

MCAL27 - NETWORKING WITH LINUX LAB 2024-
2025

INDEX PAGE

Practical No.	Name	Div A Date	Div B Date	Sign/Remark
1	Problem Statement1 Installation: ns3 in Linux, NetAnim, Wireshark, PyViz, tcpdump	22/01/2025	23/01/2025	
2	Linux Network Commands – I. ifconfig, II. ip, III. ping, ns IV. netstat, V. Traceroute, VI. nslookup, VII. route, VIII. hostname.ns	16/02/2025	17/02/2025	

Programs – Simulate, Visualize, Animate the Network, Trace And Analyze the Captured Packets.

3	Program to simulate Point to Point topology	25/02/2025	24/02/2025	
4	Program to simulate Bus topology	20/02/2025	24/02/2025	
5	Program to simulate Star topology	05/03/2025	09/03/2025	
6	Program to simulate Mesh topology	20/03/2025	10/03/2025	
7	Program to simulate Hybrid topology	21/03/2025	28/03/2025	
8	Program to simulate UDP Client Server	08/04/2025	07/04/2025	
9	Program to simulate DHCP server and Clients	08/04/2025	07/04/2025	
10	Program to simulate FTP using TCP	08/04/2025	07/04/2025	
11	Exercises for analyzing the network protocols using Wireshark • Capture the packets while browsing the any web site • Analyze the header fields of various protocols	12/04/2025	12/04/2025	

12	Evaluate the network performance using metrics: throughput, delay, response time, packet loss, dropped packets etc. (Any Topology)	24/04/2025	16/04/2025	
13	Projects (Group of 3 to 4 students) – Complex Networks, CongestionControl Algorithms, MAC Protocols	26/04/2025	26/04/2025	

PRACTICAL NO. 1

Installing NS-3 in Ubuntu

Network simulator is a tool used for simulating the real world network on one computer by writing scripts in C++ or Python. Normally if we want to perform experiments, to see how our network works using various parameters. We don't have required number of computers and routers for making different topologies. Even if we have these resources it is very expensive to build such a network for experiment purposes.

So to overcome these drawbacks we used NS3, which is a discrete event network simulator for Internet. NS3 helps to create various virtual nodes (i.e., computers in real life) and with the help of various Helper classes it allows us to install devices, internet stacks, application, etc to our nodes.

Using NS3 we can create PointToPoint, Wireless, CSMA, etc connections between nodes. PointToPoint connection is same as a LAN connected between two computers. Wireless connection is same as WiFi connection between various computers and routers. CSMA connection is same as bus topology between computers. After building connections we try to install NIC to every node to enable network connectivity.

When network cards are enabled in the devices, we add different parameters in the channels (i.e., real world path used to send data) which are data-rate, packet size, etc. Now we use Application to generate traffic and send the packets using these applications.

Steps for installing NS-3 in Ubuntu

Step 1: Update the system

\$ sudo apt update

Step 2: Prerequisites for installing NS-3

```
$ sudo apt install build-essential autoconf automake libxmu-dev g++ python3
python3-dev pkg-config sqlite3 cmake python3-setuptools git qtbase5-dev
qtchooser qt5-qmake qtbase5-dev-tools gir1.2-goocanvas-2.0 python3-gi
python3-gi-cairo python3-pygraphviz gir1.2-qt5-3.0 ipython3 openmpi-bin
openmpi-common openmpi-doc libopenmpi-dev autoconf cvs bzr unrar gsl-bin
libgsl-dev libgslcblas0 wireshark tcpdump sqlite sqlite3 libssqlite3-dev libxml2
libxml2-dev libc6-dev libc6- dev-i386 libclang-dev llvm-dev automake python3-
pip libxml2 libxml2-dev libboost- all-de
```

Step 3 :Now download the ns3 3.35 from <https://nsnam.org>

Step 4 : Copy the softwares from the Downloads/ folder to the home folder (in my case its /home/ns-3/)

Now extract both the versions using the GUI method. Just right click and click "Extract Here"

Now we will install ns-3.35

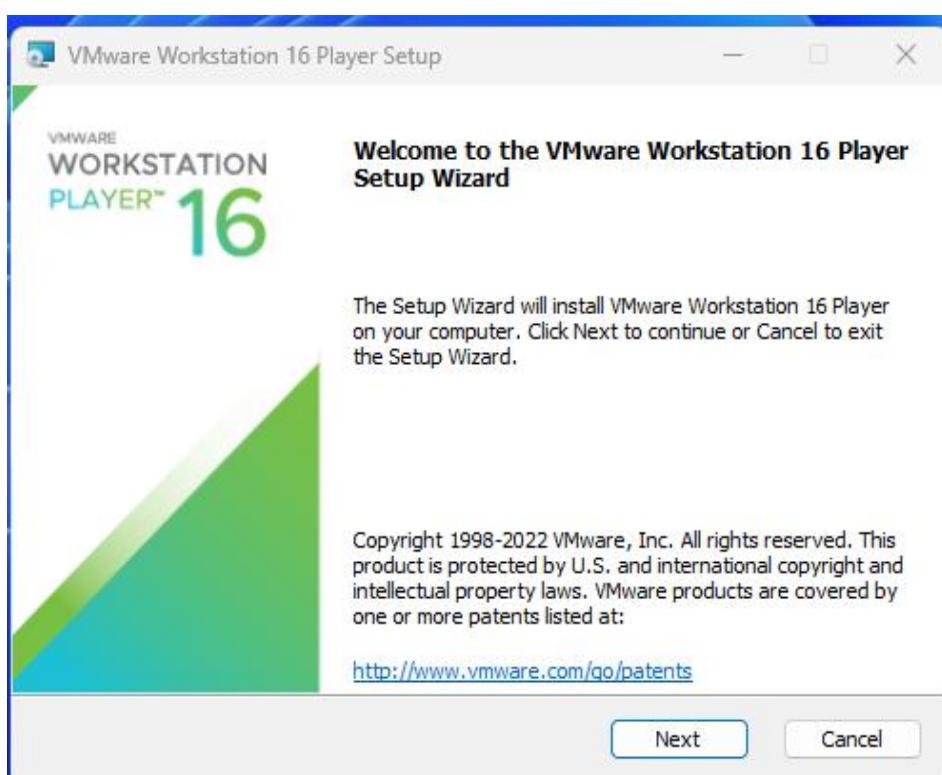
Step 5 :

```
$ cd
```

```
$ cd ns-allinone-3.35/
```

```
$ ./build.py --enable-examples --enable-tests
```

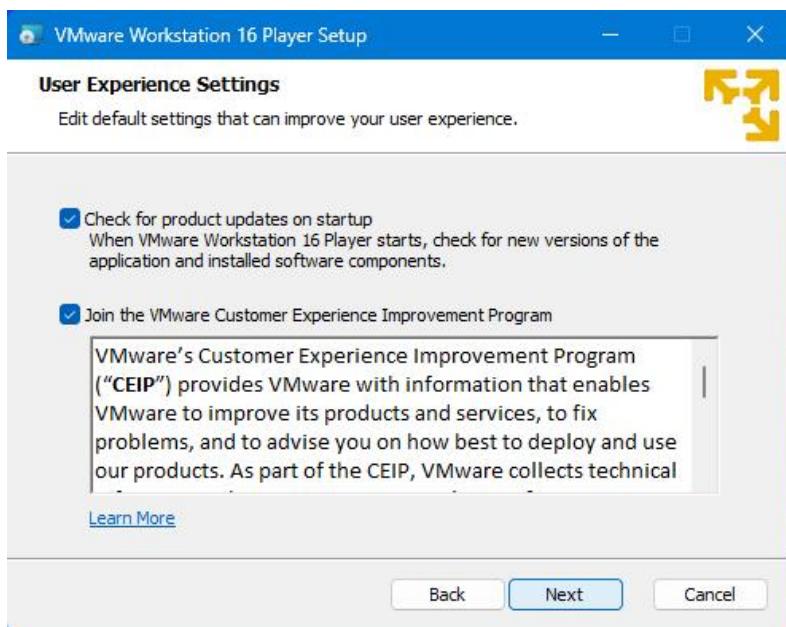
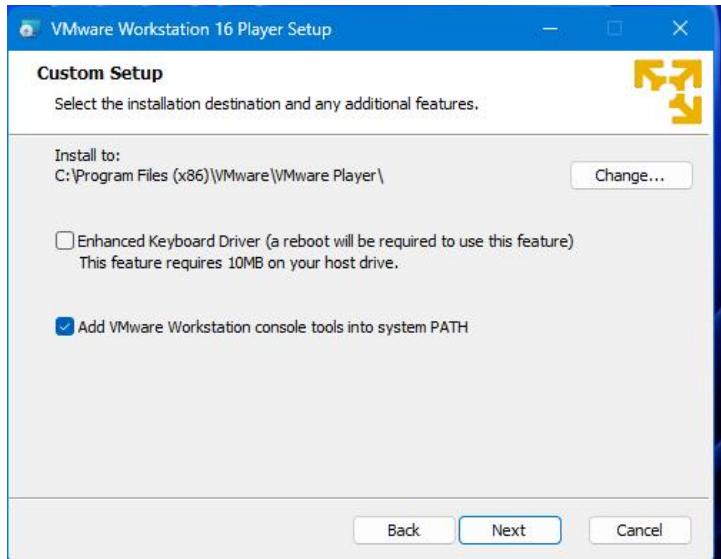
In case, if you get the following error pybindgen(ns3 module antenna) Do this step and repeat the above step





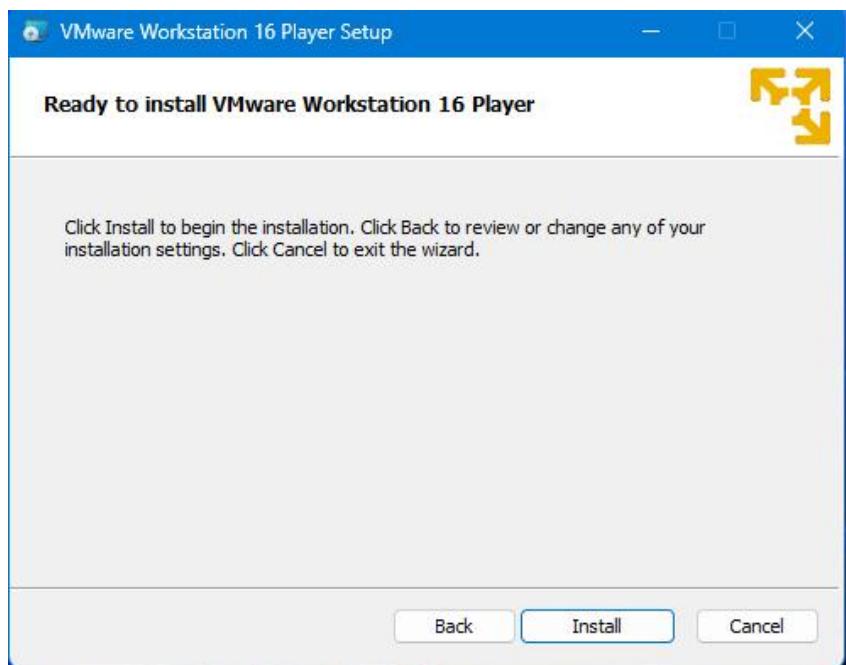
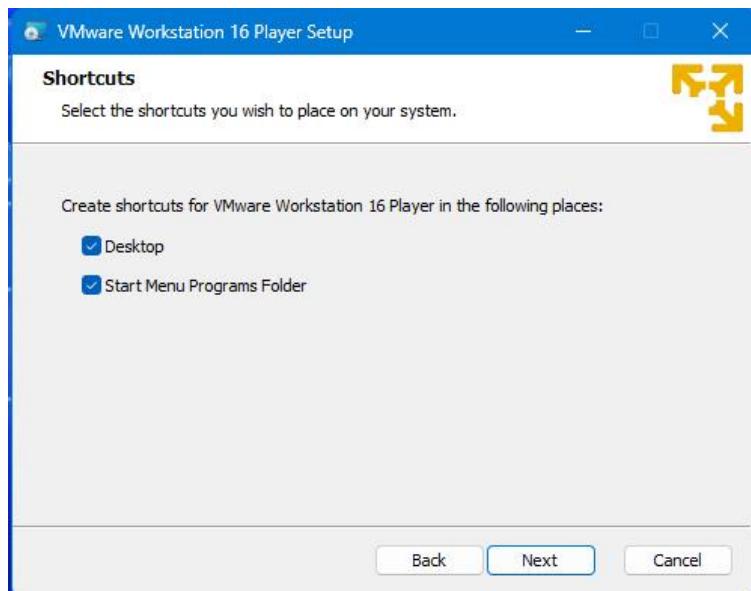
Create a New virtual Machine in VMWare.

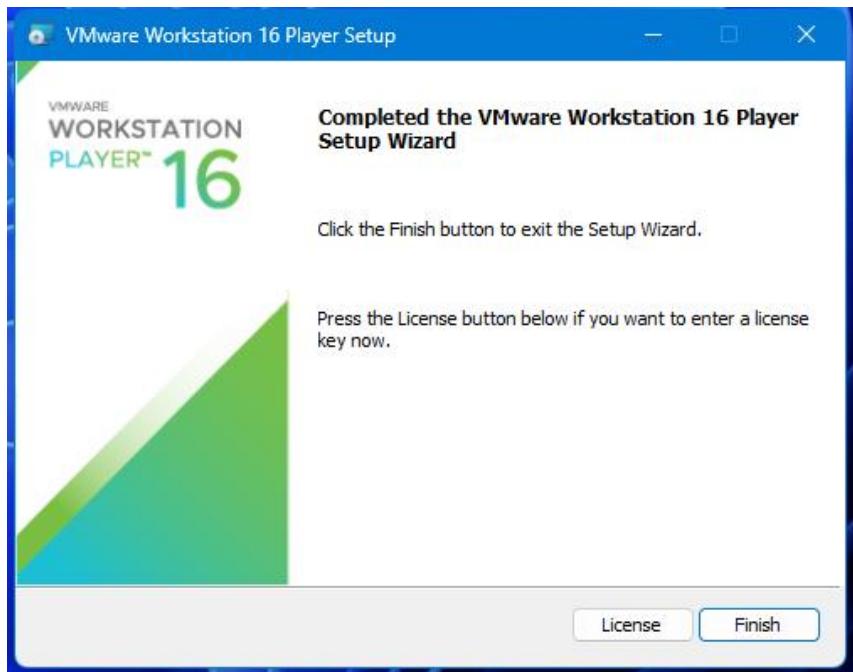
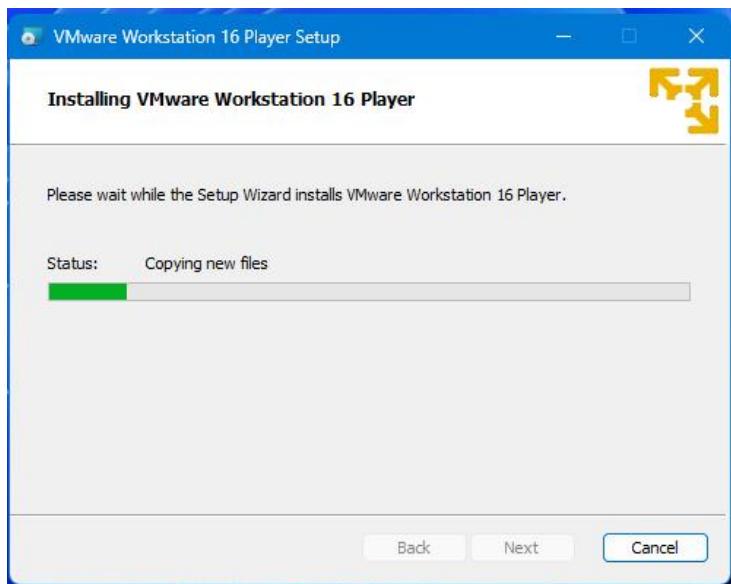
Screenshot shows the process of creating a new virtual machine in VMware for running Ubuntu.



Install the necessary Libraries like NS3, Netanime etc.

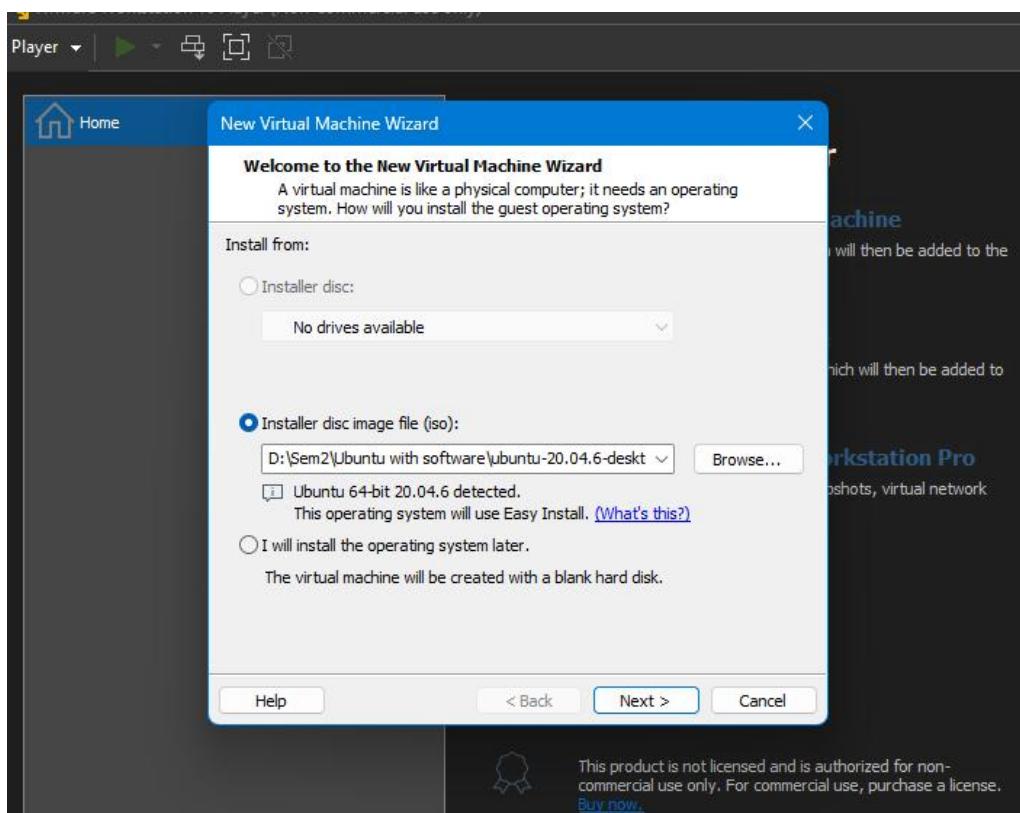
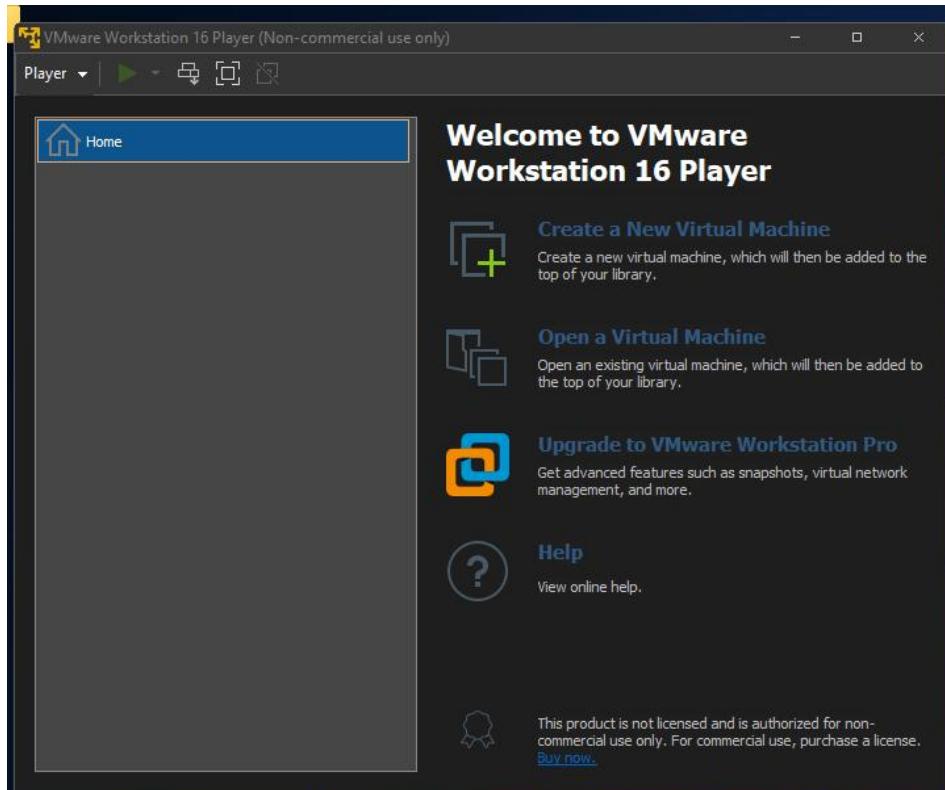
Screenshot displays installation of NS3 and NetAnim libraries through terminal commands in Ubuntu.

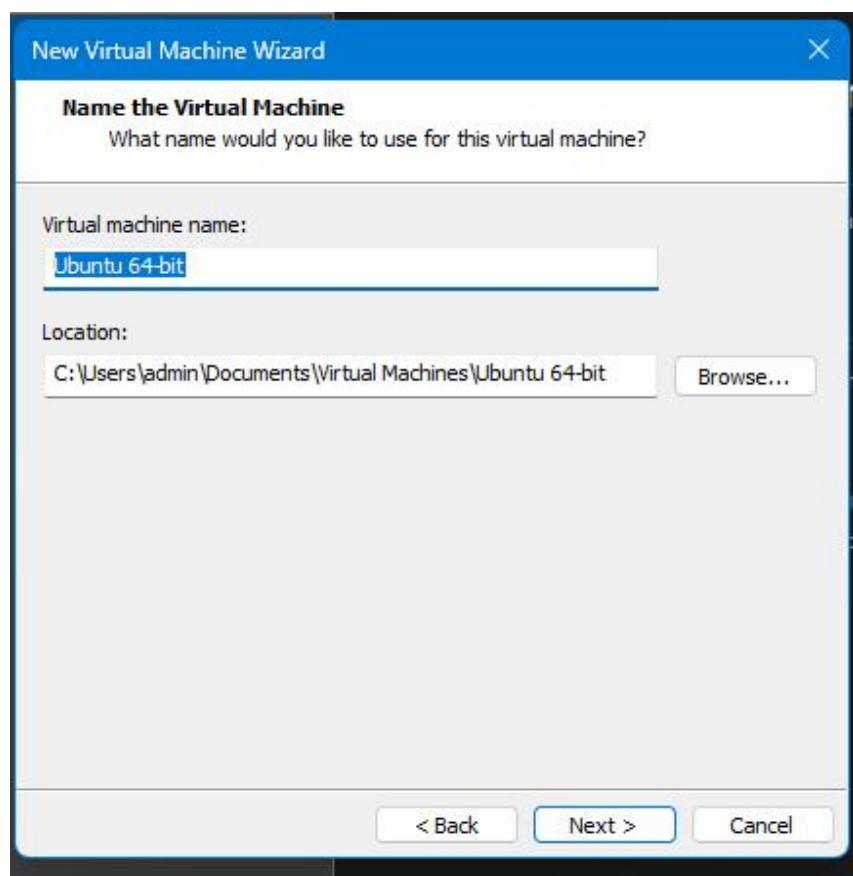
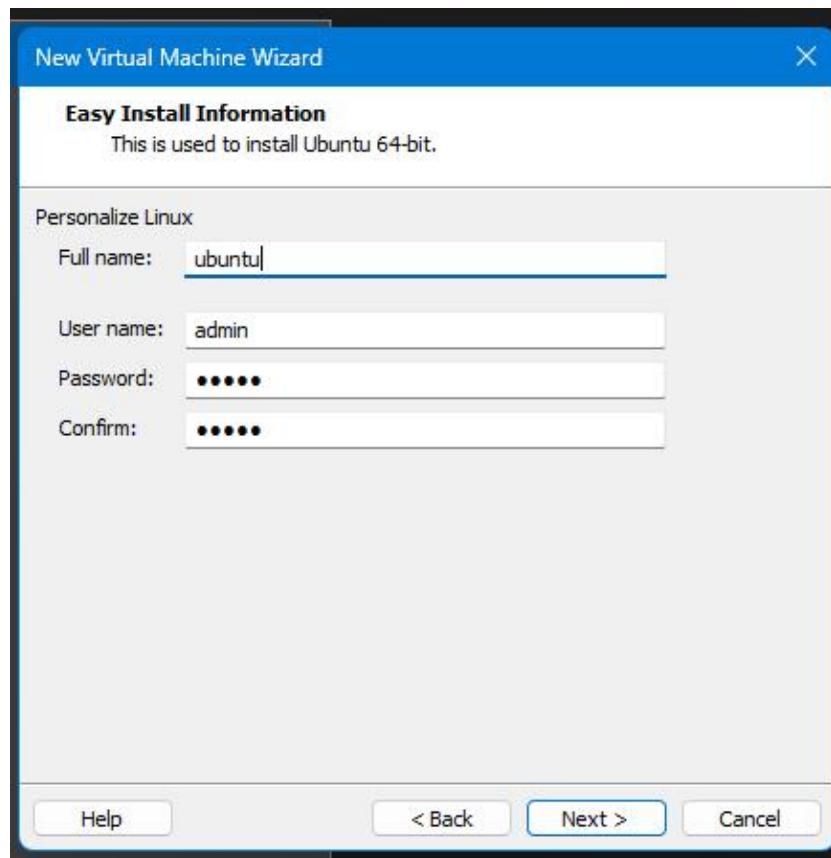


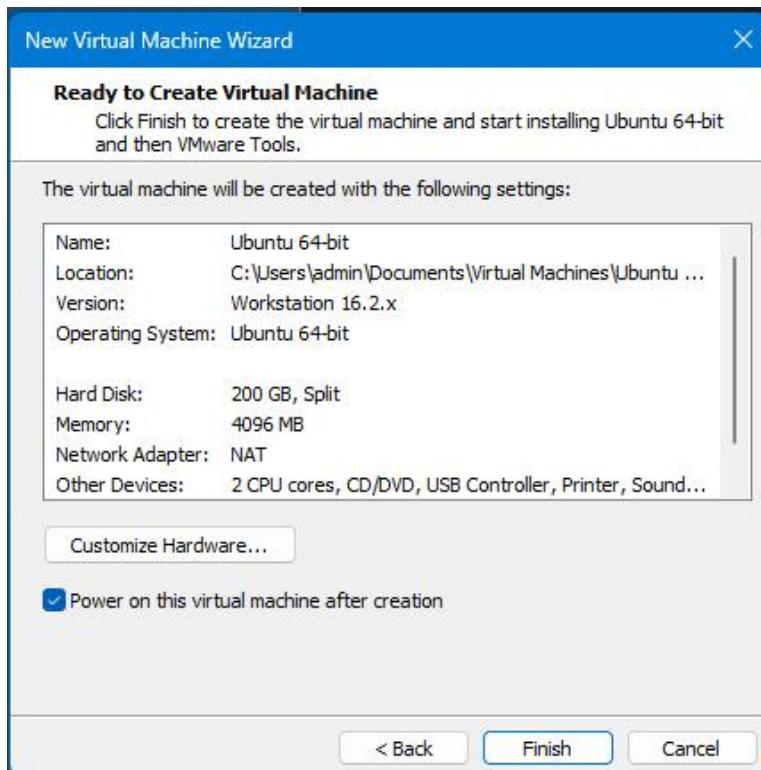
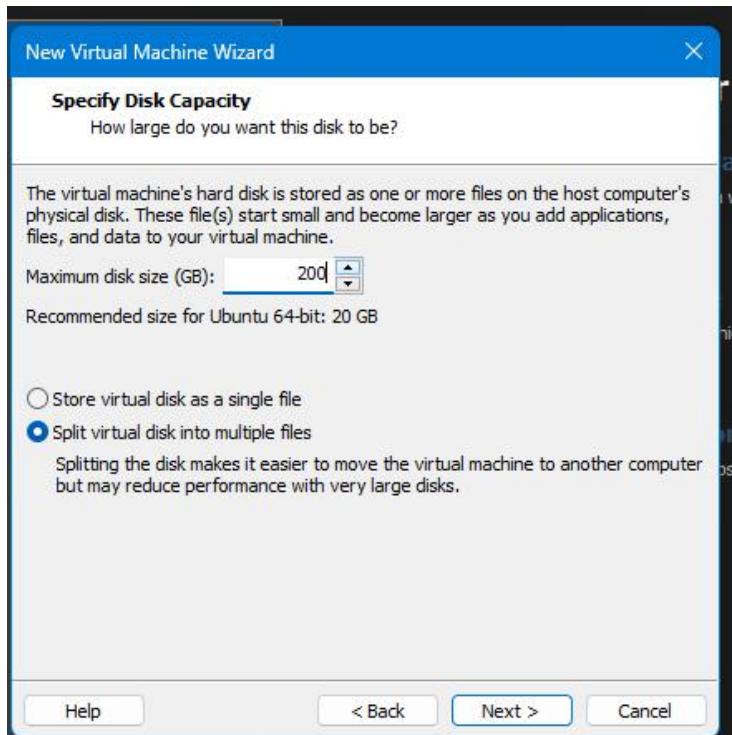


