



Bharati Vidyapeeth's  
**Institute of Management & Information Technology**  
C.B.D. Belapur, Navi Mumbai – 400614

**Vision:**

Providing high quality, innovative and value-based education in information technology to build competent professionals.

**Mission**

- M1. Technical Skills: To provide solid technical foundation theoretically as well as practically capable of providing quality services to industry.
- M2. Development: -Department caters to the needs of students through comprehensive educational programs and promotes lifelong learning in the field of computer Applications.
- M3. Ethical leadership: Department develops ethical leadership insight in the students to succeed in industry, government and academia.

**CERTIFICATE**

This is to certify that the journal is the work of **Ms. Shravani Mahesh Swami** Roll No. **60** of MCA (Sem-2 Div: A) for the academic year 2022 - 2024

Subject Code: **MCAL26**

Subject Name: **Networking with Linux Lab**

Subject-in-charge

Date: 16/06/2023

Principal

External Examiner

# **INDEX**

## **INDEX**

Sr No.	TOPIC	DATE	SIGN
1	Installation of NS-3 in Linux		
2	Installation and configuration of NetAnim		
3	Installation of Wireshark		
4	Program to simulate traffic between two nodes		
5	Animate traffic between two nodes in NetAnim and analyse pcap files using wireshark		
6	Program to implement a Bus topology.		
7	Animate Bus Topology in NetAnim.		
8	Analyse BUS topology simulation pcap files using wireshark		
9	Program to simulate star topology		
10	Animate Star Topology in NetAnim		
11	Program to simulate Mesh topology		
12	Program to simulate Hybrid topology		
13	Program to simulate UDP client-server		
14	Program to simulate DHCP server and n clients		
15	Program to simulate FTP using TCP protocol		
16	Program to simulate Three way handshake for TCP connection.		
17	Analyze the network traffic using Wireshark		
18	Mini Project – Dynamic Global Routing		

## Practical 1 Installation of NS3 in Linux

### 1. sudo apt upgrade

```
bvimit@bvimit:~$ sudo apt upgrade
[sudo] password for bvimit:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libfwupdplugin1
  libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  libfwupdplugin5 libopengl0
The following packages will be upgraded:
  alsu-ucm-conf apport apport-gtk bash bind9-dnsutils bind9-host bind9-libs
  bolt command-not-found firefox fonts-opensymbol fwupd fwupd-signed
  gir1.2-gtk-3.0 gir1.2-javascriptcoregtk-4.0 gir1.2-polkit-1.0
  gir1.2-webkit2-4.0 gtk-update-icon-cache gzip klibc-utils libarchive13
  libc-bin libdrm-amdgpu1 libdrm-common libdrm-intel1 libdrm-nouveau2
  libdrm-radeon1 libdrm2 libegl-mesa0 libegl1 libevdev2 libexpat1
  libfprint-2-2 libfprint-2-tod1 libfribidi0 libfwupd2 libgbm1 libgl1
  libgl1-mesa-dri libglapi-mesa libgles2 libglvnd0 libglx-mesa0 libglx0
  libgtk-3-0 libgtk-3-bin libgtk-3-common libinput-bin libinput10
  libjavascriptcoregtk-4.0-18 libjcat1 libjuh-java libjurt-java libklibc
  liblvm12 liblzma5 libnetplan0 libnss-systemd libpam-systemd
```

### 2. sudo apt update

```
bvimit@bvimit:~$ sudo apt update
Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
```

### 3. Minimal requirements for C++ users

```
sudo apt-get install g++ python3
```

```
bvimit@bvimit:~$ sudo apt-get install g++ python3
Reading package lists... Done
Building dependency tree
Reading state information... Done
g++ is already the newest version (4:9.3.0-1ubuntu2).
g++ set to manually installed.
python3 is already the newest version (3.8.2-0ubuntu2).
python3 set to manually installed.
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libfwupdplugin1 libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

```
bvmit@bvmit:~$ sudo apt install python3-gi python3-gi-cairo python3-pygraphviz python3-gi python3-gi-cairo
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-gi is already the newest version (3.36.0-1).
python3-gi set to manually installed.
python3-gi-cairo is already the newest version (3.36.0-1).
python3-gi-cairo set to manually installed.
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libfwupdplugin1 libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  graphviz libbanno libcdt5 libgraph6 libgts-0.7-5 libgts-bin libgvc6 libgvpr2 liblab-gamut1 libpathplan4
Suggested packages:
  gsfonts graphviz-doc python-pygraphviz-doc
The following NEW packages will be installed:
  graphviz libbanno libcdt5 libgraph6 libgts-0.7-5 libgts-bin libgvc6 libgvpr2 liblab-gamut1 libpathplan4 python3-pygraphviz
0 upgraded, 11 newly installed, 0 to remove and 0 not upgraded.
Need to get 1,952 kB of archives.
After this operation, 9,624 kB of additional disk space will be used.
Do you want to continue? [Y/n] ■
```

#### 4. Minimal requirements for Python API users

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

```
bvmit@bvmit:~$ sudo apt-get install g++ python3 python3-dev pkg-config sqlite3
Reading package lists... Done
Building dependency tree
Reading state information... Done
g++ is already the newest version (4:9.3.0-1ubuntu2).
pkg-config is already the newest version (0.29.1-0ubuntu4).
pkg-config set to manually installed.
python3 is already the newest version (3.8.2-0ubuntu2).
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libfwupdplugin1 libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libexpat1-dev libpython3-dev libpython3.8-dev python3-distutils python3.8-dev zlib1g-dev
Suggested packages:
  sqlite3-doc
The following NEW packages will be installed:
  libexpat1-dev libpython3-dev libpython3.8-dev python3-dev python3-distutils python3.8-dev sqlite3 zlib1g-dev
0 upgraded, 8 newly installed, 0 to remove and 0 not upgraded.
Need to get 5,748 kB of archives.
After this operation, 26.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y ■
```

#### 5. Netanim animator: qt5 development tools are needed for Netanim animator

**sudo apt-get install qt5-default mercurial**

```
bvmit@bvmit:~$ sudo apt-get install qt5-default mercurial
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libfwupdplugin1 libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libdouble-conversion3 libegl-dev libgl1-dev libglu1-mesa-dev libglx-dev libpcre2-16-0 libpthread-stubs0-dev libpython2.7-stub libpython2.7-min libpython2.7-stub libqt5concurrent5 libqt5core5a libqt5dbus libqt5gui5 libqt5network5 libqt5opengl5 libqt5opengl5-dev libqt5printsupport5 libqt5sql5-sqlite libqt5svg5 libqt5test5 libqt5widgets5 libqt5xml5 libvulkan-dev libxi1-dev libxau-dev libxcb-xinerama0 libxcb-xinput0 libxcb-libext-dev mercurial-common python2 python2-minimal python2.7 python2.7-minimal qt5-gtk-platformtheme qt5-qmake qtbase5-dev qtchooser qttranslations5-l10n xiproto-core-dev xiproto-dev xiproto-ext-dev xorg-sgml-doctools xtrans-dev
Suggested packages:
  qt5-image-formats-plugins qtwayland5 libxi1-doc libxcb-doc libxext-doc kdiff3 | kompare | meld | tkcvs | mgdiff qct python-mysqld python-pymgments wish python2-doc python-tk python2.7-doc binfmt-support default libmysqlclient-dev firebird-dev libpq-dev libsqlite3-dev unix
The following NEW packages will be installed:
  libdouble-conversion3 libegl-dev libgl1-dev libglu1-mesa-dev libglx-dev libpcre2-16-0 libpthread-stubs0-dev libpython2.7-stub libpython2.7-min libpython2.7-stub libqt5concurrent5 libqt5core5a libqt5dbus libqt5gui5 libqt5network5 libqt5opengl5 libqt5opengl5-dev libqt5printsupport5 libqt5sql5-sqlite libqt5svg5 libqt5test5 libqt5widgets5 libqt5xml5 libvulkan-dev libxi1-dev libxau-dev libxcb-xinerama0 libxcb-xinput0 libxcb-libext-dev mercurial-common python2 python2-minimal python2.7 python2.7-minimal qt5-gtk-platformtheme qt5-qmake qtbase5-dev-tools qtchooser qttranslations5-l10n xiproto-core-dev xiproto-dev xiproto-ext-dev xorg-sgml-doctools xtrans-dev
0 upgraded, 51 newly installed, 0 to remove and 0 not upgraded.
Need to get 23.0 MB of archives.
After this operation, 117 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

#### 6. ns-3-pyviz visualizer

**sudo apt-get install gir1.2-goocanvas-2.0**

**sudo apt-get install python3-gi python3-gi-cairo python3-pygraphviz python3-gi python3-gi-cairo**

**sudo apt-get install python3-pygraphviz gir1.2-gtk-3.0 ipython3 ipython3**

```
bvinit@bvinit:~$ sudo apt-get install python3-pygraphviz gir1.2-gtk-3.0 ipython3 ipython
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-pygraphviz is already the newest version (1.5-4build1).
gir1.2-gtk-3.0 is already the newest version (3.24.20-0ubuntu1.1).
gir1.2-gtk-3.0 set to manually installed.
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libfwupdplugin1 libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  python3-backcall python3-decorator python3-ipython python3-ipython-genutils python3-jedi python3-parso python3-pickleshare python3-prompt
  python3-traitlets python3-wcwidth
Suggested packages:
  python-ipython-doc python-pymgments-doc ttf-bitstream-vera
The following NEW packages will be installed:
  ipython3 python3-backcall python3-decorator python3-ipython python3-ipython-genutils python3-jedi python3-parso python3-pickleshare python3-pygments
  python3-traitlets python3-wcwidth
0 upgraded, 12 newly installed, 0 to remove and 0 not upgraded.
Need to get 1,990 kB of archives.
After this operation, 12.1 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

## 7. Debugging:

**sudo apt-get install gdb valgrind**

```
bvinit@bvinit:~$ sudo apt-get install gdb valgrind
Reading package lists... Done
Building dependency tree
Reading state information... Done
gdb is already the newest version (9.2-0ubuntu1~20.04.1).
gdb set to manually installed.
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libfwupdplugin1 libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
Suggested packages:
  valgrind-dbg valgrind-mpi kcachegrind alleypo valkyrie
The following NEW packages will be installed:
  valgrind
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 20.3 MB of archives.
After this operation, 90.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 valgrind amd64 1:3.15.0-1ubuntu9.1 [20.3 MB]
17% [1 valgrind 4,323 kB/20.3 MB 21%]
```

## 8. Doxygen and related inline documentation:

**sudo apt-get install doxygen graphviz imagemagick**

```
bvinit@bvinit:~$ sudo apt-get install doxygen graphviz imagemagick
Reading package lists... Done
Building dependency tree
Reading state information... Done
graphviz is already the newest version (2.42.2-3build2).
graphviz set to manually installed.
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libfwupdplugin1 libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  gsfonts imagemagick-6-common imagemagick-6.q16 libclang1-10 libilmbase24 libl LLVM10 liblqr-1-0 libmagickcore-6.
  libmagickwand-6.q16-6 libnetpbm10 libopenexr24 libxapian30 netpbm
Suggested packages:
  doxygen-latex doxygen-doc doxygen-gui imagemagick-doc autotrace curl enscript ffmpeg gimp gnuplot grads hp2xx
  texlive-base-bin transfig ufraw-batch inkscape libjxr-tools xaptan-tools
The following NEW packages will be installed:
  doxygen gsfonts imagemagick imagemagick-6-common imagemagick-6.q16 libclang1-10 libilmbase24 libl LLVM10 liblqr-
  libmagickwand-6.q16-6 libnetpbm10 libopenexr24 libxapian30 netpbm
0 upgraded, 16 newly installed, 0 to remove and 0 not upgraded.
Need to get 40.6 MB of archives.
After this operation, 177 MB of additional disk space will be used.
Do you want to continue? [Y/n] ■
```

**sudo apt-get install texlive texlive-extra-utils texlive-latex-extra texlive-font-utils dvipng latexmk**

