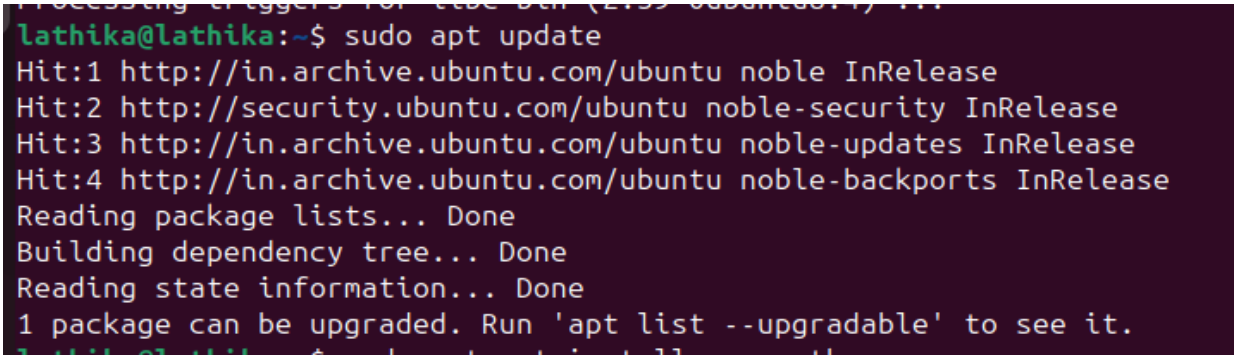
Practical No - 1

Aim: Installation: ns3 in Linux, NetAnim, Wireshark, PyViz, tcpdump

Steps:

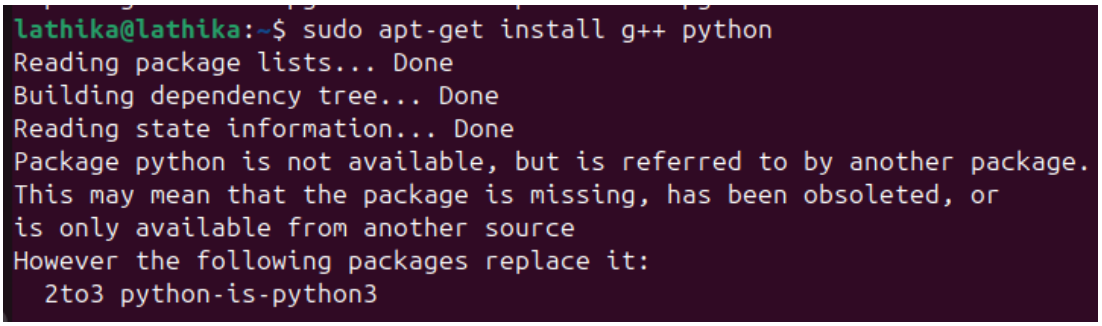
- **sudo apt upgrade**

- **sudo apt update**



PyViz

- **sudo apt-get install g++ python**



- **sudo apt-get install g++ python3 python3-dev pkg-config sqlite3**

```
lathika@lathika:~$ sudo apt-get install g++ python3  python3-dev pkg-config sqlite3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.12.3-0ubuntu2).
python3 set to manually installed.
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu g++-13 g++-13-x86-64-linux-gnu g++-x86-64-linux-gnu gcc gcc-13
  gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu javascript-common libasan8 libbinutils libcc1-0 libctf-nobfd0 libctf0
  libexpat1-dev libgcc-13-dev libgprofng0 libhwasan0 libitm1 libjs-jquery libjs-sphinxdoc libjs-underscore liblsan0
  libpkgconf3 libpython3-dev libpython3.12-dev libquadmath0 libsframe1 libstdc++-13-dev libtsan2 libubsan1 pkgconf
  pkgconf-bin python3.12-dev zlib1g-dev
```

- **sudo apt install -y qtcreator qtbase5-dev qt5-qmake cmake**

```
lathika@lathika:~$ sudo apt install -y qtcreator qtbase5-dev qt5-qmake cmake
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  assistant-qt6 clang-18 clang-tidy clang-tidy-18 clang-tools-18 clangd clangd-18 cmake-data designer-qt6 icu-devtools
  lib32gcc-s1 lib32stdc++6 libb2-1 libc6-i386 libclang-common-18-dev libclang-cpp15t64 libclang-rt-18-dev
  libclang1-15t64 libdouble-conversion3 libegl-dev libffi-dev libgc1 libgl-dev libglu1-mesa-dev libglx-dev
  libgrpc++1.51t64 libgrpc29t64 libgumbo2 libicu-dev libjsoncpp25 liblitehtml0t64 libllvm15t64 libmd4c0 libmng2
  libncurses-dev libobjc-13-dev libobjc4 libopengl-dev libpcre2-16-0 libpfm4 libprotoc32t64 libpthread-stubs0-dev
  libqt5concurrent5t64 libqt5core5t64 libqt5dbus5t64 libqt5gui5t64 libqt5network5t64 libqt5opengl5-dev
  libqt5opengl5t64 libqt5printsupport5t64 libqt5qml5 libqt5qmlmodels5 libqt5quick5 libqt5sql5-sqlite libqt5sql5t64
  libqt5svg5 libqt5test5t64 libqt5waylandclient5 libqt5waylandcompositor5 libqt5widgets5t64 libqt5xml5t64
  libqt6concurrent6t64 libqt6core5compat6 libqt6core6t64 libqt6dbus6t64 libqt6designer6 libqt6designercomponents6
  libqt6gui6t64 libqt6help6 libqt6jsonrpc6 libqt6languageserver6 libqt6network6t64 libqt6opengl6t64
  libqt6openglwidgets6t64 libqt6printsupport6t64 libqt6qml6 libqt6qmlcompiler6 libqt6qmlmodels6 libqt6qmlworkerscript6
  libqt6quick6 libqt6quickcontrols2-6 libqt6quickcontrols2impl6 libqt6quicklayouts6 libqt6quicktemplates2-6
  libqt6quicktest6 libqt6quickwidgets6 libqt6serialport6 libqt6sql6-sqlite libqt6sql6t64 libqt6svg6 libqt6svgwidgets6
  libqt6test6t64 libqt6uitools6 libqt6waylandclient6 libqt6waylandcompositor6 libqt6waylandeglclienthwintegration6
  libqt6waylandeglcompositorhwintegration6 libqt6widgets6t64 libqt6wlshellintegration6 libqt6xml6t64 libre2-10
  librhash0 libts0t64 libvulkan-dev libx11-dev libxau-dev libxcb-xinerama0 libxcb-xinput0 libxcb1-dev libxdmcp-dev
  libxext-dev libxml2-dev libyaml-cpp0.8 libz3-4 libz3-dev linguist-qt6 llvm-18 llvm-18-dev llvm-18-linker-tools
```

- **sudo apt-get install gir1.2-goocanvas-2.0 python3-pygraphviz python3-gi-cairo python3-pygraphviz gir1.2-gtk-3.0**

```
lathika@lathika:~$ sudo apt-get install gir1.2-goocanvas-2.0 python3-pygraphviz python3-gi-cairo python3-pygraphviz gir1
.2-gtk-3.0
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gir1.2-gtk-3.0 is already the newest version (3.24.41-4ubuntu1.3).
gir1.2-gtk-3.0 set to manually installed.
The following additional packages will be installed:
  fonts-liberation2 graphviz libann0 libcdt5 libcgraph6 libgoocanvas-2.0-9 libgoocanvas-2.0-common libgts-0.7-5t64
  libgts-bin libgvc6 libgvpr2 liblab-gamut1 libpathplan4
Suggested packages:
  gsfonts graphviz-doc python-pygraphviz-doc
The following NEW packages will be installed:
  fonts-liberation2 gir1.2-goocanvas-2.0 graphviz libann0 libcdt5 libcgraph6 libgoocanvas-2.0-9
  libgoocanvas-2.0-common libgts-0.7-5t64 libgts-bin libgvc6 libgvpr2 liblab-gamut1 libpathplan4 python3-gi-cairo
  python3-pygraphviz
0 upgraded, 16 newly installed, 0 to remove and 1 not upgraded.
Need to get 4,080 kB of archives.
After this operation, 11.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu noble/universe amd64 fonts-liberation2 all 1:2.1.5-3 [3,418 B]
Get:2 http://in.archive.ubuntu.com/ubuntu noble/universe amd64 libgoocanvas-2.0-common all 2.0.4-1build3 [120 kB]
```

- **sudo apt-get install gdbvalgrind**

```
lathika@lathika:~$ sudo apt-get install gdb valgrind
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gdb is already the newest version (15.0.50.20240403-0ubuntu1).
gdb set to manually installed.
Suggested packages:
  valgrind-dbg valgrind-mpi kcachegrind alleyoop valkyrie
The following NEW packages will be installed:
  valgrind
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 14.9 MB of archives.
After this operation, 78.8 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu noble/main amd64 valgrind amd64 1:3.22.
Fetched 14.9 MB in 9s (1,754 kB/s)
Selecting previously unselected package valgrind.
(Reading database ... 167864 files and directories currently installed.)
Preparing to unpack .../valgrind_1%3a3.22.0-0ubuntu3_amd64.deb ...
Unpacking valgrind (1:3.22.0-0ubuntu3) ...
Setting up valgrind (1:3.22.0-0ubuntu3) ...
Processing triggers for man-db (2.12.0-4build2) ...
```

- **sudo apt-get install doxygengraphvizimagemagick**

```
lathika@lathika:~$ sudo apt-get install  doxygen graphviz imagemagick
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
graphviz is already the newest version (2.42.2-9ubuntu0.1).
graphviz set to manually installed.
The following additional packages will be installed:
  imagemagick-6-common imagemagick-6.q16 libfftw3-double3 libfmt9 libimath-3-1-29t64 libjxr-tools libjxr0t64
  liblqr-1-0 libmagickcore-6.q16-7-extra libmagickcore-6.q16-7t64 libmagickwand-6.q16-7t64 libnetpbm11t64
  libopenexr-3-1-30 libraw23t64 libxapian30 netpbm
Suggested packages:
  doxygen-latex doxygen-doc doxygen-gui imagemagick-6-doc autotrace curl enscript ffmpeg gimp gnuplot grads hp2xx
  html2ps libwmf-bin mplayer povray radiance texlive-base-bin transfig libraw-bin libfftw3-bin libfftw3-dev inkscape
  xapian-tools
The following NEW packages will be installed:
  doxygen imagemagick imagemagick-6-common imagemagick-6.q16 libfftw3-double3 libfmt9 libimath-3-1-29t64 libjxr-tools
  libjxr0t64 liblqr-1-0 libmagickcore-6.q16-7-extra libmagickcore-6.q16-7t64 libmagickwand-6.q16-7t64 libnetpbm11t64
  libopenexr-3-1-30 libraw23t64 libxapian30 netpbm
0 upgraded, 18 newly installed, 0 to remove and 1 not upgraded.
```

- **sudo apt install python3-pip**

```
lathika@lathika:~$ sudo apt install python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  build-essential bzip2 dpkg-dev fakeroot libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl
  libdpkg-perl libfakeroot libfile-fcntllock-perl lto-disabled-list python3-setuptools python3-wheel
Suggested packages:
  bzip2-doc debian-keyring git bzr python-setuptools-doc
The following NEW packages will be installed:
  build-essential bzip2 dpkg-dev fakeroot libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl
  libdpkg-perl libfakeroot libfile-fcntllock-perl lto-disabled-list python3-pip python3-setuptools python3-wheel
0 upgraded, 14 newly installed, 0 to remove and 1 not upgraded.
Need to get 3,355 kB of archives.
After this operation, 13.8 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 libdpkg-perl all 1.22.6ubuntu6.1 [269 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 bzip2 amd64 1.0.8-5.1build0.1 [34.5 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu noble/main amd64 lto-disabled-list all 47 [12.4 kB]
```

-

- **pip install ipython**

```
(myenv) lathika@lathika:~$ pip install ipython
Collecting ipython
  Downloading ipython-9.1.0-py3-none-any.whl.metadata (4.4 kB)
Collecting decorator (from ipython)
  Downloading decorator-5.2.1-py3-none-any.whl.metadata (3.9 kB)
Collecting ipython-pygments-lexers (from ipython)
  Downloading ipython_pygments_lexers-1.1.1-py3-none-any.whl.metadata (1.1 kB)
Collecting jedi>=0.16 (from ipython)
  Downloading jedi-0.19.2-py2.py3-none-any.whl.metadata (22 kB)
Collecting matplotlib-inline (from ipython)
  Downloading matplotlib_inline-0.1.7-py3-none-any.whl.metadata (3.9 kB)
Collecting pexpect>4.3 (from ipython)
  Downloading pexpect-4.9.0-py2.py3-none-any.whl.metadata (2.5 kB)
Collecting prompt_toolkit<3.1.0,>=3.0.41 (from ipython)
  Downloading prompt_toolkit-3.0.51-py3-none-any.whl.metadata (6.4 kB)
Collecting pygments>=2.4.0 (from ipython)
  Downloading pygments-2.19.1-py3-none-any.whl.metadata (2.5 kB)
```