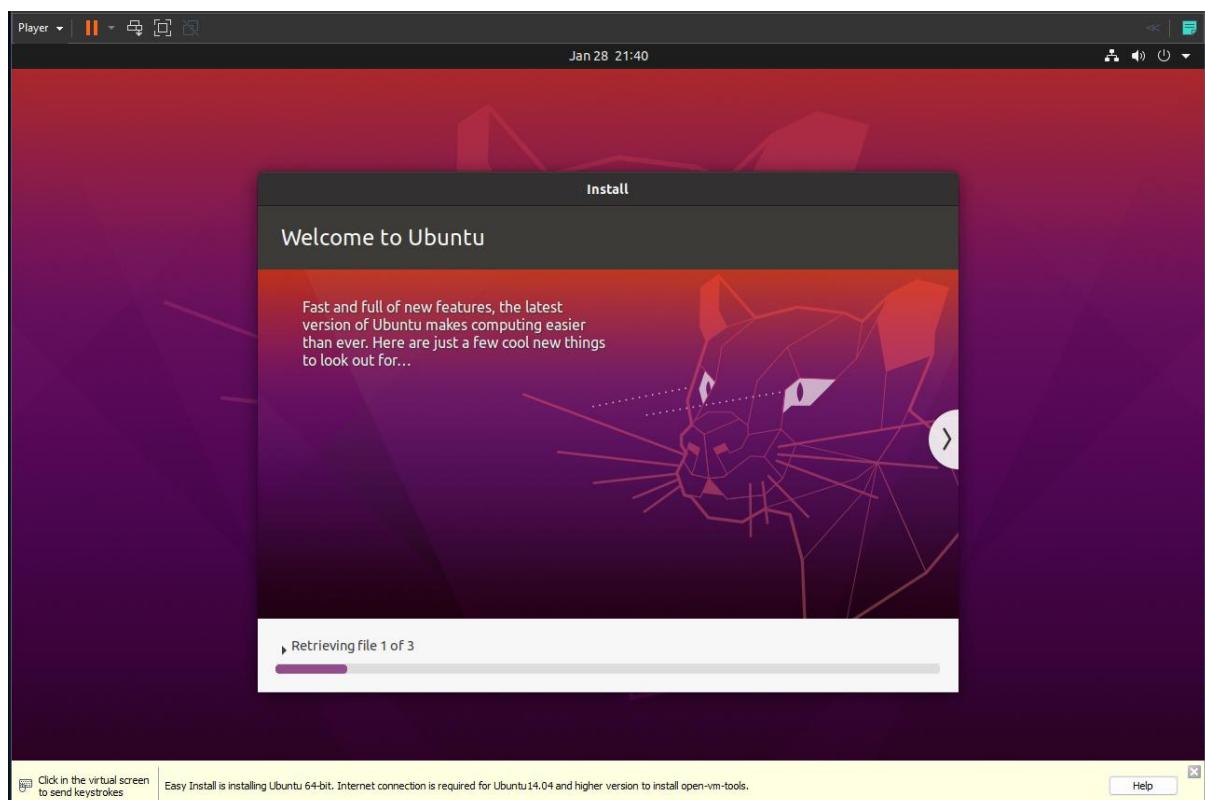
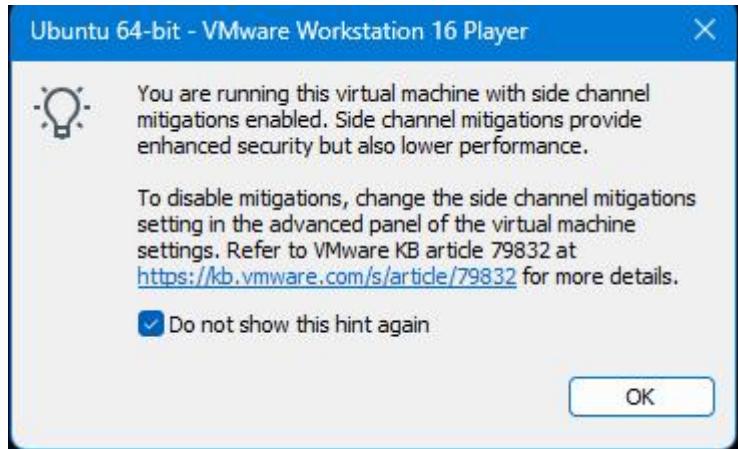


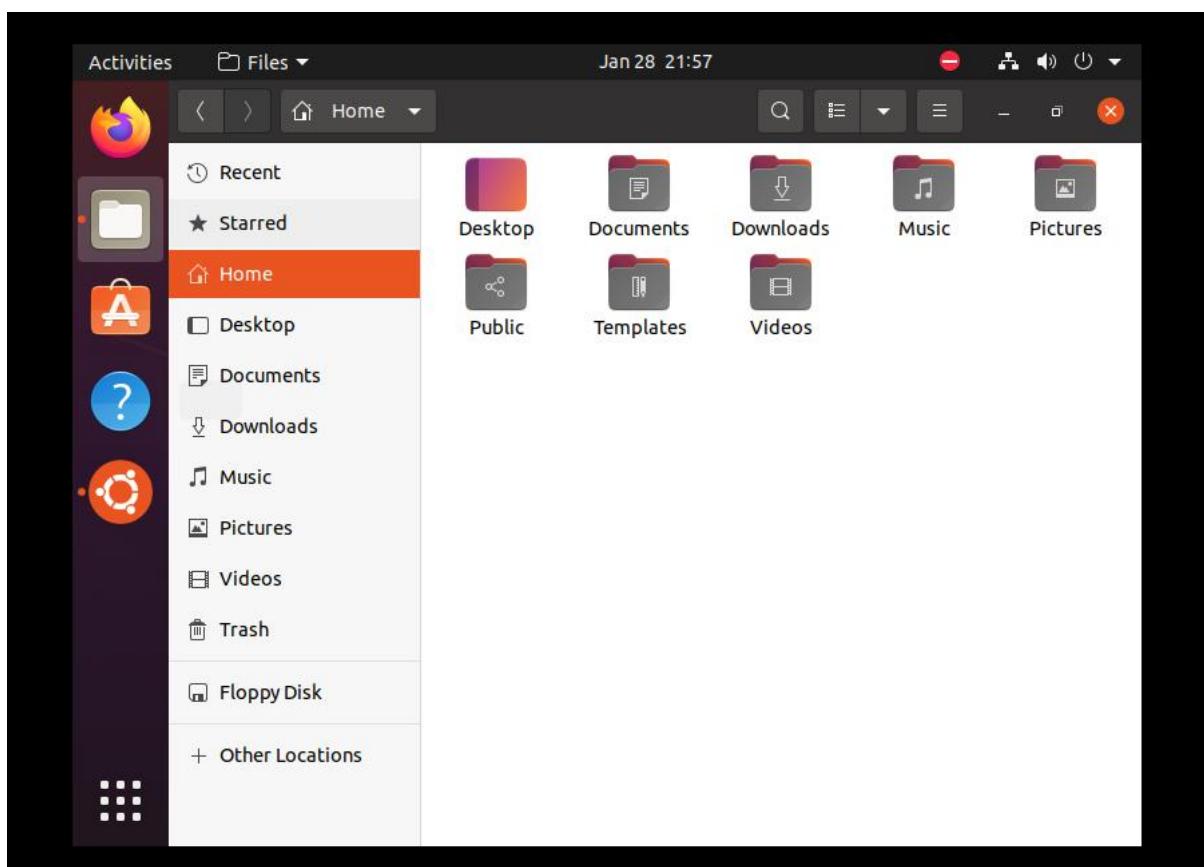
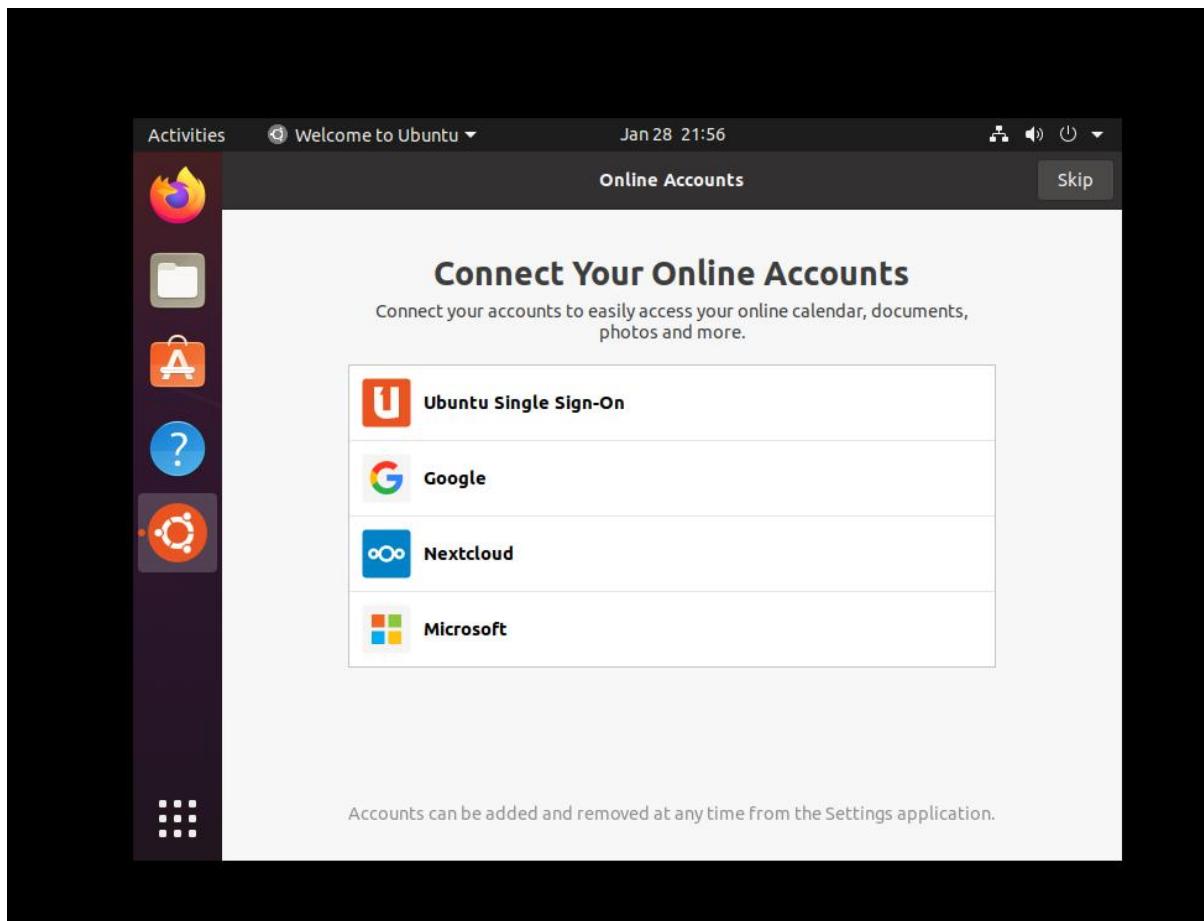
Create a New virtual Machine in VMWare.



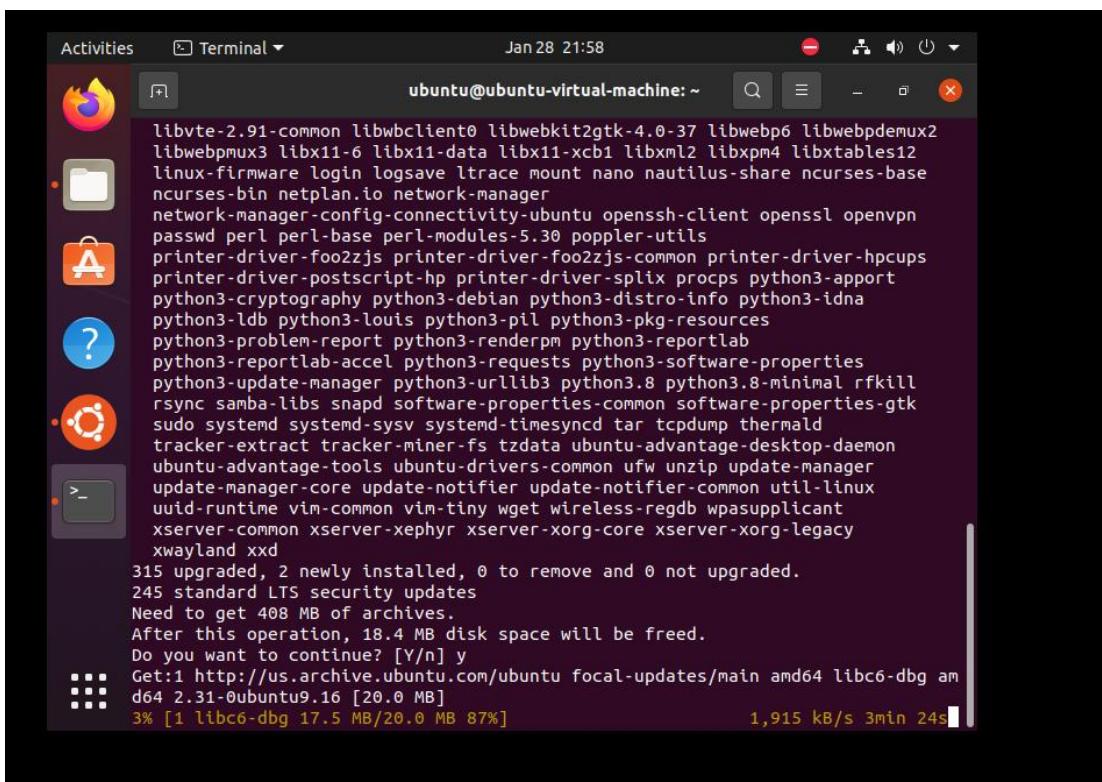
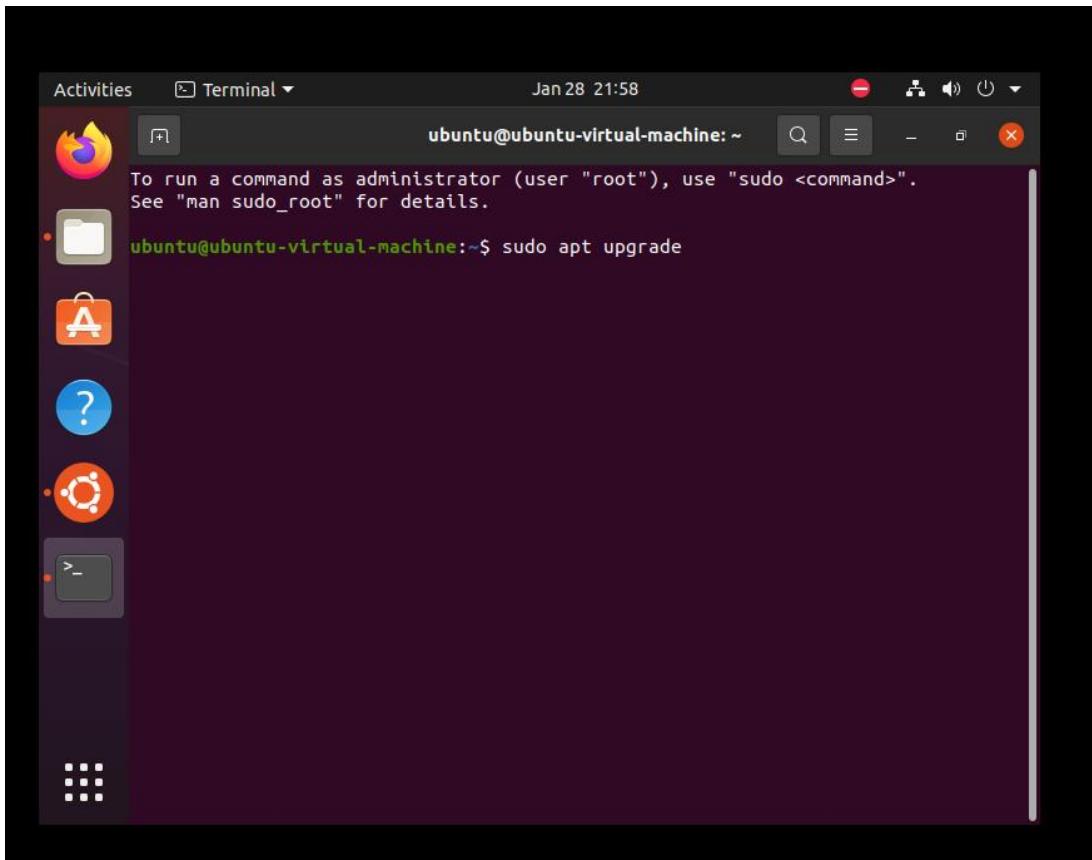


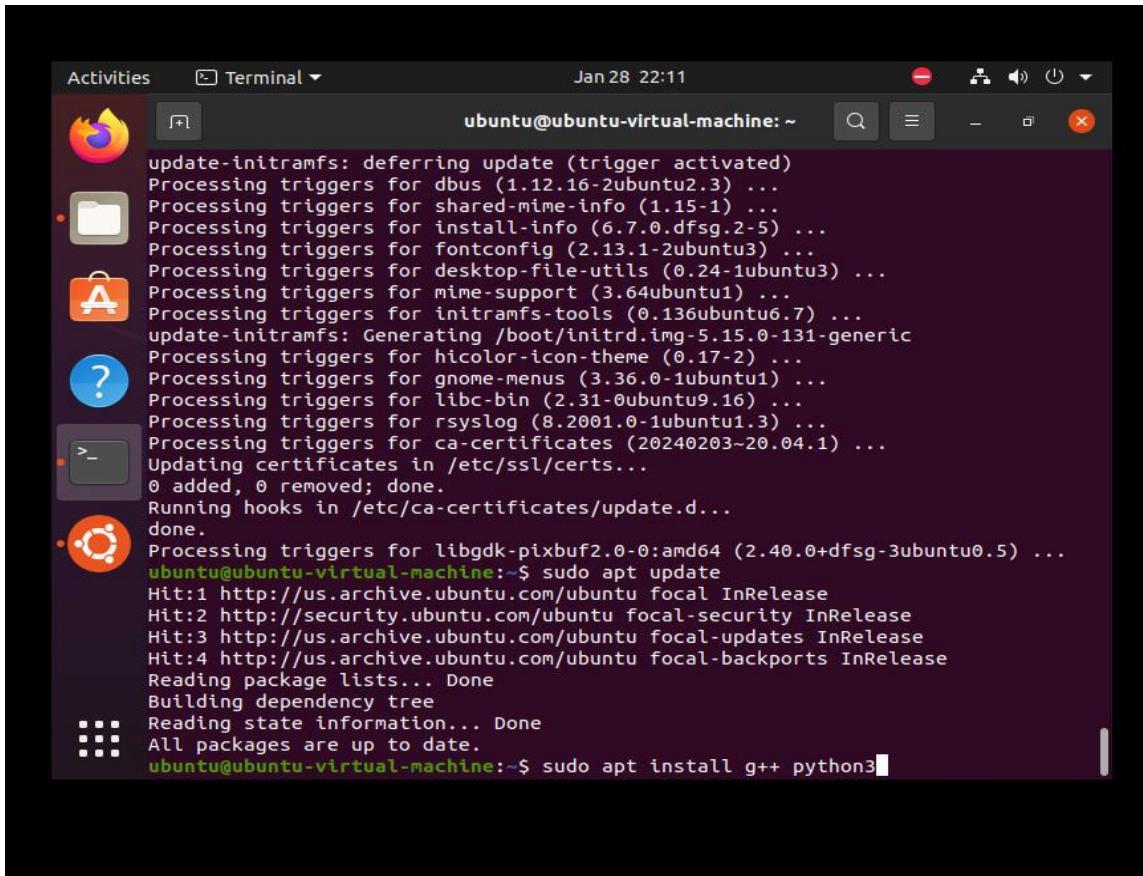






Install the necessary Libraries like NS3, Netanime etc.





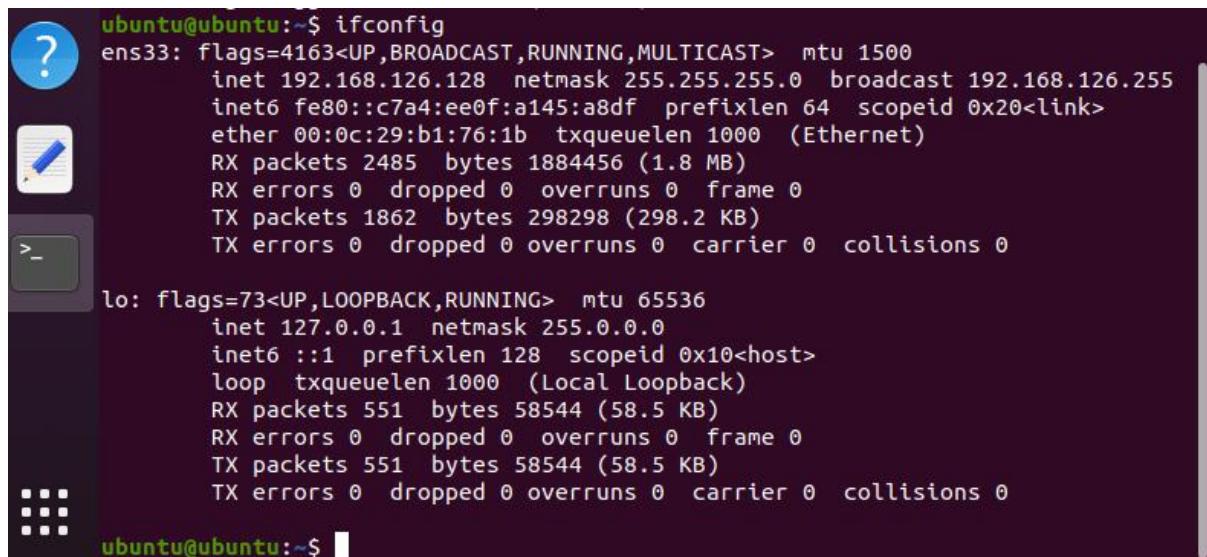
A screenshot of an Ubuntu desktop environment. On the left, there's a dock with icons for the Dash, Home, Applications, and Help. The main area shows a terminal window titled "Terminal". The terminal output is as follows:

```
Activities Terminal ▾ Jan 28 22:11
ubuntu@ubuntu-virtual-machine: ~
update-initramfs: deferring update (trigger activated)
Processing triggers for dbus (1.12.16-2ubuntu2.3) ...
Processing triggers for shared-mime-info (1.15-1) ...
Processing triggers for install-info (6.7.0.dfsg.2-5) ...
Processing triggers for fontconfig (2.13.1-2ubuntu3) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for initramfs-tools (0.136ubuntu6.7) ...
update-initramfs: Generating /boot/initrd.img-5.15.0-131-generic
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.16) ...
Processing triggers for rsyslog (8.2001.0-1ubuntu1.3) ...
Processing triggers for ca-certificates (20240203~20.04.1) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
Processing triggers for libgdk-pixbuf2.0-0:amd64 (2.40.0+dfsg-3ubuntu0.5) ...
ubuntu@ubuntu-virtual-machine: $ sudo apt update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
ubuntu@ubuntu-virtual-machine: $ sudo apt install g++ python3
```

PRACTICAL NO. 2

Linux network commands – ifconfig, ip, ping, netstat, traceroute, nslookup, route, hostname.

Ifconfig: used for **configuring and displaying network interfaces**.