```

bvmit@bvmit:~$ sudo apt-get install texlive texlive-extra-utils texlive-latex-extra texlive-font-utils dvipng latexmk
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libfwupdplugin1 libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  dvipsvg fonts-lmodern fonts-texgyre javascript-common libalgorithm-c3-perl libapache-pom-java libbb-hooks-endoscope-perl libclass-c3-perl libclass-c3-xs-perl libclass-data-inheritable-perl libclass-method-modifiers-perl libclass-xsaccessor-perl libcommons-parent-java libdata-optlist-perl libdevel-callchecker-perl libdevel-caller-perl libdevel-globaldestruction-perl libdevel-le libdevel-stacktrace-perl libdist-checkconflicts-perl libdynaloader-functions-perl libemail-date-format-perl libeval-closure-perl libexecfile-perl libfile-which-perl libfontbox-java libipc-shareable-perl libjs-jquery liblog-dispatch-perl liblog-log4perl-perl libmime charset-perl libmime-lite-perl libmime-types-perl libmodule-implementation-perl libmodule-runtime-perl libmro-compat-perl libnamespace-clean-perl libpackage-stash-perl libpackage-stash-xs-perl libpadwalker-perl libparams-classify-perl libparams-util-perl libparams-validationcompiler-perl libpdfbox-java libptexenc1 libreadonly-perl libref-util-xs-perl librole-tiny-perl libspecto-perl libsub-exporter-perl libsub-exporter-progressive-perl libsub-identify-perl libsub-install-perl libsub-name-perl libsub-libsys-hostname-long-perl libtcl8.6 libteckito libtexlua53 libtexluajit2 libtk8.6 libunicode-linebreak-perl libvariable-magic-perl libyaml-tiny-perl libzzip-0.13 lmodern preview-latex-style ps2eps rake ruby ruby-minitest ruby-net-telnet ruby-power-assert ruby-test-rubygems-integration t1utils tcl tc18.6 tex-common tex-gyre texlive-base texlive-binaries texlive-fonts-recommended texlive-latex-base texlive-pictures texlive-plain-generic tipa tk tk8.6
Suggested packages:
  apache2 | lighttpd | httpd libavalon-framework-java libcommons-logging-java-doc libcalibr-logkit-java liblog4j1.2-java libdbd-csv-perl libblog-dispatch-filerotate-perl librds-perl libxml-dom-perl libencode-hanextra-perl libpod2-base-perl default-mta | mail-transport-agent libscalar-number-perl libtest-fatal-perl ri ruby-dev bundler tcl-tclreadline debhelper perl-tk xzdec chktex dvidvt fragmaster lacheck lcdf-type tools psutils texlive-fonts-recommended-doc texlive-latex-base-doc icc-profiles libspreadsheet-parseexcel-perl texlive-latex-texlive-latex-recommended-doc texlive-luatex texlive-nstricks dot2tex prrex ruhy-tcltk | libtcltk-ruhy texlive-pictures-doc vorrex d

```

9. The ns-3 manual and tutorial are written in reStructuredText for Sphinx (doc/tutorial, doc/manual, doc/models), and figures typically in dia (also needs the texlive packages above):

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

```

bvmit@bvmit:~$ sudo apt-get install python3-sphinx dia
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libfwupdplugin1 libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  dia-common dia-shapes docutils-common gsfonts-x11 libart-2.0-2 libjs-sphinxdoc libjs-uglifyjs python3-docutils python3-imaging-size python3-jinja2 python3-packaging python3-pyparsing
Suggested packages:
  docutils-doc fonts-linux-libertine | ttf-linux-libertine texlive-lang-french python-jinja2 python3-stemmer sphinx-doc
The following NEW packages will be installed:
  dia dia-common dia-shapes docutils-common gsfonts-x11 libart-2.0-2 libjs-sphinxdoc libjs-uglifyjs python3-docutils python3-imaging-size python3-jinja2 python3-packaging python3-pyparsing
0 upgraded, 19 newly installed, 0 to remove and 0 not upgraded.
Need to get 14.9 MB of archives.
After this operation, 68.4 MB of additional disk space will be used.
Do you want to continue? [Y/n]

```

10. To read pcap packet traces:

**sudo apt-get install tcpdump**

```
bvimit@bvimit:~$ sudo apt-get install tcpdump
Reading package lists... Done
Building dependency tree
Reading state information... Done
tcpdump is already the newest version (4.9.3-4ubuntu0.1).
tcpdump set to manually installed.
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libfwupdplugin1 libgstreamer-pl
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Support for generating modified python bindings:

```
sudo apt-get install cmake libc6-dev libc6-dev-i386 libclang- 6.0-dev llvm-6.0-dev automake python3-pip
```

```
bvimit@bvimit:~$ sudo apt-get install cmake libc6-dev libc6-dev-i386 libclang-6.0-dev llvm-6.0-dev automake python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
libc6-dev is already the newest version (2.31-0ubuntu9.7).
libc6-dev set to manually installed.
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libfwupdplugin1 libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  autoconf autotools-dev binfmt-support cmake-data gcc-9-multilib gcc-multilib lib32asan5 lib32atomic1 lib32gcc-9-dev
  lib32stdc++6 lib32ubsan1 libc6-dev-x32 libc6-i386 libc6-x32 libclang-common-6.0-dev libclang1-6.0 libffi-dev libjso
  libobjc4 librhash0 libsigsegv2 libtinfo-dev libx32asan5 libx32atomic1 libx32gcc-9-dev libx32gcc-s1 libx32omp1 libx
  libx32ubsan1 llvm-6.0 llvm-6.0-runtime m4 python-pip-whl python3-setuptools python3-wheel
Suggested packages:
  autoconf-archive gnu-standards autoconf-doc libtool gettext cmake-doc ninja-build ncurses-doc m4-doc python-setupto
The following NEW packages will be installed:
  autoconf automake autotools-dev binfmt-support cmake cmake-data gcc-9-multilib gcc-multilib lib32asan5 lib32atomic1
  lib32quadmath0 lib32stdc++6 lib32ubsan1 libc6-dev-i386 libc6-dev-x32 libc6-i386 libc6-x32 libclang-6.0-dev libclang
  libjsoncpp1 libllvm6.0 libncurses-dev libobjjc-9-dev libobjc4 librhash0 libsigsegv2 libtinfo-dev libx32asan5 libx32a
  libx32itm1 libx32quadmath0 libx32stdc++6 libx32ubsan1 llvm-6.0 llvm-6.0-dev llvm-6.0-runtime m4 python-pip-whl pyth
0 upgraded, 50 newly installed, 0 to remove and 0 not upgraded.
Need to get 107 MB of archives.
After this operation, 780 MB of additional disk space will be used.
```

```
python3 -m pip install --user cxxfilt
```

```
bvimit@bvimit:~$ python3 -m pip install --user cxxfilt
Collecting cxxfilt
  Downloading cxxfilt-0.3.0-py2.py3-none-any.whl (4.6 kB)
Installing collected packages: cxxfilt
Successfully installed cxxfilt-0.3.0
bvimit@bvimit:~$
```

After installing the required packages, create a folder named workspace in the home directory and then put the NS3 tar package into the workspace.

Go to terminal and input these commands consecutively after each command finishes executing: ( go to the folder workspace where ns3 is installed)

```
cd workspace
```

`tarxjf<name of NS3 downloaded file name>` (to unzip the file) otherwise you can unzip by right clicking and selecting explore option)

```
cd<name of extracted NS3>
```

```
// go to the ns3allinone folder
```

bvimit@bvimit:~/workspace/ns-allinone-3.32\$

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

```
bvimit@bvimit:~/workspace/ns-allinone-3.32$ ./build.py --enable-examples --enable-tests
# Build NetAnim
Entering directory `netanim-3.108'
  => qmake -v
QMake version 3.1
Using Qt version 5.12.8 in /usr/lib/x86_64-linux-gnu
qmake found
  => qmake NetAnim.pro
Info: creating stash file /home/bvimit/workspace/ns-allinone-3.32/netanim-3.108/.qmake.stash
  => make
g++ -c -pipe -O2 -Wall -W -D_REENTRANT -fPIC -DNS3_LOG_ENABLE -DQT_NO_DEBUG -DQT_PRINTSUPPORT_LIB -DQT_WIDGETS_LIB -DQT_GUI_LIB -DQT_CORE_LIB -I. -Iqtpropertybrowser/src -isystem /usr/include/x86_64-linux-gnu/qt5/QtPrintSupport -isystem /usr/include/x86_64-linux-gnu/qt5/QtWidgets -isystem /usr/include/x86_64-linux-gnu/qt5/QtGui -isystem /usr/include/x86_64-linux-gnu/qt5/QtCore -I. -I/usr/lib/x86_64-linux-gnu/qt5/mkspecs/linux-g++-main.o main.cpp
g++ -c -pipe -O2 -Wall -W -D_REENTRANT -fPIC -DNS3_LOG_ENABLE -DQT_NO_DEBUG -DQT_PRINTSUPPORT_LIB -DQT_WIDGETS_LIB -DQT_GUI_LIB -DQT_CORE_LIB -I. -Iqtpropertybrowser/src -isystem /usr/include/x86_64-linux-gnu/qt5/QtPrintSupport -isystem /usr/include/x86_64-linux-gnu/qt5/QtWidgets -isystem /usr/include/x86_64-linux-gnu/qt5/QtGui -isystem /usr/include/x86_64-linux-gnu/qt5/QtCore -I. -I/usr/lib/x86_64-linux-gnu/qt5/mkspecs/linux-g++-main.o main.cpp
(It takes time)
```

Test the NS3 build and installation success by running test.py in the ns directory using the following commands:

```
cd ns-<version number> //go to the folder // ns3 folder in ns3allinone bvimit@bvimit:~/workspace/ns-allinone-3.32/ns-3.32$
```

then type,

./test.py

```
bvimit@bvimit:~/Workspace/ns-allinone-3.32/ns-3.32$ ./test.py
Waf: Entering directory `/home/bvimit/Workspace/ns-allinone-3.32/ns-3.32/build'
Waf: Leaving directory `/home/bvimit/Workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (10.047s)

Modules built:
antenna           aodv                  applications
bridge            buildings             config-store
core              csma                 csma-layout
dsdv              dsr                  energy
fd-net-device     flow-monitor         internet
internet-apps    lr-wpan              lte
mesh               mobility             netanim
network           nix-vector-routing   olsr
point-to-point    point-to-point-layout propagation
sixlowpan          spectrum            stats
tap-bridge         test (no Python)   topology-read
traffic-control   uan                 virtual-net-device
visualizer        wave                wifi
wimax

Modules not built (see ns-3 tutorial for explanation):
brite              click               dpdk-net-device
```

\$] ./waf --run hello-simulator

This will display "Hello Simulator" Which indicates that ns3 is installed successfully.

```
bvimit@bvimit:~/Workspace/ns-allinone-3.32/ns-3.32$ ./waf --run hello-simulator
Waf: Entering directory `/home/bvimit/Workspace/ns-allinone-3.32/ns-3.32/build'
Waf: Leaving directory `/home/bvimit/Workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (4.136s)
Hello Simulator
bvimit@bvimit:~/Workspace/ns-allinone-3.32/ns-3.32$
```

## Practical 2

### Installation of NetAnim

Step1: Open terminal and type sudo apt upgrade .

```
kirtee@kirtee-VirtualBox:~$ sudo apt upgrade
[sudo] password for kirtee:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following packages will be upgraded:
  base-files evolution-data-server evolution-data-server-common
  firmware-sof-signed fonts-opensymbol gdm3 gir1.2-gdm-1.0
  gir1.2-gnomedesktop-3.0 gir1.2-gtk-4.0 gjs gnome-desktop3-data
  gnome-settings-daemon gnome-settings-daemon-common libcamel-1.2-63
  libebackend-1.2-10 libebook-1.2-20 libebook-contacts-1.2-3 libecal-2.0-1
  libedata-book-1.2-26 libedata-cal-2.0-1 libedataserver-1.2-26
  libedataserverui-1.2-3 libgdm1 libgjs0g libgnome-bq-4-1
```

Step 2: sudo apt update

```
kirtee@kirtee-VirtualBox:~$ sudo apt update
[sudo] password for kirtee:
Hit:1 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [185 kB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [43.6 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [11.4 kB]
Fetched 350 kB in 3s (129 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.
```

Step 3: sudo ./waf configure

```
Lab2@Lab2-B250M-D2V:~$ cd workspace
Lab2@Lab2-B250M-D2V:~/workspace$ cd ns-allinone-3.32/ns-3.32
Lab2@Lab2-B250M-D2V:~/workspace/ns-allinone-3.32/ns-3.32$ sudo ./waf configure
[sudo] password for lab2:
Setting top to : /home/lab2/workspace/ns-allinone-3.32/ns-3.32
Setting out to : /home/lab2/workspace/ns-allinone-3.32/ns-3.32/build
Checking for 'gcc' (C compiler) : /usr/bin/gcc
Checking for cc version : 9.4.0
```