- **sudo apt-get install python3-sphinx dia**

```
lathika@lathika:~$ sudo apt-get install python3-sphinx dia
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  dia-common dia-shapes docutils-common libcommon-sense-perl libemf1 libjson-perl libjson-xs-perl
  libtypes-serialiser-perl python3-alabaster python3-docutils python3-imagesize python3-packaging python3-roman
  python3-snowballstemmer sphinx-common
Suggested packages:
  docutils-doc fonts-linuxlibertine | ttf-linux-libertine texlive-lang-french texlive-latex-base
  texlive-latex-recommended python3-stemmer dvipng fonts-freefont-otf latexmk libjs-mathjax python3-lib2to3
  python3-sphinx-rtd-theme sphinx-doc tex-gyre texlive-fonts-recommended texlive-latex-extra texlive-plain-generic
The following NEW packages will be installed:
  dia dia-common dia-shapes docutils-common libcommon-sense-perl libemf1 libjson-perl libjson-xs-perl
  libtypes-serialiser-perl python3-alabaster python3-docutils python3-imagesize python3-packaging python3-roman
  python3-snowballstemmer python3-sphinx sphinx-common
0 upgraded, 17 newly installed, 0 to remove and 1 not upgraded.
```

tcpdump

- **sudo apt-get install tcpdump**

```
lathika@lathika:~$ sudo apt-get install tcpdump
[sudo] password for lathika:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
tcpdump is already the newest version (4.99.4-3ubuntu4).
tcpdump set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
```

- **sudo apt-get install-y llvm-11 llvm-11-dev clang-11 llvm-11-tools**

```
lathika@lathika:~$ sudo apt-get install -y llvm-11 llvm-11-dev clang-11 llvm-11-tools
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package clang-11 is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

E: Unable to locate package llvm-11
E: Unable to locate package llvm-11-dev
E: Package 'clang-11' has no installation candidate
E: Unable to locate package llvm-11-tools
```

- **pip install cxxfilt**

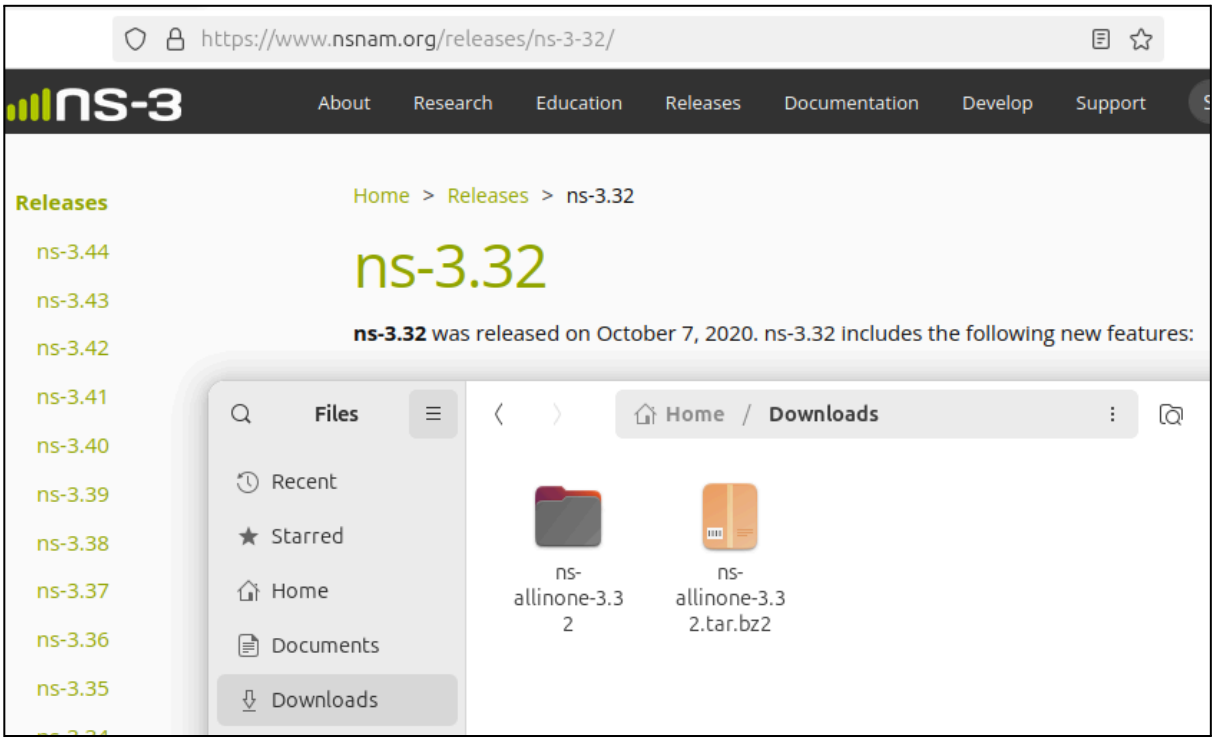
```
(myenv) lathika@lathika:~$ pip install cxxfilt
Collecting cxxfilt
  Downloading cxxfilt-0.3.0-py2.py3-none-any.whl.metadata (3.5 kB)
Downloading cxxfilt-0.3.0-py2.py3-none-any.whl (4.6 kB)
Installing collected packages: cxxfilt
Successfully installed cxxfilt-0.3.0
```

ns3 in Linux

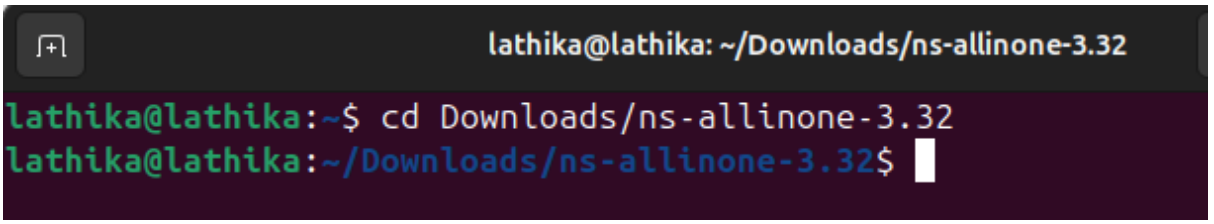
- **sudo apt install ns3**

```
lathika@lathika:~$ sudo apt install opensh-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package opensh-server
E: Unable to locate package -y
lathika@lathika:~$ sudo apt install ns3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libgsl27 libgslcblas0 libns3.41t64
Suggested packages:
  gsl-ref-psdoc | gsl-doc-pdf | gsl-doc-info | gsl-ref-html
The following NEW packages will be installed:
  libgsl27 libgslcblas0 libns3.41t64 ns3
0 upgraded, 4 newly installed, 0 to remove and 1 not upgraded.
```

- **Download NS3 from https://www.nsnam.org and extract the folder.**



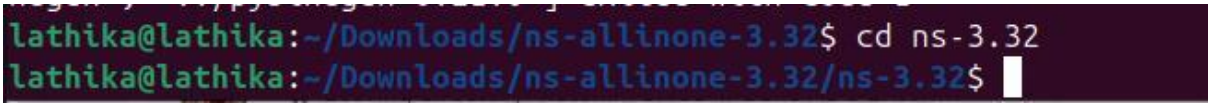
- **cd Downloads/ ns-allinone-3.32**



- **sudo ./build.py**

```
lathika@lathika:~/Downloads/ns-allinone-3.32$ sudo ./build.py
[sudo] password for lathika:
# Build NetAnim
Entering directory `netanim-3.108'
 =>  qmake -v
QMake version 3.1
Using Qt version 5.15.13 in /usr/lib/x86_64-linux-gnu
qmake found
 =>  qmake NetAnim.pro
 =>  make
g++ -c -pipe -O2 -Wall -Wextra -D_REENTRANT -fPIC -DNS3_LOG_ENABLE -DQT_NO_DEBUG
 -DQT_PRINTSUPPORT_LIB -DQT_WIDGETS_LIB -DQT_GUI_LIB -DQT_CORE_LIB -I. -Iqtprope
rtybrowser/src -I/usr/include/x86_64-linux-gnu/qt5 -I/usr/include/x86_64-linux-g
nu/qt5/QtPrintSupport -I/usr/include/x86_64-linux-gnu/qt5/QtWidgets -I/usr/inclu
de/x86_64-linux-gnu/qt5/QtGui -I/usr/include/x86_64-linux-gnu/qt5/QtCore -I. -I/
usr/lib/x86_64-linux-gnu/qt5/mkspecs/linux-g++ -o routingstatsscene.o routingsta
tsscene.cpp
g++ -c -pipe -O2 -Wall -Wextra -D_REENTRANT -fPIC -DNS3_LOG_ENABLE -DQT_NO_DEBUG
 -DQT_PRINTSUPPORT_LIB -DQT_WIDGETS_LIB -DQT_GUI_LIB -DQT_CORE_LIB -I. -Iqtprope
rtybrowser/src -I/usr/include/x86_64-linux-gnu/qt5 -I/usr/include/x86_64-linux-g
nu/qt5/QtPrintSupport -I/usr/include/x86_64-linux-gnu/qt5/QtWidgets -I/usr/inclu
de/x86_64-linux-gnu/qt5/QtGui -I/usr/include/x86_64-linux-gnu/qt5/QtCore -I. -I/
usr/lib/x86_64-linux-gnu/qt5/mkspecs/linux-g++ -o interfacestatsscene.o interfac
```

- **cd ns-3.36.1**



- **./text.py**

Installation of Wireshark

1. **sudo apt upgrade**

```
lathika@lathika:~$ sudo apt upgrade
[sudo] password for lathika:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following upgrades have been deferred due to phasing:
  ubuntu-drivers-common
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
```

2. **Sudo apt update**

```
lathika@lathika:~$ sudo apt update
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu noble InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu noble InRelease
Get:4 http://in.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:5 http://in.archive.ubuntu.com/ubuntu noble-backports InRelease
Fetched 126 kB in 3s (45.2 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
lathika@lathika:~$ cd Downloads/ns-allinone-3.32
```

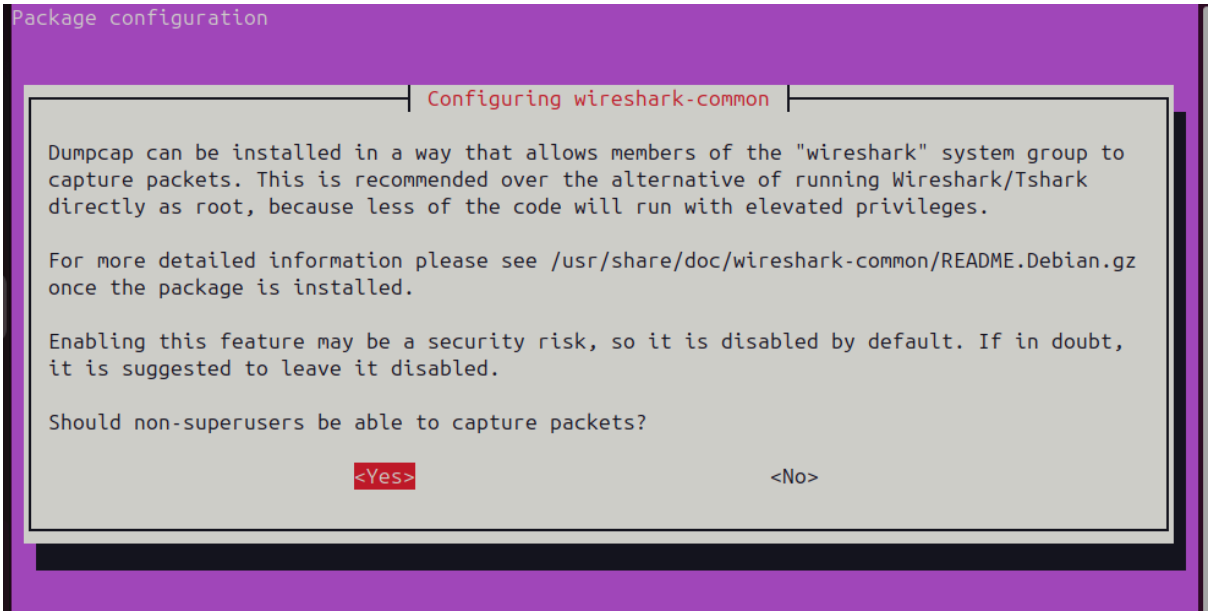
3. **Sudo add-apt-repository ppa:wireshark-dev/stable**

```
lathika@lathika:~$ sudo add-apt-repository ppa:wireshark-dev/stable
PPA publishes dbgsym, you may need to include 'main/debug' component
Repository: 'Types: deb
URIs: https://ppa.launchpadcontent.net/wireshark-dev/stable/ubuntu/
Suites: noble
Components: main
'
Description:
Latest stable Wireshark releases back-ported from Debian package versions.

Back-porting script is available at https://github.com/rbalint/pkg-wireshark-ubuntu-ppa

From Ubuntu 16.04 you also need to enable "universe"  repository, see:
http://askubuntu.com/questions/148638/how-do-i-enable-the-universe-repository
```