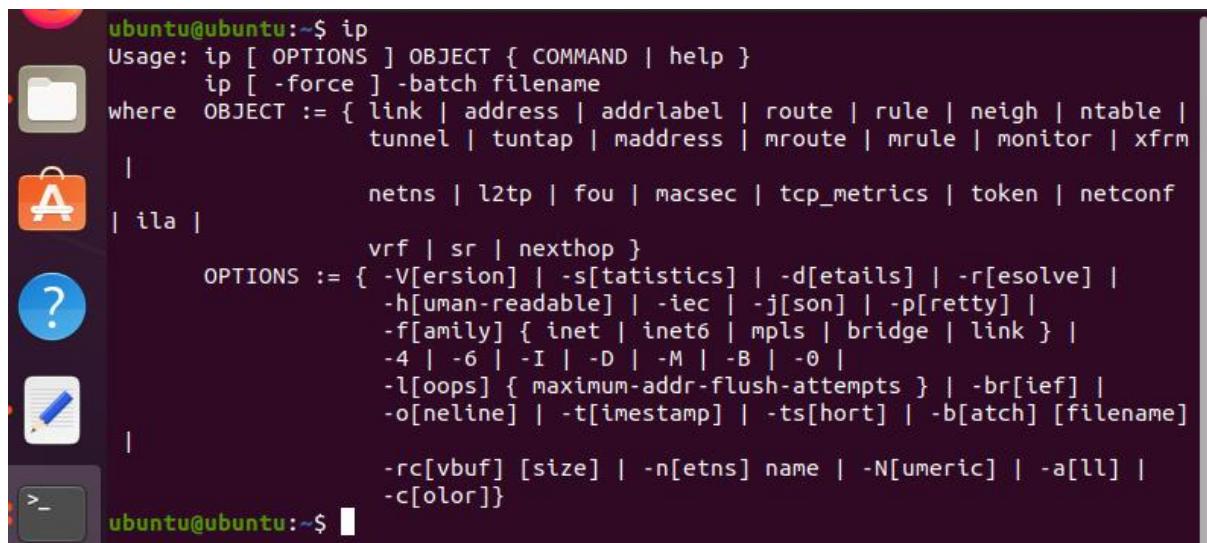


```
ubuntu@ubuntu:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.126.128 netmask 255.255.255.0 broadcast 192.168.126.255
      inet6 fe80::c7a4:ee0f:a145:a8df prefixlen 64 scopeid 0x20<link>
        ether 00:0c:29:b1:76:1b txqueuelen 1000 (Ethernet)
          RX packets 2485 bytes 1884456 (1.8 MB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 1862 bytes 298298 (298.2 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 551 bytes 58544 (58.5 KB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 551 bytes 58544 (58.5 KB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ubuntu@ubuntu:~$
```

ip : Displays and manipulates routing, devices and tunnels.



```
ubuntu@ubuntu:~$ ip
Usage: ip [ OPTIONS ] OBJECT { COMMAND | help }
      ip [ -force ] -batch filename
where OBJECT := { link | address | addrlabel | route | rule | neigh | ntable |
                 tunnel | tuntap | maddress | mroute | mrule | monitor | xfrm
                 |
                 netns | l2tp | fou | macsec | tcp_metrics | token | netconf
                 |
                 vrf | sr | nexthop }
OPTIONS := { -V[ersion] | -s[tatistics] | -d[etails] | -r[esolve] |
            -h[uman-readable] | -iec | -j[son] | -p[retty] |
            -f[amily] { inet | inet6 | mpls | bridge | link } |
            -4 | -6 | -I | -D | -M | -B | -0 |
            -l[oops] { maximum-addr-flush-attempts } | -br[ief] |
            -o[neline] | -t[imestamp] | -ts[hort] | -b[atch] [filename]
            |
            -rc[vbuf] [size] | -n[etns] name | -N[umeric] | -a[ll] |
            -c[olor]} |
ubuntu@ubuntu:~$
```

Ip -v : Shows version and output of ip commands

```
ubuntu@ubuntu:~$ ip -V
ip utility, iproute2-ss200127
ubuntu@ubuntu:~$ █
```

Ip addr : Shows ip addresses

```
ubuntu@ubuntu:~$ ip addr
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
            valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:b1:76:1b brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.126.128/24 brd 192.168.126.255 scope global dynamic noprefixroute ens33
        valid_lft 1537sec preferred_lft 1537sec
        inet6 fe80::c7a4:ee0f:a145:a8df/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
ubuntu@ubuntu:~$ █
```

Ip addr show : Displays detailed ip configuration

```
ubuntu@ubuntu:~$ ip addr show lo
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
            valid_lft forever preferred_lft forever
Show Applications | forever preferred_lft forever
ubuntu@ubuntu:~$ █
```

Ip link : Shows and configures network interfaces

```
ubuntu@ubuntu:~$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 00:0c:29:b1:76:1b brd ff:ff:ff:ff:ff:ff
    altname enp2s1
```

Ip route show : Displays the current routing table

```
ubuntu@ubuntu:~$ ip route show
default via 192.168.126.2 dev ens33 proto dhcp metric 100
169.254.0.0/16 dev ens33 scope link metric 1000
192.168.126.0/24 dev ens33 proto kernel scope link src 192.168.126.128 metric 1
00
ubuntu@ubuntu:~$
```

Ifconfig -s : Shows a short summary of network interfaces

```
ubuntu@ubuntu:~$ ifconfig -s
Iface      MTU     RX-OK RX-ERR RX-DRP RX-OVR     TX-OK TX-ERR TX-DRP TX-OVR Flg
ens33     1500    6189     0     0 0       3081     0     0 0       0 BMRU
lo        65536    605     0     0 0       605     0     0 0       0 LRU
ubuntu@ubuntu:~$
```

Dig : Queries DNS Servers for information.

```
ubuntu@ubuntu:~$ dig google.com

; <>> DiG 9.18.30-0ubuntu0.20.04.2-Ubuntu <>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62423
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
; Text Editor
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;google.com.           IN      A

;; ANSWER SECTION:
google.com.          5       IN      A      142.251.42.110

;; Query time: 3 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Mon Feb 17 22:21:22 PST 2025
;; MSG SIZE  rcvd: 55
```

Nslookup : Queries internet name servers interactively for domain name .

```
ubuntu@ubuntu:~$ nslookup google.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.183.110
Name:   google.com
Address: 2404:6800:4009:823::200e
```

Netstat -at : Displays all active TCP Connections

```
ubuntu@ubuntu:~$ netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 localhost:domain         0.0.0.0:*
tcp      0      0 localhost:ipp           0.0.0.0:*
tcp      0      0 ubuntu:47758            93.243.107.34.bc.:https ESTABLISHED
tcp      0      0 ubuntu:47750            93.243.107.34.bc.:https ESTABLISHED
tcp      0      0 ubuntu:46064            209.100.149.34.bc:https ESTABLISHED
tcp6     0      0 ip6-localhost:ipp       [::]:*
```

Host: performs DNS lookup to find ip addresses

```
ubuntu@ubuntu:~$ host google.com
google.com has address 142.251.42.110
google.com has IPv6 address 2404:6800:4009:832::200e
google.com mail is handled by 10 smtp.google.com.
ubuntu@ubuntu:~$
```

Ping : sends ICMP echo request to test network

```
ubuntu@ubuntu:~$ ping -c 5 google.com
PING google.com (142.250.183.110) 56(84) bytes of data.
64 bytes from bom12s13-in-f14.1e100.net (142.250.183.110): icmp_seq=1 ttl=128 time=2.13 ms
64 bytes from bom12s13-in-f14.1e100.net (142.250.183.110): icmp_seq=2 ttl=128 time=2.02 ms
64 bytes from bom12s13-in-f14.1e100.net (142.250.183.110): icmp_seq=3 ttl=128 time=2.14 ms
64 bytes from bom12s13-in-f14.1e100.net (142.250.183.110): icmp_seq=4 ttl=128 time=2.23 ms
64 bytes from bom12s13-in-f14.1e100.net (142.250.183.110): icmp_seq=5 ttl=128 time=2.11 ms

--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 2.018/2.126/2.231/0.068 ms
ubuntu@ubuntu:~$
```

Traceroute : Traces the route packets take to reach a network host

```
ubuntu@ubuntu:~$ sudo traceroute -T 184.95.56.34
traceroute to 184.95.56.34 (184.95.56.34), 30 hops max, 60 byte packets
 1 _gateway (192.168.126.2)  0.092 ms  0.051 ms  0.081 ms
 2 speedtest.phoenixnap.com (184.95.56.34)  0.523 ms  0.503 ms  0.740 ms
ubuntu@ubuntu:~$ sudo traceroute -T 127.0.0.53
traceroute to 127.0.0.53 (127.0.0.53), 30 hops max, 60 byte packets
 1 localhost (127.0.0.53)  0.020 ms  0.004 ms  0.004 ms
ubuntu@ubuntu:~$
```

Route : Displays or modifies the ip routing table

```
ubuntu@ubuntu:~$ route
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
default        _gateway        0.0.0.0         UG    100    0        0 ens33
link-local     0.0.0.0         255.255.0.0     U     1000   0        0 ens33
192.168.126.0  0.0.0.0         255.255.255.0   U     100    0        0 ens33
```

Arp : Displays or modifies the ARP

```
ubuntu@ubuntu:~$ arp
Address           HWtype  HWaddress            Flags Mask       Ifac
e
192.168.126.254 ether   00:50:56:e2:86:ba   C          ens3
3
_gateway         ether   00:50:56:eb:a2:d7   C          ens3
3
ubuntu@ubuntu:~$
```

Iwconfig : Configures wireless network interfaces

```
ubuntu@ubuntu:~$ iwconfig
lo      no wireless extensions.

ens33    no wireless extensions.

ubuntu@ubuntu:~$
```

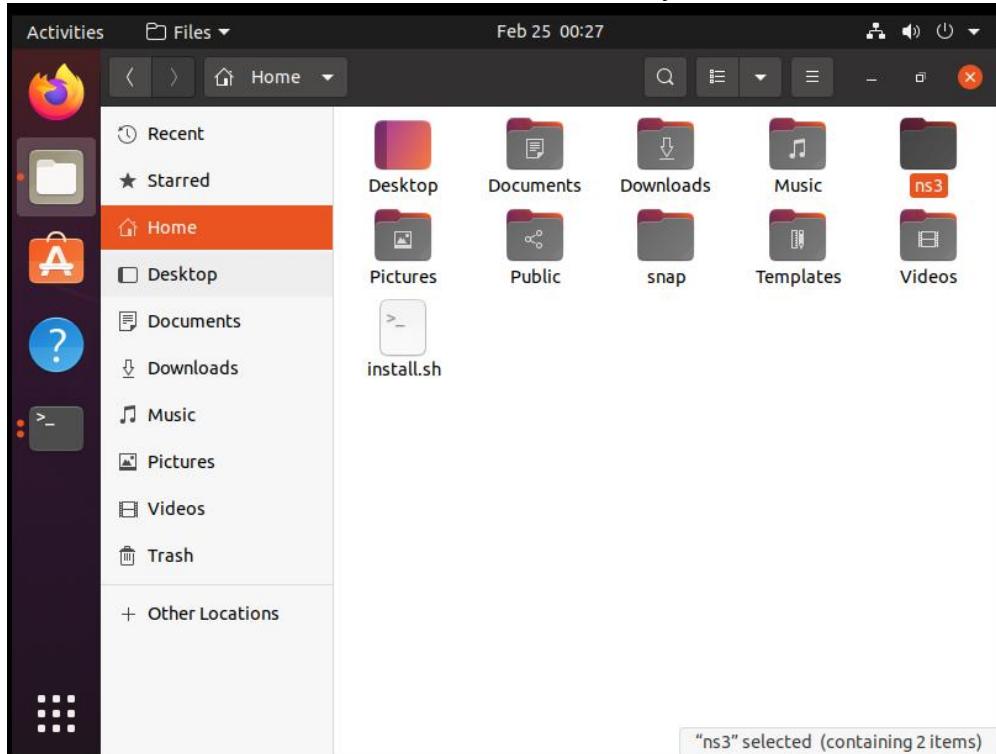
Whois : Fetches domain registration information.

```
ubuntu@ubuntu:~$ whois google.com
Domain Name: GOOGLE.COM
Registry Domain ID: 2138514_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2019-09-09T15:39:04Z
Creation Date: 1997-09-15T04:00:00Z
Registry Expiry Date: 2028-09-14T04:00:00Z
Registrar: MarkMonitor Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2086851750
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
Name Server: NS1.GOOGLE.COM
Name Server: NS2.GOOGLE.COM
Name Server: NS3.GOOGLE.COM
```

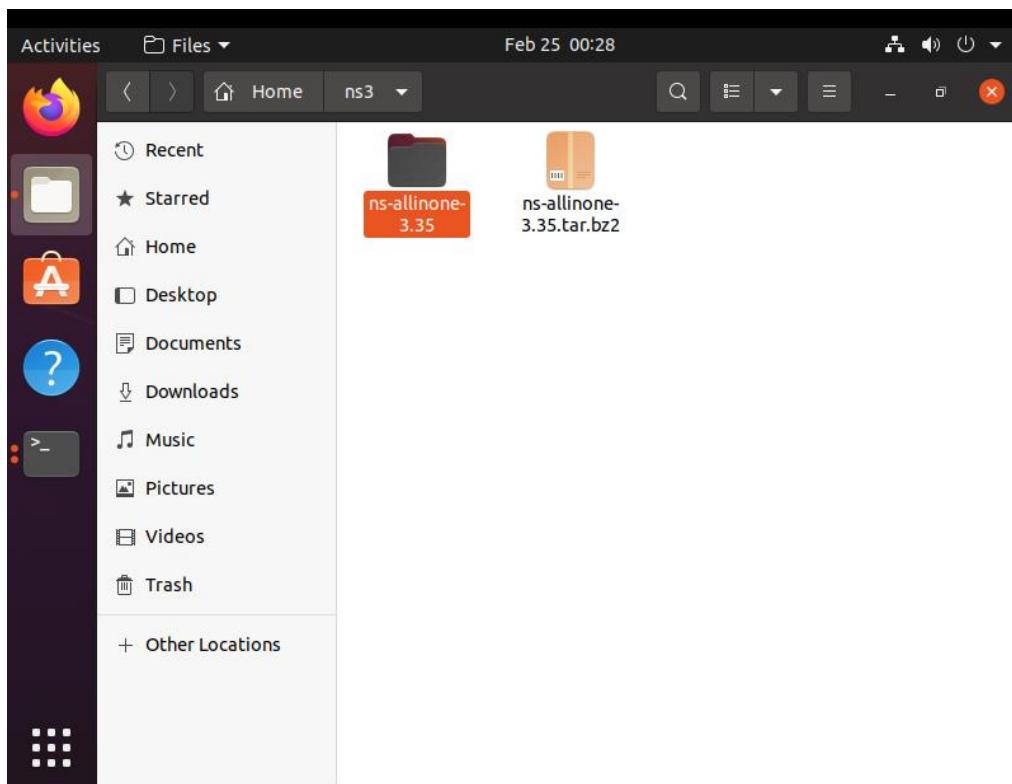
PRACTICAL NO. 3

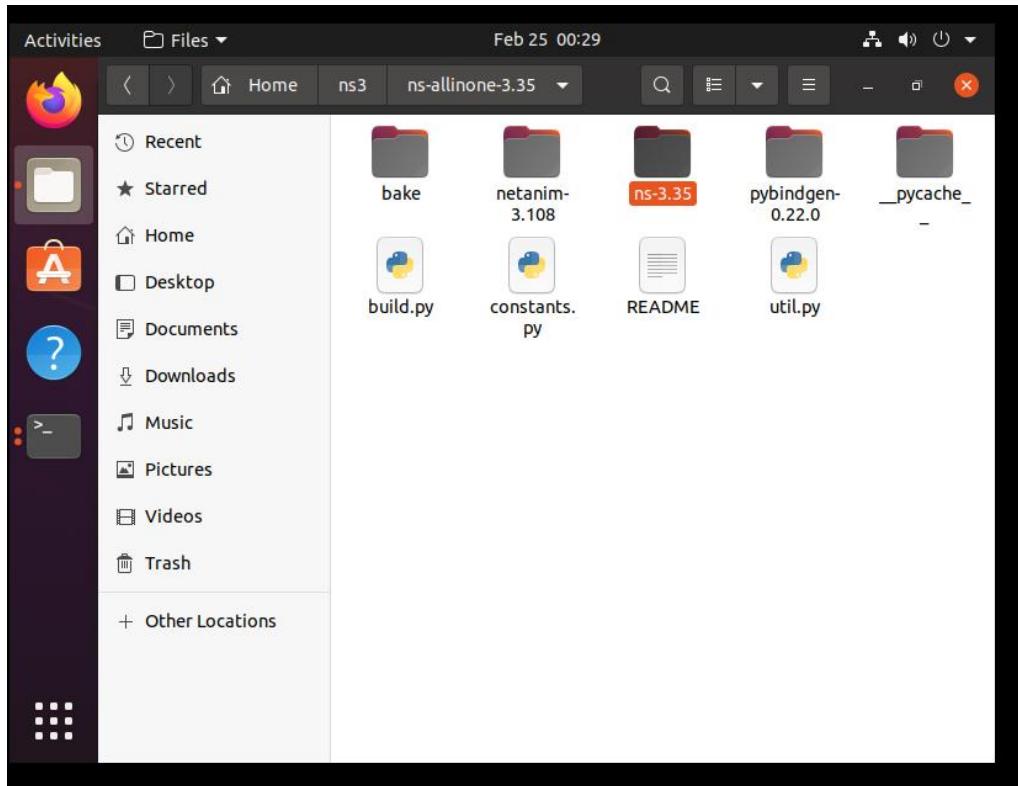
Program to simulate Point to Point Topology.

Start Ubuntu virtual machine , Go to home directory and click on ns3.

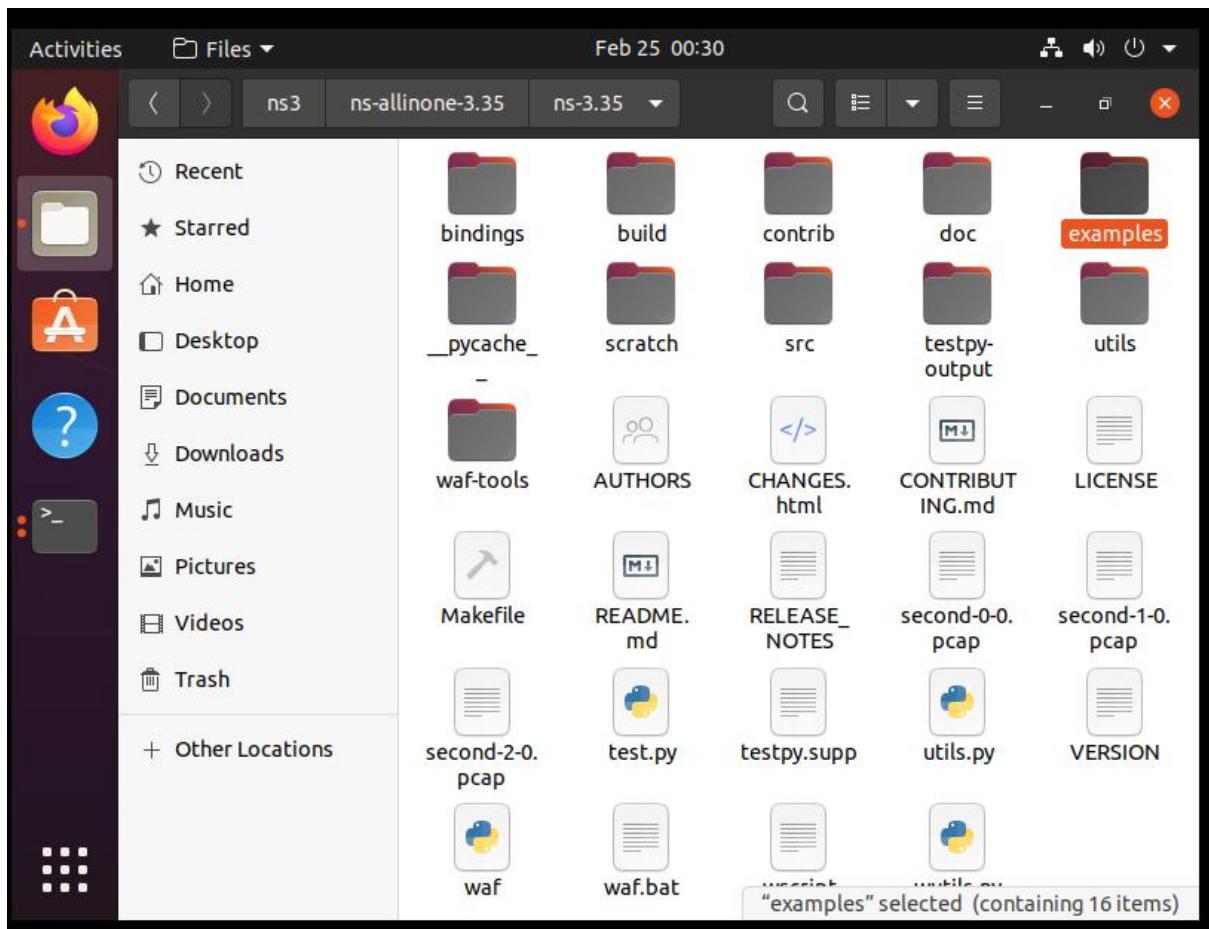


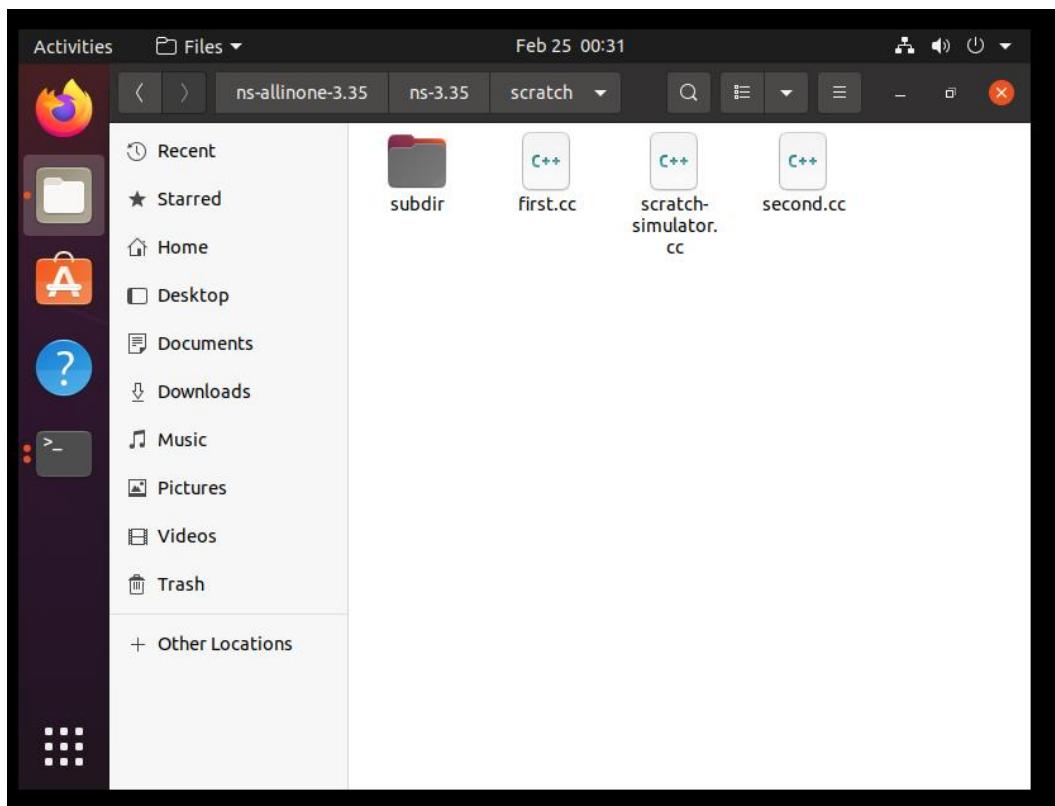
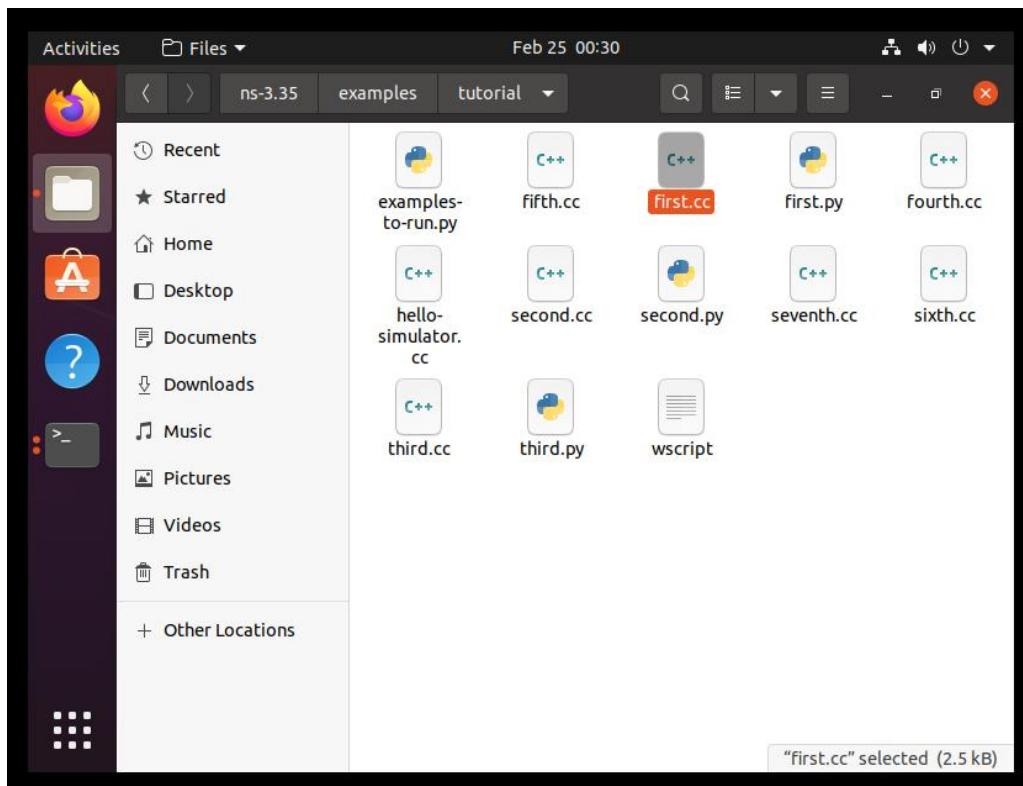
In ns3 folder open nsallinone folder →open ns 3.35 folder.



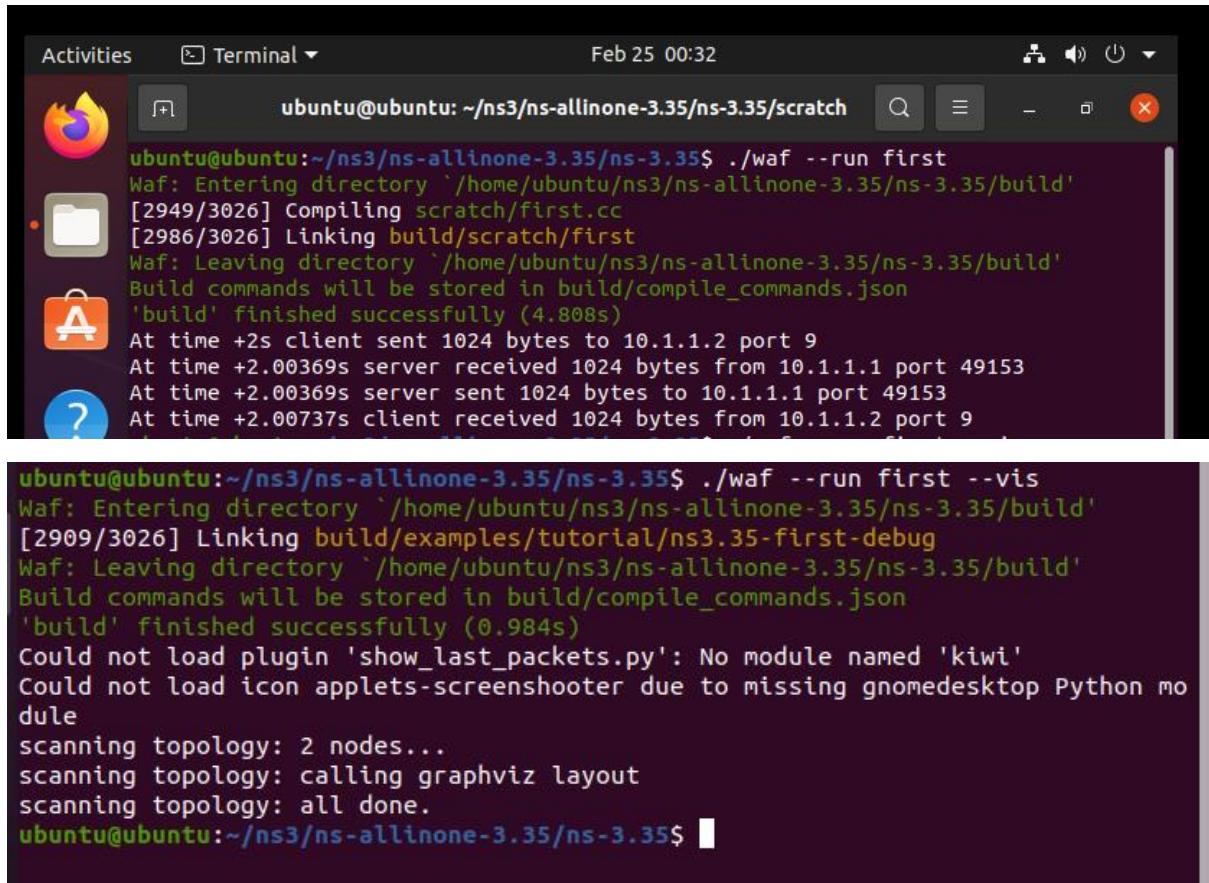


Open examples folder ,go to tutorial and copy the first.cc file into scratch folder of ns3.35.