Step 4: sudo ./waf build

```
lab2@lab2-B250M-D2V:~/workspace/ns-allinone-3.32/ns-3.32$ sudo ./waf build
waf: Entering directory '/home/lab2/workspace/ns-allinone-3.32/ns-3.32/build'
waf: Leaving directory '/home/lab2/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.670s)

Modules built:
antenna           aodv               applications
bridge            buildings          config-store
core              csma               csma-layout
dsdv              dsr                energy
fd-net-device     flow-monitor      internet
internet-apps    lr-wpan           lte
mesh               mobility           netanim
network           nix-vector-routing olsr
point-to-point    point-to-point-layout propagation
sixlowpan         spectrum          stats
tap-bridge        test (no Python) topology-read
traffic-control   uan               virtual-net-device
visualizer        wave               wifi
wimax
```

Step 5: cd netanim-3.108 & ls

```
lab2@lab2-B250M-D2V:~/workspace/ns-allinone-3.32/ns-3.32$ cd
lab2@lab2-B250M-D2V:~$ cd workspace/ns-allinone-3.32
lab2@lab2-B250M-D2V:~/workspace/ns-allinone-3.32$ cd netanim-3.108
lab2@lab2-B250M-D2V:~/workspace/ns-allinone-3.32/netanim-3.108$ cd scratch
bash: cd: scratch: No such file or directory
lab2@lab2-B250M-D2V:~/workspace/ns-allinone-3.32/netanim-3.108$ ls
abort.h           moc_fileedit.cpp
animatorconstants.h  moc_fileeditfactory.cpp
animator_fileopen.svg  moc_fileeditfactory.o
```

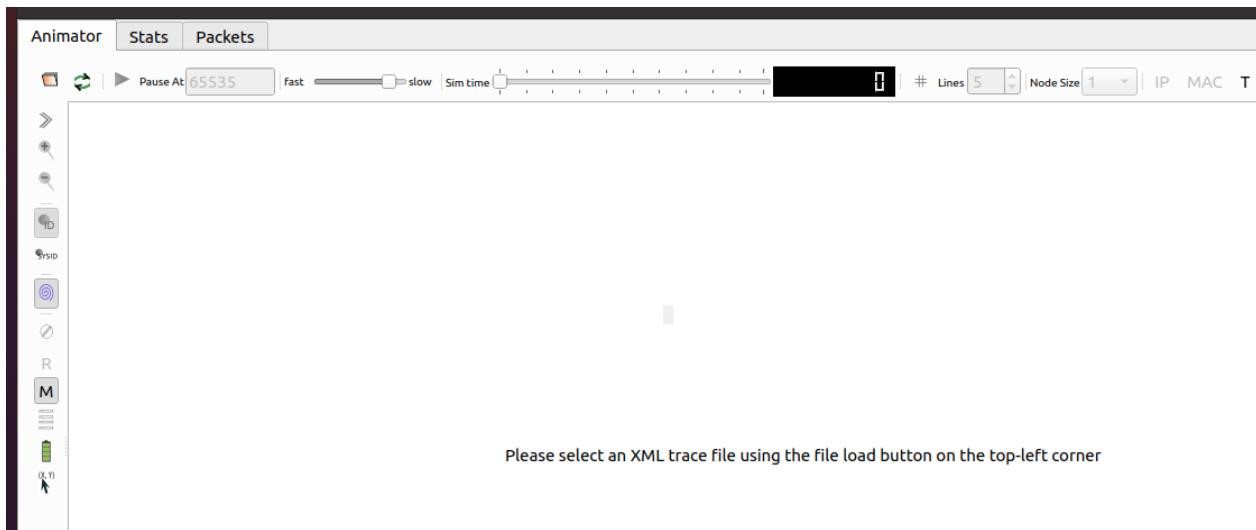
Step 6: ./NetAnim

```
lab2-114@lab2114-B250M-D2V:~/workspace/ns-allinone-3.32$ cd netanim-3.108
lab2-114@lab2114-B250M-D2V:~/workspace/ns-allinone-3.32/netanim-3.108$ ./NetAnim
```

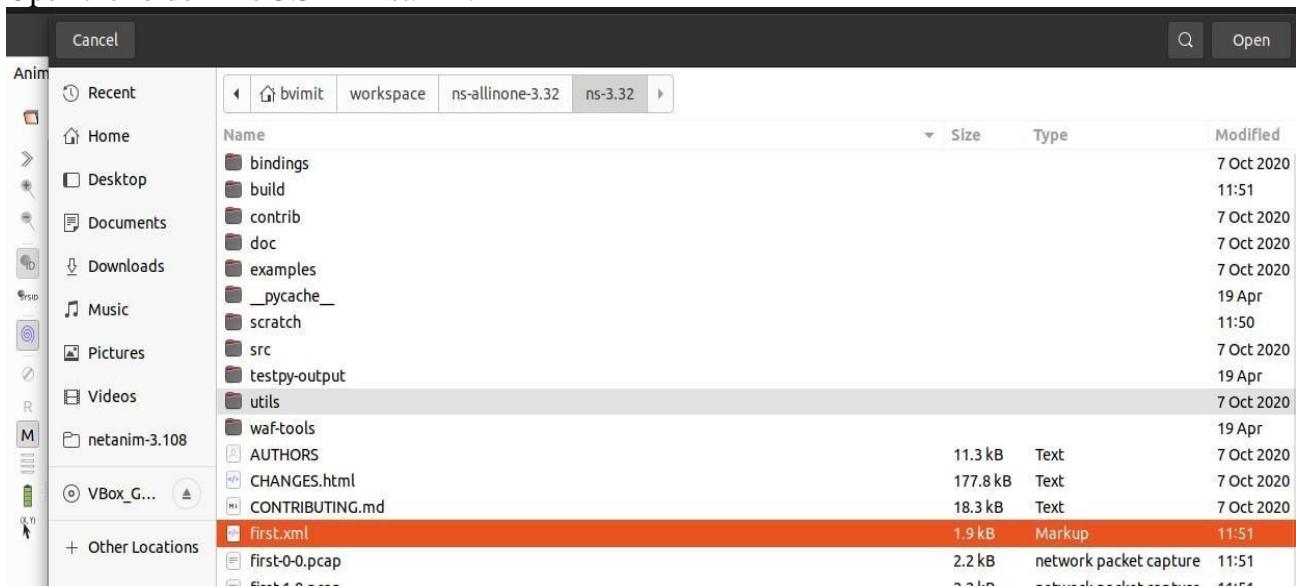
Successfully the NetAnim will be installed

Step 7:- ./NetAnim

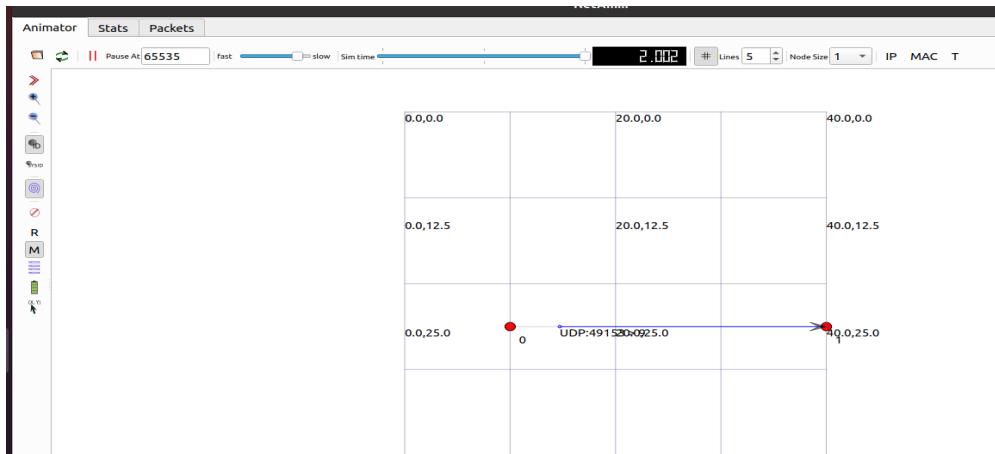
```
bvimit@bvimit:~/workspace/ns-allinone-3.32/netanim-3.108$ ./NetAnim
```



Open the folder->ns-3.32->first.xml.



Then Start the simulation



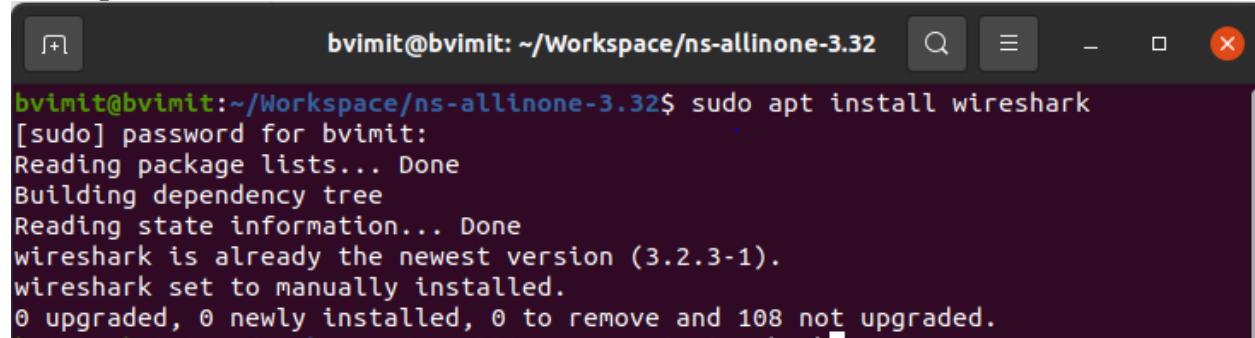
## Practical 3

### Installation of Wireshark

#### A] For Linux:

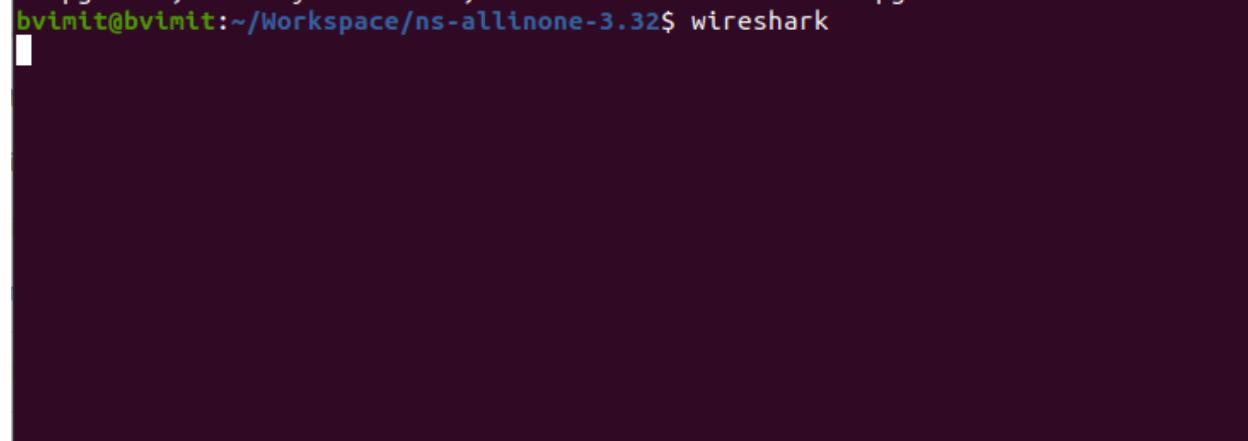
Open Terminal and type the command

**sudo apt install wireshark**

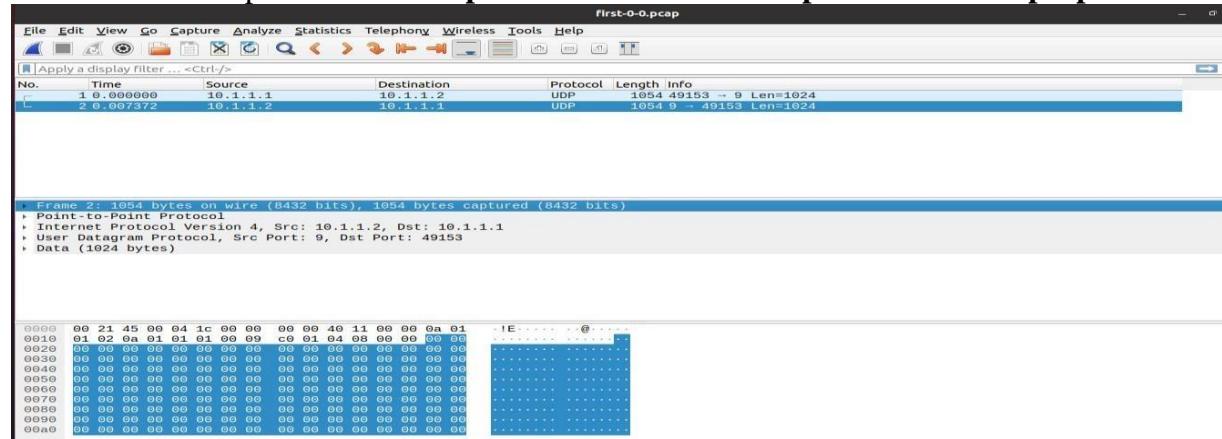


```
bvimit@bvimit:~/Workspace/ns-allinone-3.32$ sudo apt install wireshark
[sudo] password for bvimit:
Reading package lists... Done
Building dependency tree
Reading state information... Done
wireshark is already the newest version (3.2.3-1).
wireshark set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 108 not upgraded.
```