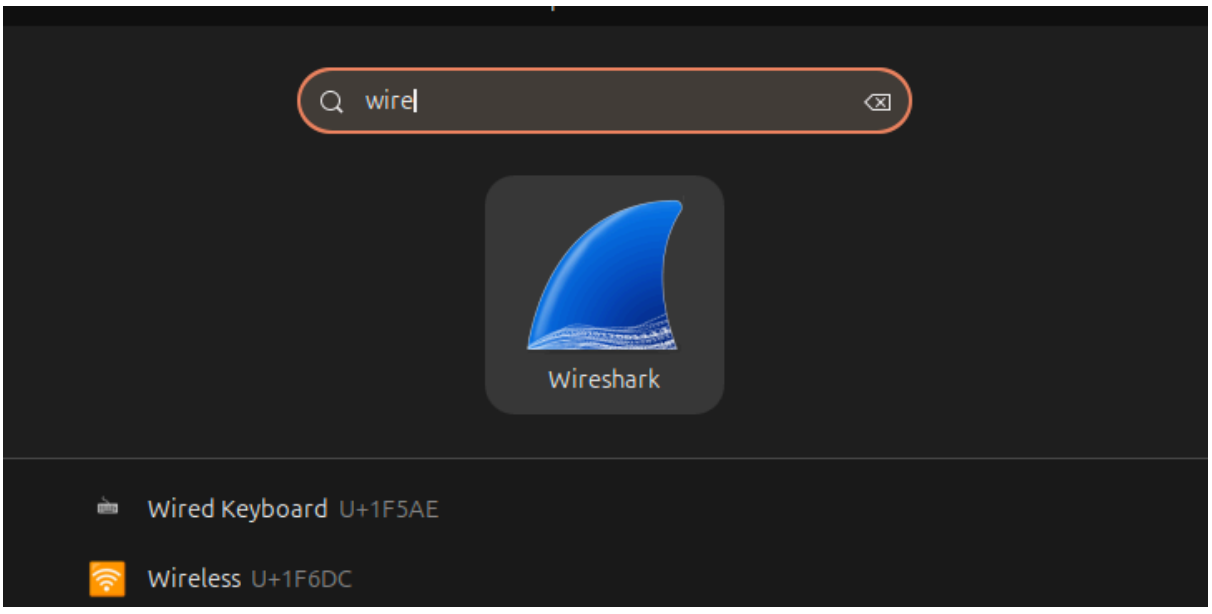
4. sudoaptinstallwireshark

```
lathika@lathika:~$ sudo apt install wireshark
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  i965-va-driver intel-media-va-driver libaacs0 libavcodec60 libavformat60 libavutil58
  libbcg729-0 libbdplus0 libbluray2 libchromaprint1 libcjson1 libcodec2-1.2 libdav1d7 libgme0
  libgsm1 libhwy1t64 libigdgmm12 libjxl0.7 libmbedcrypto7t64 libminizip1t64 libnghttp3-3
  libnorm1t64 libopencore-amrnb0 libopenmpt0t64 libpgm-5.3-0t64 libqt6multimedia6 librabbitmq4
  librav1e0 librist4 libshine3 libsmi2t64 libsnappy1v5 libsodium23 libsoxr0 libspandsp2t64
  libsrt1.5-gnutls libssh-gcrypt-4 libsvtav1enc1d1 libswresample4 libswscale7 libudfread0
  libva-drm2 libva-x11-2 libva2 libvdpau1 libvpl2 libwireshark-data libwireshark18 libwiretap15
  libwsutil16 libx264-164 libx265-199 libxvidcore4 libzmq5 libzvbi-common libzvbi0t64
  mesa-va-drivers mesa-vdpau-drivers ocl-icd-libopencl1 va-driver-all vdpau-driver-all
  wireshark-common
```

5. ConfiguringWireshark>Clickyes



Installs Wireshark



NetAnim Installation

1. **cd**

```
sims@sims-virtual-machine:~/Downloads/ns-allinone-3.32/netanim-3.108$ cd ..
sims@sims-virtual-machine:~/Downloads/ns-allinone-3.32$ cd netanim-3.108
```

2. **make clean**

```
sims@sims-virtual-machine:~/Downloads/ns-allinone-3.32/netanim-3.108$ make clea
n
rm -f qrc_resources.cpp qrc_qtpropertybrowser.cpp
rm -f moc_predefs.h
rm -f moc_animatorscene.cpp moc_animpacket.cpp moc_netanim.cpp moc_animatormode
.cpp moc_statsmode.cpp moc_qtvariantproperty.cpp moc_qttreepropertybrowser.cpp
moc_qtpropertymanager.cpp moc_qtpropertybrowserutils_p.cpp moc_qtpropertybrowse
r.cpp moc_qtgroupboxpropertybrowser.cpp moc_qteditorfactory.cpp moc_qtbuttonpro
pertybrowser.cpp moc_animpropertybrowser.cpp moc_filepathmanager.cpp moc_fileed
itfactory.cpp moc_fileedit.cpp moc_packetsmode.cpp moc_table.cpp moc_qcustomplo
t.cpp
rm -f qttreepropertybrowser.moc qtpropertymanager.moc qteditorfactory.moc
rm -f main.o log.o fatal-error.o fatal-impl.o logqt.o resizeableitem.o animnode
.o animatorscene.o animpacket.o netanim.o animatormode.o mode.o animxmlparser.o
 animatorview.o animlink.o animresource.o statsview.o statsmode.o routingxmlpar
ser.o routingstatsscene.o interfacestatsscene.o flowmonxmlparser.o flowmonstats
scene.o textbubble.o qtvariantproperty.o qttreepropertybrowser.o qtpropertymana
```

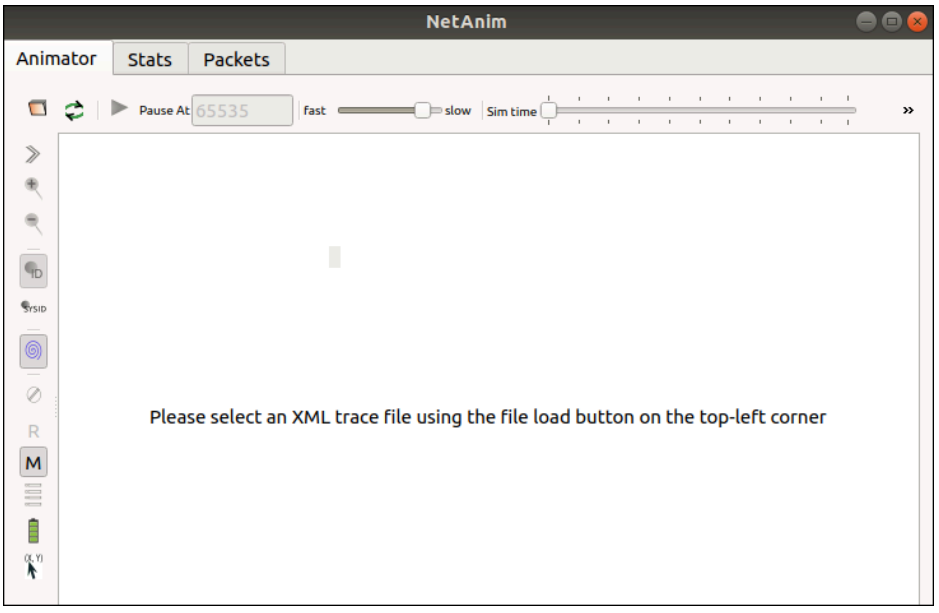
3. **qmake NetAnim.pro**

```
sims@sims-virtual-machine:~/Downloads/ns-allinone-3.32/netanim-3.108$ qmake Net
Anim.pro
```

4. **qmake NetAnim.pro**
   ./NetAnim

```
sims@sims-virtual-machine:~/Downloads/ns-allinone-3.32/netanim-3.108$ qmake Net
Anim.pro
sims@sims-virtual-machine:~/Downloads/ns-allinone-3.32/netanim-3.108$ ./NetAnim
```

OUTPUT :



Practical No -2

Aim: Linux Network Commands – ifconfig, ip, ping, netstat, traceroute, nslookup, route, hostname.

Code:

```
lathika@lathika:~$ sudo apt install net-tools
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 204 kB of archives.
```

Command: ifconfig

```
lathika@lathika: ~
lathika@lathika:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fd00::66f3:797e:4171:e709  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::a00:27ff:fed6:b58f  prefixlen 64  scopeid 0x20<link>
        inet6 fd00::a00:27ff:fed6:b58f  prefixlen 64  scopeid 0x0<global>
        ether 08:00:27:d6:b5:8f  txqueuelen 1000  (Ethernet)
        RX packets 73  bytes 21342 (21.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 165  bytes 23977 (23.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 163  bytes 14641 (14.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 163  bytes 14641 (14.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Command: ip

```
lathika@lathika:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen
1000
    link/ether 08:00:27:d6:b5:8f brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 84372sec preferred_lft 84372sec
    inet6 fd00::f96a:d63a:dbc0:c36d/64 scope global temporary dynamic
       valid_lft 86056sec preferred_lft 14056sec
    inet6 fd00::a00:27ff:fed6:b58f/64 scope global dynamic mngtmpaddr
       valid_lft 86056sec preferred_lft 14056sec
    inet6 fe80::a00:27ff:fed6:b58f/64 scope link
       valid_lft forever preferred_lft forever
```

Command: ping

```
lathika@lathika:~$ ping google.com
PING google.com (142.250.192.238) 56(84) bytes of data.
64 bytes from del11s13-in-f14.1e100.net (142.250.192.238): icmp_seq=1 ttl=255 time=54.3 ms
64 bytes from del11s13-in-f14.1e100.net (142.250.192.238): icmp_seq=2 ttl=255 time=52.9 ms
64 bytes from del11s13-in-f14.1e100.net (142.250.192.238): icmp_seq=3 ttl=255 time=47.5 ms
64 bytes from del11s13-in-f14.1e100.net (142.250.192.238): icmp_seq=4 ttl=255 time=55.9 ms
^C
--- google.com ping statistics ---
5 packets transmitted, 4 received, 20% packet loss, time 4010ms
rtt min/avg/max/mdev = 47.458/52.653/55.949/3.189 ms
```

Command: netstart

```
lathika@lathika:~$ netstat -tuln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.54:53           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp6       0      0 ::1:631                 :::*                    LISTEN
udp        0      0 127.0.0.54:53           0.0.0.0:*
udp        0      0 127.0.0.53:53           0.0.0.0:*
udp        0      0 0.0.0.0:5353            0.0.0.0:*
udp        0      0 0.0.0.0:56285           0.0.0.0:*
udp6       0      0 :::5353                 :::*
udp6       0      0 :::37222                :::*
```

Command: traceroute

```
lathika@lathika:~$ traceroute -n 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  10.0.2.2  0.508 ms  0.236 ms  0.249 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
```

Command: nslookup

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

Non-authoritative answer:
Name:   google.com
Address: 142.250.192.206
Name:   google.com
Address: 2404:6800:4002:817::200e
```

Command: route

```
lathika@lathika:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.0.2.2        0.0.0.0         UG    100    0        0 enp0s3
10.0.2.0        0.0.0.0         255.255.255.0   U     100    0        0 enp0s3
```

Command: hostname

```
lathika@lathika:~$ hostname
lathika
```

Practical No - 3

Aim: Program to simulate Point to Point topology

Code:
first.cc

```
#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/netanim-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[])
{
 CommandLinecmd (__FILE__);
 cmd.Parse (argc, argv);

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

 // Take n number of computer
 NodeContainer nodes;
 nodes.Create (2);

 // Choose your technology to communicate
 PointToPointHelperpointToPoint;
 pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
 pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));

 // Install technology on computers
 NetDeviceContainer devices;
 devices = pointToPoint.Install (nodes);
```

```
 // Asking to follow rules
 InternetStackHelper stack;
 stack.Install (nodes);
 // Assign Ip address to communicate
 Ipv4AddressHelper address;
 address.SetBase ("10.1.1.0", "255.255.255.0");
 Ipv4InterfaceContainer interfaces = address.Assign (devices);

 // Create a x type of server on port x
 UdpEchoServerHelperechoServer (9);

 // Install server on a node then Start and Stop the server
 ApplicationContainerserverApps = echoServer.Install (nodes.Get (1));
 serverApps.Start (Seconds (1.0));
 serverApps.Stop (Seconds (10.0));

 // Create x type of client and set its attributes
 UdpEchoClientHelperechoClient (interfaces.GetAddress (1), 9);
 echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
 echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
 echoClient.SetAttribute ("PacketSize", UintegerValue (1024));
 // Install the server then Start and Stop it.
 ApplicationContainerclientApps = echoClient.Install (nodes.Get (0));
 clientApps.Start (Seconds (2.0));
 clientApps.Stop (Seconds (10.0));
 // NetAnim
 AnimationInterfaceanim("NetAnimFolder/first.xml");
 anim.SetConstantPosition(nodes.Get(0), 10.0, 10.0);
 anim.SetConstantPosition(nodes.Get(1), 20.0, 20.0);

 // Run the Simulation
 Simulator::Run ();
 Simulator::Destroy ();
 return 0;
}
```

./waf –run scratch/first

```
sims@sims-virtual-machine:~/Downloads/ns-allinone-3.32/ns-3.32$ ./waf --run scr
atch/first
Waf: Entering directory `/home/sims/Downloads/ns-allinone-3.32/ns-3.32/build'
Waf: Leaving directory `/home/sims/Downloads/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.381s)
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
```