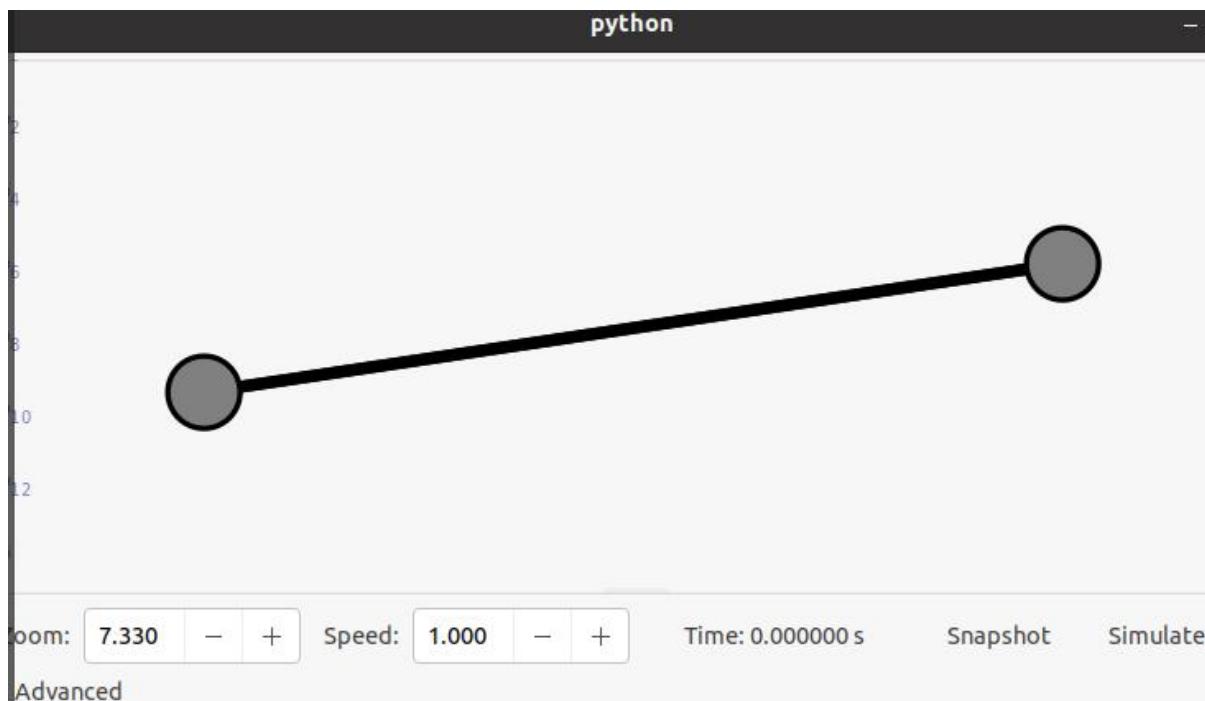
Go to ns 3.35 and then click on open in terminal.



The screenshot shows a terminal window titled "Terminal" with the command line "ubuntu@ubuntu: ~/ns3/ns-allinone-3.35/ns-3.35/scratch". The terminal displays the output of running the "first" example using Waf. The output shows the build process, network traffic exchange between nodes, and the successful completion of the build. Below this, another command is run to run the "first" example with visualization, which fails to load certain Python modules ("kiwi" and "applets-screenshooter").

```
ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35$ ./waf --run first
Waf: Entering directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
[2949/3026] Compiling scratch/first.cc
[2986/3026] Linking build/scratch/first
Waf: Leaving directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (4.808s)
At time +2s client sent 1024 bytes to 10.1.1.2 port 9
At time +2.00369s server received 1024 bytes from 10.1.1.1 port 49153
At time +2.00369s server sent 1024 bytes to 10.1.1.1 port 49153
At time +2.00737s client received 1024 bytes from 10.1.1.2 port 9

ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35$ ./waf --run first --vis
Waf: Entering directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
[2909/3026] Linking build/examples/tutorial/ns3.35-first-debug
Waf: Leaving directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.984s)
Could not load plugin 'show_last_packets.py': No module named 'kiwi'
Could not load icon applets-screenshooter due to missing gnomedesktop Python module
scanning topology: 2 nodes...
scanning topology: calling graphviz layout
scanning topology: all done.
ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35$
```



CODE:

```
/* -- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -- /
/


- This program is free software; you can redistribute it and/or modify
- it under the terms of the GNU General Public License version 2 as
- published by the Free Software Foundation;
- 
- This program is distributed in the hope that it will be useful,
- but WITHOUT ANY WARRANTY; without even the implied warranty of
- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
- GNU General Public License for more details.
- 
- You should have received a copy of the GNU General Public License
- along with this program; if not, write to the Free Software
- Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

 */
#include "ns3/core-module.h"
#include "ns3/network-module.h"
```

```
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
//netanimation
#include "ns3/netanim-module.h"
#include "ns3/mobility-module.h"
[Screenshot Placeholder: Insert NetAnim or Wireshark output image here]

// Default Network Topology
//
// 10.1.1.0
// n0 n1
// point-to-point
//

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("FirstScriptExample");

int
main (int argc, char *argv[])
{
CommandLine cmd ( FILE );
cmd.Parse (argc, argv);

Time::SetResolution (Time::NS);
LogComponentEnable ("UdpEchoClientApplication", LOG_LEVEL_INFO);
LogComponentEnable ("UdpEchoServerApplication", LOG_LEVEL_INFO);

NodeContainer nodes;
nodes.Create (2);

PointToPointHelper pointToPoint;
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));

NetDeviceContainer devices;
devices = pointToPoint.Install (nodes);

InternetStackHelper stack;
stack.Install (nodes);

Ipv4AddressHelper address;
address.SetBase ("10.1.1.0", "255.255.255.0");

Ipv4InterfaceContainer interfaces = address.Assign (devices);

UdpEchoServerHelper echoServer (9);
```

```
ApplicationContainer serverApps = echoServer.Install (nodes.Get (1));
serverApps.Start (Seconds (1.0));
serverApps.Stop (Seconds (10.0));

UdpEchoClientHelper echoClient (interfaces.GetAddress (1), 9);
echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
echoClient.SetAttribute ("PacketSize", UintegerValue (1024));

ApplicationContainer clientApps = echoClient.Install (nodes.Get (0));
clientApps.Start (Seconds (2.0));
clientApps.Stop (Seconds (10.0));

MobilityHelper mobility;
mobility.SetMobilityModel("ns3::ConstantPositionMobilityModel");
mobility.Install(nodes);
AnimationInterface anim("first.xml");
AnimationInterface::SetConstantPosition(nodes.Get(0),10,25);
AnimationInterface::SetConstantPosition(nodes.Get(1),40,25);
anim.EnablePacketMetadata(true);
pointToPoint.EnablePcapAll("first");
Simulator::Run ();
Simulator::Destroy ();
return 0;
}
```

[Screenshot Placeholder: Insert NetAnim or Wireshark output image here]

This NS-3 code simulates a simple point-to-point network between two nodes, where one node acts as a UDP echo client and the other as a UDP echo server. The nodes are connected via a link with a data rate of 5 Mbps and 2 ms delay. IP addresses are assigned, and the server listens on port 9 while the client sends one 1024-byte packet after 2 seconds. Mobility models fix the nodes' positions, and the simulation includes NetAnim support for visualization and PCAP tracing for packet capture. The simulation runs for 10 seconds, demonstrating basic end-to-end communication, animation, and tracing.

[Screenshot Placeholder: Insert NetAnim or Wireshark output image here]

To edit the given .cc file open the terminal in scratch folder and perform the gedit command.

A screenshot of a Linux desktop environment. On the left, there's a dock with various icons: a browser, a file manager, a terminal, a help center, a dash, and a text editor. The main window is a 'Text Editor' showing a C++ code file named 'first.cc'. The code sets up a point-to-point link between two nodes, configures them with 5Mbps data rate and 2ms delay, installs Internet stacks, assigns IP addresses (10.1.1.0/255.255.255.0), starts a UDP echo server on port 15, and installs application containers for both client and server. The terminal window at the bottom shows the execution of the 'waf' command to build the project, followed by network traffic logs indicating a successful communication session between the two nodes.

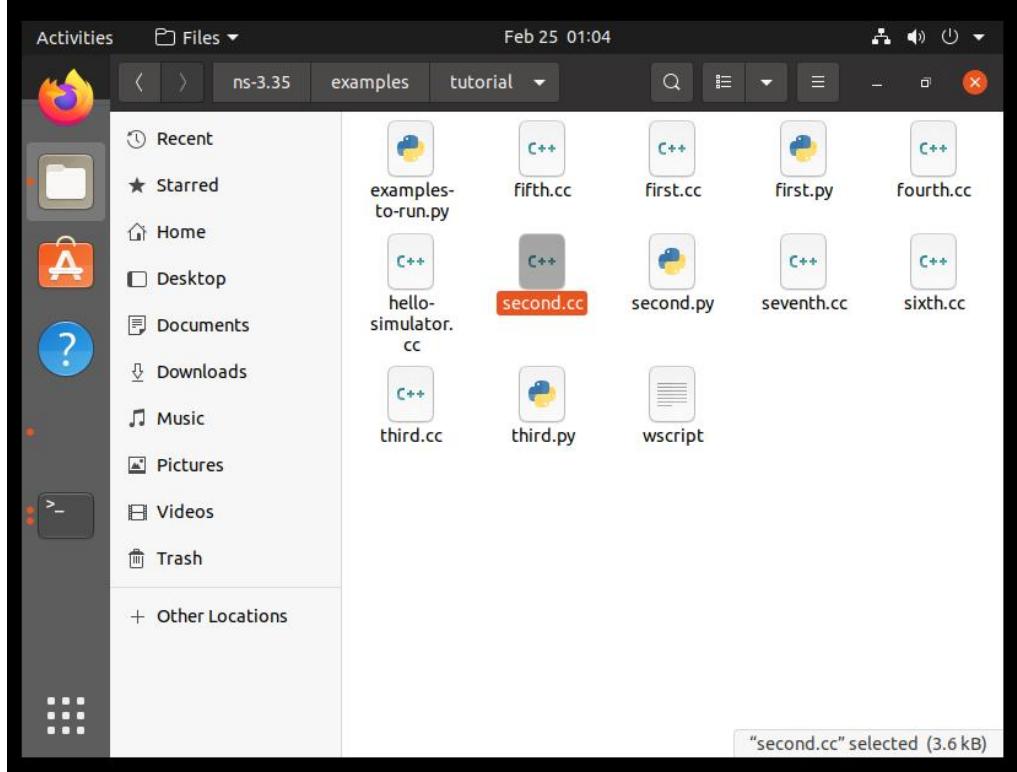
```
46 PointToPointHelper pointToPoint;
47 pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
48 pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));
49
50 NetDeviceContainer devices;
51 devices = pointToPoint.Install (nodes);
52
53 InternetStackHelper stack;
54 stack.Install (nodes);
55
56 Ipv4AddressHelper address;
57 address.SetBase ("10.1.1.0", "255.255.255.0");
58
59 Ipv4InterfaceContainer interfaces = address.Assign (devices);
60
61 UdpEchoServerHelper echoServer (15);
62
63 ApplicationContainer serverApps = echoServer.Install (nodes.Get (1));
64 serverApps.Start (Seconds (1.0));
65 serverApps.Stop (Seconds (10.0));
66
67 UdpEchoClientHelper echoClient (interfaces.GetAddress (1), 15);
68 echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
69 echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
70 echoClient.SetAttribute ("PacketSize", UintegerValue (1024));
71
72 ApplicationContainer clientApps = echoClient.Install (nodes.Get (0));
73
```

```
ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35$ ./waf --run scratch/first
Waf: Entering directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
[2912/3026] Linking build/examples/tutorial/ns3.35-first-debug
[2956/3026] Compiling scratch/first.cc
[2986/3026] Linking build/scratch/first
Waf: Leaving directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (5.326s)
At time +2s client sent 1024 bytes to 10.1.1.2 port 15
At time +2.00369s server received 1024 bytes from 10.1.1.1 port 49153
At time +2.00369s server sent 1024 bytes to 10.1.1.1 port 49153
At time +2.00737s client received 1024 bytes from 10.1.1.2 port 15
ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35$
```

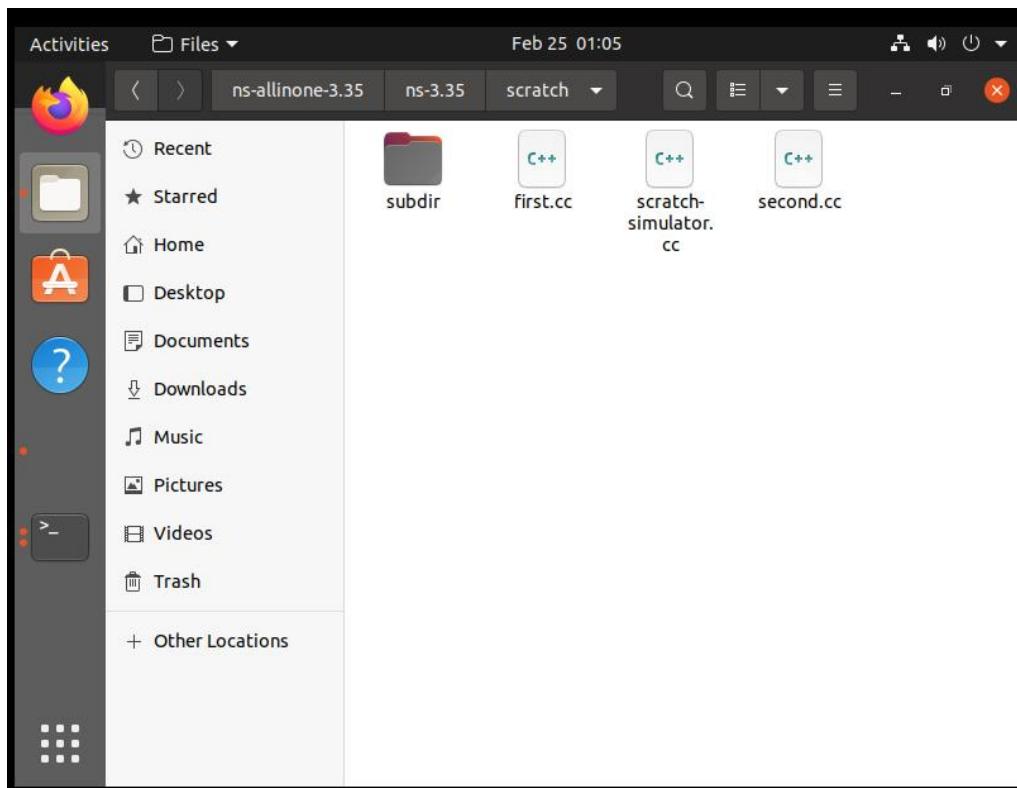
PRACTICAL NO. 4

Program to simulate Bus topology.

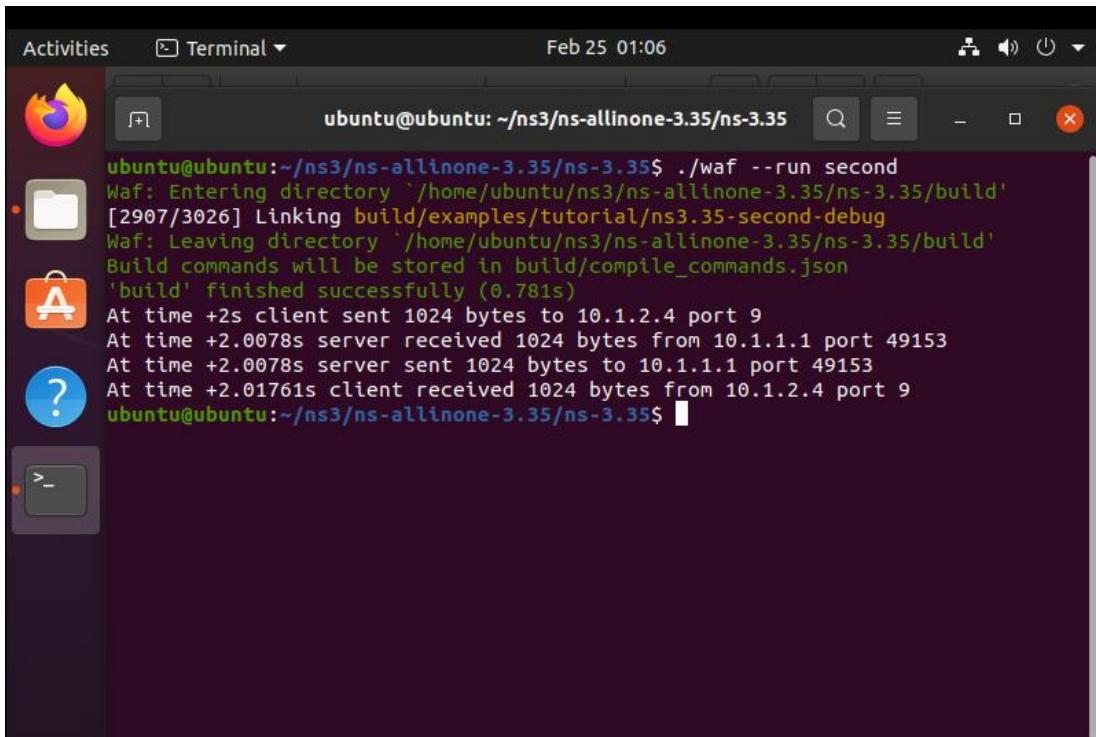
Open the ns3.35 folder and in examples → tutorial → second .cc



Copy this file and paste it in scratch folder.

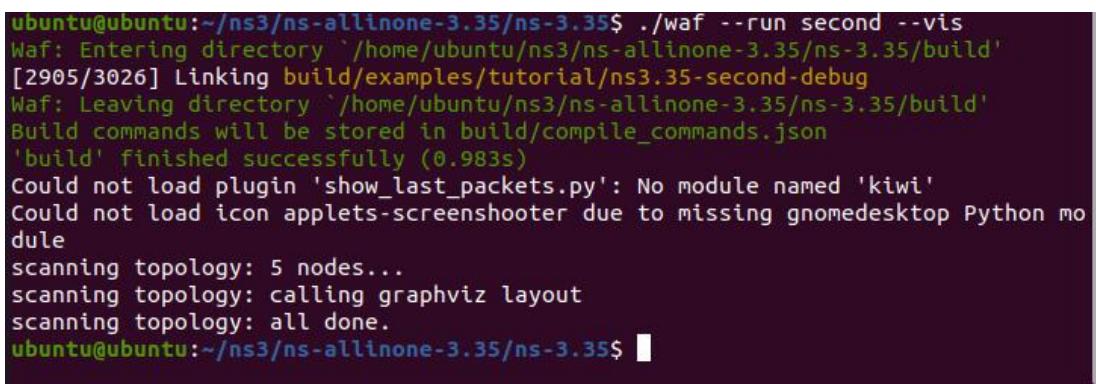


Open the terminal in ns3.35 and perform the commands.



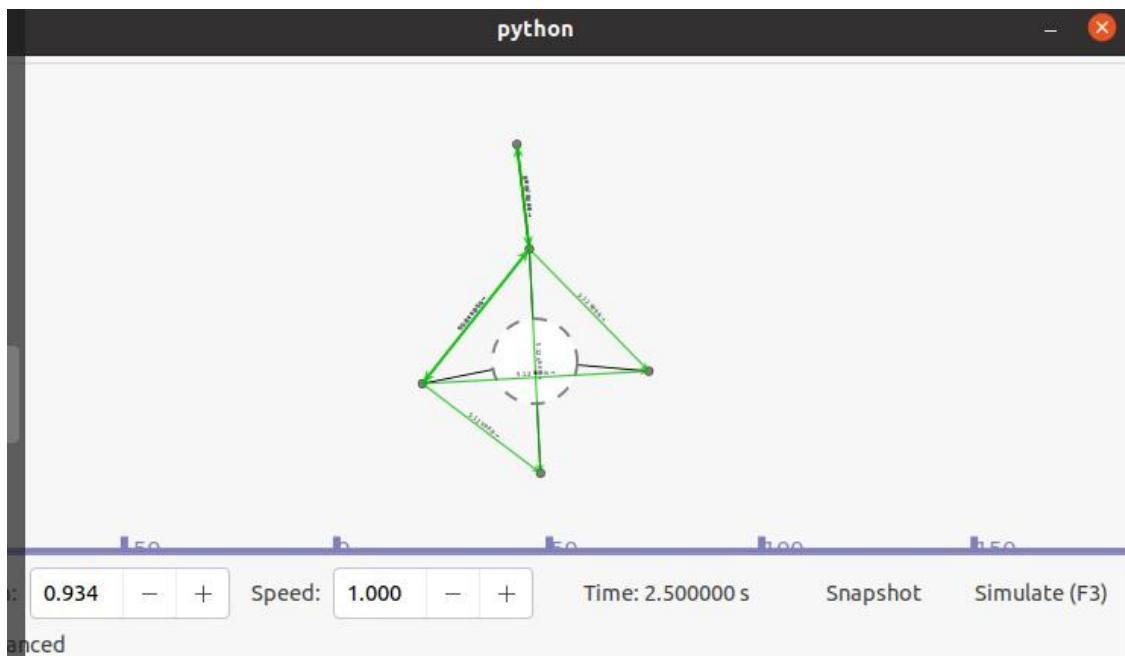
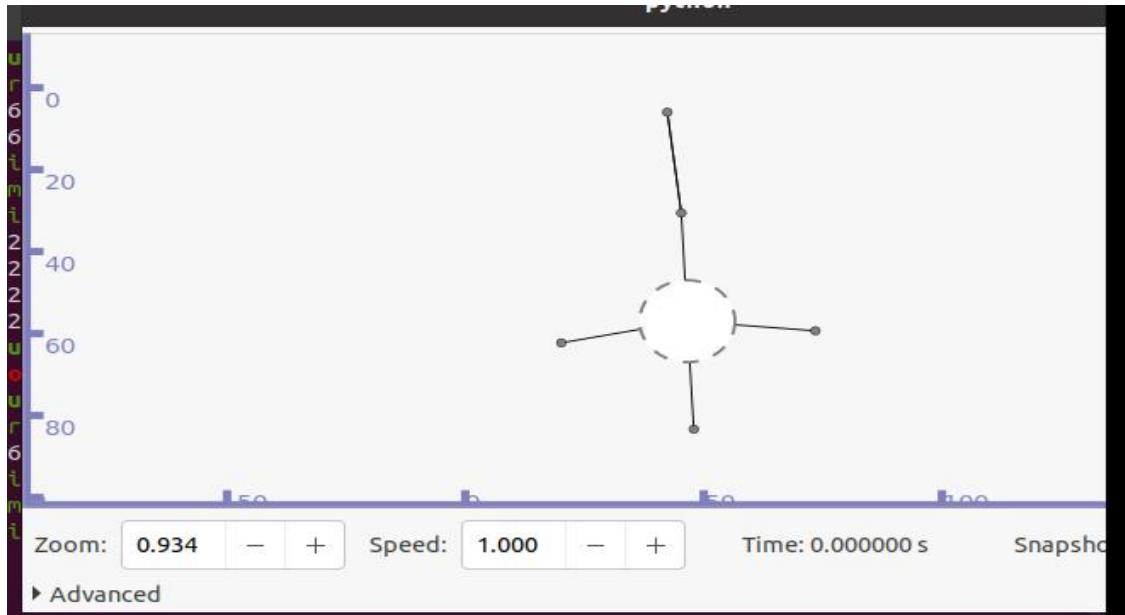
A screenshot of an Ubuntu desktop environment. In the top left, there's a dock with icons for the Dash, Home, and several application windows. One window is a terminal titled "Terminal". The terminal window shows the command line and its output. The date and time at the top right of the terminal window are "Feb 25 01:06".

```
ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35$ ./waf --run second
Waf: Entering directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
[2907/3026] Linking build/examples/tutorial/ns3.35-second-debug
Waf: Leaving directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.781s)
At time +2s client sent 1024 bytes to 10.1.2.4 port 9
At time +2.0078s server received 1024 bytes from 10.1.1.1 port 49153
At time +2.0078s server sent 1024 bytes to 10.1.1.1 port 49153
At time +2.0176is client received 1024 bytes from 10.1.2.4 port 9
ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35$
```



A screenshot of a terminal window showing the execution of the command `./waf --run second --vis`. The terminal output includes the build process, network traffic details, and a message indicating it could not load the 'show_last_packets.py' plugin due to a missing module named 'kiwi'. It also shows the topology scanning process.

```
ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35$ ./waf --run second --vis
Waf: Entering directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
[2905/3026] Linking build/examples/tutorial/ns3.35-second-debug
Waf: Leaving directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.983s)
Could not load plugin 'show_last_packets.py': No module named 'kiwi'
Could not load icon applets-screenshooter due to missing gnomedesktop Python module
scanning topology: 5 nodes...
scanning topology: calling graphviz layout
scanning topology: all done.
ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35$
```



CODE:

```
/* -- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -- */  
/*  
 * This program is free software; you can redistribute it and/or modify  
 * it under the terms of the GNU General Public License version 2 as  
 * published by the Free Software Foundation;
```

*

- * This program is distributed in the hope that it will be useful,
- * but WITHOUT ANY WARRANTY; without even the implied warranty of
- * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
- * GNU General Public License for more details.

*

- * You should have received a copy of the GNU General Public License
- * along with this program; if not, write to the Free Software

* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

*/

```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/csma-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
#include "ns3/ipv4-global-routing-helper.h"
```

```
// Default Network Topology
```

```
//
```

```
//      10.1.1.0
```

```
// n0 ----- n1  n2  n3  n4
```

```
//   point-to-point |  |  |  |
```

```
//               =====
```

```
//      LAN 10.1.2.0
```

```
using namespace ns3;
```

```
NS_LOG_COMPONENT_DEFINE ("SecondScriptExample");

int
main (int argc, char *argv[])
{
    bool verbose = true;
    uint32_t nCsma = 3;

    CommandLine cmd (_FILE_);
    cmd.AddValue ("nCsma", "Number of \"extra\" CSMA nodes/devices", nCsma);
    cmd.AddValue ("verbose", "Tell echo applications to log if true", verbose);

    cmd.Parse (argc, argv);

    if (verbose)
    {
        LogComponentEnable ("UdpEchoClientApplication", LOG_LEVEL_INFO);
        LogComponentEnable ("UdpEchoServerApplication", LOG_LEVEL_INFO);
    }

    nCsma = nCsma == 0 ? 1 : nCsma;

    NodeContainer p2pNodes;
    p2pNodes.Create (2);

    NodeContainer csmaNodes;
    csmaNodes.Add (p2pNodes.Get (1));
    csmaNodes.Create (nCsma);

    PointToPointHelper pointToPoint;
```

```
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));
```

```
NetDeviceContainer p2pDevices;
p2pDevices = pointToPoint.Install (p2pNodes);
```

```
CsmaHelper csma;
csma.SetChannelAttribute ("DataRate", StringValue ("100Mbps"));
csma.SetChannelAttribute ("Delay", TimeValue (NanoSeconds (6560)));
```

```
NetDeviceContainer csmaDevices;
csmaDevices = csma.Install (csmaNodes);
```

```
InternetStackHelper stack;
stack.Install (p2pNodes.Get (0));
stack.Install (csmaNodes);
```

```
Ipv4AddressHelper address;
address.SetBase ("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer p2pInterfaces;
p2pInterfaces = address.Assign (p2pDevices);
```

```
address.SetBase ("10.1.2.0", "255.255.255.0");
Ipv4InterfaceContainer csmaInterfaces;
csmaInterfaces = address.Assign (csmaDevices);
```

```
UdpEchoServerHelper echoServer (9);
```

```
ApplicationContainer serverApps = echoServer.Install (csmaNodes.Get (nCsma));
serverApps.Start (Seconds (1.0));
```

```
serverApps.Stop (Seconds (10.0));  
  
UdpEchoClientHelper echoClient (csmaInterfaces.GetAddress (nCsma), 9);  
echoClient.SetAttribute ("MaxPackets", UintegerValue (1));  
echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));  
echoClient.SetAttribute ("PacketSize", UintegerValue (1024));
```

```
ApplicationContainer clientApps = echoClient.Install (p2pNodes.Get (0));  
clientApps.Start (Seconds (2.0));  
clientApps.Stop (Seconds (10.0));
```

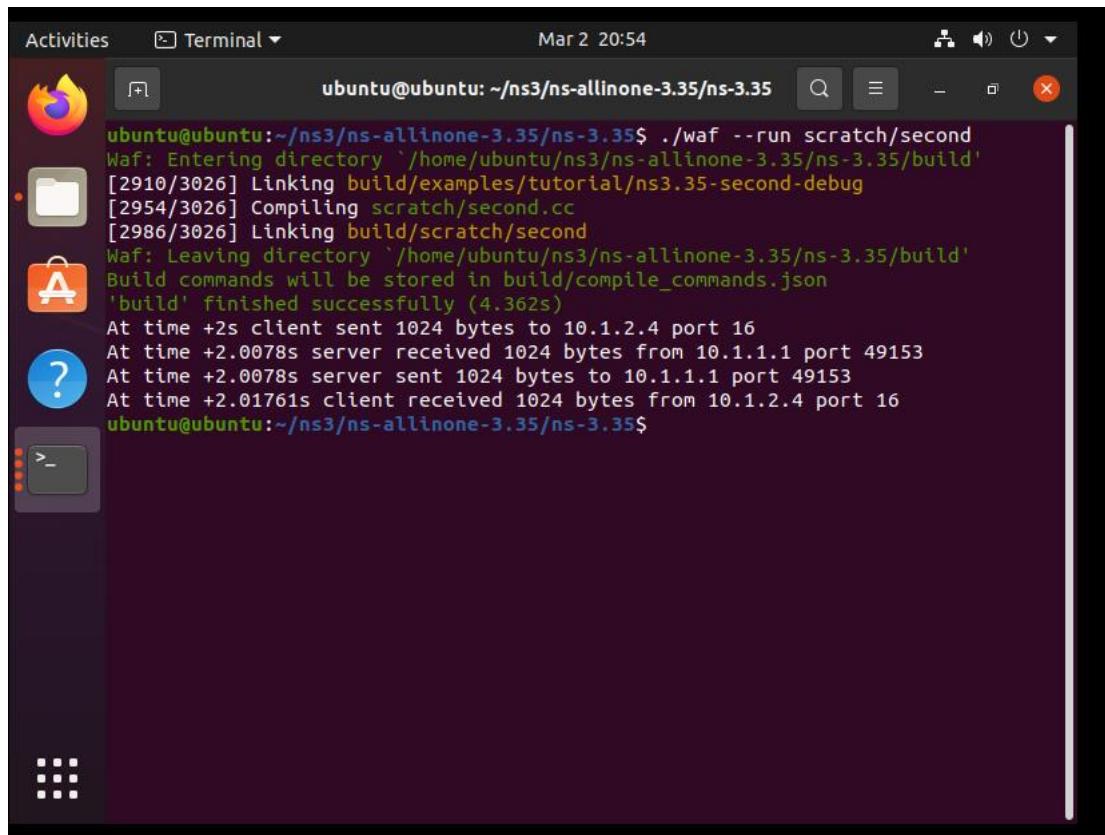
```
Ipv4GlobalRoutingHelper::PopulateRoutingTables ();
```

```
pointToPoint.EnablePcapAll ("second");  
csma.EnablePcap ("second", csmaDevices.Get (1), true);
```

```
Simulator::Run ();  
Simulator::Destroy ();  
return 0;  
}
```

A P2P link connects two nodes: n0 and n1. Node n1 then connects to a group of CSMA nodes (n2, n3, n4, etc.) via a shared CSMA channel, which simulates a bus-like structure where all nodes are connected to the same medium. The CSMA segment acts as the shared backbone, similar to a classic bus topology. The server runs on one of the CSMA nodes, and a client runs on the P2P node n0, sending a UDP echo request to the server. This setup demonstrates communication between nodes across a mixed topology with bus-like characteristics.