Then again type **wireshark** in terminal then directly it will open the wireshark



When wireshark open then **file -> open-> Go to ns-3.2 and open the first-0-0.pcap folder**



## B] For Windows:

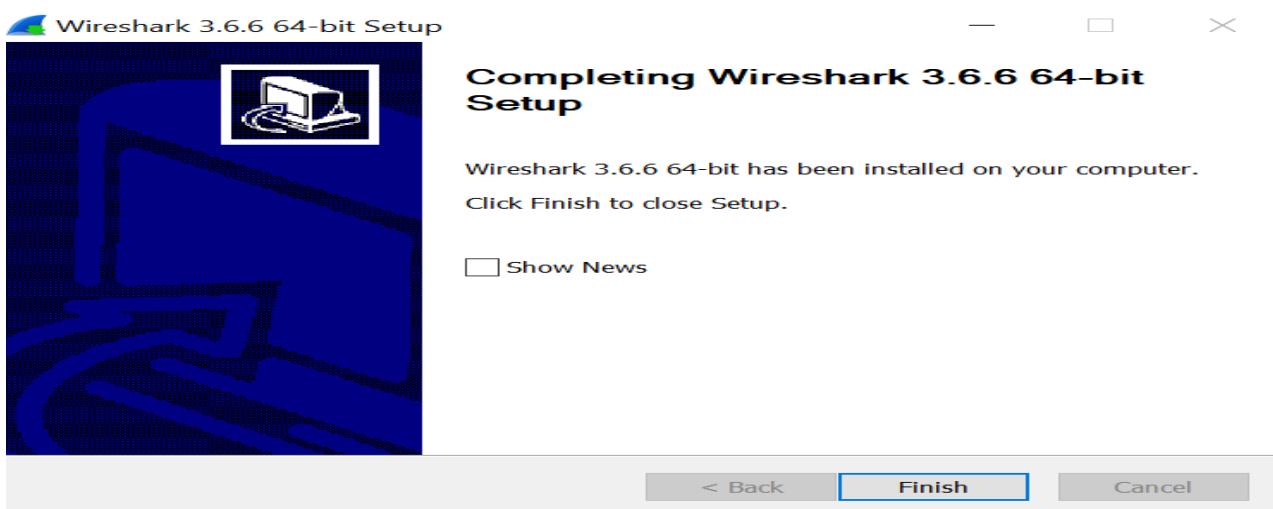
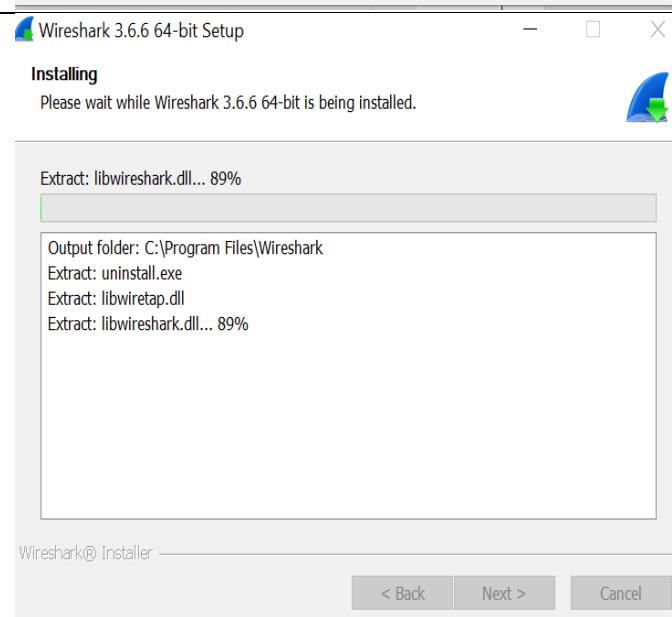
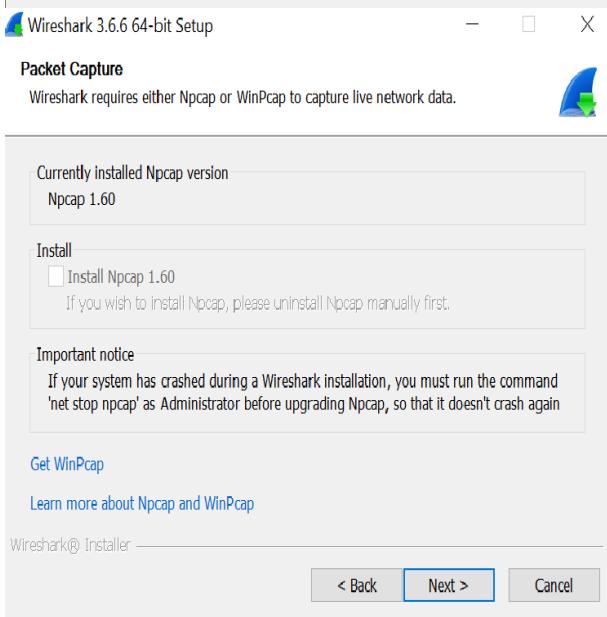
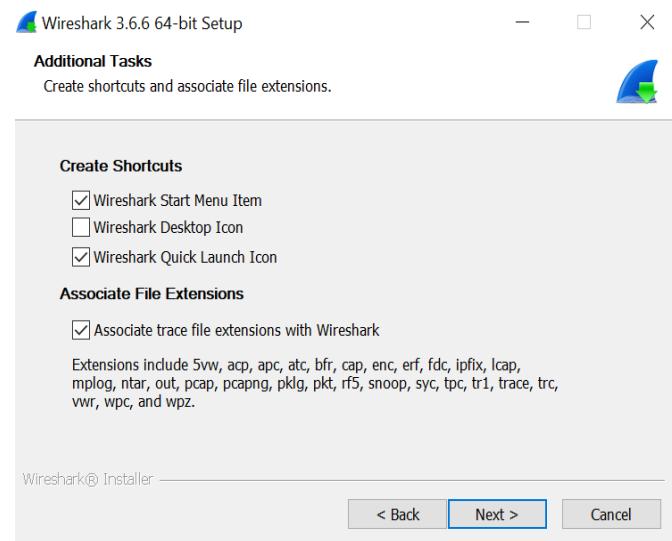
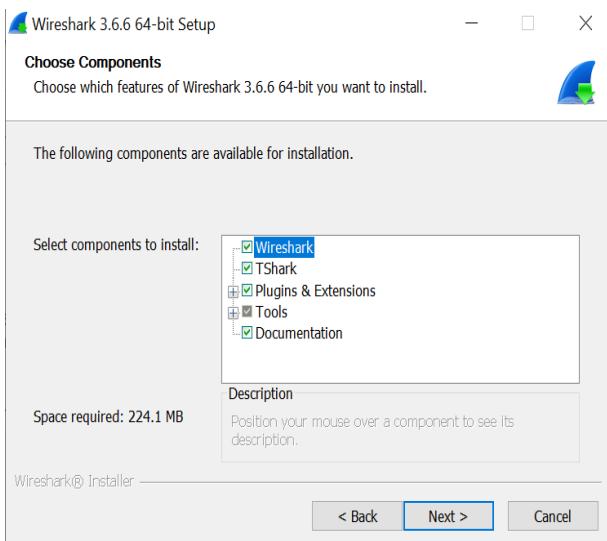
Step 1: Download the latest version of Wireshark on your windows operating system.

The screenshot shows the official Wireshark download page. At the top right, there is a "NEWS" link. Below it, the "Download Wireshark" section is visible, stating "The current stable release of Wireshark is 3.6.6. It supersedes all previous releases. You can also download the latest development release (3.7.1) and documentation." A blue header bar indicates "Stable Release (3.6.6)". Underneath, there is a list of download options:

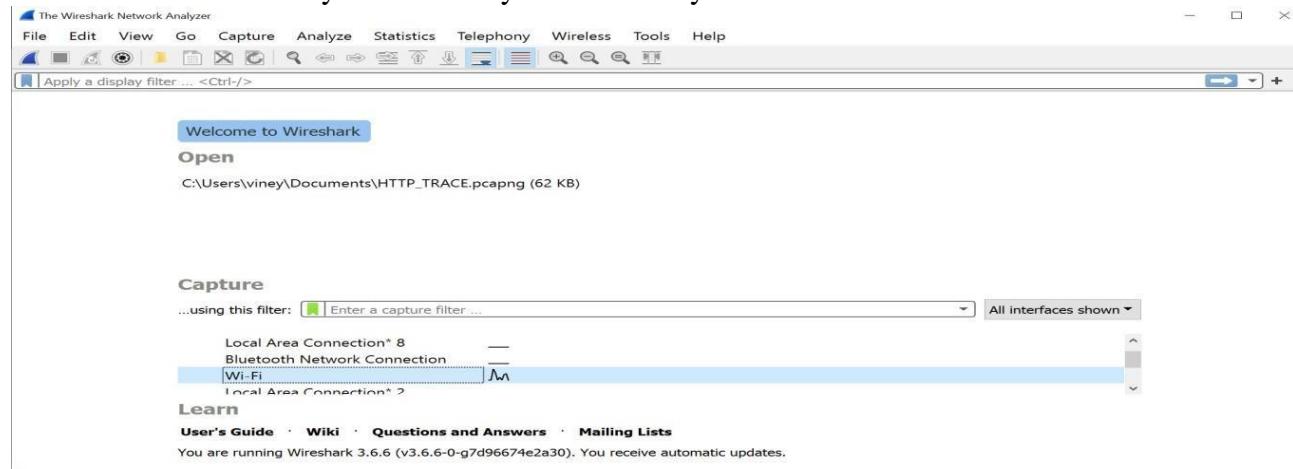
- Windows Installer (64-bit)
- Windows Installer (32-bit)
- Windows PortableApps® (64-bit)
- Windows PortableApps® (32-bit)
- macOS Arm 64-bit .dmg
- macOS Intel 64-bit .dmg
- Source Code

Step 2: Install Wireshark on your system.

The screenshot shows two windows from the Wireshark 3.6.6 64-bit Setup wizard. The left window is the "Welcome to Wireshark 3.6.6 64-bit Setup" screen, which includes a large blue background image of a shark fin, a small Wireshark logo icon, and text explaining the setup wizard's purpose, prerequisites, and the next step. The right window is the "License Agreement" screen, which displays the GNU General Public License text. It features a scrollable text area with sections for Part I, Part II, and Part III, followed by a note about the EULA being informational only. Navigation buttons like "Next >" and "Cancel" are at the bottom of both windows.



Wireshark is successfully installed on your window system.