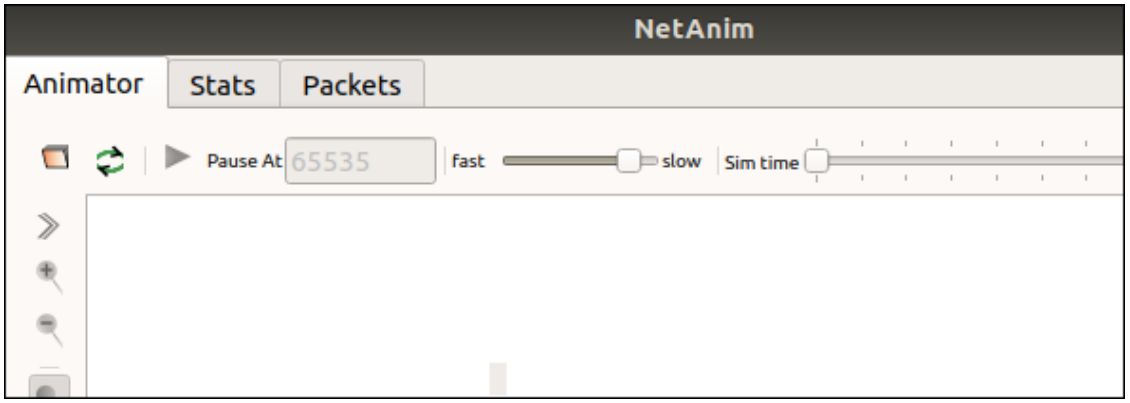
In new terminal :

- **qmake NetAnim.pro**
- **make**

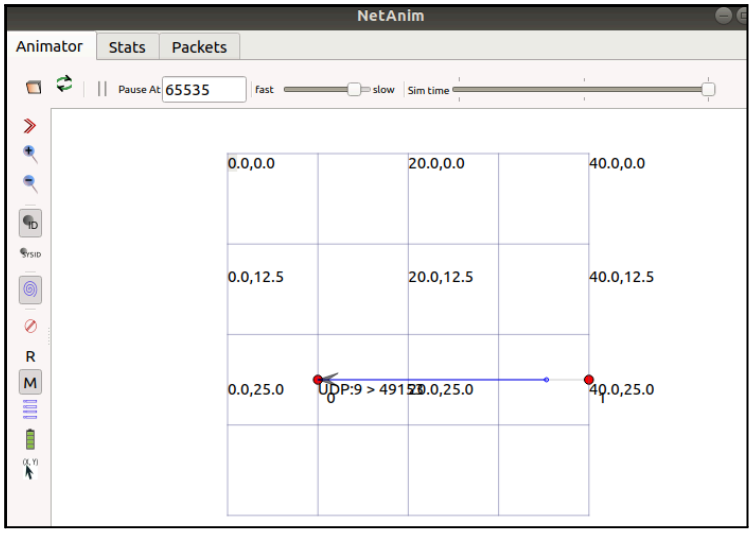
```
sims@sims-virtual-machine:~/Downloads/ns-allinone-3.32/netanim-3.108$ qmake Net
Anim.pro
Info: creating stash file /home/sims/Downloads/ns-allinone-3.32/netanim-3.108/.
qmake.stash
sims@sims-virtual-machine:~/Downloads/ns-allinone-3.32/netanim-3.108$ sudo apt
install qt5-qmake
Reading package lists... Done
Building dependency tree
Reading state information... Done
qt5-qmake is already the newest version (5.9.5+dfsg-0ubuntu2.6).
qt5-qmake set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
sims@sims-virtual-machine:~/Downloads/ns-allinone-3.32/netanim-3.108$ qmake Net
Anim.pro
sims@sims-virtual-machine:~/Downloads/ns-allinone-3.32/netanim-3.108$ make
g++ -c -pipe -O2 -Wall -W -D_REENTRANT -fPIC -DNS3_LOG_ENABLE -DQT_NO_DEBUG -DQ
T_PRINTSUPPORT_LIB -DQT_WIDGETS_LIB -DQT_GUI_LIB -DQT_CORE_LIB -I. -Iqtproperty
browser/src -isystem /usr/include/x86_64-linux-gnu/qt5 -isystem /usr/include/x8
6_64-linux-gnu/qt5/QtPrintSupport -isystem /usr/include/x86_64-linux-gnu/qt5/Qt
```

./NetAnim

Go to file and open first.xml



Output:



Practical No - 4

Aim: Program to simulate Bus topology

Code:
second.cc

```
#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"

#include "ns3/netanim-module.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;

  CommandLinecmd (__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;
```

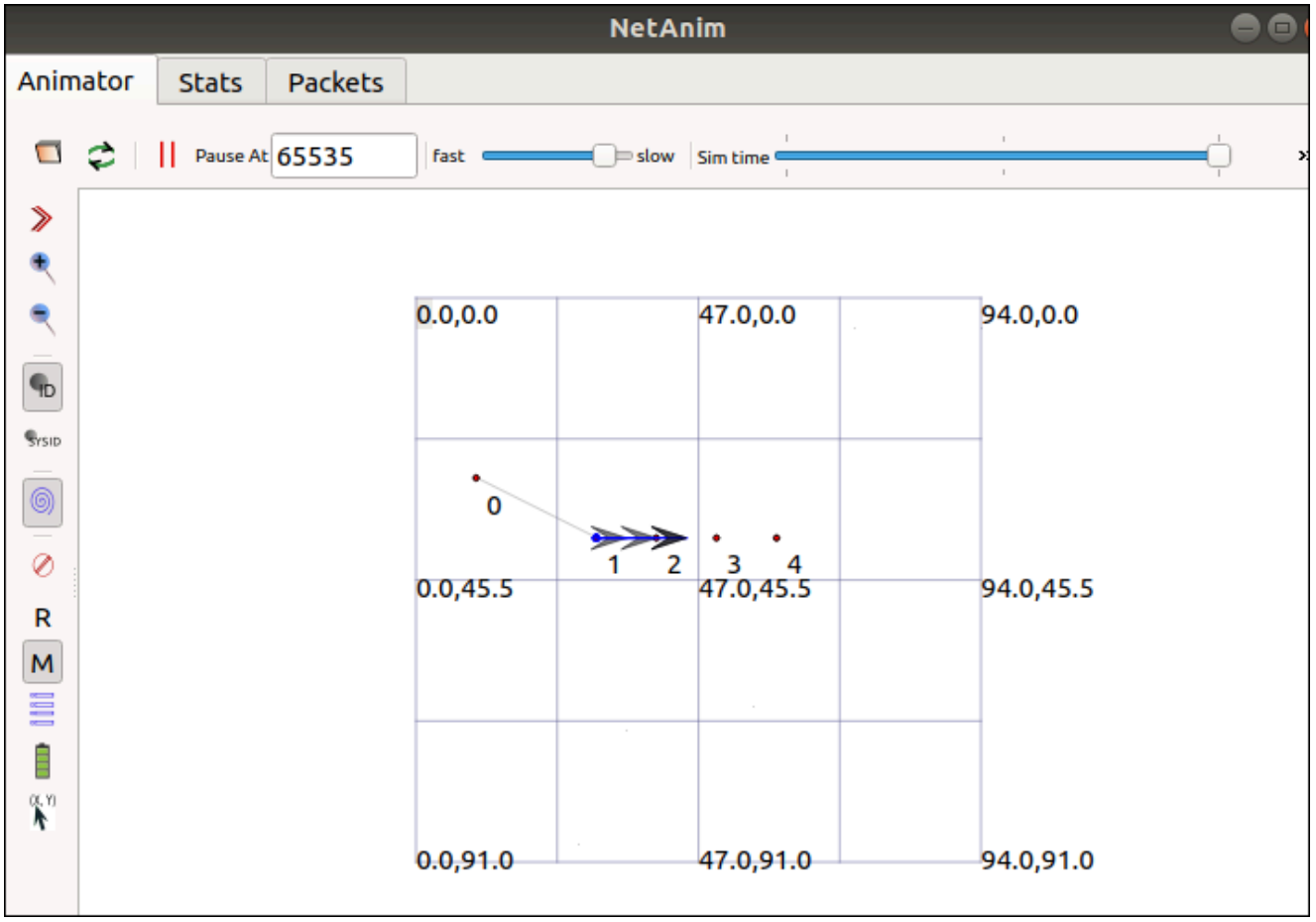
```
p2pNodes.Create (2);

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

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

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

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

NetDeviceContainercsmaDevices;
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);

UdpEchoServerHelperechoServer (9);

ApplicationContainerserverApps = echoServer.Install (csmaNodes.Get (nCsma));
serverApps.Start (Seconds (1.0));
serverApps.Stop (Seconds (10.0));

UdpEchoClientHelperechoClient (csmaInterfaces.GetAddress (nCsma), 9);
echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
echoClient.SetAttribute ("PacketSize", UintegerValue (1024));

ApplicationContainerclientApps = 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);

 // NetAnim Simulation Output
 AnimationInterfaceanim("second.xml");

 // Position point-to-point nodes
 anim.SetConstantPosition(p2pNodes.Get(0), 10.0, 30.0);
 anim.SetConstantPosition(p2pNodes.Get(1), 20.0, 30.0);

 // Position CSMA (bus) nodes in a row to visualize the bus topology
 for (uint32_t i = 0; i<csmaNodes.GetN(); ++i)
 {
    anim.SetConstantPosition(csmaNodes.Get(i), 30.0 + 10 * i, 40.0);
 }

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

Output:

```
sims@sims-virtual-machine:~/Downloads/ns-allinone-3.32/ns-3.32$ ./waf --run scr
atch/second
Waf: Entering directory `/home/sims/Downloads/ns-allinone-3.32/ns-3.32/build'
[1995/2068] Compiling scratch/second.cc
[2028/2068] Linking build/scratch/second
Waf: Leaving directory `/home/sims/Downloads/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (4.430s)
```

```
sims@sims-virtual-machine:~/Downloads/ns-allinone-3.32/netanim-3.108$ ./NetAnim
Gtk-Message: 16:55:28.255: GtkDialog mapped without a transient parent. This is
 discouraged.
```


NetAnim
Animator
Stats
Packets
Pause At
65535
fast
slow
Sim time
0.0,0.0
47.0,0.0
94.0,0.0
0
1
2
3
4
0.0,45.5
47.0,45.5
94.0,45.5
0.0,91.0
47.0,91.0
94.0,91.0

Practical No - 5

Aim: Program to simulate Star topology

Code:
star.cc

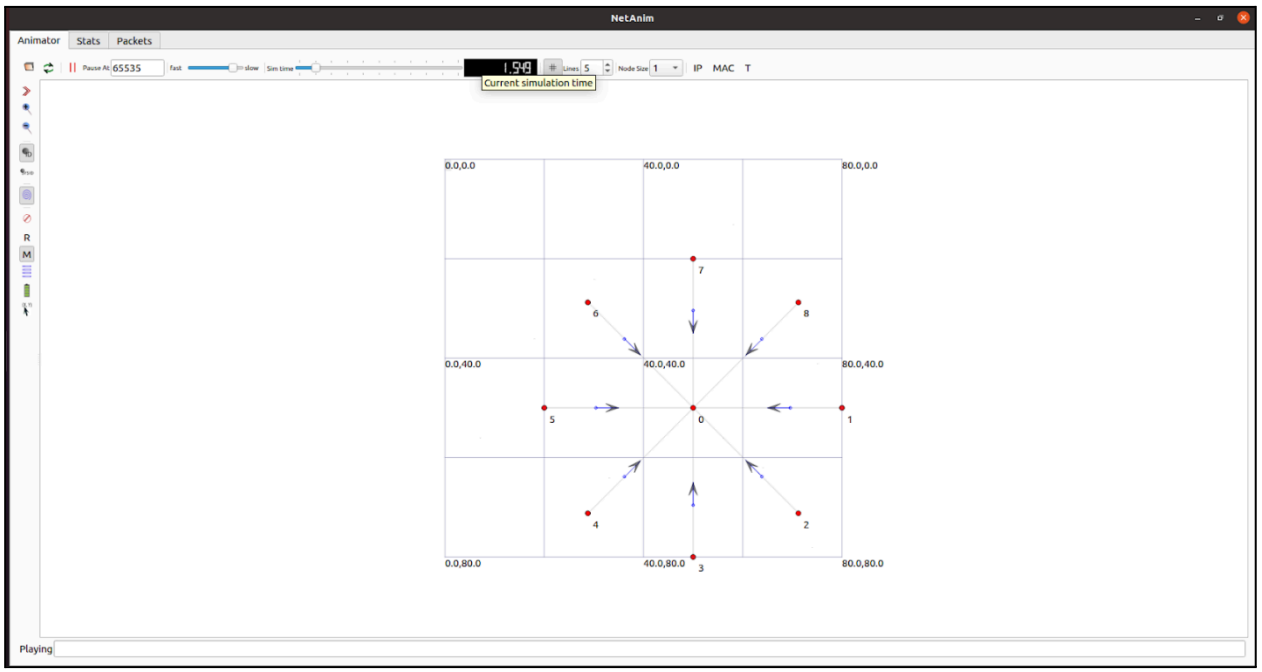
```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/netanim-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
#include "ns3/point-to-point-layout-module.h"

// 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;