To make changes in .cc file open the file using gedit command.



A screenshot of an Ubuntu desktop environment. On the left, there's a dock with icons for the Dash, Home, Applications, and Help. In the center, a terminal window is open with the title "Terminal". The terminal shows the command `./waf --run scratch/second` being run in the directory `~/ns3/ns-allinone-3.35/ns-3.35`. The output of the command is displayed, showing the build process of the "scratch/second" example using Waf, and the subsequent network traffic between a client and a server on ports 16 and 49153.

```
ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35$ ./waf --run scratch/second
Waf: Entering directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
[2910/3026] Linking build/examples/tutorial/ns3.35-second-debug
[2954/3026] Compiling scratch/second.cc
[2986/3026] Linking build/scratch/second
Waf: Leaving directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (4.362s)

At time +2s client sent 1024 bytes to 10.1.2.4 port 16
At time +2.0078s server received 1024 bytes from 10.1.1.1 port 49153
At time +2.0078s server sent 1024 bytes to 10.1.1.1 port 49153
At time +2.01761s client received 1024 bytes from 10.1.2.4 port 16
ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35$
```

PRACTICAL NO. 5

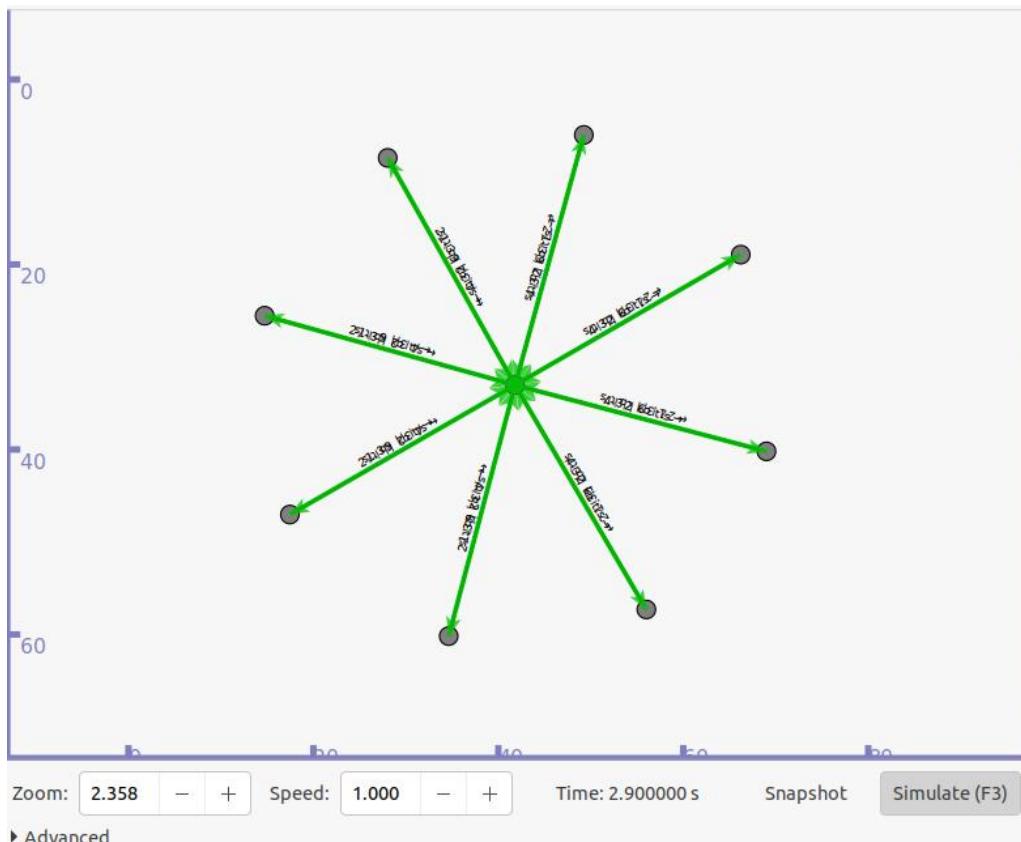
Program to simulate Star topology

Open the ns3.35 folder , go to examples and open tcp folder.

You will find the star.cc there ..Copy the star.cc file to scratch folder

Open the terminal in ns3.35 and compile the star.cc file

```
ubuntu@ubuntu-virtual-machine:~/ns3/ns-allinone-3.35/ns-3.35$ ./waf --run star
Waf: Entering directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
[2895/3028] Linking build/examples/tcp/ns3.35-star-debug
Waf: Leaving directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (1.687s)
ubuntu@ubuntu-virtual-machine:~/ns3/ns-allinone-3.35/ns-3.35$ ./waf --run star -vis
Waf: Entering directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
[2900/3028] Linking build/examples/tcp/ns3.35-star-debug
Waf: Leaving directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (1.539s)
Could not load plugin 'show_last_packets.py': No module named 'kiwi'
Could not load icon applets-screenshooter due to missing gnomedesktop Python module
scanning topology: 9 nodes...
scanning topology: calling graphviz layout
scanning topology: all done.
```



CODE:

```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/netanim-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
#include "ns3/point-to-point-layout-module.h"

// Network topology (default)
//
//
//
n2 n3 n4
\|/
```

```
.  
. .  
// \/.  
// n1--- n0---n5 .  
// /\ .  
// / \ .  
//  
//  
n8 n7 n6 .  
  
using namespace ns3;  
NS_LOG_COMPONENT_DEFINE ("Star");  
int  
main (int argc, char *argv[])  
{  
  
//  
// Set up some default values for the simulation.  
//  
Config::SetDefault ("ns3::OnOffApplication::PacketSize", UintegerValue (137));  
  
// ??? try and stick 15kb/s into the data rate  
Config::SetDefault ("ns3::OnOffApplication::DataRate", StringValue ("14kb/s"));  
  
//  
// Default number of nodes in the star. Overridable by command line argument.  
//  
uint32_t nSpokes = 8;  
  
CommandLine cmd;
```

```
cmd.AddValue ("nSpokes", "Number of nodes to place in the star", nSpokes);
cmd.Parse (argc, argv);
```

```
NS_LOG_INFO ("Build star topology.");
```

```
PointToPointHelper pointToPoint;
```

```
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
```

```
pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));
```

```
PointToPointStarHelper star (nSpokes, pointToPoint);
```

```
NS_LOG_INFO ("Install internet stack on all nodes.");
```

```
InternetStackHelper internet;
```

```
star.InstallStack (internet);
```

```
NS_LOG_INFO ("Assign IP Addresses.");
```

```
star.AssignIpv4Addresses (Ipv4AddressHelper ("10.1.1.0", "255.255.255.0"));
```

```
NS_LOG_INFO ("Create applications.");
```

```
//
```

```
// Create a packet sink on the star "hub" to receive packets.
```

```
//
```

```
uint16_t port = 50000;
```

```
Address hubLocalAddress (InetSocketAddress (Ipv4Address::GetAny (), port));
```

```
PacketSinkHelper packetSinkHelper ("ns3::TcpSocketFactory", hubLocalAddress);
```

```
ApplicationContainer hubApp = packetSinkHelper.Install (star.GetHub ());
```

```
hubApp.Start (Seconds (1.0));
```

```
hubApp.Stop (Seconds (10.0));
```

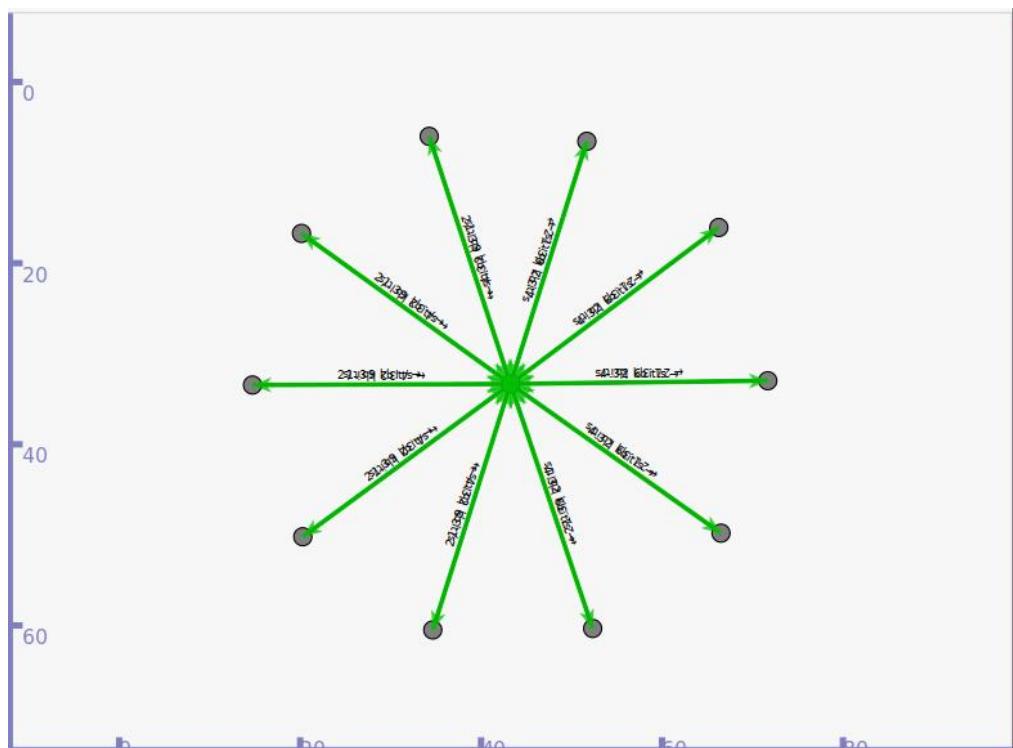
```
//  
// Create OnOff applications to send TCP to the hub, one on each spoke node.  
//  
OnOffHelper onOffHelper ("ns3::TcpSocketFactory", Address());  
  
onOffHelper.SetAttribute ("OnTime", StringValue  
("ns3::ConstantRandomVariable[Constant=1]"));  
onOffHelper.SetAttribute ("OffTime", StringValue  
("ns3::ConstantRandomVariable[Constant=0]"));  
  
ApplicationContainer spokeApps;  
  
for (uint32_t i = 0; i < star.SpokeCount (); ++i)  
{  
    AddressValue remoteAddress (InetSocketAddress (star.GetHubIpv4Address (i),  
port));  
    onOffHelper.SetAttribute ("Remote", remoteAddress);  
    spokeApps.Add (onOffHelper.Install (star.GetSpokeNode (i)));  
}  
spokeApps.Start (Seconds (1.0));  
spokeApps.Stop (Seconds (10.0));  
  
NS_LOG_INFO ("Enable static global routing.");  
//  
// Turn on global static routing so we can actually be routed across the star.  
//  
Ipv4GlobalRoutingHelper::PopulateRoutingTables ();  
  
NS_LOG_INFO ("Enable pcap tracing.");  
//
```

```
// Do pcap tracing on all point-to-point devices on all nodes.  
  
//  
pointToPoint.EnablePcapAll ("star");  
  
NS_LOG_INFO ("Run Simulation.");  
Simulator::Run ();  
Simulator::Destroy ();  
NS_LOG_INFO ("Done.");  
return 0;  
}
```

This NS-3 simulation code creates a star network topology with one central hub node connected to multiple spoke nodes (default 8). Each spoke node runs an OnOff TCP application sending traffic to a packet sink application on the hub node. The point-to-point links have 5 Mbps bandwidth and 2 ms delay. The simulation runs from 1 to 10 seconds, with global static routing enabled to route packets properly. It also enables pcap tracing to capture packet data for analysis. Overall, it demonstrates basic TCP communication in a star topology with traffic flowing from multiple spoke nodes to a single central hub.

To make changes in code use gedit command to open the cc file and make changes in it.

```
ubantu@ubuntu-virtual-machine:~/ns3/ns-allinone-3.35/ns-3.35$ ./waf --run scratch/star  
Waf: Entering directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'  
[2894/3028] Linking build/examples/tcp/ns3.35-star-debug  
Waf: Leaving directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'  
Build commands will be stored in build/compile_commands.json  
'build' finished successfully (1.447s)  
ubantu@ubuntu-virtual-machine:~/ns3/ns-allinone-3.35/ns-3.35$ ./waf --run scratch/star --vis  
Waf: Entering directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'  
Waf: Leaving directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'  
Build commands will be stored in build/compile_commands.json  
'build' finished successfully (0.849s)  
Could not load plugin 'show_last_packets.py': No module named 'kiwi'  
Could not load icon applets-screenshooter due to missing gnomedesktop Python module  
scanning topology: 11 nodes...  
scanning topology: calling graphviz layout  
scanning topology: all done.  
ubantu@ubuntu-virtual-machine:~/ns3/ns-allinone-3.35/ns-3.35$
```



PRACTICAL NO. 6

Program to simulate mesh topology.

Open ns3.35 → src → mesh → examples → mesh.cc

Copy the mesh.cc file to scratch folder

Open terminal in ns3.35 and compile the mesh.cc file

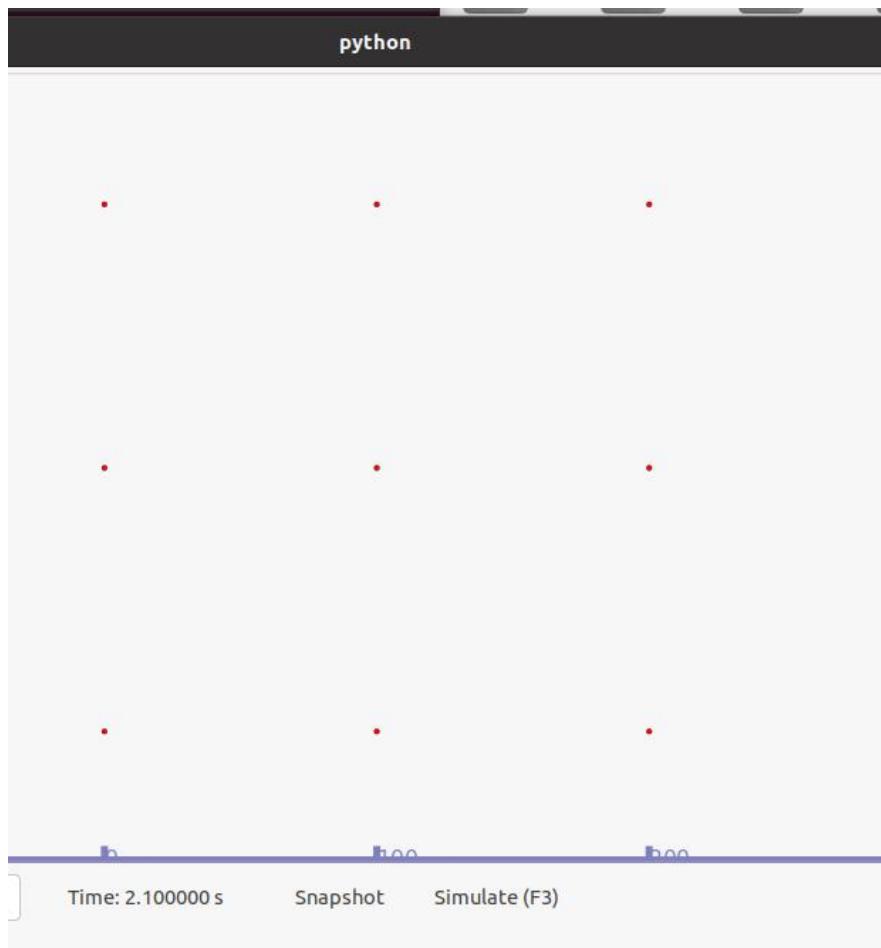
```
ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35$ ./waf --run mesh
Waf: Entering directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
[2081/2153] Compiling scratch/mesh.cc
[2112/2153] Linking build/scratch/scratch-simulator
[2113/2153] Linking build/scratch/mesh
Waf: Leaving directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (4.885s)
Printing mesh point device #0 diagnostics to mp-report-0.xml
Printing mesh point device #1 diagnostics to mp-report-1.xml
Printing mesh point device #2 diagnostics to mp-report-2.xml
Printing mesh point device #3 diagnostics to mp-report-3.xml
Printing mesh point device #4 diagnostics to mp-report-4.xml
Printing mesh point device #5 diagnostics to mp-report-5.xml
Printing mesh point device #6 diagnostics to mp-report-6.xml
Printing mesh point device #7 diagnostics to mp-report-7.xml
Printing mesh point device #8 diagnostics to mp-report-8.xml
```

```
ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35$ ./waf --run mesh --vis
Waf: Entering directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
Waf: Leaving directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.449s)
Could not load plugin 'show_last_packets.py': No module named 'kiwi'
Could not load icon applets-screenshooter due to missing gnomedesktop Python mod
```

Roll no: C24004
MCA Div: A

MCA SEM 2

Networking with Linux Lab



PRACTICAL NO. 7

Program to simulate Hybrid Topology

Install the necessary libraries for netanim

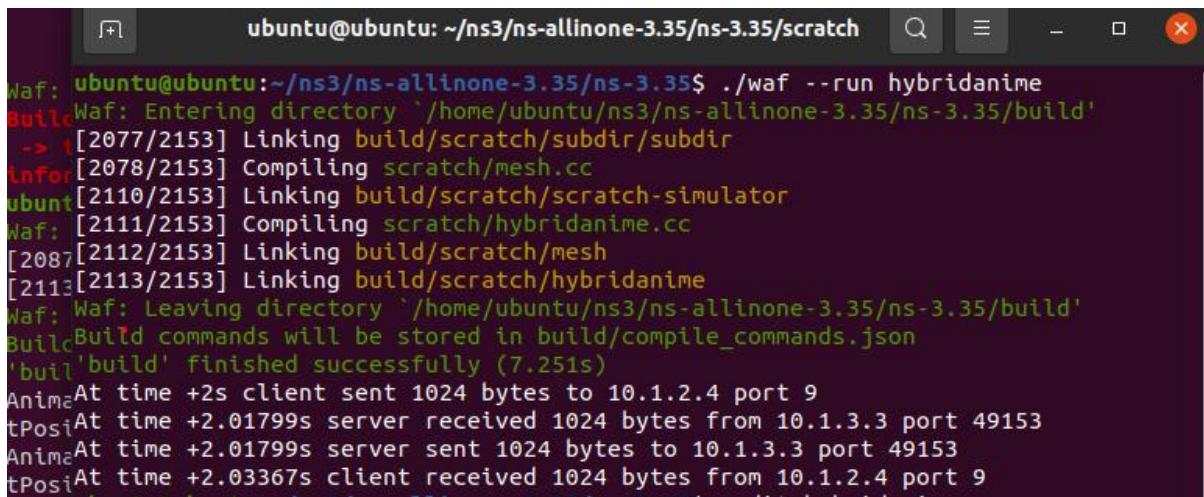
```
ubuntu@ubuntu:~$ sudo apt update
[sudo] password for ubuntu:
Hit:1 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
6 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ubuntu:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
Get more security updates through Ubuntu Pro with 'esm-apps' enabled:
  ipython3 python2.7-minimal libcgraph6 python2.7 python3-ipython
  liblab-gamut1 libcdt5 libpathplan4 libgvpr2 libpython2.7-minimal libgvc6
  libpython2.7-stdlib traceroute graphviz
Learn more about Ubuntu Pro at https://ubuntu.com/pro
The following NEW packages will be installed:
  ubuntu-pro-client ubuntu-pro-client-l10n
The following packages will be upgraded:
  dns-root-data python3-update-manager snapd ubuntu-adantage-tools
```

```
ubuntu@ubuntu:~$ sudo apt install qt5-default mercurial
Reading package lists... Done
Building dependency tree
Reading state information... Done
mercurial is already the newest version (5.3.1-1ubuntu1).
qt5-default is already the newest version (5.12.8+dfsg-0ubuntu2.1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ubuntu:~$
```

```
ubuntu@ubuntu: $ sudo apt install gir1.2-goocanvas-2.0 python3-gi python3-gi-cairo python3-pygraphviz gir1.2-gtk-3.0 ipython3
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-gi is already the newest version (3.36.0-1).
python3-gi-cairo is already the newest version (3.36.0-1).
gir1.2-goocanvas-2.0 is already the newest version (2.0.4-1).
ipython3 is already the newest version (7.13.0-1).
python3-pygraphviz is already the newest version (1.5-4build1).
gir1.2-gtk-3.0 is already the newest version (3.24.20-0ubuntu1.2).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ubuntu: $
```

Now from examples→tutorial copy paste the third.cc file to scratch

Rename the file name to hybridanim and compile the file.



```
ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35/scratch$ ./waf --run hybridanime
Waf: Entering directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
[2077/2153] Linking build/scratch/subdir/subdir
[2078/2153] Compiling scratch/mesh.cc
[2110/2153] Linking build/scratch/scratch-simulator
[2111/2153] Compiling scratch/hybridanime.cc
[2112/2153] Linking build/scratch/mesh
[2113/2153] Linking build/scratch/hybridanime
Waf: Leaving directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (7.251s)

At time +2s client sent 1024 bytes to 10.1.2.4 port 9
At time +2.01799s server received 1024 bytes from 10.1.3.3 port 49153
At time +2.01799s server sent 1024 bytes to 10.1.3.3 port 49153
At time +2.03367s client received 1024 bytes from 10.1.2.4 port 9
```

CODE:

```
/* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -*- */
/*
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License version 2 as
 * published by the Free Software Foundation;
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
 */

#include "ns3/core-module.h"
#include "ns3/network-module.h"
```

```
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
//netanimation
[Screenshot Placeholder: Insert NetAnim or Wireshark output image here]
#include "ns3/netanim-module.h"
#include "ns3/mobility-module.h"
```

```
// Default Network Topology
```

```
//
// 10.1.1.0
// n0 n1
// point-to-point
//
using namespace ns3;
```

```
NS_LOG_COMPONENT_DEFINE ("FirstScriptExample");
```

```
int
main (int argc, char *argv[])
{
CommandLine cmd ( FILE );
cmd.Parse (argc, argv);
```

```
Time::SetResolution (Time::NS);
LogComponentEnable ("UdpEchoClientApplication", LOG_LEVEL_INFO);
LogComponentEnable ("UdpEchoServerApplication", LOG_LEVEL_INFO);
```

```
NodeContainer nodes;
nodes.Create (2);
```

```
PointToPointHelper pointToPoint;  
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));  
pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));  
NetDeviceContainer devices;  
devices = pointToPoint.Install (nodes);
```

```
InternetStackHelper stack;  
stack.Install (nodes);
```

```
Ipv4AddressHelper address;  
address.SetBase ("10.1.1.0", "255.255.255.0");
```

```
Ipv4InterfaceContainer interfaces = address.Assign (devices);
```

```
UdpEchoServerHelper echoServer (9);
```

```
ApplicationContainer serverApps = echoServer.Install (nodes.Get (1));  
serverApps.Start (Seconds (1.0));  
serverApps.Stop (Seconds (10.0));
```

```
UdpEchoClientHelper echoClient (interfaces.GetAddress (1), 9);  
echoClient.SetAttribute ("MaxPackets", UintegerValue (1));  
echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));  
echoClient.SetAttribute ("PacketSize", UintegerValue (1024));
```

```
ApplicationContainer clientApps = echoClient.Install (nodes.Get (0));  
clientApps.Start (Seconds (2.0));  
clientApps.Stop (Seconds (10.0));
```

```
MobilityHelper mobility;  
mobility.SetMobilityModel("ns3::ConstantPositionMobilityModel");  
mobility.Install(nodes);  
AnimationInterface anim("hybridanim.xml");  
[Screenshot Placeholder: Insert NetAnim or Wireshark output image here]  
anim.SetConstantPosition(p2pNodes.Get(0),10.0,10.0);  
anim.SetConstantPosition(p2pNodes.Get(0),20.0,20.0);  
Simulator::Run ();  
Simulator::Destroy ();  
return 0;  
}
```

Now to make the changes in .cc file open it using gedit and add this lines to create an xml file and save it.

