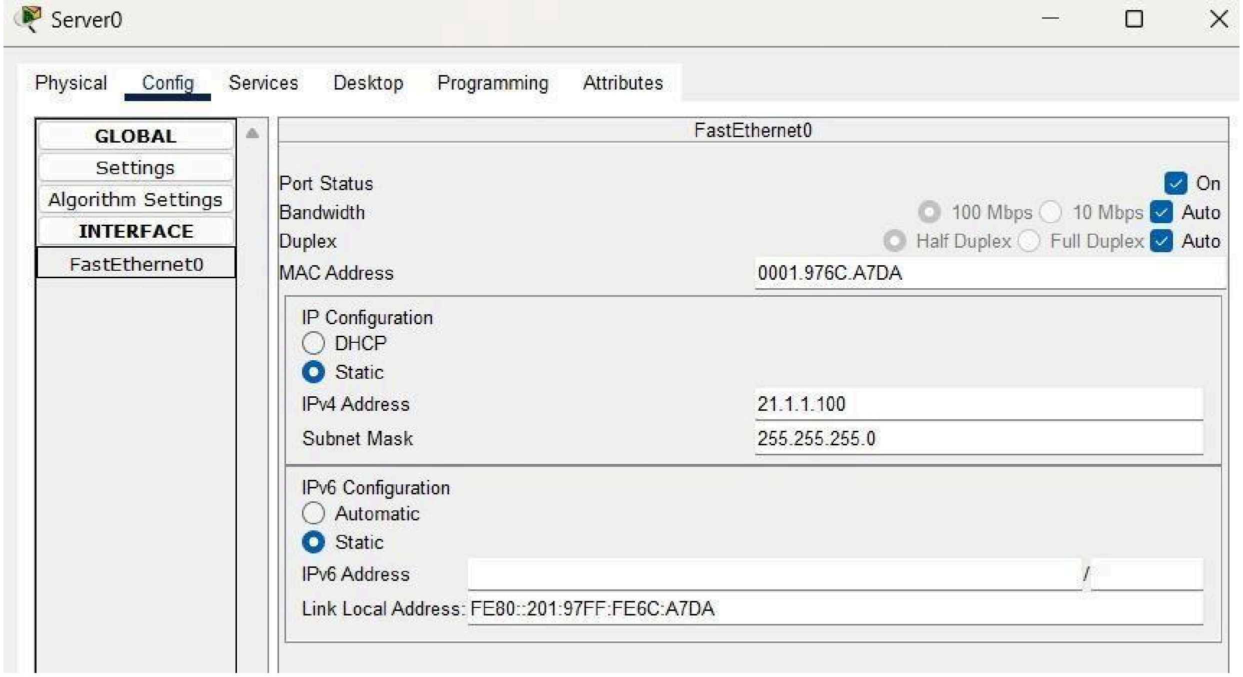
MAC Address 0001.C7A3.9D95

IP Configuration

IPv4 Address 21.1.1.1

Subnet Mask 255.255.255.0

Tx Ring Limit 10



FastEthernet0

Port Status ☑ On

Bandwidth ○ 100 Mbps ○ 10 Mbps ☑ Auto

Duplex ○ Half Duplex ○ Full Duplex ☑ Auto

MAC Address 0001.976C.A7DA

IP Configuration

○ DHCP

● Static

IPv4 Address 21.1.1.100

Subnet Mask 255.255.255.0

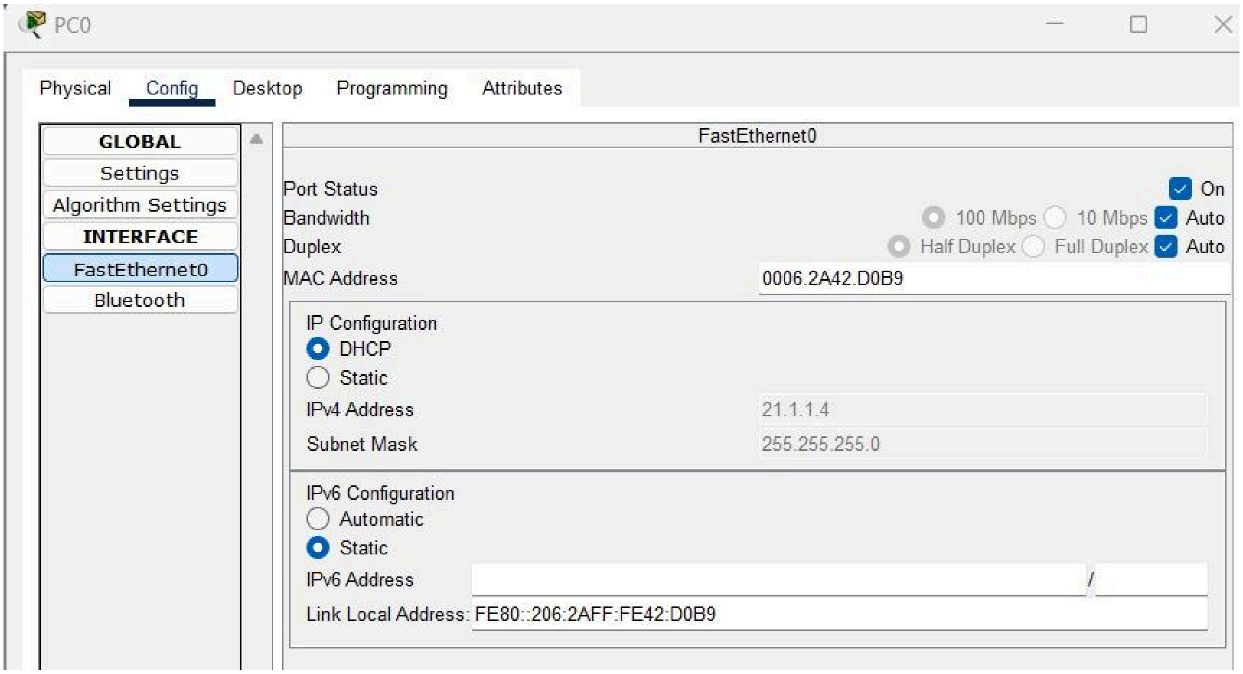
IPv6 Configuration

○ Automatic

● Static

IPv6 Address /