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

```
  NS_LOG_INFO ("Build star topology.");
  PointToPointHelperpointToPoint;
  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));
  PacketSinkHelperpacketSinkHelper ("ns3::TcpSocketFactory", hubLocalAddress);
  ApplicationContainerhubApp = 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.
  //
  OnOffHelperonOffHelper ("ns3::TcpSocketFactory", Address ());
  onOffHelper.SetAttribute ("OnTime", StringValue
("ns3::ConstantRandomVariable[Constant=1]"));
  onOffHelper.SetAttribute ("OffTime", StringValue
("ns3::ConstantRandomVariable[Constant=0]"));

  ApplicationContainerspokeApps;
  for (uint32_t i = 0; i<star.SpokeCount (); ++i)
    {
      AddressValueremoteAddress (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.");

  Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

  NS_LOG_INFO ("Enable pcap tracing.");
```

```
 pointToPoint.EnablePcapAll ("star");
 AnimationInterfaceanim("NetAnimFolder/star.xml");

 anim.SetConstantPosition(star.GetHub(), 50.0, 50.0);
 // Arrange spoke nodes in a circle around the hub
 double radius = 30.0;
 for (uint32_t i = 0; i<star.SpokeCount(); ++i)
 {
  double angle = i * 2 * M_PI / star.SpokeCount();
  double x = 50.0 + radius * std::cos(angle); // center at (50,50)
  double y = 50.0 + radius * std::sin(angle);
  anim.SetConstantPosition(star.GetSpokeNode(i), x, y);
 }
 NS_LOG_INFO ("Run Simulation.");
 Simulator::Run ();
 Simulator::Destroy ();
 NS_LOG_INFO ("Done.");
 return 0;
}
```

Output:

```
sims@sims-virtual-machine:~/Downloads/ns-allinone-3.32/ns-3.32$ ./waf --run scr
atch/star
Waf: Entering directory `/home/sims/Downloads/ns-allinone-3.32/ns-3.32/build'
Waf: Leaving directory `/home/sims/Downloads/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.373s)
```



Practical No - 6

Aim: Program to simulate Mesh topology

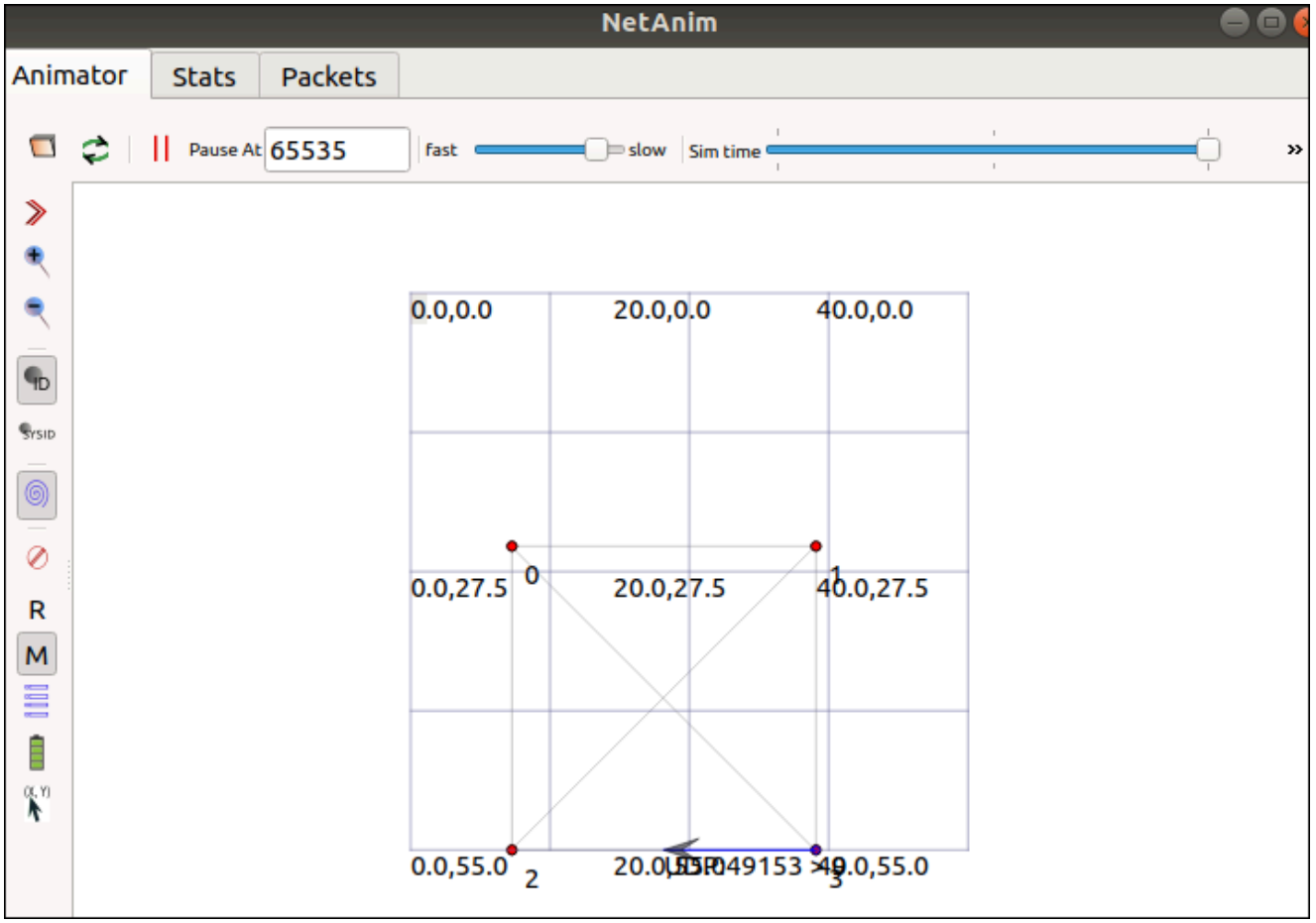
Code:
mesh-topology.cc

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

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("MeshTopologyExample");

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

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

 // Create 4 nodes for the mesh topology (2x2 grid)
 NodeContainer nodes;
 nodes.Create (4);

 // Set up point-to-point links between all pairs of nodes
 PointToPointHelperpointToPoint;
 pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
 pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));

 NetDeviceContainer devices;
 // Create the links between all pairs of nodes
 for (uint32_t i = 0; i<nodes.GetN(); ++i)
  {
   for (uint32_t j = i + 1; j <nodes.GetN(); ++j)
    {
     NetDeviceContainerlinkDevices = pointToPoint.Install (nodes.Get (i), nodes.Get (j));
     devices.Add (linkDevices);
    }
  }

 // Install the Internet stack on the nodes
```

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

 // Assign IP addresses to each device
 Ipv4AddressHelper address;
 address.SetBase ("10.1.1.0", "255.255.255.0");
 Ipv4InterfaceContainer interfaces = address.Assign (devices);

 // Set up UDP Echo server on the last node (node 3)
 UdpEchoServerHelperechoServer (9);
 ApplicationContainerserverApps = echoServer.Install (nodes.Get (3));
 serverApps.Start (Seconds (1.0));
 serverApps.Stop (Seconds (10.0));

 // Set up UDP Echo client on the first node (node 0)
 UdpEchoClientHelperechoClient (interfaces.GetAddress (3), 9);
 echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
 echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
 echoClient.SetAttribute ("PacketSize", UintegerValue (1024));

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

 // Set mobility model for all nodes (constant position model)
 MobilityHelper mobility;
 mobility.SetMobilityModel("ns3::ConstantPositionMobilityModel");
 mobility.Install(nodes);

 // Set constant positions for the nodes to create a 2x2 grid
 AnimationInterfaceanim("mesh_topology.xml");
 // Positioning nodes in a 2x2 grid layout
 AnimationInterface::SetConstantPosition (nodes.Get(0), 10, 25); // Node 0 at (10, 25)
 AnimationInterface::SetConstantPosition (nodes.Get(1), 40, 25); // Node 1 at (40, 25)
 AnimationInterface::SetConstantPosition (nodes.Get(2), 10, 55); // Node 2 at (10, 55)
 AnimationInterface::SetConstantPosition (nodes.Get(3), 40, 55); // Node 3 at (40, 55)

 // Enable packet metadata in the animation output
 anim.EnablePacketMetadata(true);

 // Run the simulation
 Simulator::Run ();
 Simulator::Destroy ();
 return 0;
}
```

Output:

```
sims@sims-virtual-machine:~/Downloads/ns-allinone-3.32/ns-3.32$ ./waf --run scr
atch/mesh-topology
Waf: Entering directory `/home/sims/Downloads/ns-allinone-3.32/ns-3.32/build'
[1999/2070] Compiling scratch/mesh-topology.cc
[2030/2070] Linking build/scratch/mesh-topology
Waf: Leaving directory `/home/sims/Downloads/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (3.223s)
At time +2s client sent 1024 bytes to 10.1.1.4 port 9
```



Practical No - 7

Aim: Program to simulate Hybrid topology

Code:
third.cc

```
#include "ns3/core-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/network-module.h"
#include "ns3/applications-module.h"
#include "ns3/mobility-module.h"
#include "ns3/csma-module.h"
#include "ns3/internet-module.h"
#include "ns3/yans-wifi-helper.h"
#include "ns3/ssid.h"

#include "ns3/netanim-module.h"

// Default Network Topology
//
//   Wifi 10.1.3.0
//                 AP
// *    *    *    *
// |    |    |    |    10.1.1.0
// n5   n6   n7   n0 -------------- n1   n2   n3   n4
//                  point-to-point  |    |    |    |
//                                  ================
//                                    LAN 10.1.2.0

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("ThirdScriptExample");

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

  CommandLinecmd (__FILE__);
  cmd.AddValue ("nCsma", "Number of \"extra\" CSMA nodes/devices", nCsma);
  cmd.AddValue ("nWifi", "Number of wifi STA devices", nWifi);
  cmd.AddValue ("verbose", "Tell echo applications to log if true", verbose);
  cmd.AddValue ("tracing", "Enable pcap tracing", tracing);

  cmd.Parse (argc,argv);
```

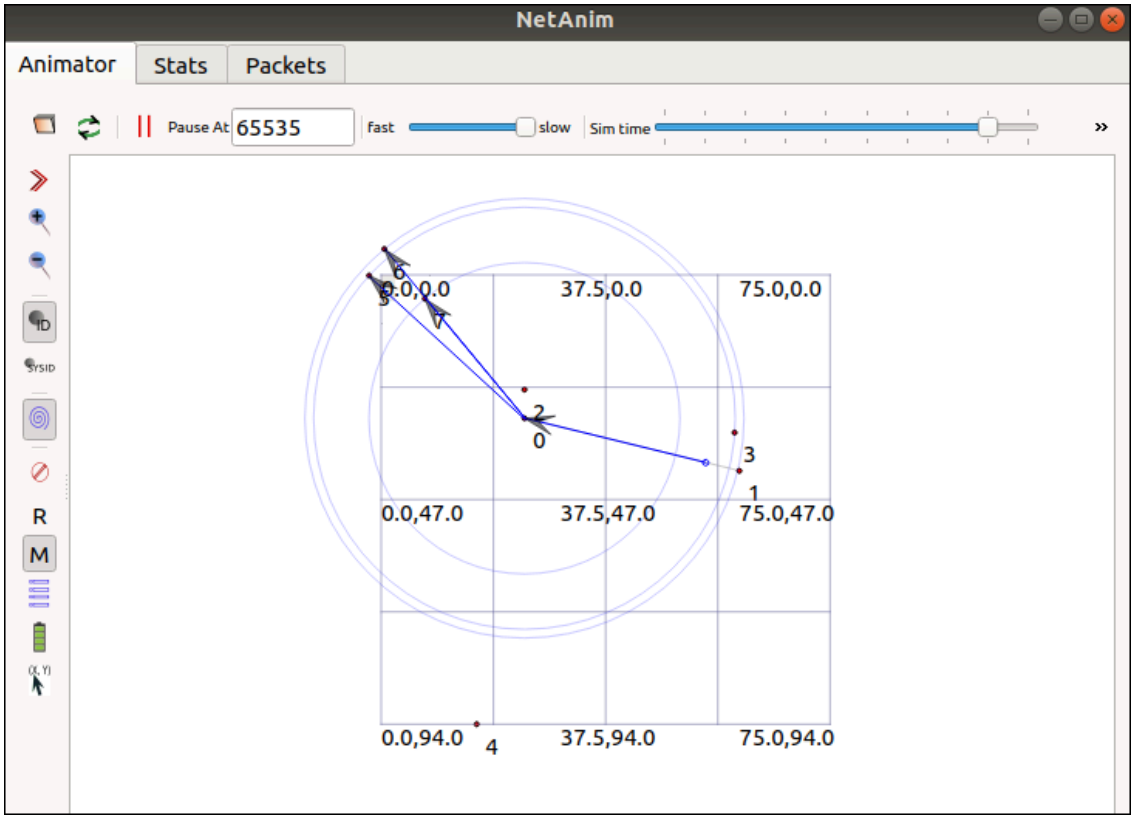
```
 // The underlying restriction of 18 is due to the grid position
 // allocator's configuration; the grid layout will exceed the
 // bounding box if more than 18 nodes are provided.
 if (nWifi> 18)
  {
   std::cout<< "nWifi should be 18 or less; otherwise grid layout exceeds the bounding box"
<<std::endl;
   return 1;
  }

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

 NodeContainer p2pNodes;
 p2pNodes.Create (2);

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

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

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

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

 NetDeviceContainercsmaDevices;
 csmaDevices = csma.Install (csmaNodes);

 NodeContainerwifiStaNodes;
 wifiStaNodes.Create (nWifi);
 NodeContainerwifiApNode = p2pNodes.Get (0);

 YansWifiChannelHelper channel = YansWifiChannelHelper::Default ();
 YansWifiPhyHelperphy = YansWifiPhyHelper::Default ();
 phy.SetChannel (channel.Create ());

 WifiHelperwifi;
 wifi.SetRemoteStationManager ("ns3::AarfWifiManager");

 WifiMacHelper mac;
```