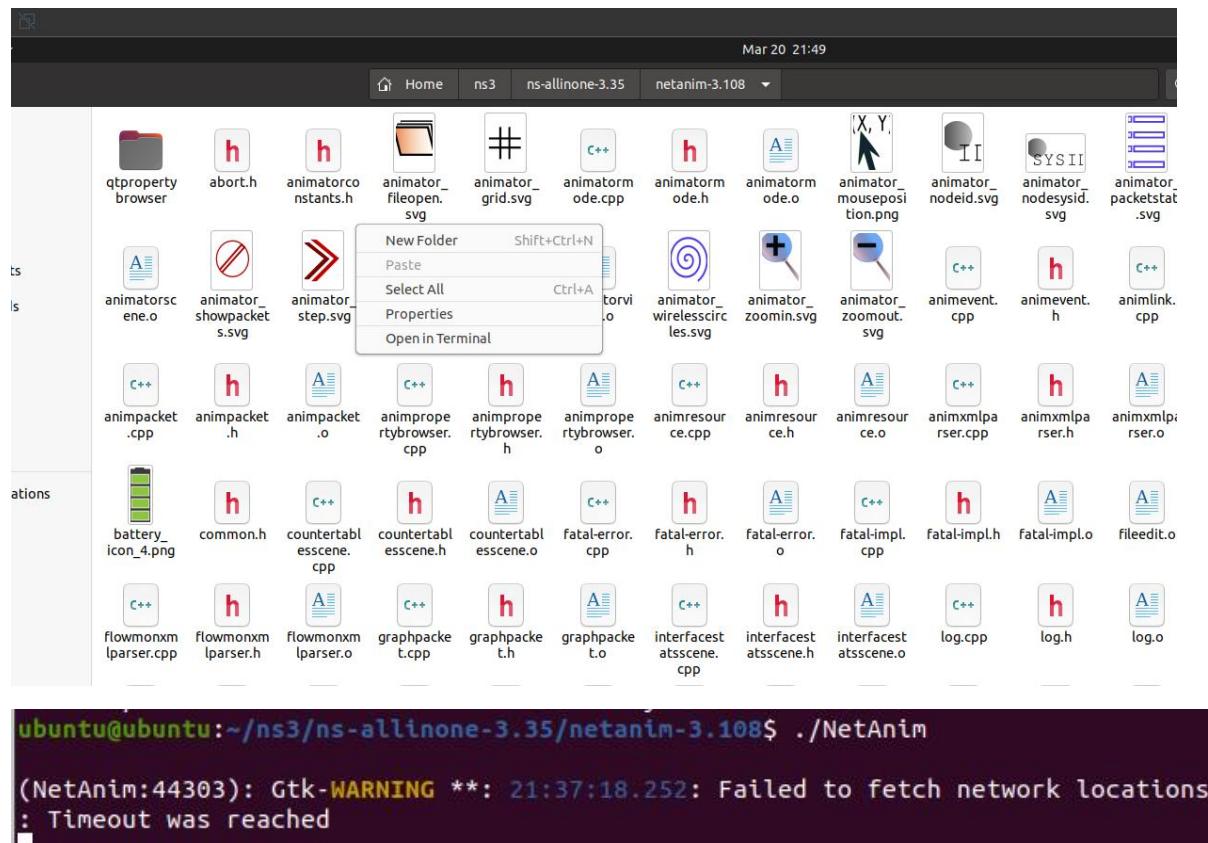
```
phy.EnablePcap ("third", apDevices.Get (0));  
csma.EnablePcap ("third", csmaDevices.Get (0), true);  
}  
AnimationInterface anim("hybridanim.xml");  
anim.SetConstantPosition(p2pNodes.Get(0),10.0,10.0);  
anim.SetConstantPosition(p2pNodes.Get(0),20.0,20.0);  
Simulator::Run ();  
Simulator::Destroy ();  
return 0;  
}
```

Compile the saved file.

```
ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35$ ./waf --run scratch/hybridanim
Waf: Entering directory '/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
[2087/2153] Compiling scratch/hybridanim.cc
[2113/2153] Linking build/scratch/hybridanim
Waf: Leaving directory '/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (4.716s)
AnimationInterface WARNING:Node:1 Does not have a mobility model. Use SetConstan
tPosition if it is stationary
AnimationInterface WARNING:Node:2 Does not have a mobility model. Use SetConstan
tPosition if it is stationary
AnimationInterface WARNING:Node:3 Does not have a mobility model. Use SetConstan
tPosition if it is stationary
AnimationInterface WARNING:Node:4 Does not have a mobility model. Use SetConstan
tPosition if it is stationary
AnimationInterface WARNING:Node:1 Does not have a mobility model. Use SetConstan
tPosition if it is stationary
AnimationInterface WARNING:Node:2 Does not have a mobility model. Use SetConstan
tPosition if it is stationary
AnimationInterface WARNING:Node:3 Does not have a mobility model. Use SetConstan
tPosition if it is stationary
AnimationInterface WARNING:Node:4 Does not have a mobility model. Use SetConstan
tPosition if it is stationary
```

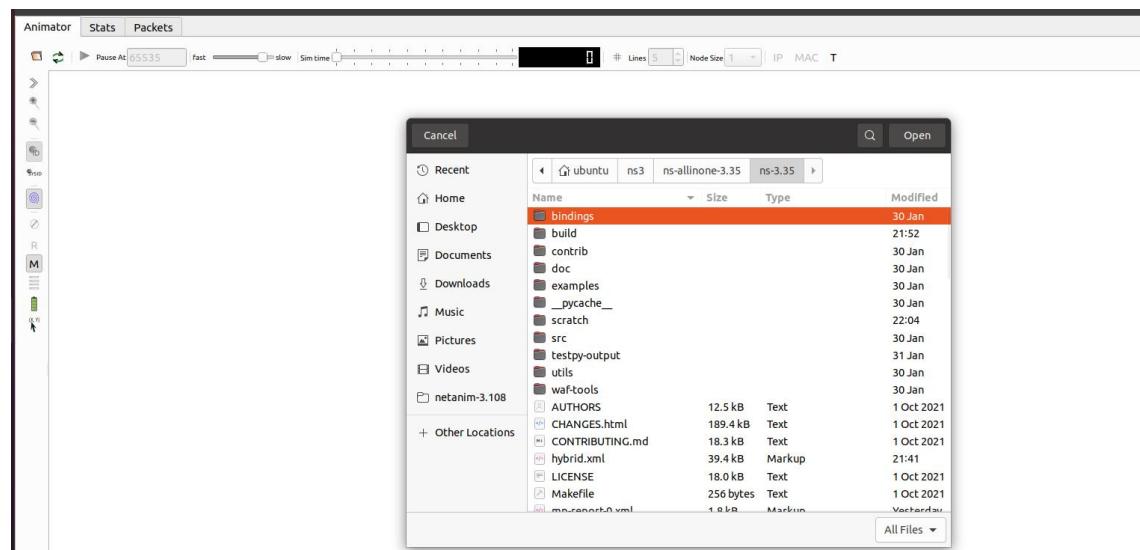
Open Netanim

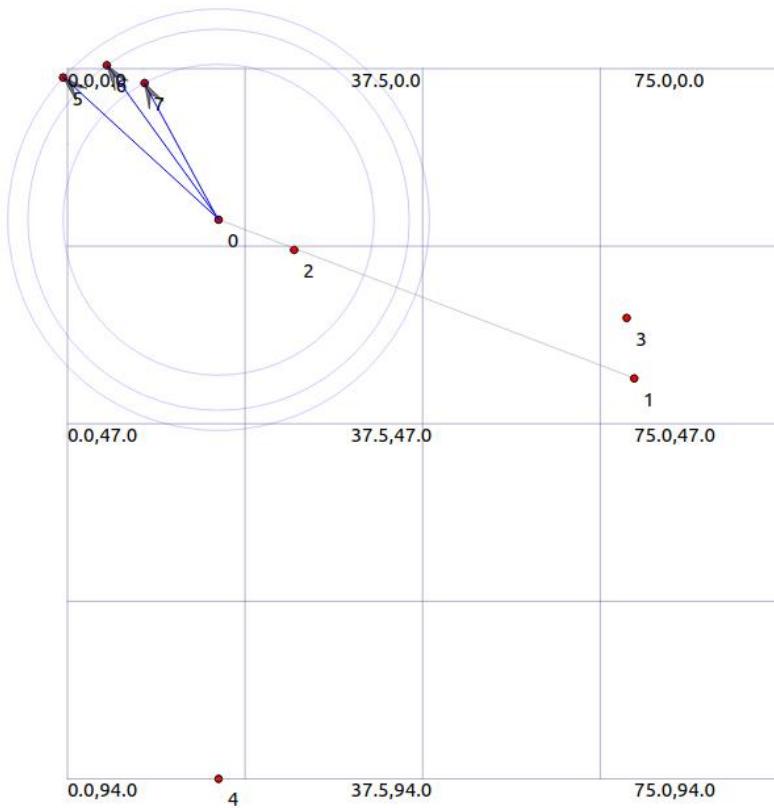


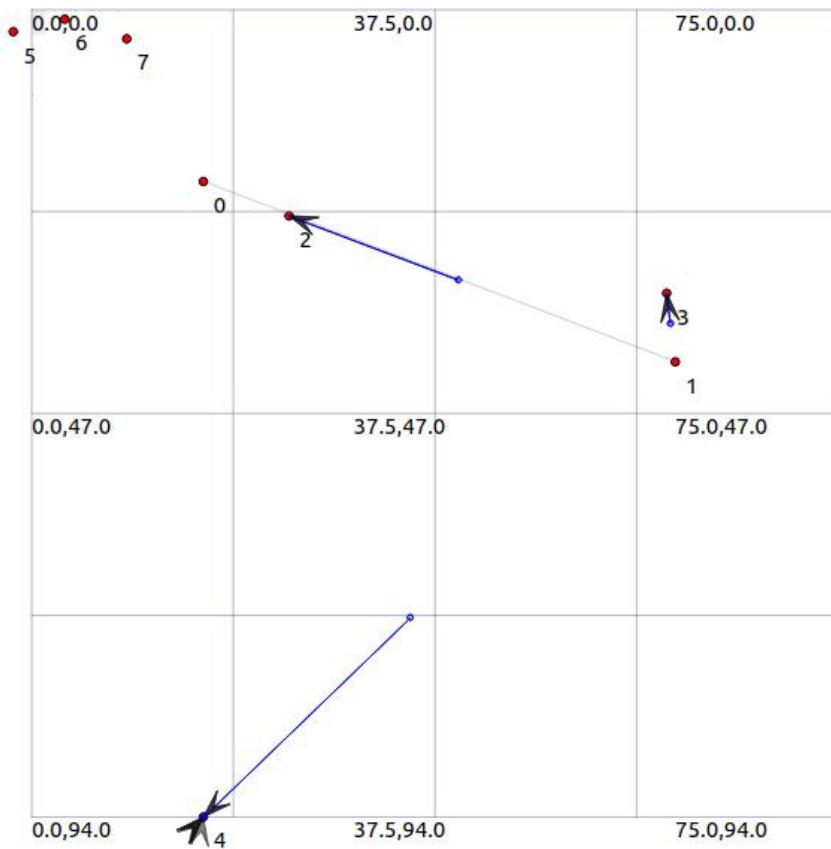


Search for the Xml file created in Netanim

Open the file.



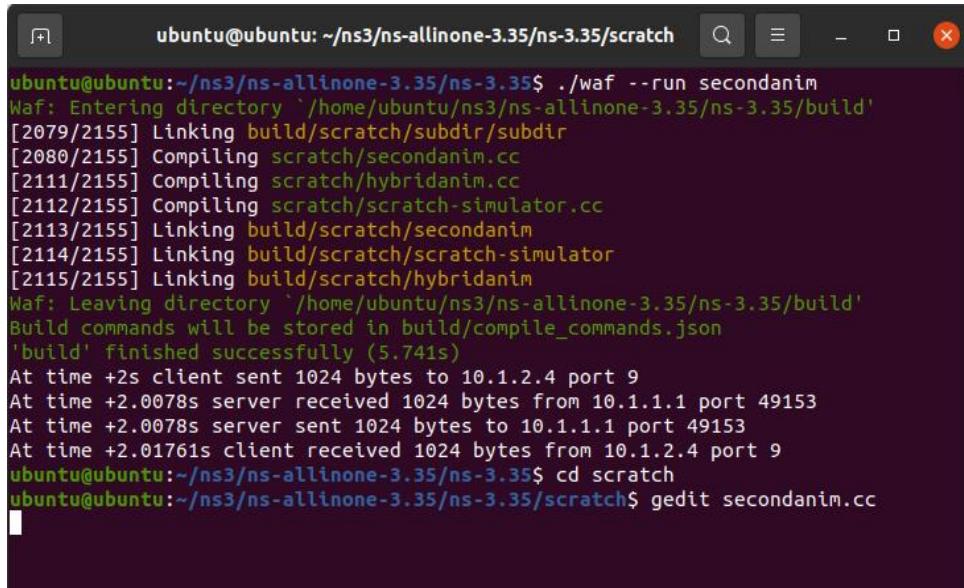




For bus topology

Copy paste third.cc to scratch folder from examples folder and rename it.

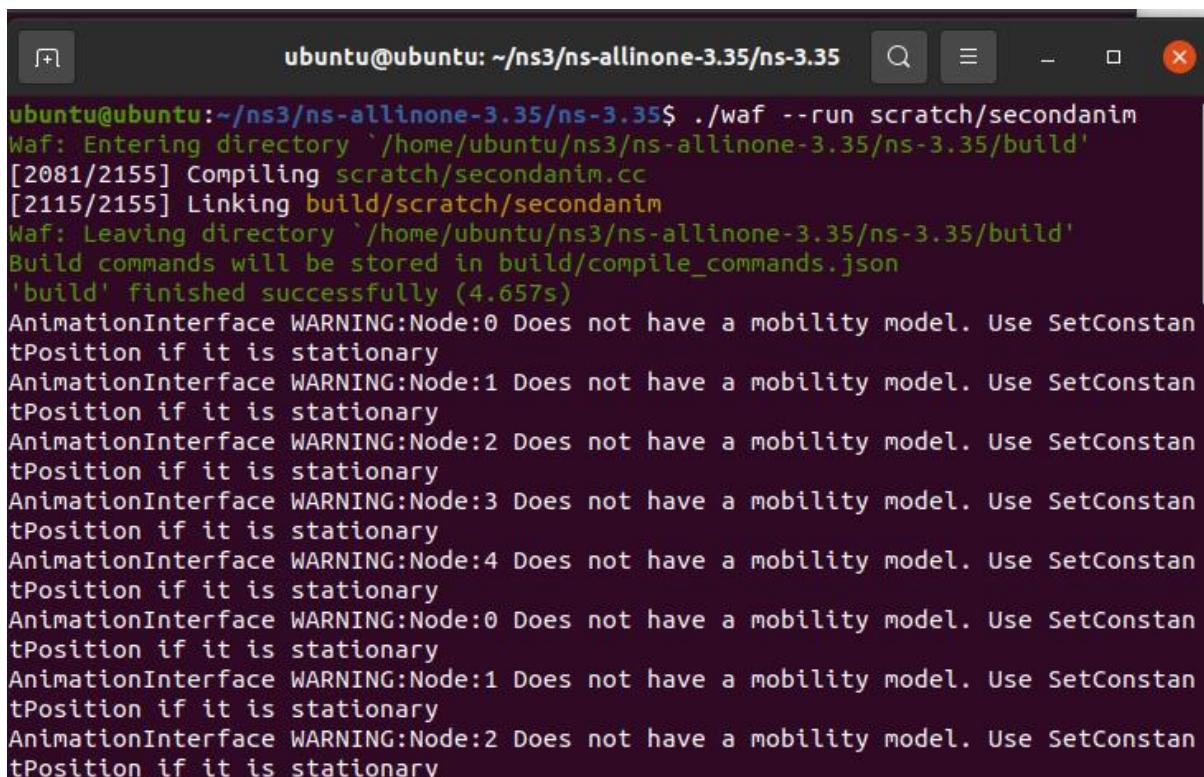
Compile the file.



A terminal window titled "ubuntu@ubuntu: ~/ns3/ns-allinone-3.35/ns-3.35/scratch". The window shows the command "ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35\$./waf --run secondanim" being run. The output of the command is displayed, showing the compilation of various source files like scratch/secondanim.cc, scratch/hybridanim.cc, and scratch-simulator.cc, followed by the linking process. It also shows network traffic between a client and a server. The terminal ends with the command "ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35\$ gedit secondanim.cc".

Make changes in the code and Also write the command to create xml file.

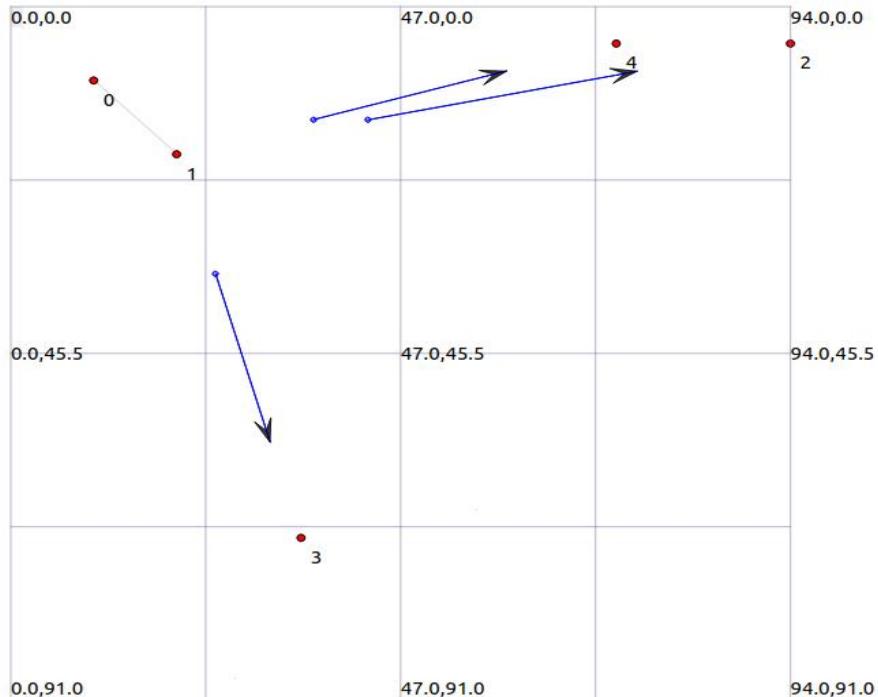
Compile the changed file



A terminal window titled "ubuntu@ubuntu: ~/ns3/ns-allinone-3.35/ns-3.35/scratch/secondanim". The window shows the command "ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35\$./waf --run scratch/secondanim" being run. The output shows the same compilation and linking process as before, but includes multiple "WARNING" messages from AnimationInterface indicating that nodes do not have a mobility model and suggesting the use of SetConstantPosition if they are stationary. The terminal ends with the command "ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35\$".

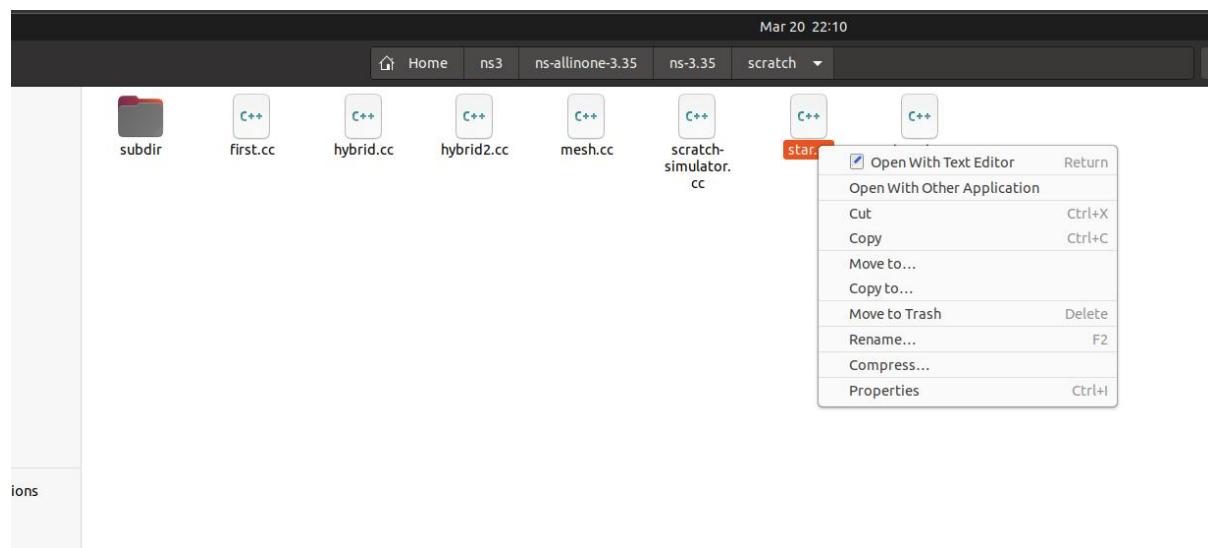
Open Netanim

Search for created xml file and open the xml file.



For Netanim Star Topology

Copy paste Star.cc file to scratch folder and rename it.

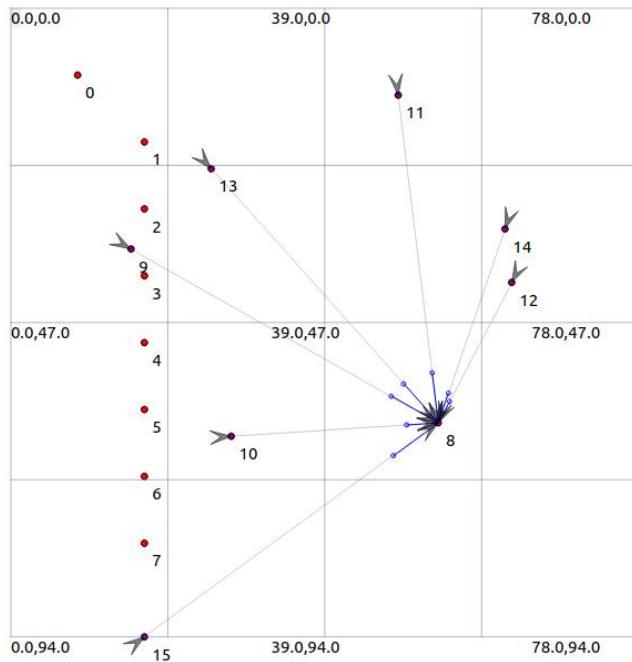


The terminal window shows the command `./waf --run hybridstar` being executed. The output indicates the build process for the `hybridstar` module, which includes linking and compiling various source files like `mesh.cc`, `hybrid.cc`, and `hybridstar.cc`. The build is successful, and the commands are stored in `build/compile_commands.json`.

```
ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35$ ./waf --run hybridstar
Waf: Entering directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35'
[2952/3034] Linking build/scratch/scratch-simulator
[2953/3034] Compiling scratch/mesh.cc
[2990/3034] Compiling scratch/hybrid.cc
[2991/3034] Compiling scratch/hybridstar.cc
[2992/3034] Linking build/scratch/mesh
[2993/3034] Linking build/scratch/hybrid
[2994/3034] Linking build/scratch/hybridstar
Waf: Leaving directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (6.709s)
ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35$
```

The terminal window shows the command `./NetAnim` being executed. The output is currently blank, indicating the program is running or has not yet completed its output.

```
ubuntu@ubuntu:~/ns3/ns-allinone-3.35/netanim-3.108$ ./NetAnim
```



PRACTICAL NO. 8

Program to simulate UDP Client server.

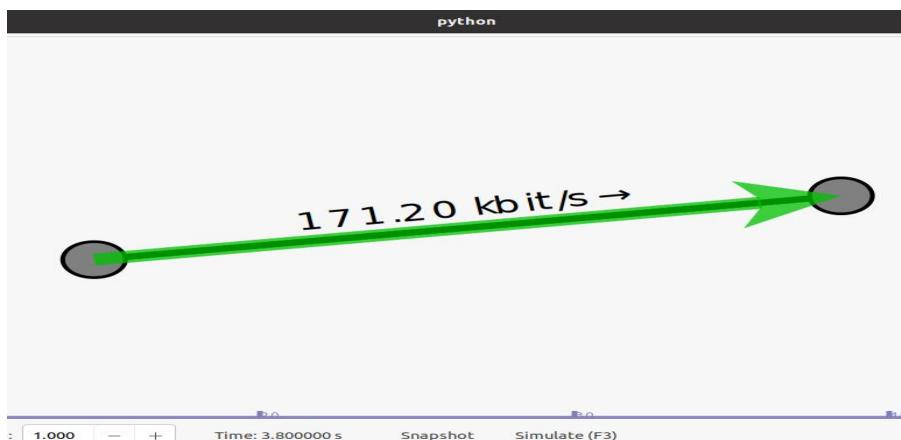
UDP (User Datagram Protocol) is a simple, connectionless transport layer protocol used for fast data transmission. Unlike TCP, it doesn't establish a connection before sending data and doesn't guarantee delivery, order, or error checking. This makes UDP faster and more efficient, ideal for real-time applications like video streaming, online gaming, and VoIP where speed is more important than reliability.

Open the ns3.35 folder and copy paste the udp-client-server file to scratch.

Compile the file.

```
ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35$ ./waf --run udp-client-server
Waf: Entering directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
[2076/2161] Compiling scratch/udp-client-server.cc
[2079/2161] Linking build/scratch/subdir/subdir
[2110/2161] Linking build/scratch/mfourth
[2111/2161] Compiling scratch/secondanim.cc
[2112/2161] Compiling scratch/hybridanim.cc
[2113/2161] Compiling scratch/scratch-simulator.cc
[2114/2161] Linking build/scratch/udp-client-server
[2115/2161] Linking build/scratch/secondanim
[2116/2161] Linking build/scratch/hybridanim
[2117/2161] Linking build/scratch/scratch-simulator
[2118/2161] Compiling scratch/staranim.cc
[2119/2161] Compiling scratch/Hybrid.cc
[2120/2161] Linking build/scratch/staranim
[2121/2161] Linking build/scratch/Hybrid
```

```
ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35$ ./waf --run udp-client-server --vis
Waf: Entering directory '/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
Waf: Leaving directory '/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.462s)
Could not load plugin 'show_last_packets.py': No module named 'kiwi'
Could not load icon applets-screenshooter due to missing gnomedesktop Python module
scanning topology: 2 nodes...
scanning topology: calling graphviz layout
scanning topology: all done.
TraceDelay TX 1024 bytes to 10.1.1.2 Uid: 0 Time: +2s
TraceDelay: RX 1024 bytes from 10.1.1.1 Sequence Number: 0 Uid: 0 TXtime: +2e+09ns
RXtime: +2.01592e+09ns Delay: +1.59188e+07ns
TraceDelay TX 1024 bytes to 10.1.1.2 Uid: 11 Time: +2.05s
TraceDelay: RX 1024 bytes from 10.1.1.1 Sequence Number: 1 Uid: 11 TXtime: +2.05e+09ns
RXtime: +2.05371e+09ns Delay: +3.712e+06ns
TraceDelay TX 1024 bytes to 10.1.1.2 Uid: 14 Time: +2.1s
```



PRACTICAL NO. 9

Program to simulate DHCP Server and clients.

DHCP (Dynamic Host Configuration Protocol) is a network protocol used to automatically assign IP addresses and other network settings (like gateway and DNS) to devices on a network. Instead of manually configuring each device, a DHCP server dynamically provides this information, making network management easier and reducing configuration errors. It's commonly used in home routers, offices, and enterprise networks.

CODE:

```
/* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -*- */
/*
 * Copyright (c) 2011 UPB
 * Copyright (c) 2017 NITK Surathkal
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License version 2 as
 * published by the Free Software Foundation;
```

*

* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.

*

* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

*

* Author: Radu Lupu <rlupu@elcom.pub.ro>
* Ankit Deepak <adadeepak8@gmail.com>
* Deepti Rajagopal <deoptir96@gmail.com>

*

*/

/*

* Network layout:

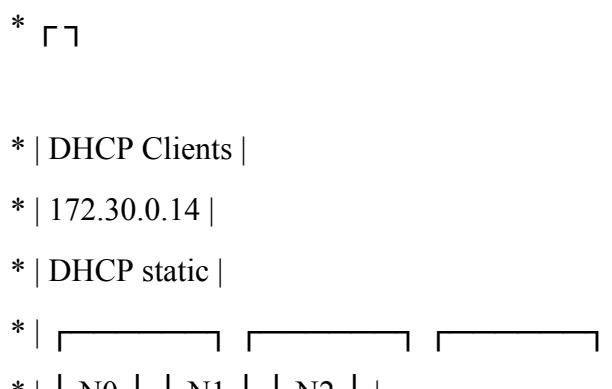
*

* R0 is a DHCP server. The DHCP server announced R1 as the default router.