## **Practical 4** **Program to simulate traffic between two nodes**

Using Visualization first.cc

Code:-

```
//pre defined modules
#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"

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

//The ns-3 project is implemented in a C++ namespace called ns3. This groups all ns -3-related declarations in a scope outside the global namespace, which will help with integration with other code.  
using namespace ns3;

//Enable the log for this program.This line declares a logging component called FirstScriptExample that allows you to enable and disable console message logging by reference to the name.  
NS\_LOG\_COMPONENT\_DEFINE ("FirstScriptExample");

//It is just the declaration of the main function of your program (script).  
int main (int argc, char \*argv[])
{

//This command is used to read and parse command line argument.  
CommandLine cmd ( FILE );
cmd.Parse (argc, argv); Time::SetResolution (Time::NS);

//These two lines of the script are used to enable two logging components that are built into the Echo Client and Echo Server applications:

```
LogComponentEnable ("UdpEchoClientApplication", LOG_LEVEL_INFO);
LogComponentEnable ("UdpEchoServerApplication", LOG_LEVEL_INFO);
```

//The NodeContainer topology helper provides a convenient way to create, manage and access any Node objects that we create in order to run a simulation. The first line above just declares a NodeContainer which

we call nodes. The second line calls the Create method on the nodes object and asks the container to create two nodes

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

```
// It instantiates a PointToPointHelper object on the stack.
```

```
PointToPointHelper pointToPoint;
```

```
//It tells the PointToPointHelper object to use the value “5Mbps” (five megabits per second) as the “DataRate” when it creates a PointToPointNetDevice object.
```

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

```
//It tells the PointToPointHelper to use the value “2ms” (two milliseconds) as the value of the transmission delay of every point to point channel it subsequently creates.
```

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

```
// This line declares the device container object. NetDeviceContainer devices;
```

```
//The Install method of the PointToPointHelper takes a NodeContainer as a parameter. Internally, a NetDeviceContainer is created.
```

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

```
//The InternetStackHelper is a topology helper that is to internet stacks what the PointToPointHelper is to point-to-point net devices. The Install method takes a NodeContainer as a parameter. When it is executed, it will install an Internet Stack (TCP, UDP, IP, etc.) on each of the nodes in the node container.
```

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

```
//It declares an address helper object and tell it that it should begin allocating IP addresses from the network 10.1.1.0 using the mask 255.255.255.0 to define the allocatable bits. By default the addresses allocated will start at one and increase monotonically, so the first address allocated from this base will be 10.1.1.1, followed by 10.1.1.2, etc.
```

```
Ipv4AddressHelper address;
```

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

```
//It performs the actual address assignment.
```

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

```
//Set up a UDP echo server application on one of the nodes we have previously created.
```

```
//Declares the UdpEchoServerHelper at port 9
```

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

```
//This method actually causes the underlying echo server application to be instantiated and attached to a node.
```

```

ApplicationContainer serverApps = echoServer.Install (nodes.Get (1));

//Define the start time of server. serverApps.Start (Seconds (1.0));

//Define the stop time of server. serverApps.Stop (Seconds (10.0));

//Set up a UDP echo client application on one of the nodes we have previously created.
The first two Attributes are set during construction of the UdpEchoClientHelper. We pass parameters that are
used (internally to the helper) to set the “RemoteAddress” and “RemotePort” Attributes.

UdpEchoClientHelper echoClient (interfaces.GetAddress (1), 9);

//The “MaxPackets” Attribute tells the client the maximum number of packets we allow it to send during the
simulation.

echoClient.SetAttribute ("MaxPackets", UIntegerValue (1));

//The “Interval” Attribute tells the client how long to wait between packets, and the “PacketSize” Attribute
tells the client how large its packet payloads should be. With this particular combination of Attributes, we are
telling the client to send one 1024-byte packet.

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

//This method actually causes the underlying echo client application to be instantiated.
ApplicationContainer clientApps = echoClient.Install (nodes.Get (0));

//This define the start time. clientApps.Start (Seconds (2.0));

//This define the stop time. clientApps.Stop (Seconds (10.0));

//run the simulation.
Simulator::Run ();

//destroy all of the objects that were created.
Simulator::Destroy ();
return 0;
}

```

(Put the first.cc program in scratch folder)

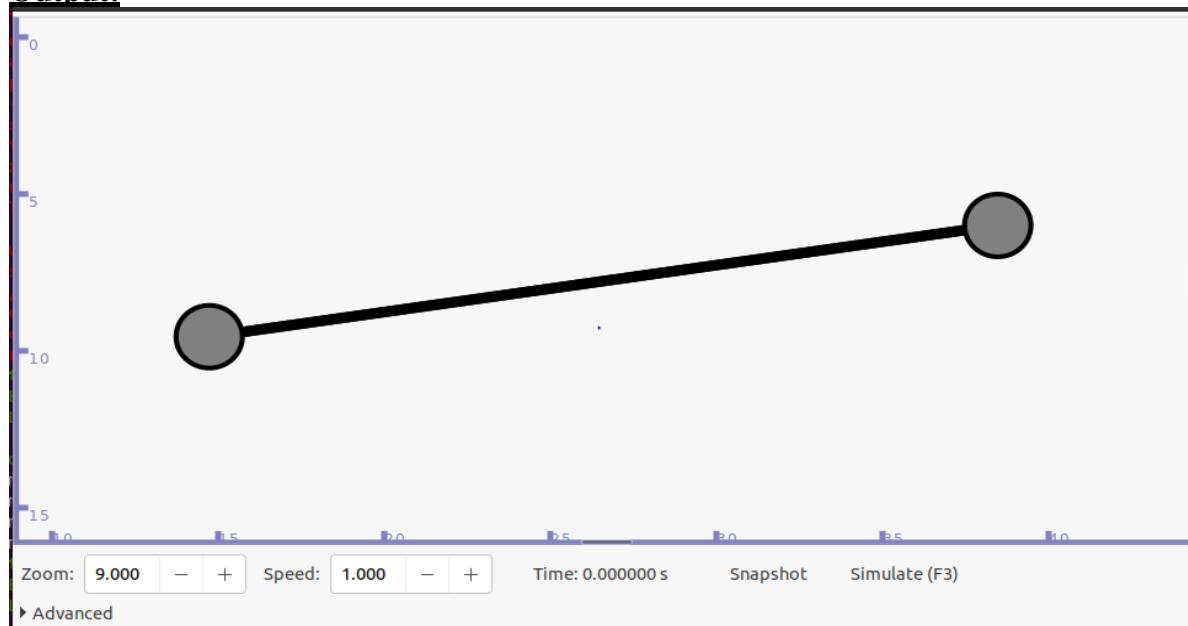
1. ./waf --run scratch/first

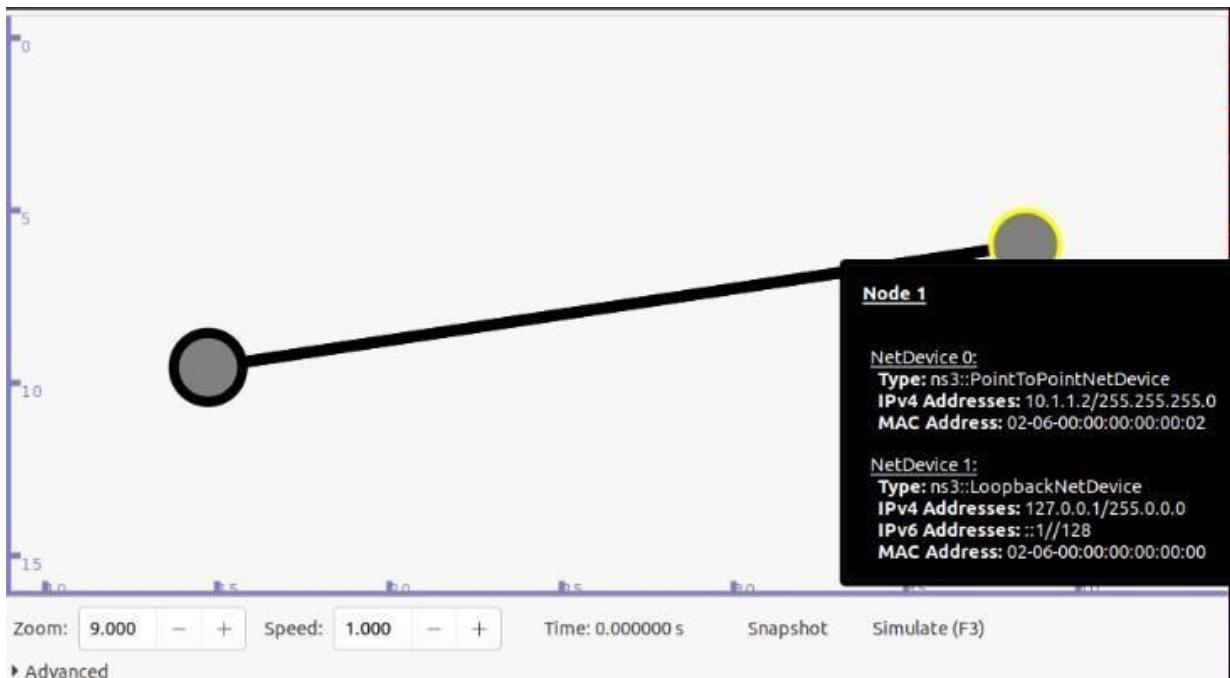
```
bvimit@bvimit-VirtualBox:~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run scratch/first
Waf: Entering directory '/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
Waf: Leaving directory '/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (2.318s)
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
```

## 2. ./waf --run scratch/first --vis

```
bvimit@bvimit-VirtualBox:~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run scratch/first --vis
Waf: Entering directory '/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
Waf: Leaving directory '/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (1.996s)
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.
```

### Output:





## Practical 5

### **Animate traffic between two nodes in NetAnim and analyse pcap files using wireshark**

**File Name:** animation.cc

**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"

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

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("FirstScriptExample");
Int main (int argc, char *argv[])
{
CommandLine cmd ( FILE );
cmd.Parse (argc, argv);
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;
}

```

## **Output:**

1. ./waf --run scratch/animation

```

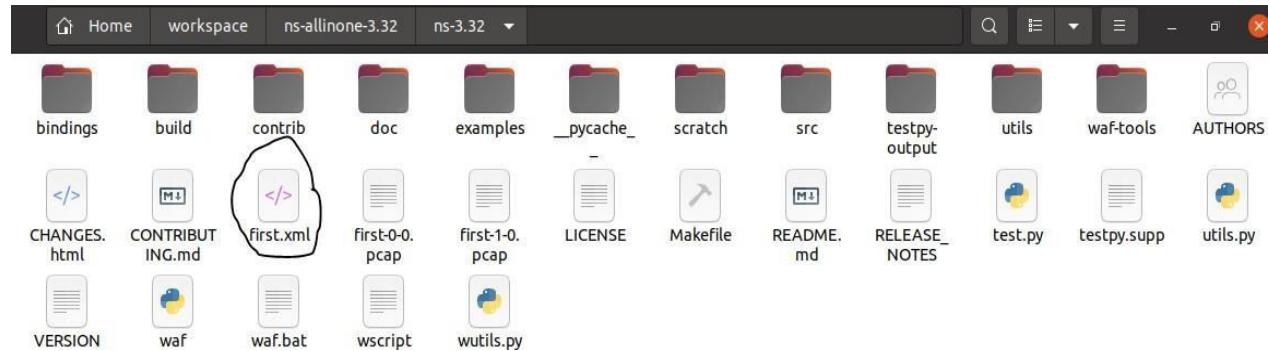
bvimit@bvimit:~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run scratch/animation
Waf: Entering directory `/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
[2830/2900] Compiling scratch/animation.cc
[2860/2900] Linking build/scratch/animation
Waf: Leaving directory `/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (7.464s)
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
bvimit@bvimit:~/workspace/ns-allinone-3.32/ns-3.32$
```

2. ls \*.xml

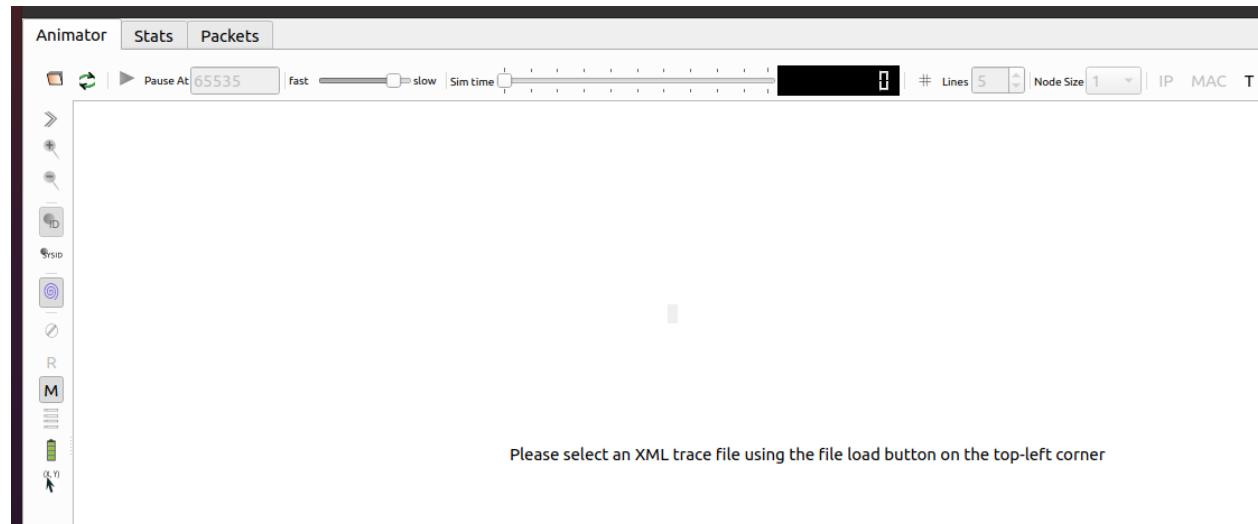
```

bvimit@bvimit:~/workspace/ns-allinone-3.32/ns-3.32$ ls *.xml
first.xml
```