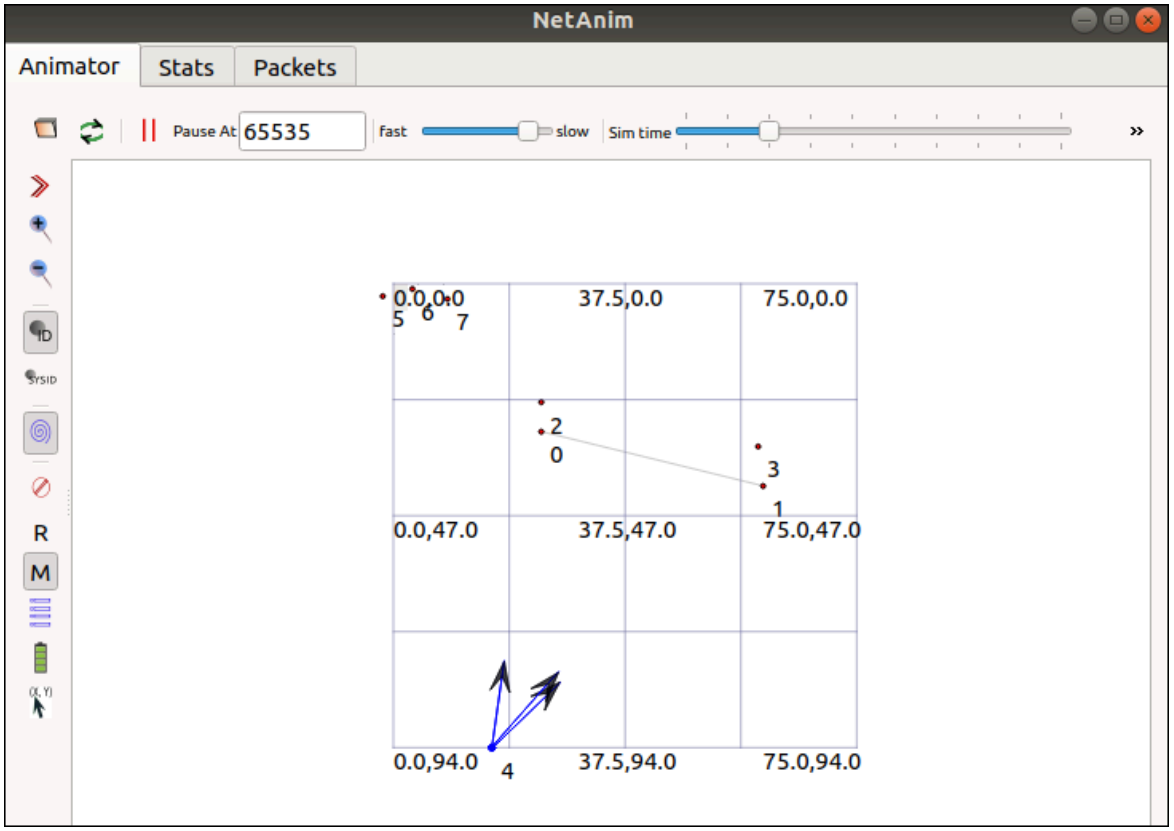
```
Ssidssid = Ssid ("ns-3-ssid");
mac.SetType ("ns3::StaWifiMac",
        "Ssid", SsidValue (ssid),
        "ActiveProbing", BooleanValue (false));

NetDeviceContainerstaDevices;
staDevices = wifi.Install (phy, mac, wifiStaNodes);

mac.SetType ("ns3::ApWifiMac",
        "Ssid", SsidValue (ssid));

NetDeviceContainerapDevices;
apDevices = wifi.Install (phy, mac, wifiApNode);

MobilityHelper mobility;

mobility.SetPositionAllocator ("ns3::GridPositionAllocator",
                    "MinX", DoubleValue (0.0),
                    "MinY", DoubleValue (0.0),
                    "DeltaX", DoubleValue (5.0),
                    "DeltaY", DoubleValue (10.0),
                    "GridWidth", UintegerValue (3),
                    "LayoutType", StringValue ("RowFirst"));

mobility.SetMobilityModel ("ns3::RandomWalk2dMobilityModel",
                "Bounds", RectangleValue (Rectangle (-50, 50, -50, 50)));
mobility.Install (wifiStaNodes);

mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
mobility.Install (wifiApNode);

InternetStackHelper stack;
stack.Install (csmaNodes);
stack.Install (wifiApNode);
stack.Install (wifiStaNodes);

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);

address.SetBase ("10.1.3.0", "255.255.255.0");
address.Assign (staDevices);
address.Assign (apDevices);
```

```
 UdpEchoServerHelperechoServer (9);

 ApplicationContainerserverApps = echoServer.Install (csmaNodes.Get (nCsma));
 serverApps.Start (Seconds (1.0));
 serverApps.Stop (Seconds (10.0));

 UdpEchoClientHelperechoClient (csmaInterfaces.GetAddress (nCsma), 9);
 echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
 echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
 echoClient.SetAttribute ("PacketSize", UintegerValue (1024));

 ApplicationContainerclientApps =
  echoClient.Install (wifiStaNodes.Get (nWifi - 1));
 clientApps.Start (Seconds (2.0));
 clientApps.Stop (Seconds (10.0));

 Ipv4GlobalRoutingHelper::PopulateRoutingTables ();
 Simulator::Stop (Seconds (10.0));
 if (tracing == true)
  {
    pointToPoint.EnablePcapAll ("third");
    phy.EnablePcap ("third", apDevices.Get (0));
    csma.EnablePcap ("third", csmaDevices.Get (0), true);
  }
 AnimationInterfaceanim("third.xml");
 anim.SetConstantPosition(p2pNodes.Get(0), 30.0, 30.0);

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

Output:

```
sims@sims-virtual-machine:~/Downloads/ns-allinone-3.32/ns-3.32$ ./waf --run scr
atch/third
Waf: Entering directory `/home/sims/Downloads/ns-allinone-3.32/ns-3.32/build'
[2011/2074] Compiling scratch/third.cc
[2034/2074] Linking build/scratch/third
Waf: Leaving directory `/home/sims/Downloads/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (3.310s)
AnimationInterface WARNING:Node:1 Does not have a mobility model. Use SetConsta
ntPosition if it is stationary
AnimationInterface WARNING:Node:2 Does not have a mobility model. Use SetConsta
ntPosition if it is stationary
AnimationInterface WARNING:Node:3 Does not have a mobility model. Use SetConsta
ntPosition if it is stationary
AnimationInterface WARNING:Node:4 Does not have a mobility model. Use SetConsta
ntPosition if it is stationary
AnimationInterface WARNING:Node:1 Does not have a mobility model. Use SetConsta
ntPosition if it is stationary
AnimationInterface WARNING:Node:2 Does not have a mobility model. Use SetConsta
```


NetAnim
Animator
Stats
Packets
Pause At 65535
fast
slow
Sim time
0.0,0.0
37.5,0.0
75.0,0.0
0.0,47.0
37.5,47.0
75.0,47.0
0.0,94.0
37.5,94.0
75.0,94.0
0
1
2
3
4
5
6
7


NetAnim
Animator
Stats
Packets
Pause At 65535
fast
slow
Sim time
0.0,0.0
37.5,0.0
75.0,0.0
0.0,47.0
37.5,47.0
75.0,47.0
0.0,94.0
37.5,94.0
75.0,94.0
0
1
2
3
4
5
6
7

Practical No - 8

Aim: Program to simulate UDP Client Server

Code:
udp-echo.cc

```
// Network topology
//
//       n0    n1   n2   n3
//       |     |    |    |
//       ================
//             LAN
//
// - UDP flows from n0 to n1 and back
// - DropTail queues
// - Tracing of queues and packet receptions to file "udp-echo.tr"

#include <fstream>
#include "ns3/core-module.h"
#include "ns3/csma-module.h"
#include "ns3/applications-module.h"
#include "ns3/internet-module.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("UdpEchoExample");

int
main (int argc, char *argv[])
{
  // Enable logging
  LogComponentEnable ("UdpEchoExample", LOG_LEVEL_INFO);
  LogComponentEnable ("UdpEchoClientApplication", LOG_LEVEL_INFO);
  LogComponentEnable ("UdpEchoServerApplication", LOG_LEVEL_INFO);
  bool useV6 = false;
  Address serverAddress;
  CommandLinecmd (__FILE__);
  cmd.AddValue ("useIpv6", "Use Ipv6", useV6);
  cmd.Parse (argc, argv);

  NS_LOG_INFO ("Creating 4 nodes.");
  NodeContainer n;
  n.Create (4);

  InternetStackHelper internet;
  internet.Install (n);

  NS_LOG_INFO ("Creating CSMA channel.");
  CsmaHelpercsma;
```

```
csma.SetChannelAttribute ("DataRate", DataRateValue (DataRate (5000000)));
csma.SetChannelAttribute ("Delay", TimeValue (MilliSeconds (2)));
csma.SetDeviceAttribute ("Mtu", UintegerValue (1400));
NetDeviceContainer d = csma.Install (n);

NS_LOG_INFO ("Assigning IP Addresses.");
if (!useV6)
{
  Ipv4AddressHelper ipv4;
  ipv4.SetBase ("10.1.1.0", "255.255.255.0");
  Ipv4InterfaceContainer i = ipv4.Assign (d);
  serverAddress = Address(i.GetAddress (1));
  NS_LOG_INFO ("Server IPv4 Address: " <<i.GetAddress(1));
}
else
{
  Ipv6AddressHelper ipv6;
  ipv6.SetBase ("2001:0000:f00d:cafe::", Ipv6Prefix (64));
  Ipv6InterfaceContainer i6 = ipv6.Assign (d);
  serverAddress = Address(i6.GetAddress (1,1));
  NS_LOG_INFO ("Server IPv6 Address: " << i6.GetAddress(1,1));
}

NS_LOG_INFO ("Installing UDP Echo Server on Node 1.");
uint16_t port = 9;
UdpEchoServerHelper server (port);
ApplicationContainer apps = server.Install (n.Get (1));
apps.Start (Seconds (1.0));
apps.Stop (Seconds (10.0));

NS_LOG_INFO ("Installing UDP Echo Client on Node 0.");
uint32_t packetSize = 1024;
uint32_t maxPacketCount = 1;
Time interPacketInterval = Seconds (1.0);
UdpEchoClientHelper client (serverAddress, port);
client.SetAttribute ("MaxPackets", UintegerValue (maxPacketCount));
client.SetAttribute ("Interval", TimeValue (interPacketInterval));
client.SetAttribute ("PacketSize", UintegerValue (packetSize));
apps = client.Install (n.Get (0));
apps.Start (Seconds (2.0));
apps.Stop (Seconds (10.0));
NS_LOG_INFO ("Tracing enabled.");
AsciiTraceHelper ascii;
csma.EnableAsciiAll (ascii.CreateFileStream ("udp-echo.tr"));
csma.EnablePcapAll ("udp-echo", false);
NS_LOG_INFO ("Running Simulation.");
Simulator::Run ();
Simulator::Destroy ();
NS_LOG_INFO ("Simulation Done.");
```

```
}
```

Output:

```
sims@sims-virtual-machine:~/Downloads/ns-allinone-3.32/ns-3.32$ ./waf --run scr
atch/udp-echo
Waf: Entering directory `/home/sims/Downloads/ns-allinone-3.32/ns-3.32/build'
[1992/2076] Compiling scratch/udp-echo.cc
[1997/2076] Linking build/scratch/ftp
[2024/2076] Compiling scratch/second.cc
[2025/2076] Compiling scratch/first.cc
[2026/2076] Compiling scratch/subdir/scratch-simulator-subdir.cc
[2027/2076] Linking build/scratch/udp-echo
[2028/2076] Linking build/scratch/second
[2029/2076] Linking build/scratch/first
[2030/2076] Linking build/scratch/subdir/subdir
[2031/2076] Compiling scratch/mesh-topology.cc
[2032/2076] Compiling scratch/star-5-node.cc
[2033/2076] Compiling scratch/third.cc
[2034/2076] Linking build/scratch/mesh-topology
[2035/2076] Linking build/scratch/star-5-node
[2036/2076] Linking build/scratch/third
Waf: Leaving directory `/home/sims/Downloads/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (10.942s)
Creating 4 nodes.
Creating CSMA channel.
Assigning IP Addresses.
```

Practical No - 9

Aim: Program to simulate DHCP Server and Clients

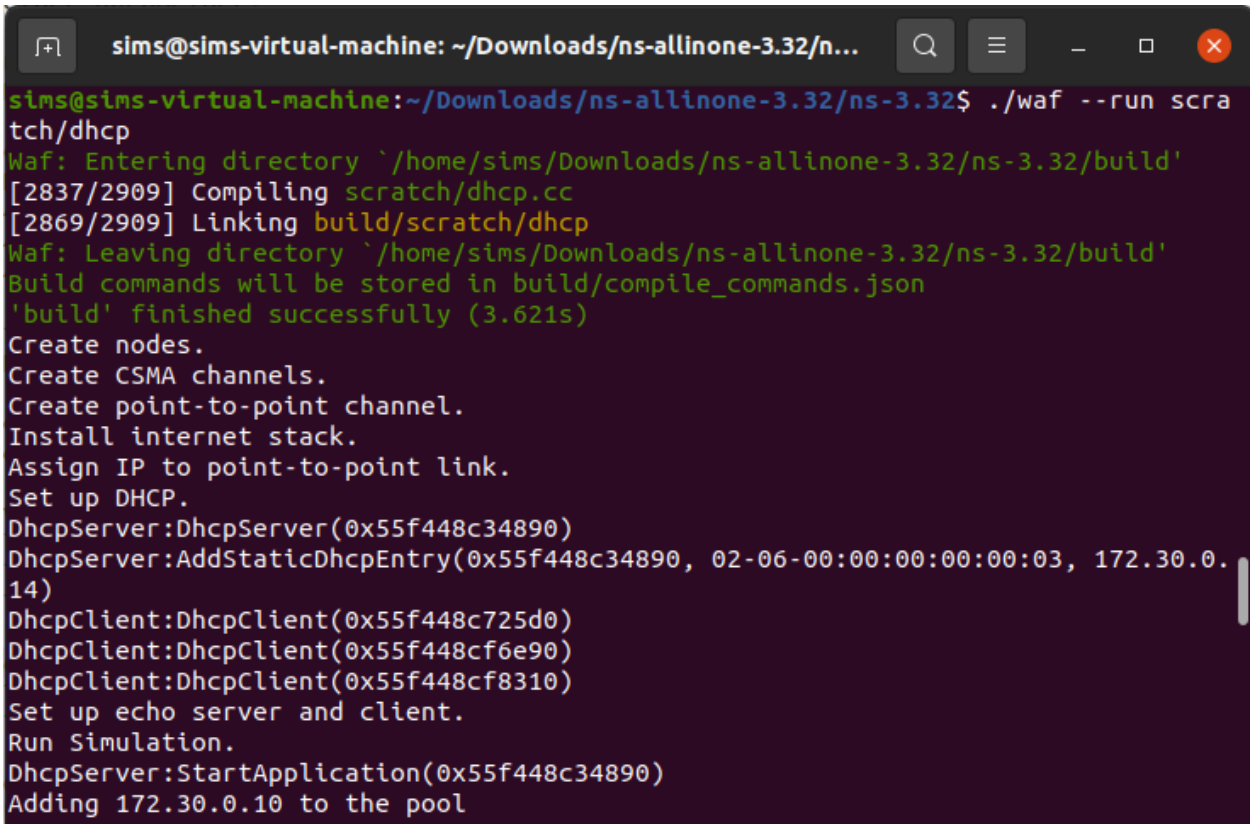
Code:
dhcp.cc

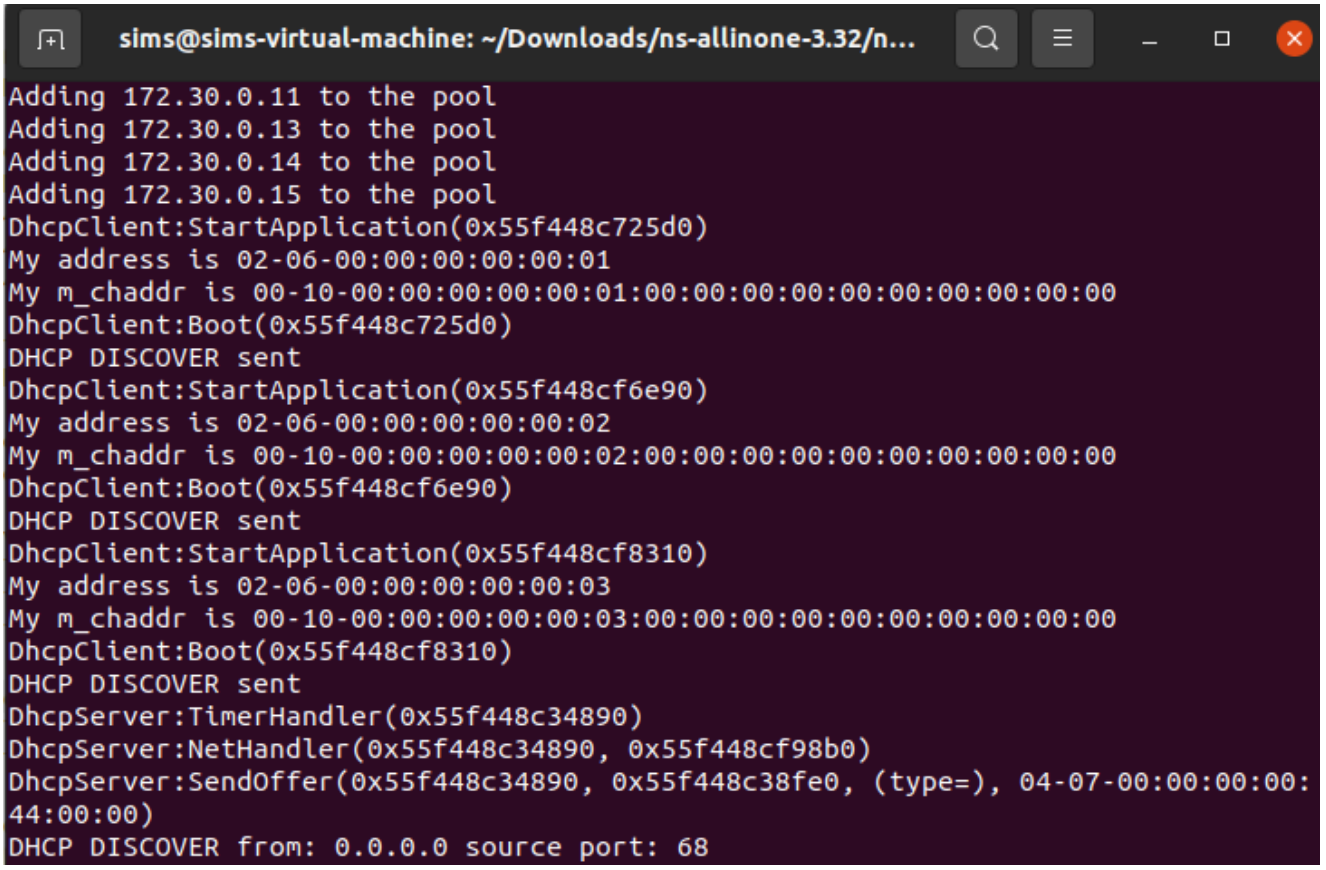
```
#include "ns3/core-module.h"
#include "ns3/internet-apps-module.h"
#include "ns3/csma-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("DhcpExample");

int
main (int argc, char *argv[])
{
 CommandLinecmd (__FILE__);

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

 cmd.Parse (argc, argv);

 if (verbose)
 {
  LogComponentEnable ("DhcpExample", LOG_LEVEL_INFO);
  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 CSMA channels.");
CsmaHelpercsma;
csma.SetChannelAttribute ("DataRate", StringValue ("5Mbps"));
csma.SetChannelAttribute ("Delay", StringValue ("2ms"));
csma.SetDeviceAttribute ("Mtu", UintegerValue (1500));
NetDeviceContainerdevNet = csma.Install (net);

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

NS_LOG_INFO ("Create point-to-point channel.");
PointToPointHelperpointToPoint;
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));

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

NS_LOG_INFO ("Install internet stack.");
InternetStackHelpertcpip;
tcpip.Install (nodes);
tcpip.Install (router);
tcpip.Install (p2pNodes.Get (1));

NS_LOG_INFO ("Assign IP to point-to-point link.");
Ipv4AddressHelper address;
address.SetBase ("172.30.1.0", "255.255.255.0");
Ipv4InterfaceContainer p2pInterfaces;
p2pInterfaces = address.Assign (p2pDevices);

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 ("Set up DHCP.");
DhcpHelperdhcpHelper;

Ipv4InterfaceContainer fixedNodes = dhcpHelper.InstallFixedAddress (
  devNet.Get (4), Ipv4Address ("172.30.0.17"), Ipv4Mask ("/24"));
fixedNodes.Get (0).first->SetAttribute ("IpForward", BooleanValue (true));

ApplicationContainerdhcpServerApp = dhcpHelper.InstallDhcpServer (
  devNet.Get (3), Ipv4Address ("172.30.0.12"),
  Ipv4Address ("172.30.0.0"), Ipv4Mask ("/24"),
  Ipv4Address ("172.30.0.10"), Ipv4Address ("172.30.0.15"),
```

```
  Ipv4Address ("172.30.0.17"));

 DynamicCast<DhcpServer> (dhcpServerApp.Get (0))->AddStaticDhcpEntry (
  devNet.Get (2)->GetAddress (), Ipv4Address ("172.30.0.14"));

 dhcpServerApp.Start (Seconds (0.0));
 dhcpServerApp.Stop (stopTime);

 NetDeviceContainerdhcpClientNetDevs;
 dhcpClientNetDevs.Add (devNet.Get (0));
 dhcpClientNetDevs.Add (devNet.Get (1));
 dhcpClientNetDevs.Add (devNet.Get (2));

 ApplicationContainerdhcpClients = dhcpHelper.InstallDhcpClient (dhcpClientNetDevs);
 dhcpClients.Start (Seconds (1.0));
 dhcpClients.Stop (stopTime);

 NS_LOG_INFO ("Set up echo server and client.");
 UdpEchoServerHelperechoServer (9);
 ApplicationContainerserverApps = echoServer.Install (p2pNodes.Get (1));
 serverApps.Start (Seconds (0.0));
 serverApps.Stop (stopTime);

 UdpEchoClientHelperechoClient (p2pInterfaces.GetAddress (1), 9);
 echoClient.SetAttribute ("MaxPackets", UintegerValue (100));
 echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
 echoClient.SetAttribute ("PacketSize", UintegerValue (1024));

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

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

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

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

Output:


sims@sims-virtual-machine: ~/Downloads/ns-allinone-3.32/n...
sims@sims-virtual-machine:~/Downloads/ns-allinone-3.32/ns-3.32$ ./waf --run scratch/dhcp
Waf: Entering directory `/home/sims/Downloads/ns-allinone-3.32/ns-3.32/build'
[2837/2909] Compiling scratch/dhcp.cc
[2869/2909] Linking build/scratch/dhcp
Waf: Leaving directory `/home/sims/Downloads/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (3.621s)
Create nodes.
Create CSMA channels.
Create point-to-point channel.
Install internet stack.
Assign IP to point-to-point link.
Set up DHCP.
DhcpServer:DhcpServer(0x55f448c34890)
DhcpServer:AddStaticDhcpEntry(0x55f448c34890, 02-06-00:00:00:00:00:03, 172.30.0.14)
DhcpClient:DhcpClient(0x55f448c725d0)
DhcpClient:DhcpClient(0x55f448cf6e90)
DhcpClient:DhcpClient(0x55f448cf8310)
Set up echo server and client.
Run Simulation.
DhcpServer:StartApplication(0x55f448c34890)
Adding 172.30.0.10 to the pool


sims@sims-virtual-machine: ~/Downloads/ns-allinone-3.32/n...
Adding 172.30.0.11 to the pool
Adding 172.30.0.13 to the pool
Adding 172.30.0.14 to the pool
Adding 172.30.0.15 to the pool
DhcpClient:StartApplication(0x55f448c725d0)
My address is 02-06-00:00:00:00:00:01
My m_chaddr is 00-10-00:00:00:00:00:01:00:00:00:00:00:00:00:00:00:00
DhcpClient:Boot(0x55f448c725d0)
DHCP DISCOVER sent
DhcpClient:StartApplication(0x55f448cf6e90)
My address is 02-06-00:00:00:00:00:02
My m_chaddr is 00-10-00:00:00:00:00:02:00:00:00:00:00:00:00:00:00:00
DhcpClient:Boot(0x55f448cf6e90)
DHCP DISCOVER sent
DhcpClient:StartApplication(0x55f448cf8310)
My address is 02-06-00:00:00:00:00:03
My m_chaddr is 00-10-00:00:00:00:00:03:00:00:00:00:00:00:00:00:00:00
DhcpClient:Boot(0x55f448cf8310)
DHCP DISCOVER sent
DhcpServer:TimerHandler(0x55f448c34890)
DhcpServer:NetHandler(0x55f448c34890, 0x55f448cf98b0)
DhcpServer:SendOffer(0x55f448c34890, 0x55f448c38fe0, (type=), 04-07-00:00:00:00:44:00:00)
DHCP DISCOVER from: 0.0.0.0 source port: 68