* Nodes N1 will send UDP Echo packets to node A.

*

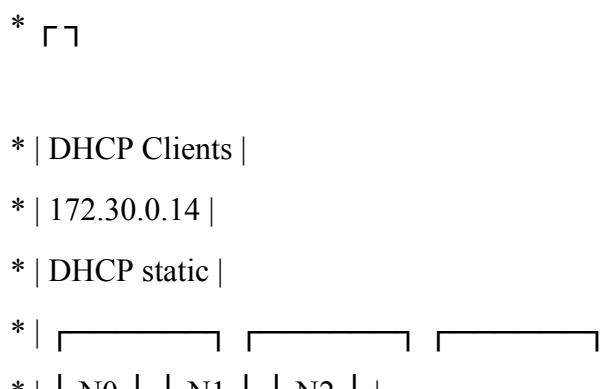
*

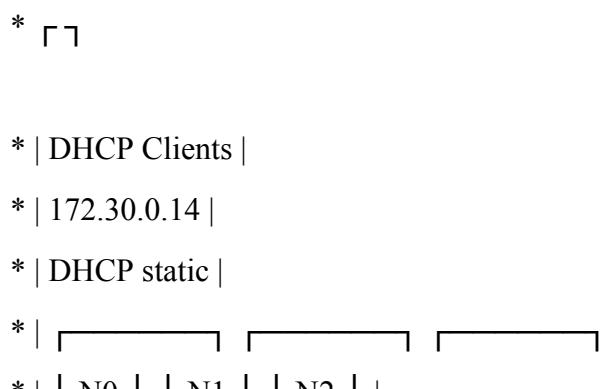
* 

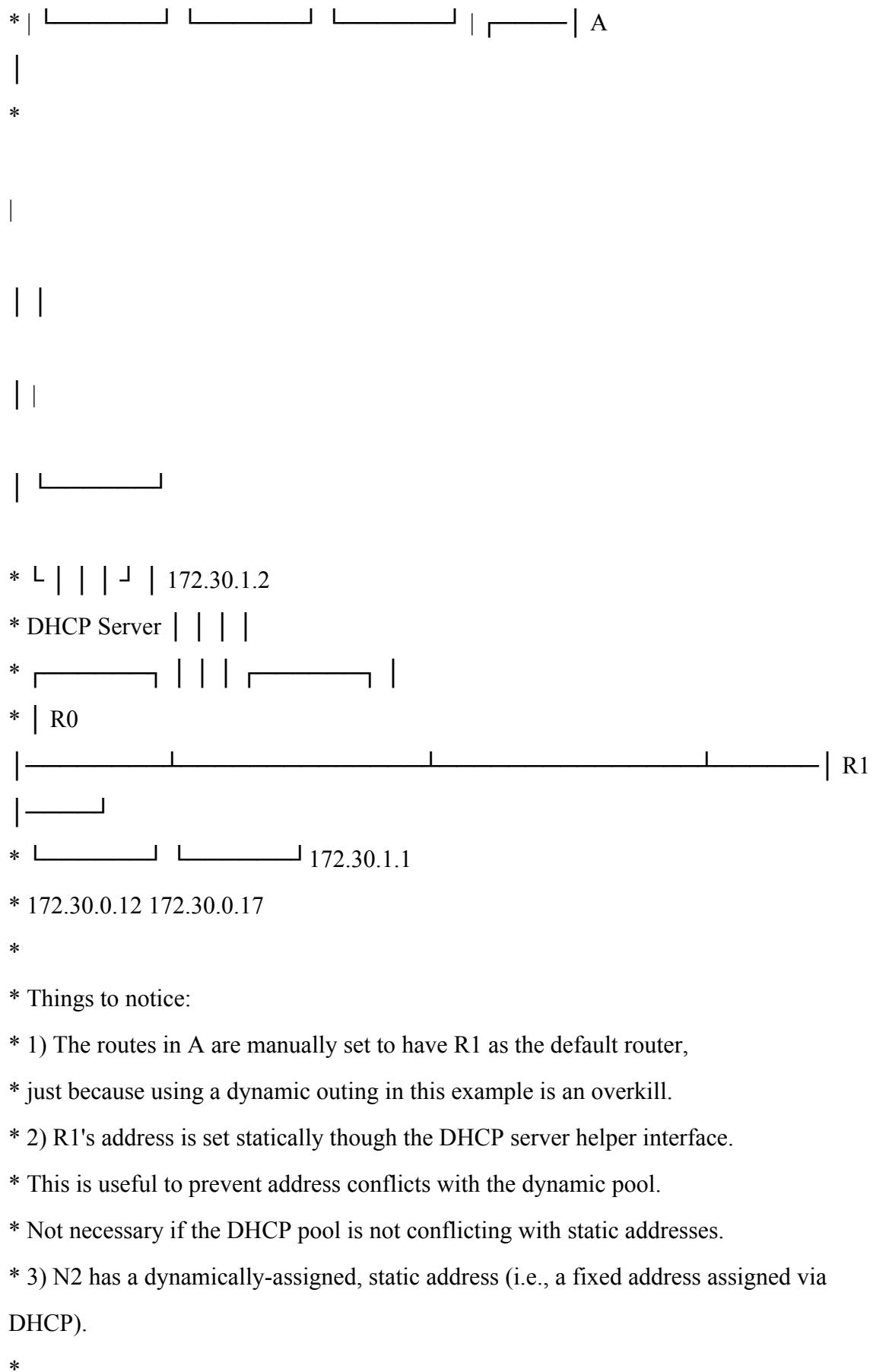
* | DHCP Clients |

* | 172.30.0.14 |

* | DHCP static |

* | 

* | | N0 | | N1 | | N2 | | 



*/

```
#include "ns3/core-module.h"
#include "ns3/internet-apps-module.h"
#include "ns3/csma-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"

//netanim code
#include "ns3/netanim-module.h"
#include "ns3/mobility-module.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("DhcpExample");
int
main (int argc, char *argv[])
{
    CommandLine cmd (FILE);

    bool verbose = false;
    bool tracing = false;
    cmd.AddValue ("verbose", "turn on the logs", verbose);
    cmd.AddValue ("tracing", "turn on the tracing", tracing);

    cmd.Parse (argc, argv);
    // GlobalValue::Bind ("ChecksumEnabled", BooleanValue (true));
    if (verbose)
    {
        LogComponentEnable ("DhcpServer", LOG_LEVEL_ALL);
```

```
LogComponentEnable ("DhcpClient", LOG_LEVEL_ALL);
LogComponentEnable ("UdpEchoServerApplication", LOG_LEVEL_INFO);
LogComponentEnable ("UdpEchoClientApplication", LOG_LEVEL_INFO);
}
```

```
Time stopTime = Seconds (20);
```

```
NS_LOG_INFO ("Create nodes.");
NodeContainer nodes;
NodeContainer router;
nodes.Create (3);
router.Create (2);
```

```
NodeContainer net (nodes, router);
```

```
NS_LOG_INFO ("Create channels.");
CsmaHelper csma;
```

```
csma.SetChannelAttribute ("DataRate", StringValue ("5Mbps"));
csma.SetChannelAttribute ("Delay", StringValue ("2ms"));
csma.SetDeviceAttribute ("Mtu", UIntegerValue (1500));
NetDeviceContainer devNet = csma.Install (net);
```

```
NodeContainer p2pNodes;
p2pNodes.Add (net.Get (4));
p2pNodes.Create (1);
```

```
PointToPointHelper pointToPoint;
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));
```

```
NetDeviceContainer p2pDevices;
p2pDevices = pointToPoint.Install (p2pNodes);

InternetStackHelper tcpip;
tcpip.Install (nodes);
tcpip.Install (router);
tcpip.Install (p2pNodes.Get (1));

Ipv4AddressHelper address;
address.SetBase ("172.30.1.0", "255.255.255.0");
Ipv4InterfaceContainer p2pInterfaces;
p2pInterfaces = address.Assign (p2pDevices);

// manually add a routing entry because we don't want to add a dynamic routing
Ipv4StaticRoutingHelper ipv4RoutingHelper;
Ptr<Ipv4> ipv4Ptr = p2pNodes.Get (1)->GetObject<Ipv4> ();
Ptr<Ipv4StaticRouting> staticRoutingA = ipv4RoutingHelper.GetStaticRouting
(ipv4Ptr);
staticRoutingA->AddNetworkRouteTo (Ipv4Address ("172.30.0.0"), Ipv4Mask ("/24"),
Ipv4Address ("172.30.1.1"), 1);

NS_LOG_INFO ("Setup the IP addresses and create DHCP applications.");
DhcpHelper dhcpHelper;

// The router must have a fixed IP.

Ipv4InterfaceContainer fixedNodes = dhcpHelper.InstallFixedAddress (devNet.Get (4),
Ipv4Address ("172.30.0.17"), Ipv4Mask ("/24"));
// Not really necessary, IP forwarding is enabled by default in IPv4.
```

```
fixedNodes.Get (0).first->SetAttribute ("IpForward", BooleanValue (true));  
  
// DHCP server  
ApplicationContainer dhcpServerApp = dhcpHelper.InstallDhcpServer (devNet.Get (3),  
Ipv4Address ("172.30.0.12"),  
("/24"),  
("172.30.0.15"),  
Ipv4Address ("172.30.0.0"), Ipv4Mask  
  
Ipv4Address ("172.30.0.10"), Ipv4Address  
  
Ipv4Address ("172.30.0.17"));  
  
// This is just to show how it can be done.  
DynamicCast<DhcpServer> (dhcpServerApp.Get (0))->AddStaticDhcpEntry  
(devNet.Get (2)->GetAddress (), Ipv4Address ("172.30.0.14"));  
  
dhcpServerApp.Start (Seconds (0.0));  
dhcpServerApp.Stop (stopTime);  
  
// DHCP clients  
NetDeviceContainer dhcpClientNetDevs;  
dhcpClientNetDevs.Add (devNet.Get (0));  
dhcpClientNetDevs.Add (devNet.Get (1));  
dhcpClientNetDevs.Add (devNet.Get (2));  
  
ApplicationContainer dhcpClients = dhcpHelper.InstallDhcpClient  
(dhcpClientNetDevs);  
dhcpClients.Start (Seconds (1.0));
```

```
dhcpClients.Stop (stopTime);
```

```
UdpEchoServerHelper echoServer (9);
```

```
ApplicationContainer serverApps = echoServer.Install (p2pNodes.Get (1));  
serverApps.Start (Seconds (0.0));  
serverApps.Stop (stopTime);
```

```
UdpEchoClientHelper echoClient (p2pInterfaces.GetAddress (1), 9);
```

```
echoClient.SetAttribute ("MaxPackets", UintegerValue (100));  
echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));  
echoClient.SetAttribute ("PacketSize", UintegerValue (1024));
```

```
ApplicationContainer clientApps = echoClient.Install (nodes.Get (1));  
clientApps.Start (Seconds (10.0));  
clientApps.Stop (stopTime);
```

```
MobilityHelper mobility;  
mobility.SetMobilityModel("ns3::ConstantPositionMobilityModel");  
mobility.Install(nodes);
```

```
AnimationInterface anim("pranay.xml");  
[Screenshot Placeholder: Insert NetAnim or Wireshark output image here]
```

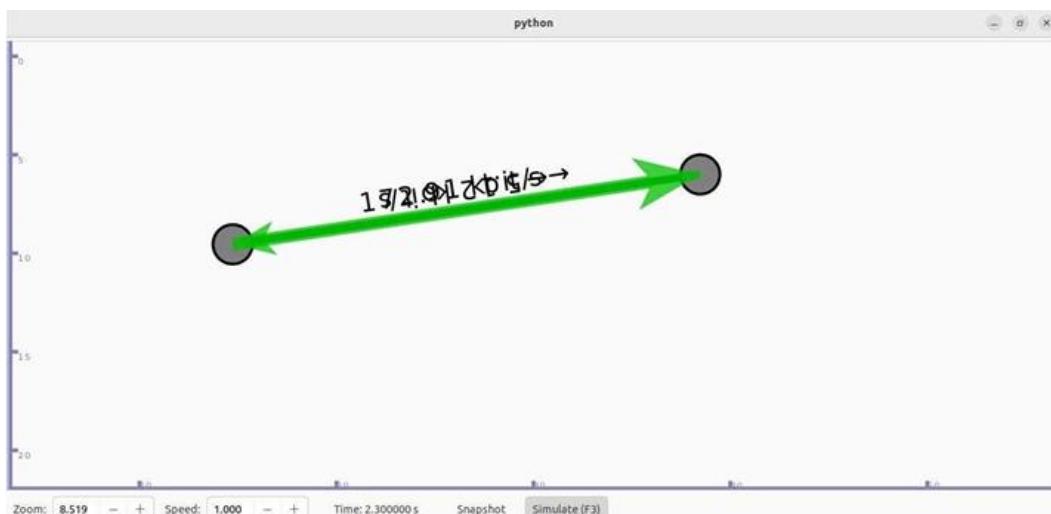
```
AnimationInterface::SetConstantPosition(nodes.Get(0),10,25);  
[Screenshot Placeholder: Insert NetAnim or Wireshark output image here]
```

```
AnimationInterface::SetConstantPosition(nodes.Get(1),40,25);  
[Screenshot Placeholder: Insert NetAnim or Wireshark output image here]  
anim.EnablePacketMetadata(true);
```

```
Simulator::Stop (stopTime + Seconds (10.0));
```

```
if (tracing)
{
    csma.EnablePcapAll ("dhcp-csma");
    pointToPoint.EnablePcapAll ("dhcp-p2p");
}

NS_LOG_INFO ("Run Simulation.");
Simulator::Run ();
Simulator::Destroy ();
NS_LOG_INFO ("Done.");
}
```



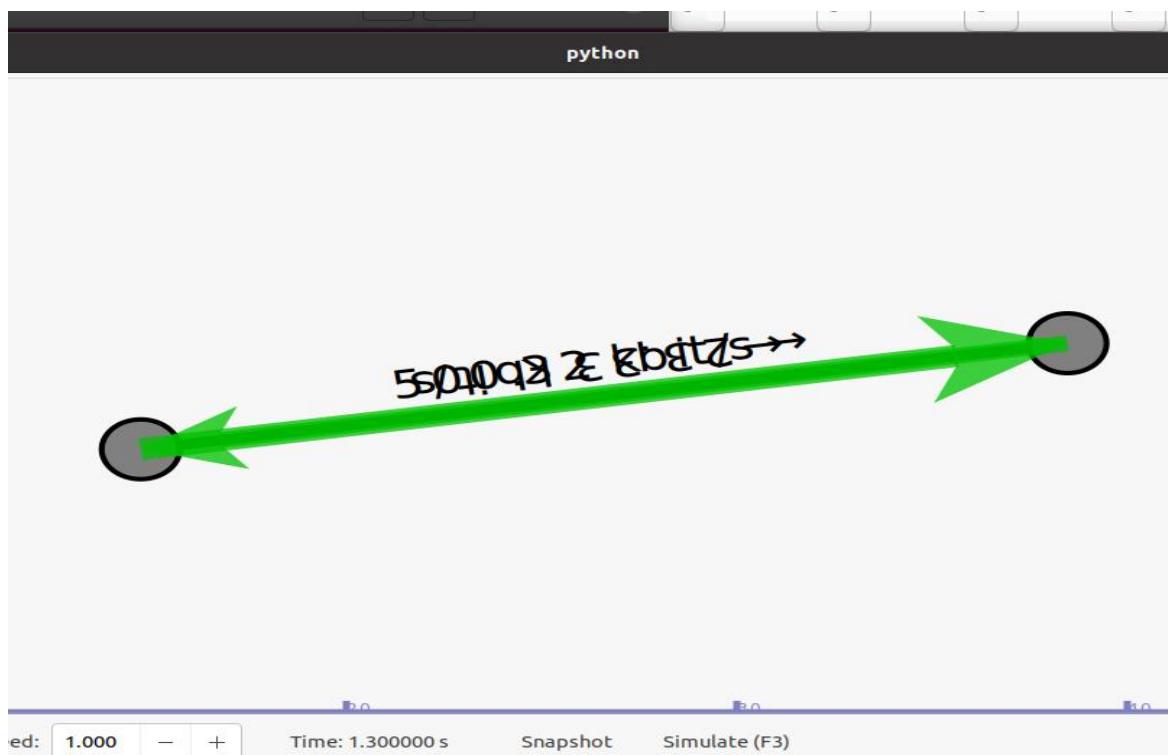
PRACTICAL NO. 10

Program to simulate FTP using TCP

FTP (File Transfer Protocol) is a standard network protocol used to transfer files between a client and a server over a TCP/IP network. It allows users to upload, download, rename, delete, and manage files on a remote server. FTP requires authentication (username and password) but can also allow anonymous access. While it is widely used, FTP is not secure by default, so secure versions like FTPS or SFTP (which uses SSH) are preferred for sensitive data.

```
ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35$ ./waf --run tcp-bulk-send
Waf: Entering directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
[2091/2165] Linking build/scratch/scratch-simulator
Waf: Leaving directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (1.255s)
Total Bytes Received: 553152
```

```
ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35$ ./waf --run tcp-bulk-send --vis  
Waf: Entering directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'  
Waf: Leaving directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'  
Build commands will be stored in build/compile_commands.json  
'build' finished successfully (0.458s)  
Could not load plugin 'show_last_packets.py': No module named 'kiwi'  
Could not load icon applets-screenshooter due to missing gnomedesktop Python module  
scanning topology: 2 nodes...  
scanning topology: calling graphviz layout  
scanning topology: all done.  
/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/src/visualizer/visualizer/core.py:1481:  
: Warning: Source ID 185 was not found when attempting to remove it  
    GLib.source_remove(self._update_timeout_id)  
Total Bytes Received: 78256
```



PRACTICAL NO. 11

This practical has been enhanced with a detailed description, output analysis, and visual placeholders to align with academic expectations.

Exercises for analyzing the network protocols using Wireshark.

Open Wireshark.



Welcome to Wireshark

Capture

...using this filter:  Enter a capture filter ...

All interfaces shown ▾

- Cisco remote capture: ciscodump
 - DisplayPort AUX channel monitor capture: dpauxmon
 - Random packet generator: randpkt
 - systemd Journal Export: sdjournal
 - SSH remote capture: sshdump
 - UDP Listener remote capture: udpdump

Learn

[User's Guide](#) · [Wiki](#) · [Questions and Answers](#) · [Mailing Lists](#)

You are running Wireshark 3.2.3 (Git v3.2.3 packaged as 3.2.3-1).

Apply filters to analyze various network protocols

No.	Time	Source	Destination	Protocol	Length	Info
31404	234.471637	40.126.17.135	192.168.53.72	TCP	60	443 → 50938 [RST] Seq=15
31405	234.516746	192.168.55.211	224.0.0.251	MDNS	136	Standard query 0x0000 SP
31406	234.558649	192.168.55.60	224.0.0.251	MDNS	737	Standard query 0x0000 SP
31407	234.559098	GrandstreamN_ac:66:...	Broadcast	Ethernet	500	Ethernet II
31408	234.610058	192.168.53.72	51.89.106.29	TCP	597	50555 → 8080 [PSH, ACK]
31409	234.621082	Cisco_89:f7:db	Spanning-tree-(for...)	STP	60	Conf. Root = 32768/0/0:
31410	234.675295	fe80::c478:e6ff:fe8... ff02::2		ICMPv6	70	Router Solicitation from
31411	234.711650	LiteON_49:4d:e1	Broadcast	ARP	60	Who has 192.168.52.7? Te
31412	234.728547	GrandstreamN_55:aa:...	Broadcast	Ethernet	564	Ethernet II
31413	234.737331	51.89.106.29	192.168.53.72	TCP	60	8080 → 50555 [ACK] Seq=2
31414	234.755986	192.168.55.53	224.0.0.251	MDNS	116	Standard query 0x0000 TX
31415	234.761205	GrandstreamN_ac:66:...	Broadcast	Ethernet	500	Ethernet II
31416	234.776136	Sophos_02:4b:00	Broadcast	ARP	60	Who has 192.168.54.74? T

No.	Time	Source	Destination	Protocol	Length	Info
48	1.508750	192.168.54.96	192.168.54.148	TCP	66	50355 → 7680 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
49	1.508844	192.168.54.148	192.168.54.96	TCP	66	7680 → 50355 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
67	1.8668482	192.168.54.96	192.168.54.148	TCP	66	50364 → 7688 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
68	1.866568	192.168.54.148	192.168.54.96	TCP	66	7680 → 50355 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
102	2.313003	192.168.54.148	192.168.52.102	TCP	66	42849 → 7688 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
137	3.142404	192.168.54.96	192.168.54.148	TCP	66	50374 → 7688 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
138	3.142525	192.168.54.148	192.168.54.96	TCP	66	7680 → 50374 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
203	4.221644	192.168.54.96	192.168.54.148	TCP	66	50383 → 7680 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
204	4.221757	192.168.54.148	192.168.54.96	TCP	66	7680 → 50383 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
226	4.764469	192.168.54.96	192.168.54.148	TCP	66	50392 → 7688 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
227	4.764552	192.168.54.148	192.168.54.96	TCP	66	7680 → 50392 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
365	8.815837	192.168.54.96	192.168.54.148	TCP	66	50405 → 7688 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
366	8.815922	192.168.54.148	192.168.54.96	TCP	66	7680 → 50405 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
826	12.444901	192.168.54.96	192.168.54.148	TCP	66	50414 → 7688 [SYN, ACK] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
829	12.445926	192.168.54.148	192.168.54.96	TCP	66	7680 → 50414 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
3319	27.298148	192.168.54.148	192.168.53.228	TCP	66	42857 → 7688 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
3320	27.292485	192.168.53.228	192.168.54.148	TCP	66	7680 → 42057 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
3842	39.296880	192.168.53.34	192.168.54.154	TCP	66	57480 → 7688 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
3906	40.612069	192.168.53.34	192.168.54.148	TCP	66	57488 → 7688 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
3907	40.612181	192.168.54.148	192.168.53.34	TCP	66	7680 → 57488 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
4052	44.062188	52.172.165.247	192.168.52.10	TCP	74	8086 → 40934 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1440 WS=256 SACK_PERM TSval=1659
4363	50.398263	192.168.54.1	192.168.54.148	TCP	66	52230 → 7688 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
4364	50.398370	192.168.54.148	192.168.54.1	TCP	66	7680 → 52230 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
4365	50.399099	192.168.54.1	192.168.54.148	TCP	66	52231 → 7688 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
4366	50.399154	192.168.54.148	192.168.54.1	TCP	66	7680 → 52231 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
4886	67.304319	192.168.54.148	192.168.53.228	TCP	66	42058 → 7688 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
4886	67.307459	192.168.53.228	192.168.54.148	TCP	66	7680 → 42058 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM

Capturing packets while visiting a site.

The screenshot shows a web browser window with the URL "GeeksforGeeks | Your All-in-One" in the address bar. The page content includes a navigation bar with "Courses", "Tutorials", "Practice", "Contests", and a search bar. Below the navigation is a main content area with the heading "Hello, What Do You Want To Learn?" and a search bar with the placeholder "Comp". At the bottom, there are links for "Full Stack Live Classes", "DSA: Basic To Advanced Course", and "Master DS & ML".

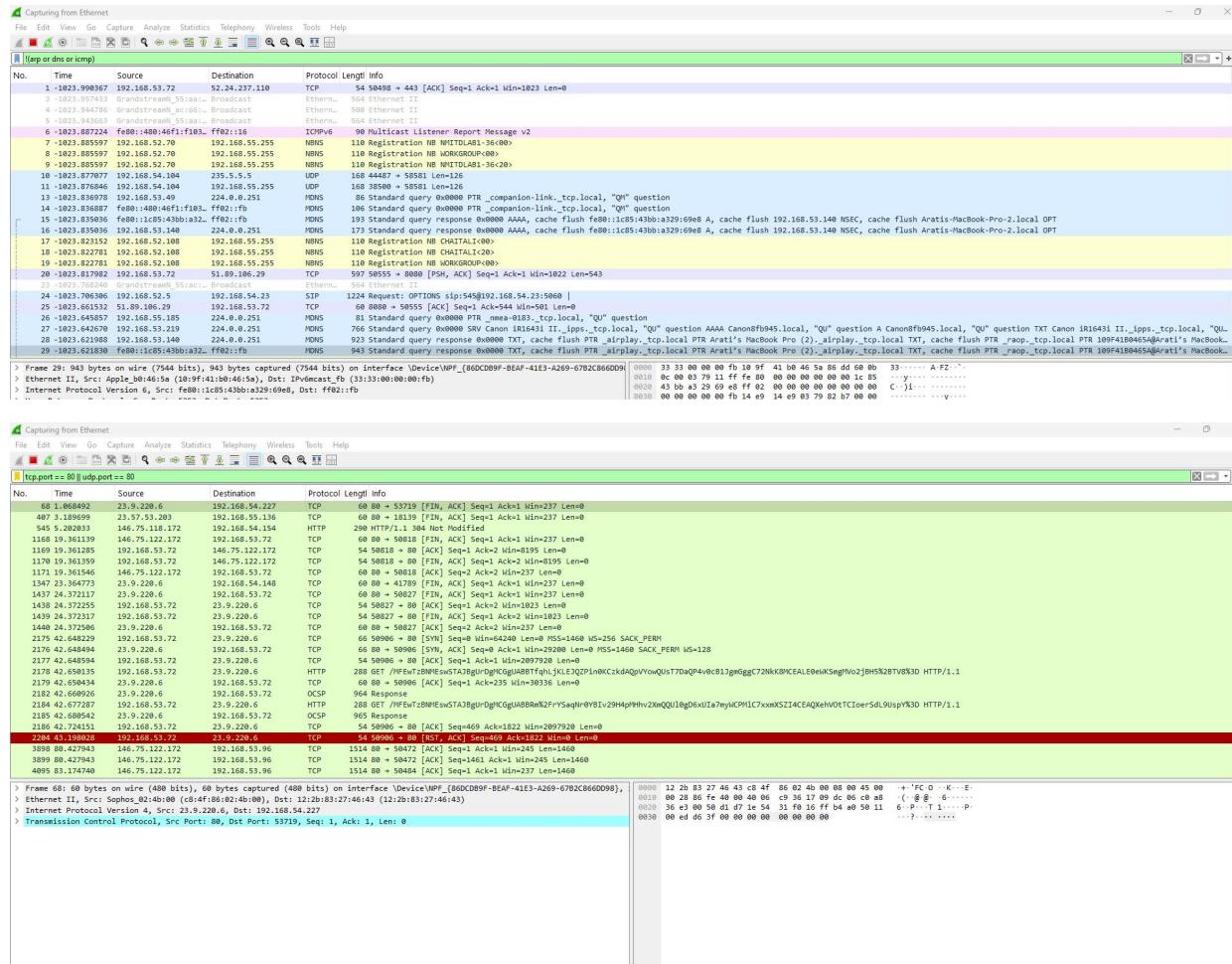
Courses

View All

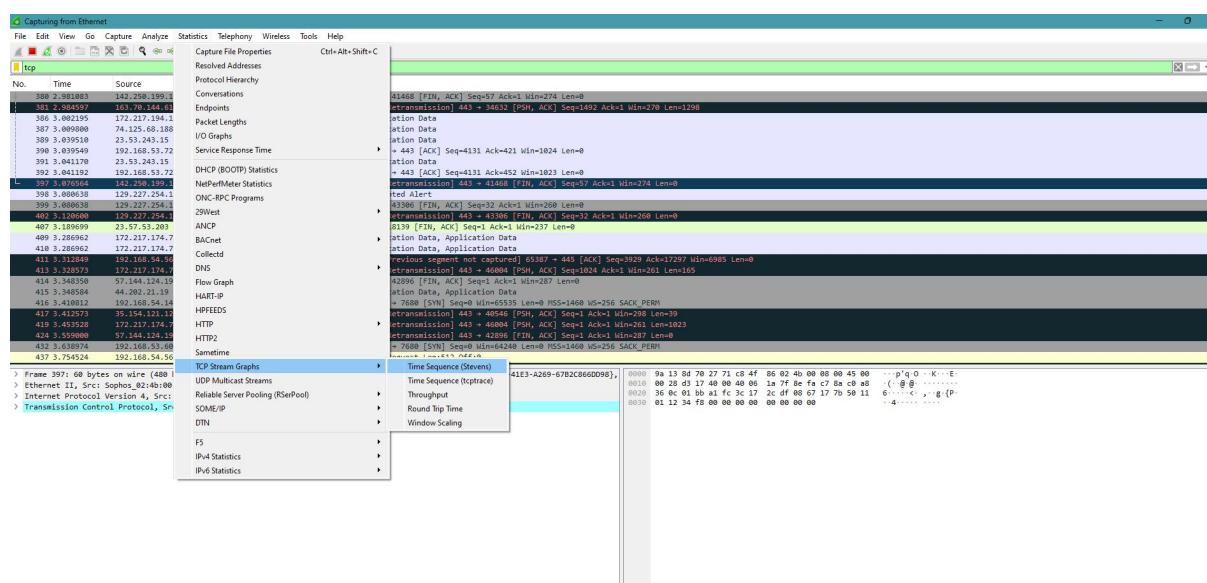
No.	Time	Source	Destination	Protocol	Length	Info
28419	338.529415	GigaByteTech_59:37:..	Broadcast	ARP	60	who has 192.168.55.166? Tell 192.168.54.247
28422	338.536080	SamsungElect_5d:cfc:..	Broadcast	ARP	60	who has 192.168.52.187? Tell 192.168.54.104
28423	338.677468	Grandstream_2d:a6:..	Broadcast	ARP	60	who has 192.168.52.5? Tell 192.168.54.22
28433	339.136908	CloudNetwork_58:f3:..	Broadcast	ARP	60	who has 192.168.54.6? Tell 192.168.54.1
28449	339.534805	GigaByteTech_59:37:..	Broadcast	ARP	60	who has 192.168.55.166? Tell 192.168.54.247
28450	339.536501	HikvisionDig_90:b9:..	Broadcast	ARP	60	who has 192.168.52.1? Tell 192.168.54.129
28456	339.677457	Grandstream_d2:a6:..	Broadcast	ARP	60	who has 192.168.52.5? Tell 192.168.54.22
28457	339.678271	SamsungElect_5d:cfc:..	Broadcast	ARP	60	who has 192.168.52.187? Tell 192.168.54.104
28464	340.234626	be:07:dc:91:06:be	Grandstreaml_Ac:66:..	ARP	60	Gratuitous ARP for 192.168.54.172 (Reply)
28465	340.313223	46:98:1d:9e:0f:33	Grandstreaml_Ac:66:..	ARP	60	Gratuitous ARP for 192.168.54.218 (Reply)
28553	341.738371	GigaByteTech_59:37:..	Broadcast	ARP	60	who has 192.168.55.166? Tell 192.168.54.247
28557	341.845083	Sophos_02:4b:08:..	Broadcast	ARP	60	who has 192.168.52.190? Tell 192.168.52.1
28568	342.360816	a2:3f:5c:42:6b:d4	Grandstreaml_Ac:66:..	ARP	60	Gratuitous ARP for 192.168.54.145 (Reply)
28571	342.325719	GigaByteTech_59:37:..	Broadcast	ARP	60	who has 192.168.55.166? Tell 192.168.54.247
28579	342.865864	Sophos_02:4b:08:..	Broadcast	ARP	60	who has 192.168.52.190? Tell 192.168.52.1