Link Local Address: FE80::201:97FF:FE6C:A7DA



FastEthernet0

Port Status ☑ On

Bandwidth ○ 100 Mbps ○ 10 Mbps ☑ Auto

Duplex ○ Half Duplex ○ Full Duplex ☑ Auto

MAC Address 0006.2A42.D0B9

IP Configuration

● DHCP

○ Static

IPv4 Address 21.1.1.4

Subnet Mask 255.255.255.0

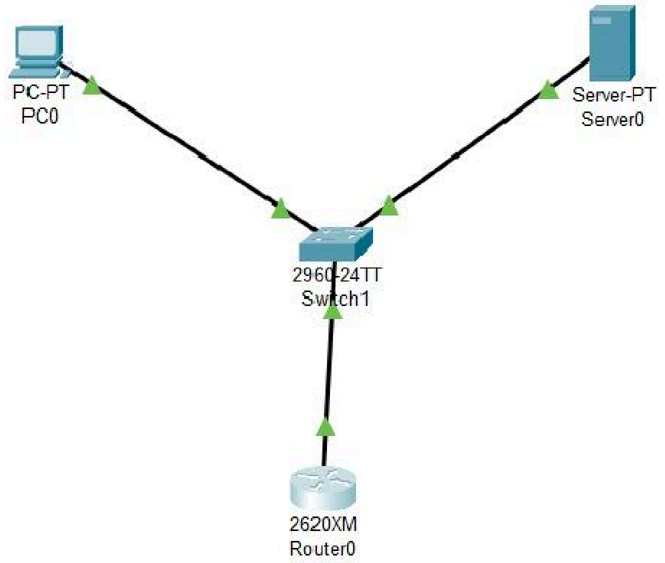
IPv6 Configuration

○ Automatic

● Static

IPv6 Address /

Link Local Address: FE80::206:2AFF:FE42:D0B9


PC-PT
PC0
Server-PT
Server0
2960-24TT
Switch1
2620XM
Router0