Practical No – 10

Aim: Program to simulate FTP using TCP

Code:
ftp.cc

```
#include <string>
#include <fstream>
#include "ns3/core-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/internet-module.h"
#include "ns3/applications-module.h"
#include "ns3/network-module.h"
#include "ns3/packet-sink.h"
using namespace ns3;
NS_LOG_COMPONENT_DEFINE("TcpBulkSendExample");
int main(int argc, char *argv[])
{
   bool tracing = false;
   uint32_t maxBytes = 0;
   //
   // Allow the user to override any of the defaults at
   // run-time, via command-line arguments
   //
   CommandLinecmd;
   cmd.AddValue("tracing", "Flag to enable/disable tracing", tracing);
   cmd.AddValue("maxBytes",
            "Total number of bytes for application to send", maxBytes);
   cmd.Parse(argc, argv);

   //
   // Explicitly create the nodes required by the topology (shown above).
   //
   NS_LOG_INFO("Create nodes.");
   NodeContainer nodes;
   nodes.Create(2);
   NS_LOG_INFO("Create channels.");
   //
   // Explicitly create the point-to-point link required by the topology (shown above).
   //
   PointToPointHelperpointToPoint;
   pointToPoint.SetDeviceAttribute("DataRate", StringValue("500Kbps"));
   pointToPoint.SetChannelAttribute("Delay", StringValue("5ms"));

   NetDeviceContainer devices;
   devices = pointToPoint.Install(nodes);
   //
   // Install the internet stack on the nodes
   //
```

```
  InternetStackHelper internet;
  internet.Install(nodes);
  //
  // We've got the "hardware" in place.  Now we need to add IP addresses.
  //
  NS_LOG_INFO("Assign IP Addresses.");
  Ipv4AddressHelper ipv4;
  ipv4.SetBase("10.1.1.0", "255.255.255.0");
  Ipv4InterfaceContainer i = ipv4.Assign(devices);
  NS_LOG_INFO("Create Applications.");
  //
  // Create a BulkSendApplication and install it on node 0
  //
  uint16_t port = 9; // well-known echo port number
  BulkSendHelpersource("ns3::TcpSocketFactory",
                  InetSocketAddress(i.GetAddress(1), port));
  // Set the amount of data to send in bytes.  Zero is unlimited.
  source.SetAttribute("MaxBytes", UintegerValue(maxBytes));
  ApplicationContainersourceApps = source.Install(nodes.Get(0));
  sourceApps.Start(Seconds(0.0));
  sourceApps.Stop(Seconds(10.0));
  //
  // Create a PacketSinkApplication and install it on node 1
  //
  PacketSinkHelpersink("ns3::TcpSocketFactory",
                  InetSocketAddress(Ipv4Address::GetAny(), port));
  ApplicationContainersinkApps = sink.Install(nodes.Get(1));
  sinkApps.Start(Seconds(0.0));
  sinkApps.Stop(Seconds(10.0));
  //
  // Set up tracing if enabled
  //
  if (tracing)
  {
     AsciiTraceHelper ascii;
     pointToPoint.EnableAsciiAll(ascii.CreateFileStream("tcp-bulk-send.tr"));
     pointToPoint.EnablePcapAll("tcp-bulk-send", false);
  }
  //
  // Now, do the actual simulation.
  //
  NS_LOG_INFO("Run Simulation.");
  Simulator::Stop(Seconds(10.0));
  Simulator::Run();
  Simulator::Destroy();
  NS_LOG_INFO("Done.");
  Ptr<PacketSink> sink1 = DynamicCast<PacketSink>(sinkApps.Get(0));
  std::cout<< "Total Bytes Received: " << sink1->GetTotalRx() <<std::endl;
}
```

Output:


sims@sims-virtual-machine:~/Downloads/ns-allinone-3.32/ns-3.32$ ./waf --run scra
tch/ftp
Waf: Entering directory `/home/sims/Downloads/ns-allinone-3.32/ns-3.32/build'
[2840/2911] Compiling scratch/ftp.cc
[2871/2911] Linking build/scratch/ftp
Waf: Leaving directory `/home/sims/Downloads/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (3.646s)
Total Bytes Received: 565480
sims@sims-virtual-machine:~/Downloads/ns-allinone-3.32/ns-3.32$

FYMCA – B
Roll no. 98

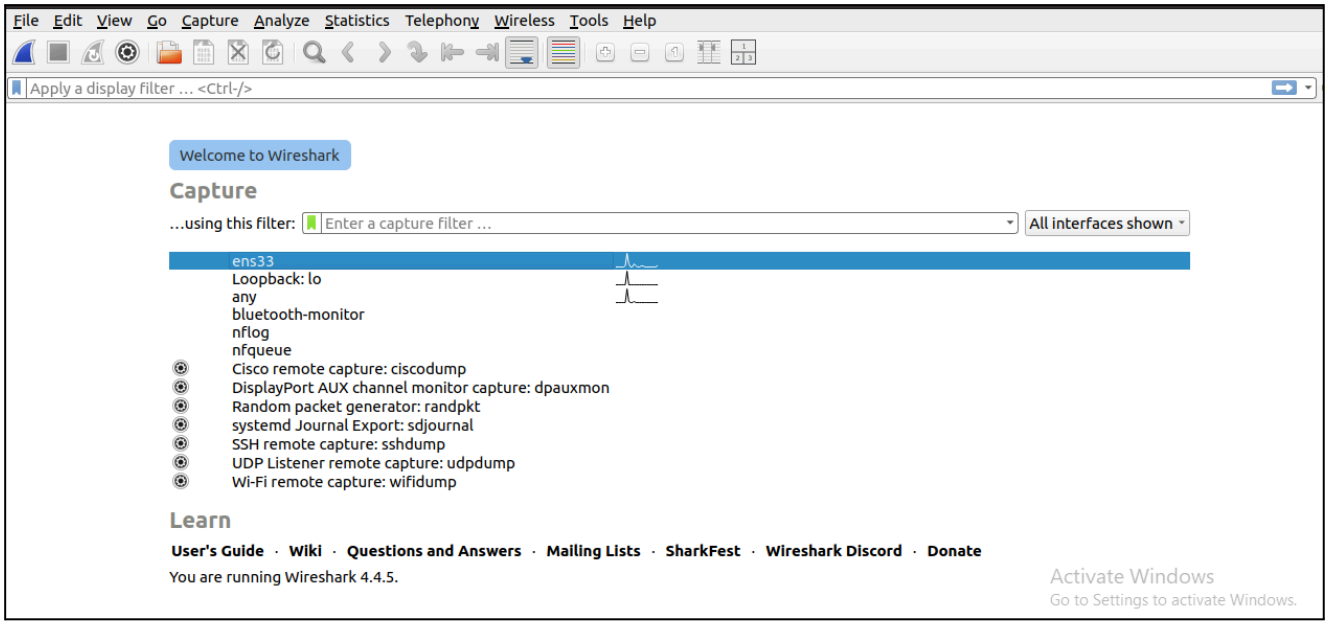
Practical No - 11

Aim: Exercises for analyzing network protocols using Wireshark:

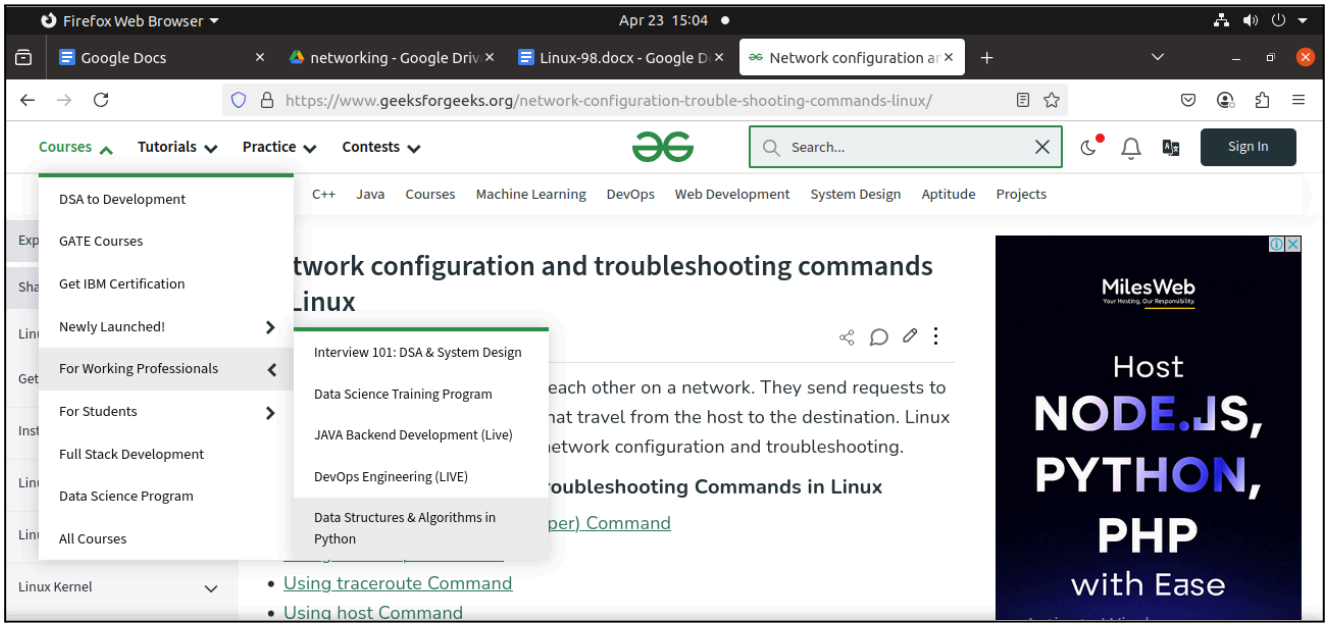
- Capture the packets while browsing any website
- Analyze the header fields of various protocols

Code:

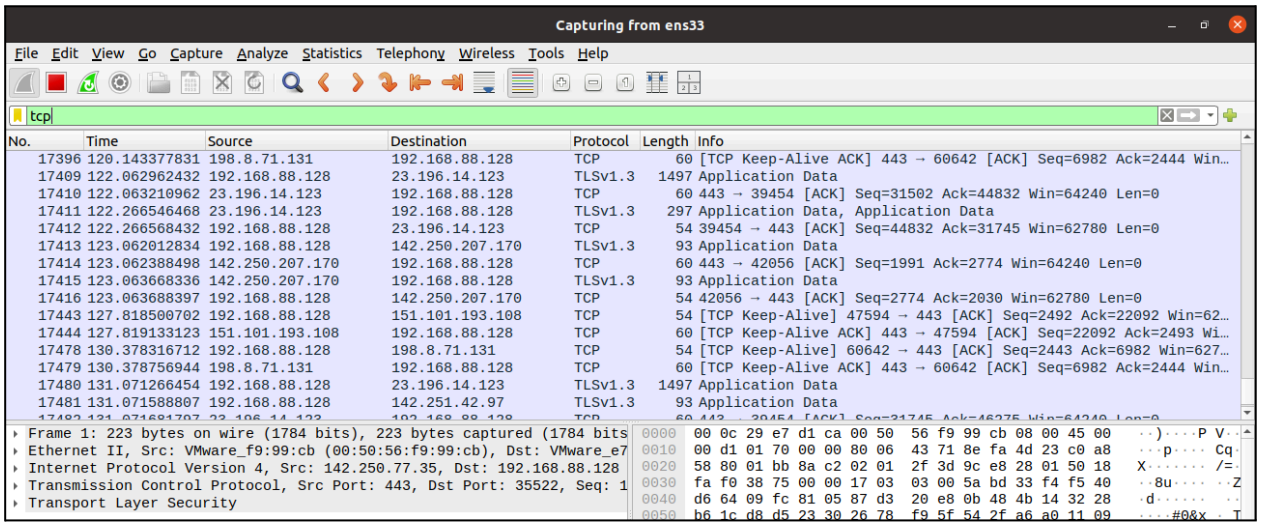
Start Wireshark



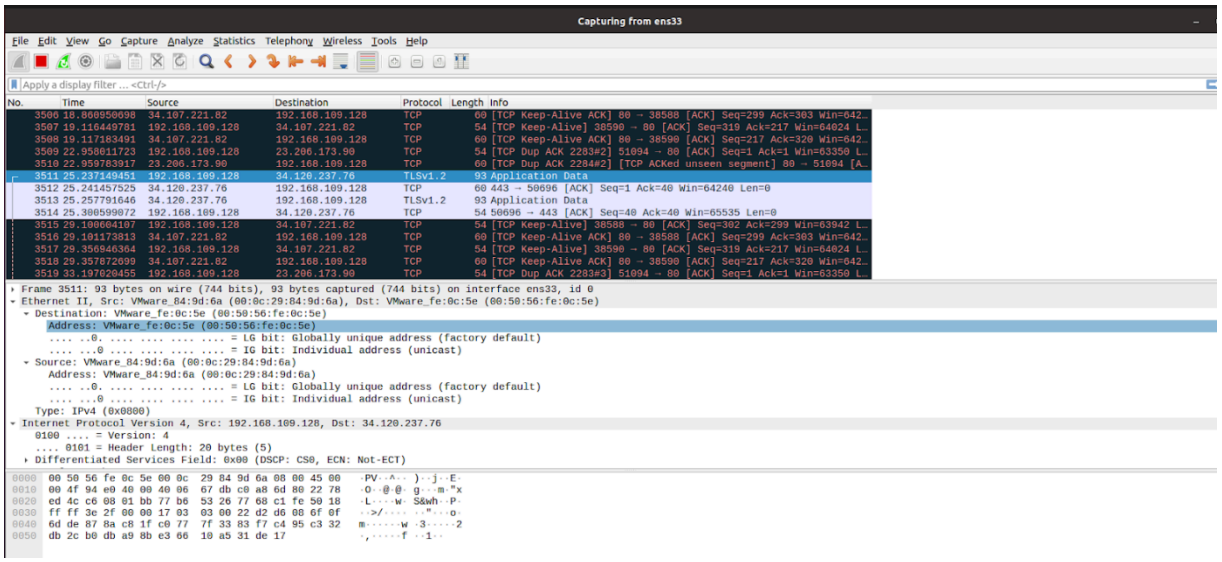
https://www.geeksforgeeks.com/



TCP



Header (Ethernet II)



Statistics (I/O Graph)