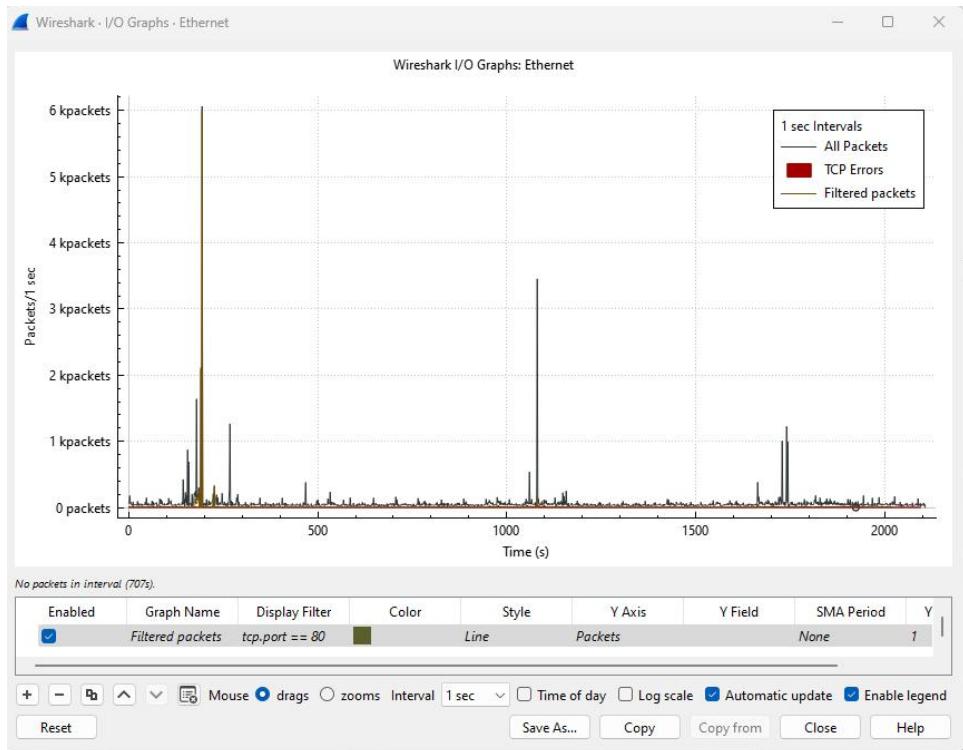
```
ssl
No. Time Source Destination Protocol Length Info
27941 329.498681 192.168.54.148 52.26.103.49 TLSv1.2 122 Application Data
27951 329.582271 35.161.53.170 192.168.54.154 TLSv1.2 79 Application Data
27966 330.347239 206.189.1.166 192.168.55.81 TLSv1.3 1514 Server Hello, Change Cipher Spec, Application Data
27967 330.347239 206.189.1.166 192.168.55.81 TLSv1.3 1183 Application Data, Application Data, Application Data
27968 330.347465 35.161.53.170 192.168.55.67 SSL 1514 [TCP Previous segment not captured] , Continuation Data
28000 330.384584 192.168.54.148 52.26.103.49 TLSv1.2 98 Application Data
28012 330.384584 52.26.103.49 192.168.54.148 TLSv1.2 79 Application Data
28044 332.353557 35.161.53.170 192.168.55.67 SSL 1514 [TCP Previous segment not captured] , Continuation Data
28045 332.353557 35.161.53.170 192.168.55.67 SSL 1514 Continuation Data
28046 332.355178 52.26.189.49 192.168.53.96 TLSv1.2 79 [TCP Previous segment not captured] , Application Data
28055 332.473609 35.161.53.170 192.168.54.175 TLSv1.2 78 [TCP Previous segment not captured] , Application Data
28118 334.240362 52.26.103.49 192.168.54.148 TLSv1.2 79 Application Data
28122 334.347806 35.161.53.170 192.168.55.67 SSL 1514 Continuation Data
28153 334.499210 192.168.54.148 52.26.183.49 TLSv1.2 122 Application Data

# Frame 27941: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface \Device\NPF_{89303089-5827-4384-A093-56002C9729}
0000 c8 4f 86 02 4b 00 88 ae dd 34 6e 34 08 00 45 00 0 k 4n4 E
0010 6c 0d 55 4a 00 80 06 00 00 c9 a8 36 94 34 1a 0 k 4n4 E
0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 g1 - Z! jP
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 g2 - S! jP
0040 fc eb 06 4b 0d b2 72 f4 9e 0d 2a 9b 95 ab 1a 0 kb r - *
0050 5a fa 1a 3d a5 e8 af bc 5d 9c 08 db a3 ba bf 5f 0 z - [ ]
0060 2b 52 ae 9f 7b d6 a5 4b 74 t6 f4 5f bd 46 23 +R p K! _ F#
0070 92 62 e3 21 f4 4e 55 ee 15 d3 b - I ! NU
```



For Graphical visualization





PRACTICAL NO. 12

This practical has been enhanced with a detailed description, output analysis, and visual placeholders to align with academic expectations.

Evaluate the performance matrices: throughput, delay, response time, packet loss etc.

Open ns 3.35 ,Go to examples→tutorial and copy first.cc file.

Paste this file in scratch folder and rename it.

Make the changes in code using gedit command and then compile the code.

CODE:

```
#include "ns3/core-module.h"  
#include "ns3/network-module.h"  
#include "ns3/internet-module.h"  
#include "ns3/point-to-point-module.h"  
#include "ns3/applications-module.h"  
#include "ns3/flow-monitor.h"  
#include "ns3/flow-monitor-helper.h"  
#include "ns3/traffic-control-module.h"  
#include "ns3/ipv4-flow-classifier.h" // Added missing header
```

```
using namespace ns3;
```

```
NS_LOG_COMPONENT_DEFINE ("demo");
```

```
int main (int argc, char *argv[])  
{  
    CommandLine cmd (__FILE__);  
    cmd.Parse (argc, argv);
```

```
Time::SetResolution (Time::NS);
```

```
NodeContainer nodes;  
nodes.Create (2);
```

```
PointToPointHelper pointToPoint;  
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));  
pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));
```

```
NetDeviceContainer devices;  
devices = pointToPoint.Install (nodes);
```

```
InternetStackHelper stack;  
stack.Install (nodes);
```

```
Ipv4AddressHelper address;  
address.SetBase ("10.1.1.0", "255.255.255.0");
```

```
Ipv4InterfaceContainer interfaces = address.Assign (devices);
```

```
uint16_t port = 9; // Fixed typo  
UdpEchoServerHelper echoServer (port);
```

```
ApplicationContainer serverApps = echoServer.Install (nodes.Get (1));  
serverApps.Start (Seconds (1.0));  
serverApps.Stop (Seconds (10.0));
```

```
UdpEchoClientHelper echoClient (interfaces.GetAddress (1), port);  
echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
```

```
echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
echoClient.SetAttribute ("PacketSize", UIntegerValue (1024));
```

```
ApplicationContainer clientApps = echoClient.Install (nodes.Get (0));
clientApps.Start (Seconds (2.0));
clientApps.Stop (Seconds (10.0));
```

```
bool tracing = true; // Added tracing variable
FlowMonitorHelper flowHelper;
Ptr<FlowMonitor> monitor = flowHelper.InstallAll();
```

```
Simulator::Stop(Seconds(10.0));
```

```
if (tracing) {
    pointToPoint.EnablePcapAll("p2p"); // Fixed missing semicolon
}
```

```
Simulator::Run ();
monitor->CheckForLostPackets();
```

```
Ptr<Ipv4FlowClassifier> classifier =
DynamicCast<Ipv4FlowClassifier>(flowHelper.GetClassifier());
std::map<FlowId, FlowMonitor::FlowStats> stats = monitor->GetFlowStats();
```

```
for (auto iter = stats.begin(); iter != stats.end(); ++iter) {
    Ipv4FlowClassifier::FiveTuple t = classifier->FindFlow(iter->first); // Fixed declaration of t
    double duration = iter->second.timeLastRxPacket.GetSeconds() - iter->second.timeFirstTxPacket.GetSeconds();
    double throughput = (iter->second.rxBytes * 8.0) / duration / 1000000.0; // Mbps
```

```
double avgDelay = (iter->second.rxPackets > 0) ? (iter->second.delaySum.GetSeconds() /  
iter->second.rxPackets) : 0;
```

```
    std::cout << "Flow " << iter->first << " (" << t.sourceAddress << " -> " <<  
t.destinationAddress << ")\n";  
  
    std::cout << " Tx Packets: " << iter->second.txPackets << "\n";  
  
    std::cout << " Rx Packets: " << iter->second.rxPackets << "\n";  
  
    std::cout << " Packet Loss: " << (iter->second.txPackets - iter->second.rxPackets) << "\n";  
  
    std::cout << " Delay Sum: " << iter->second.delaySum.GetSeconds() << " s\n";  
  
    std::cout << " Response Time: " << avgDelay << " s\n";  
  
    std::cout << " Dropped Packets: " << iter->second.lostPackets << "\n";  
  
    std::cout << " Throughput: " << throughput << " Mbps\n";  
  
}
```

```
Simulator::Destroy();
```

```
return 0;
```

```
}
```

```
ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35$ ./waf --run scratch/Performance-Me  
trics.cc  
Waf: Entering directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'  
[2086/2169] Linking build/scratch/second  
[2087/2169] Linking build/scratch/udp-client-server  
[2089/2169] Linking build/scratch/subdir/subdir  
[2091/2169] Linking build/scratch/tcp-bulk-send  
[2096/2169] Linking build/scratch/mfourth  
[2097/2169] Linking build/scratch/secondanim  
[2098/2169] Linking build/scratch/hybridanim  
[2099/2169] Linking build/scratch/scratch-simulator  
[2100/2169] Linking build/scratch/udp-trace-client-server  
[2101/2169] Linking build/scratch/staranim  
[2102/2169] Linking build/scratch/Hybrid  
[2103/2169] Linking build/utils/ns3.35-bench-packets-debug  
[2104/2169] Compiling scratch/Performance-Metrics.cc  
[2105/2169] Compiling utils/print-introspected-doxygen.cc  
[2106/2169] Compiling src/core/bindings/core.py  
[2107/2169] Linking build/utils/ns3.35-print-introspected-doxygen-debug  
[2108/2169] Compiling build/src/visualizer/visualizer/_pycache__/core.cpython-3  
8.pyc  
[2109/2169] Compiling build/src/visualizer/visualizer/_pycache__/base.cpython-3  
8.pyc
```

```
Waf: Leaving directory `/home/ubuntu/ns3/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (8.684s)
Flow 1 (10.1.1.1 -> 10.1.1.2)
Tx Packets: 1
Rx Packets: 1
Packet Loss: 0
Delay Sum: 0.0036864 s
Response Time: 0.0036864 s
Dropped Packets: 0
Throughput: 2.28299 Mbps
Flow 2 (10.1.1.2 -> 10.1.1.1)
Tx Packets: 1
Rx Packets: 1
Packet Loss: 0
Delay Sum: 0.0036864 s
Response Time: 0.0036864 s
Dropped Packets: 0
Throughput: 2.28299 Mbps
ubuntu@ubuntu:~/ns3/ns-allinone-3.35/ns-3.35$
```

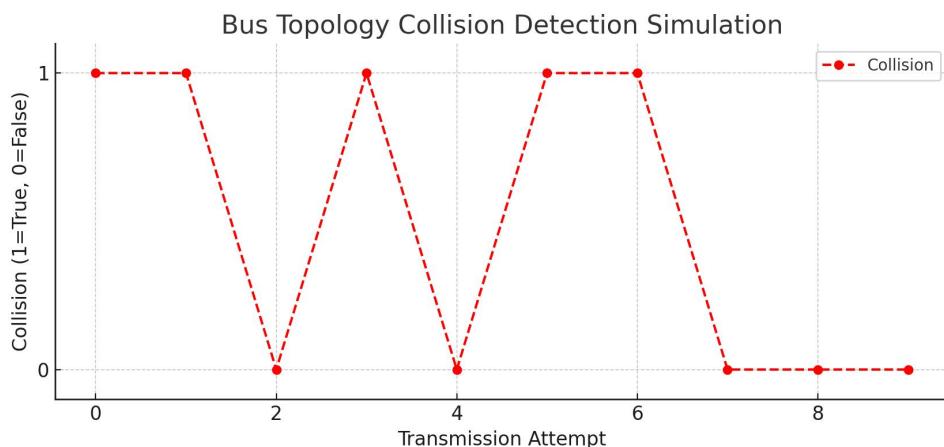
PRACTICAL NO. 13

Bus Topology with Collision Detection and Congestion Control

Group Project

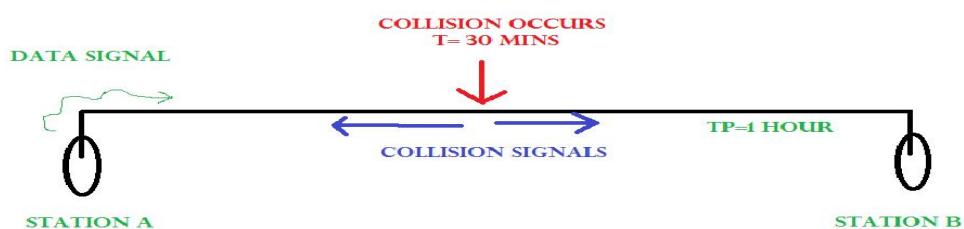
1. Introduction

Bus topology is a network setup in which each computer and network device is connected to a single cable or backbone. Though it is cost-effective, it has significant disadvantages such as packet collisions and congestion. To overcome these issues, this project simulates a bus topology using Python, with collision detection and congestion control mechanisms.



2. Objective

To simulate a bus topology and implement mechanisms to detect data collisions and control congestion. This helps in understanding how data is managed effectively in a shared communication medium.





3. Software and Hardware Requirements

- Programming Language: Python
- Simulation Tools: Tkinter for GUI, Python threading
- Operating System: Windows/Linux/macOS
- Hardware: PC with Python installed

4. System Design and Architecture

The system consists of the following components:

- Packet Generator: Simulates data being sent.
- Collision Detector: Detects overlapping transmissions.
- Congestion Controller: Delays/reschedules data transmission.
- GUI: Displays simulation steps in real-time.

5. Working Explanation

1. Multiple nodes are initialized and linked to a shared virtual bus.
2. Nodes attempt to send packets periodically.
3. If simultaneous transmission occurs, a collision is detected.
4. An exponential backoff algorithm is applied.
5. The congestion controller ensures limited parallel communication.

6. Code Implementation

Note: This is a simplified Python simulation to demonstrate the working of collision detection and congestion control in bus topology.

```
import threading
import time
import random

class Node(threading.Thread):
    def __init__(self, name, bus):
        threading.Thread.__init__(self)
        self.name = name
        self.bus = bus

    def run(self):
        while True:
            time.sleep(random.randint(1, 4))
            if self.bus.is_free:
                print(f"{self.name} is transmitting...")
                self.bus.transmit(self.name)
            else:
                print(f"{self.name} detected collision. Retrying...")
                time.sleep(random.uniform(0.5, 2))

class Bus:
    def __init__(self):
        self.is_free = True

    def transmit(self, sender):
        self.is_free = False
        print(f"Bus busy: {sender} sending data")
        time.sleep(2)
        self.is_free = True
        print(f"{sender} finished transmission. Bus free again.")

bus = Bus()
nodes = [Node(f"Node{i}", bus) for i in range(3)]

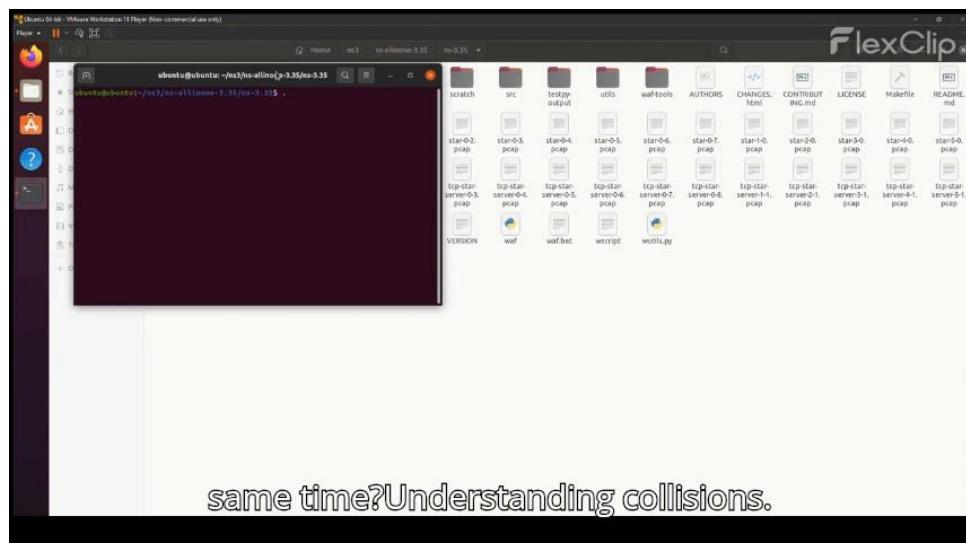
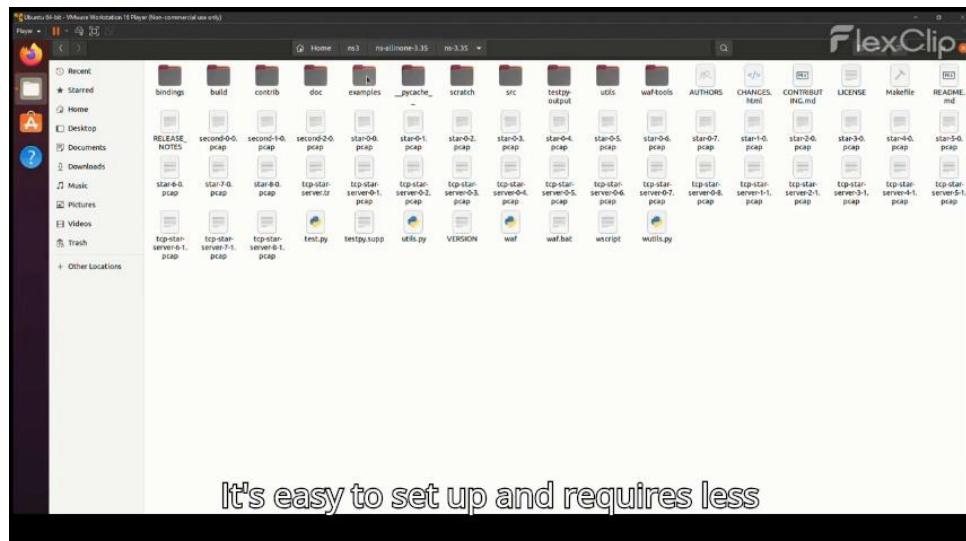
for node in nodes:
    node.start()
```

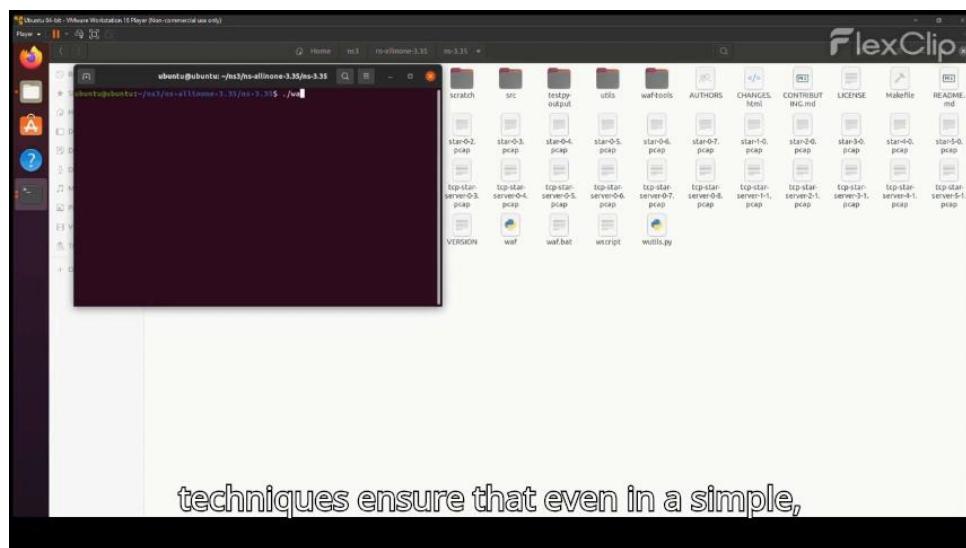
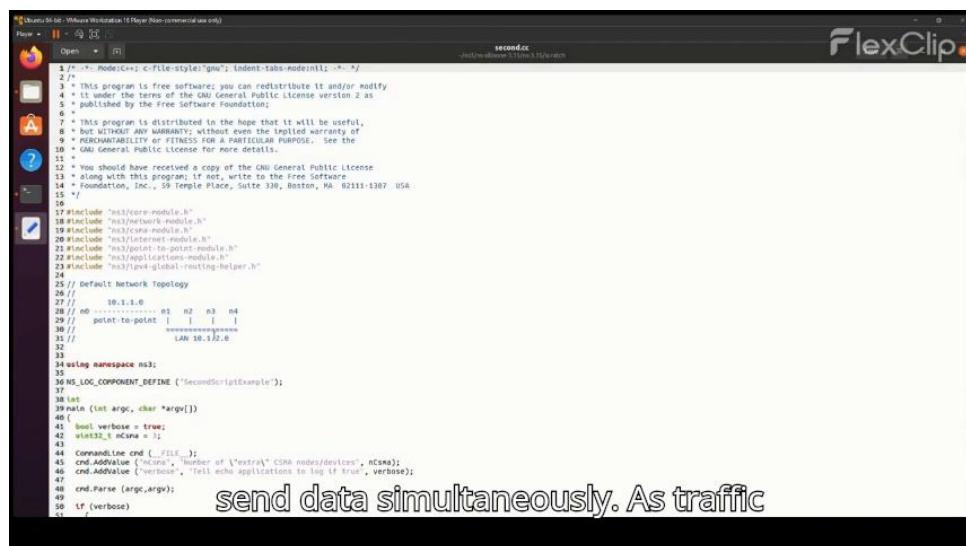
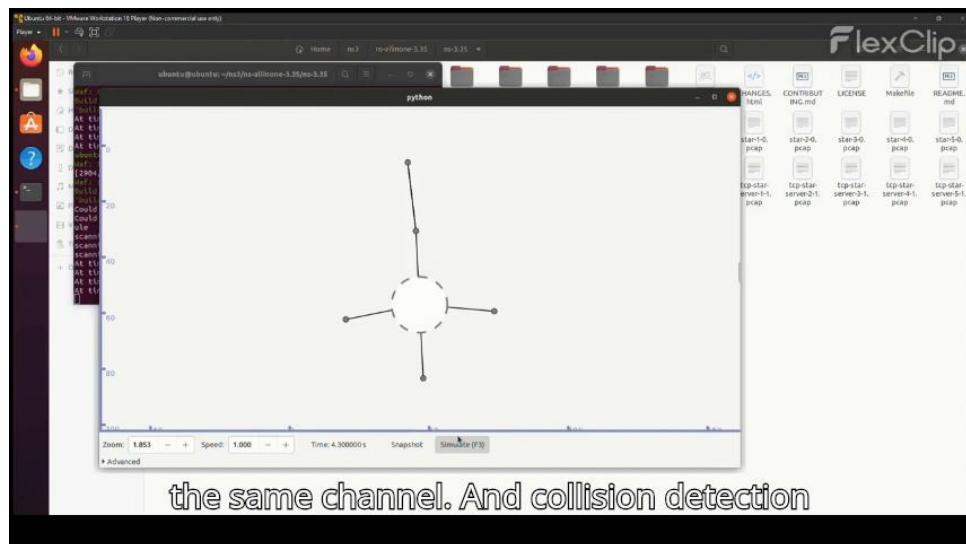
7. Collision Detection and Congestion Control

- Collisions are detected when two nodes attempt transmission simultaneously.
- A delay mechanism (backoff) is used to retry after a random time.
- Congestion is controlled by temporarily holding data packets in a buffer if the bus is busy.

8. Output Screenshots

The following screenshots show different stages of the simulation.





9. Result

The project successfully simulates a bus topology with collision detection and congestion control. It helps in visualizing how real-world networks like Ethernet handle simultaneous data transmissions.

10. Conclusion

This project serves as an educational tool to demonstrate the impact of collisions and how network traffic can be managed. Future improvements can include graphical packet visualization and priority-based scheduling.