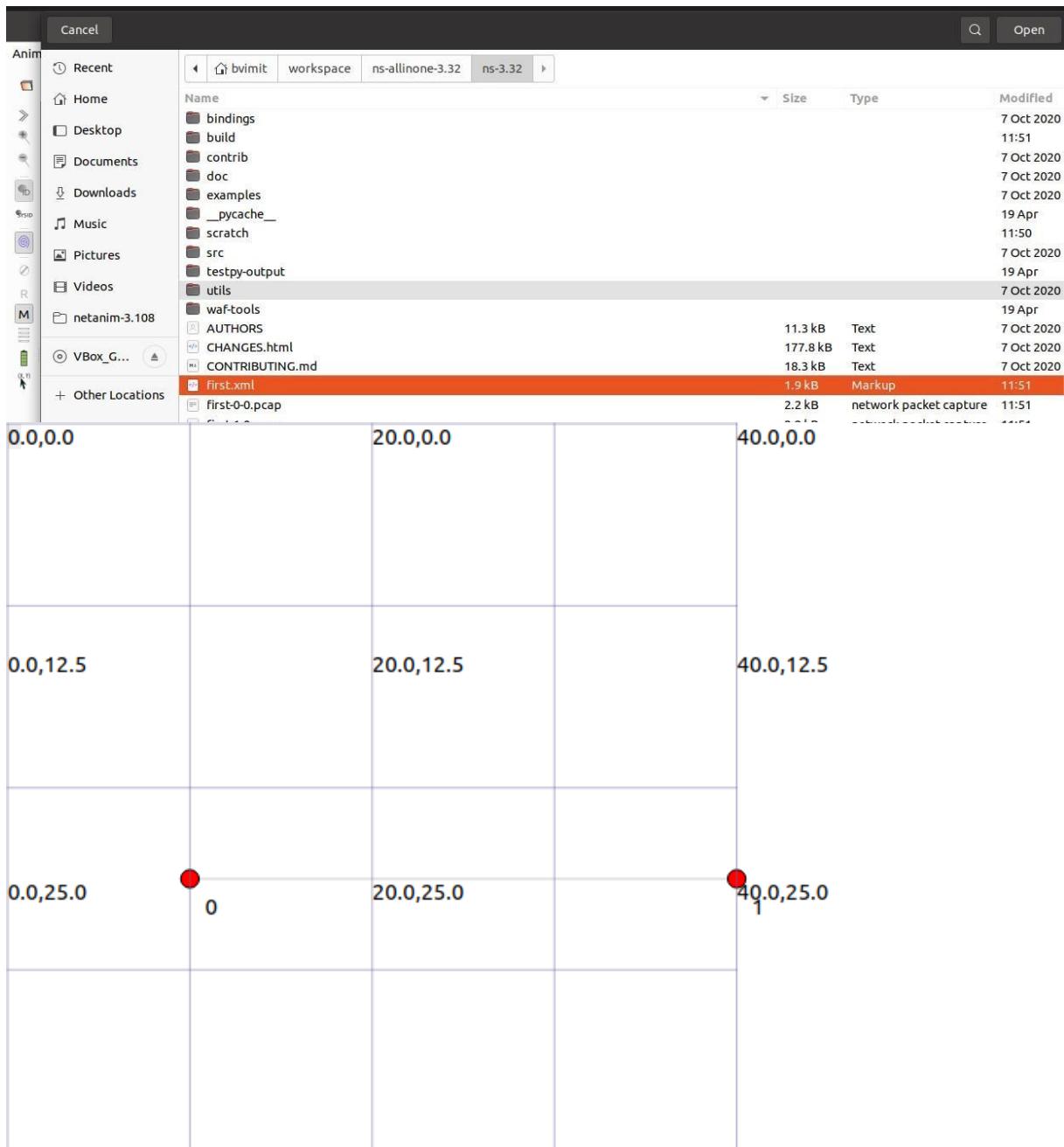
The first.xml file is created in the ns-3.32 folder

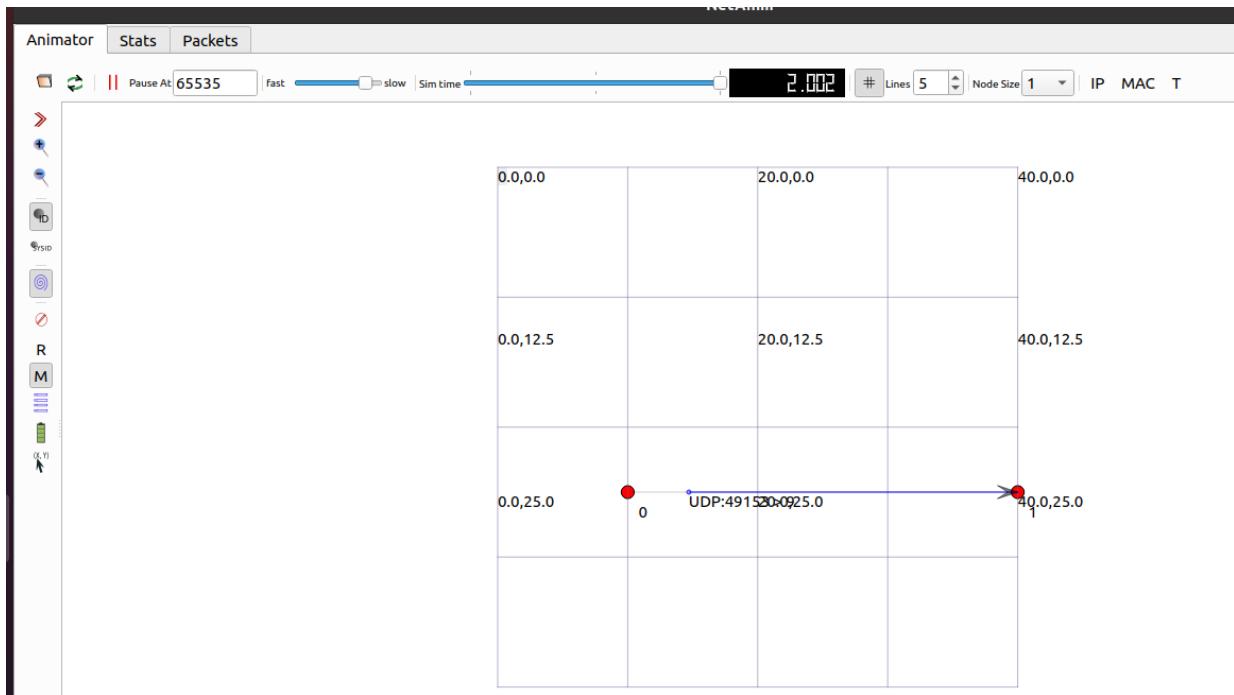


### 3. ./NetAnim



Open folder ns-3.32 -> first.xml





## **Practical 6**

### **Program to simulate bus topology**

**File Name:** Bus.cc

#### **Code:**

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

//Default Network Topology
//
//10.1.1.0
```

```

//n0.....n1
//      |
//      -----
//      |   |   |
//      n2   n3   n4
// point-to-point
//Bus Topology

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 application to log if true", verbose);
cmd.Parse(argc, argv);

if(verbose)
{
//Time::SetResolution (Time::NS);
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);
//for Net Anim

MobilityHelper mobility;
mobility.SetMobilityModel("ns3::ConstantPositionMobilityModel");
mobility.Install(p2pNodes);
mobility.Install(csmaNodes);

AnimationInterface anim("second111.xml");
AnimationInterface::SetConstantPosition(p2pNodes.Get(0), 10, 25);

```

```

AnimationInterface::SetConstantPosition(p2pNodes.Get(1), 40, 25);
AnimationInterface::SetConstantPosition(csmaNodes.Get(1), 10, 25);
AnimationInterface::SetConstantPosition(csmaNodes.Get(2), 10, 25);
AnimationInterface::SetConstantPosition(csmaNodes.Get(3), 10, 25);
anim.EnablePacketMetadata(true);

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

```

### **Output:**

./waf --run scratch/Buss

```

bvimit@bvimit-VirtualBox:~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run scratch/Buss
Waf: Entering directory `/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
[2838/2908] Compiling scratch/Buss.cc
[2868/2908] Linking build/scratch/Buss
Waf: Leaving directory `/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (6.156s)
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 +2.01761s client received 1024 bytes from 10.1.2.4 port 9

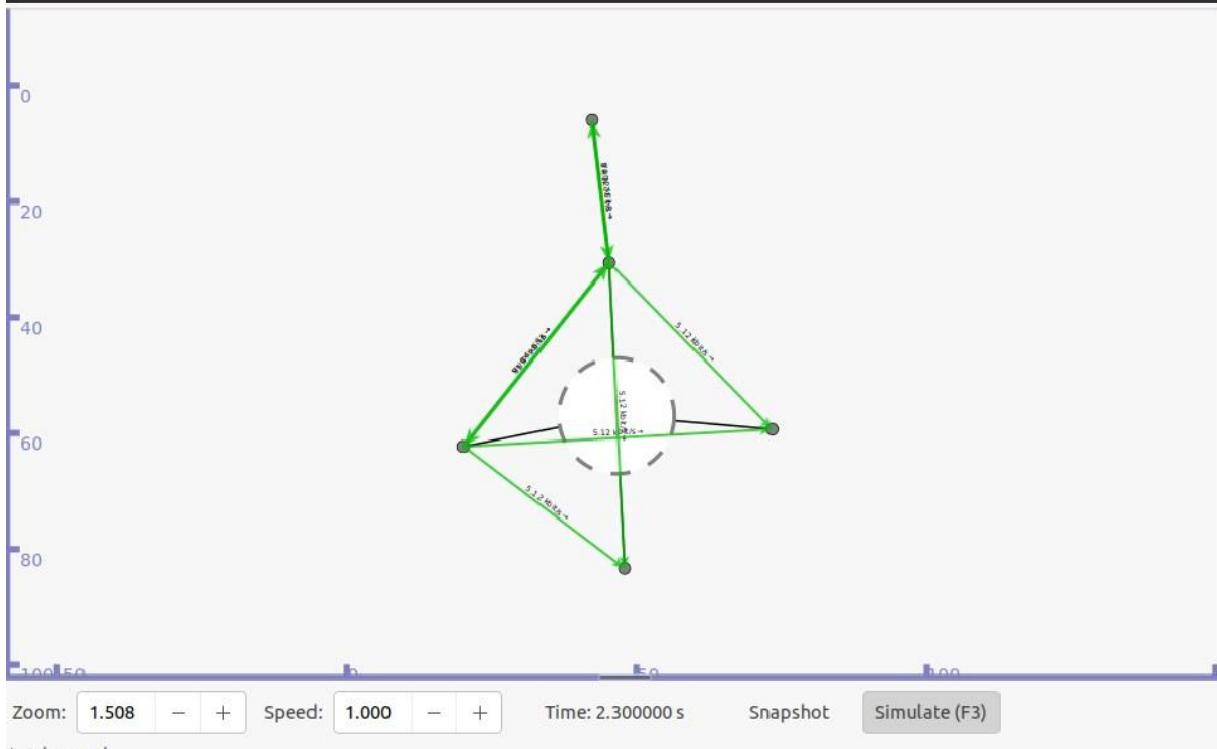
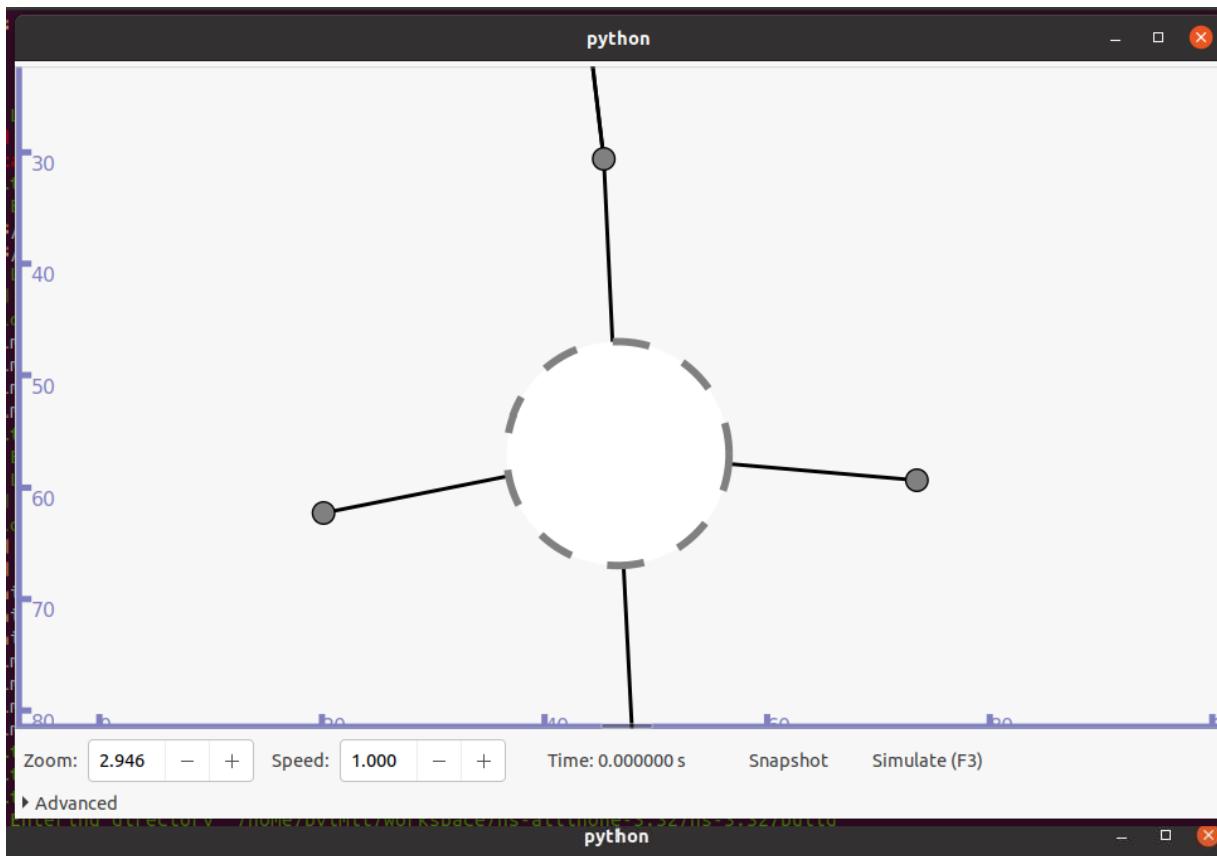
```

./waf --run scratch/Buss --vis

```

At time +2.01761s client received 1024 bytes from 10.1.2.4 port 9
bvimit@bvimit-VirtualBox:~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run scratch/Buss --vis
Waf: Entering directory `/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
Waf: Leaving directory `/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (2.130s)
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.

```



## **Practical 7**

### **Program to simulate bus topology Using NetAnim**

**File Name:** BussAnimate.cc

**Code:**

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

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 application 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); I

nternetStackHelper 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);

```

```

MobilityHelper mobility;
mobility.SetMobilityModel("ns3::ConstantPositionMobilityModel");
mobility.Install(csmaNodes);
mobility.Install(p2pNodes);

AnimationInterface anim("second.xml");
anim.SetConstantPosition(p2pNodes.Get(0), 10, 25);
anim.SetConstantPosition(p2pNodes.Get(1), 20, 25);
anim.SetConstantPosition(csmaNodes.Get(1), 40, 25);
anim.SetConstantPosition(csmaNodes.Get(2), 50, 25);
anim.SetConstantPosition(csmaNodes.Get(3), 60, 25);
anim.EnablePacketMetadata(true);

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

```

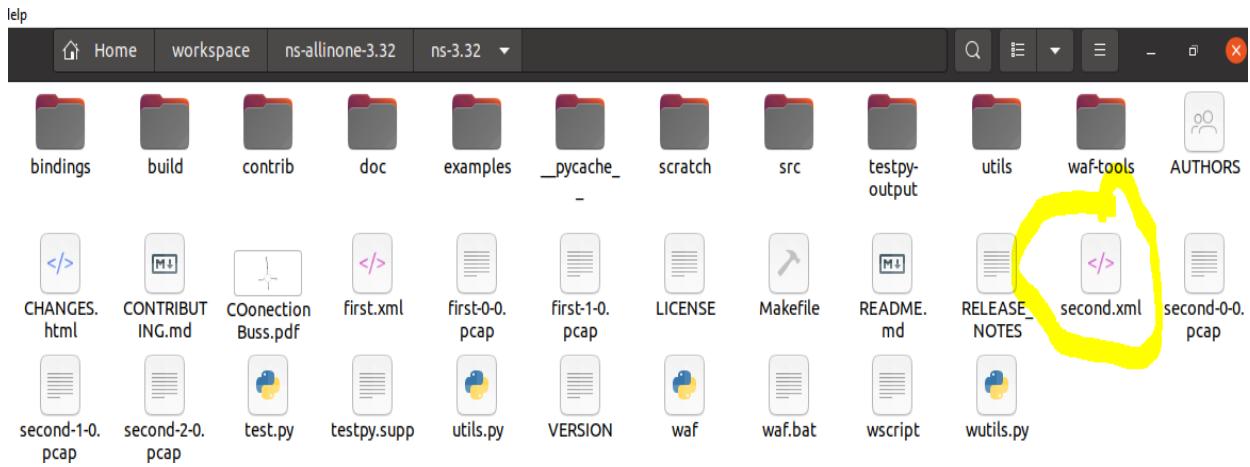
## **Output:**

./waf --run scratch/BussAnimate

```

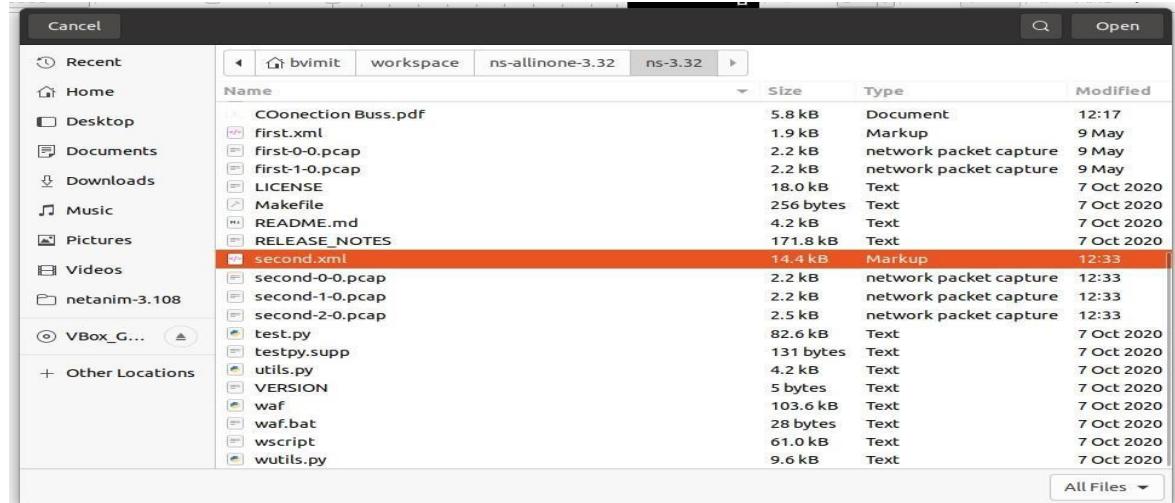
bvimit@bvimit-VirtualBox:~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run scratch/BussAnimate
Waf: Entering directory '/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
Waf: Leaving directory '/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (2.330s)
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 +2.01761s client received 1024 bytes from 10.1.2.4 port 9

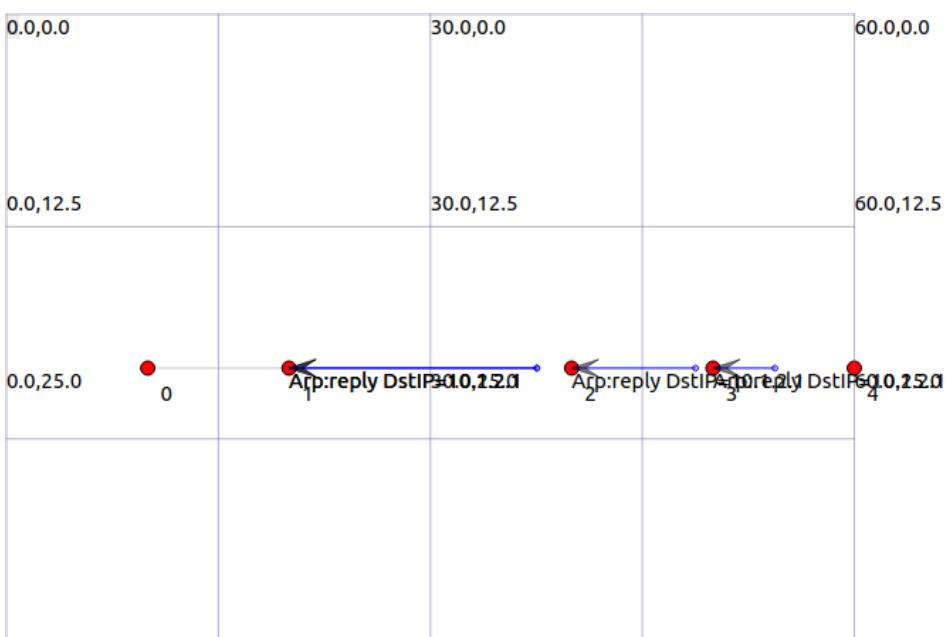
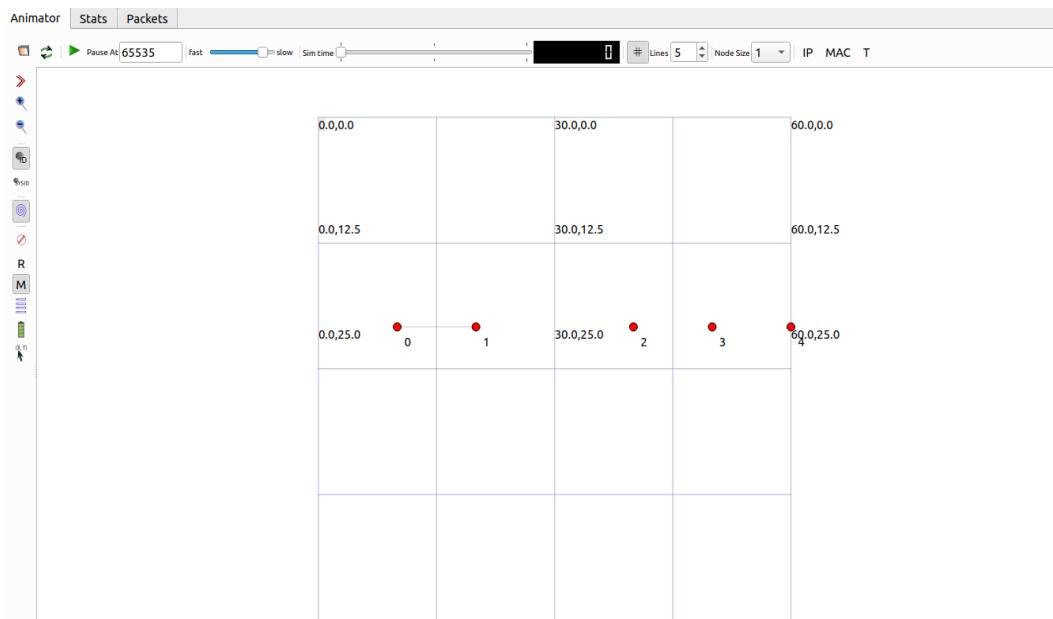
```

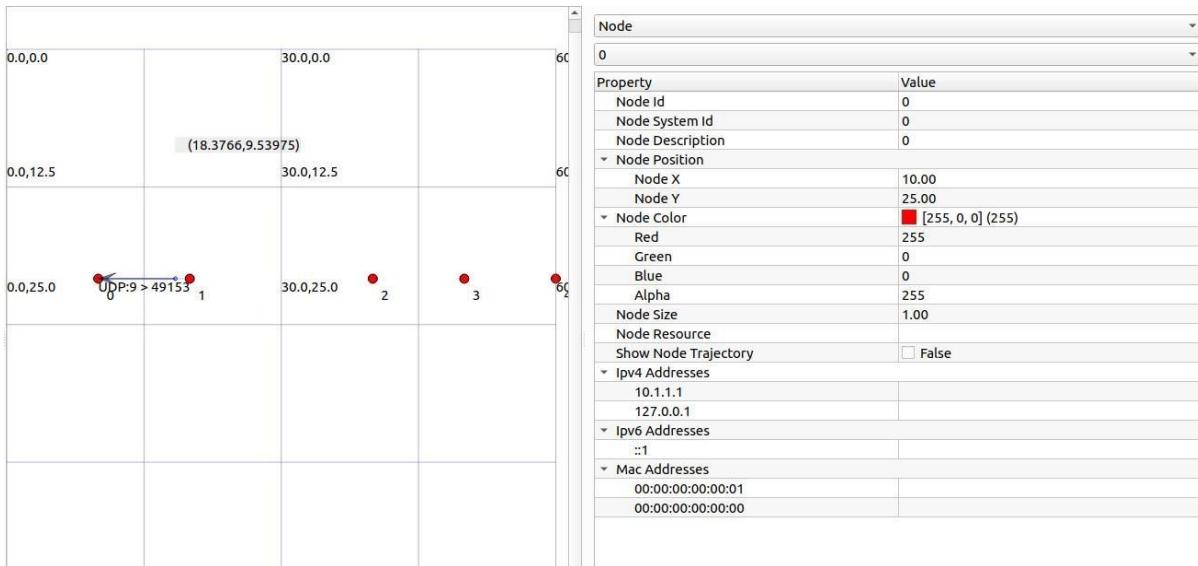


./NetAnim

```
bvimit@bvimit-VirtualBox:~/workspace/ns-allinone-3.32/netanim-3.108$ ./NetAnim
```







## **Practical 8**

### **Analyse BUS topology simulation pcap files using wireshark**

**bus\_topology.cc**

**Code:**

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

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("mCsma", "Number of \"Extra\" CSMA nodes/devices", nCsma);
    cmd.AddValue("verbose", "Tell echo application 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","255.555.255.0");

Ipv4InterfaceContainer p2pInterfaces;
p2pInterfaces = address.Assign(p2pDevices);

address.SetBase("10.1.2.0","255.555.255.0");

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();

MobilityHelper mobility;
mobility.SetMobilityModel("ns3::ConstantPositionMobilityModel");
mobility.Install(csmaNodes);
mobility.Install(p2pNodes);

```

```

AnimationInterface anim("second.xml");
anim.SetConstantPosition(p2pNodes.Get(0), 10, 25);
anim.SetConstantPosition(p2pNodes.Get(1), 20, 25);
anim.SetConstantPosition(csmaNodes.Get(1), 30, 15);
anim.SetConstantPosition(csmaNodes.Get(2), 40, 25);
anim.SetConstantPosition(csmaNodes.Get(3), 30, 35);
anim.EnablePacketMetadata(true);

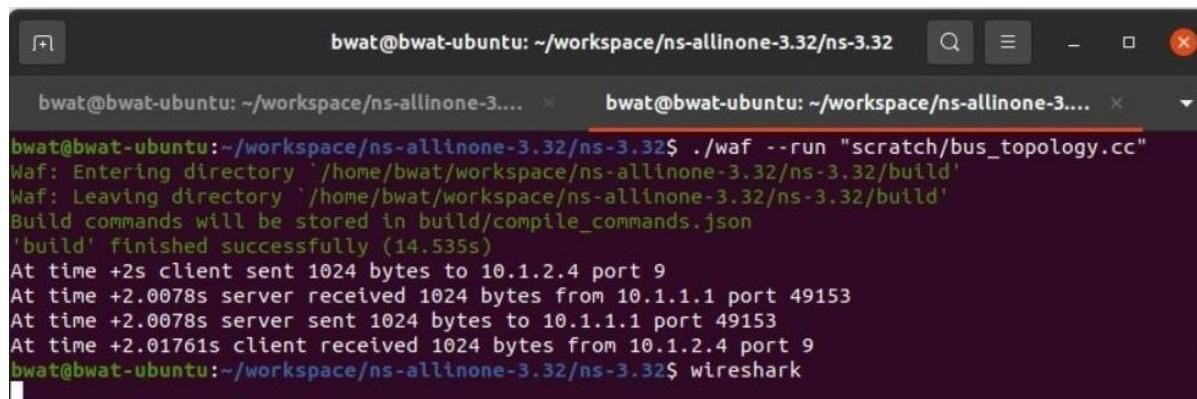
pointToPoint.EnablePcapAll("second");
csma.EnablePcap("second", csmaDevices.Get(1), true);

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

```

### **Output:**

Open terminal and type below command and then type wireshark:



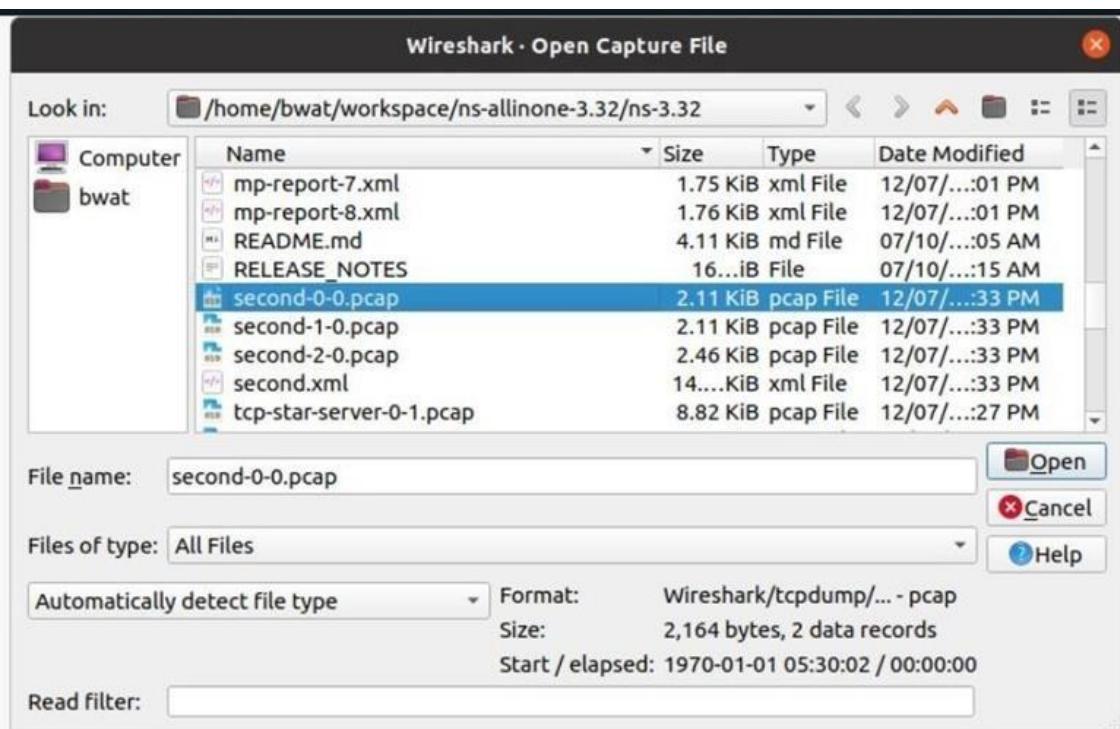
The screenshot shows a terminal window with two tabs. The left tab shows the command: `bwat@bwat-ubuntu:~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run "scratch/bus_topology.cc"`. The output of this command is displayed, including directory changes, build logs, and network traffic details. The right tab shows the command: `bwat@bwat-ubuntu:~/workspace/ns-allinone-3.32/ns-3.32$ wireshark`.

```

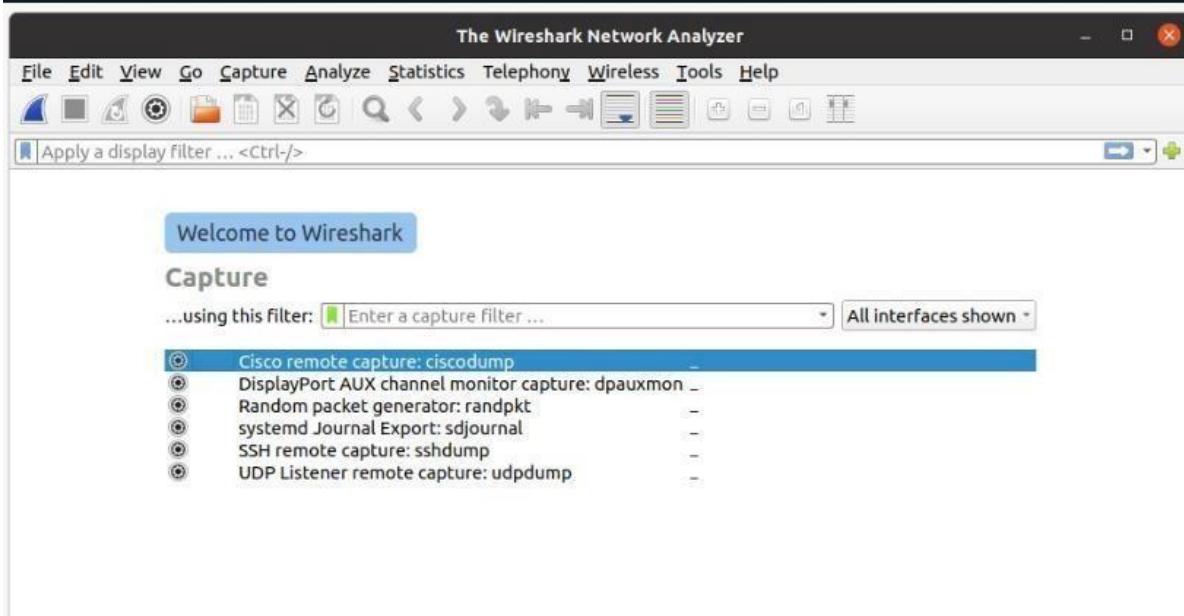
bwat@bwat-ubuntu:~/workspace/ns-allinone-3.... x bwat@bwat-ubuntu:~/workspace/ns-allinone-3.... x
bwat@bwat-ubuntu:~/workspace/ns-allinone-3.32$ ./waf --run "scratch/bus_topology.cc"
Waf: Entering directory '/home/bwat/workspace/ns-allinone-3.32/ns-3.32/build'
Waf: Leaving directory '/home/bwat/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (14.535s)
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 +2.01761s client received 1024 bytes from 10.1.2.4 port 9
bwat@bwat-ubuntu:~/workspace/ns-allinone-3.32/ns-3.32$ wireshark

```

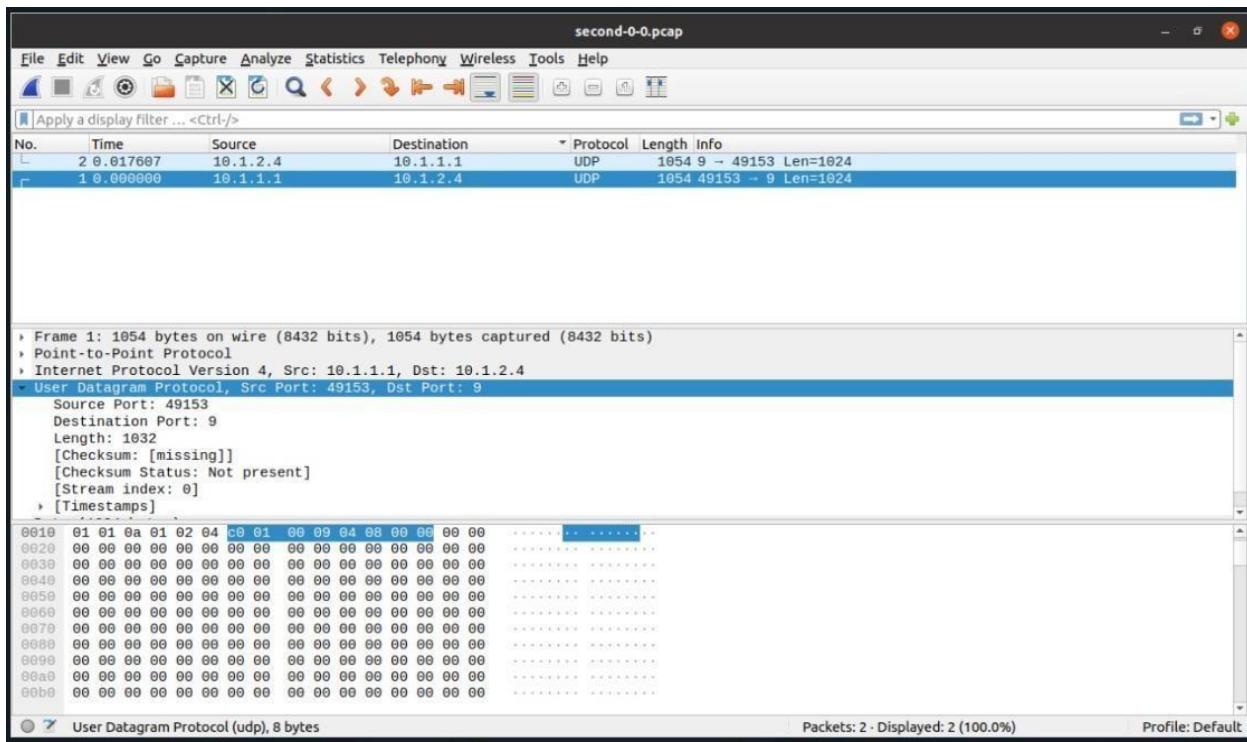
Then three pcap file as shown below will be autogenerated:



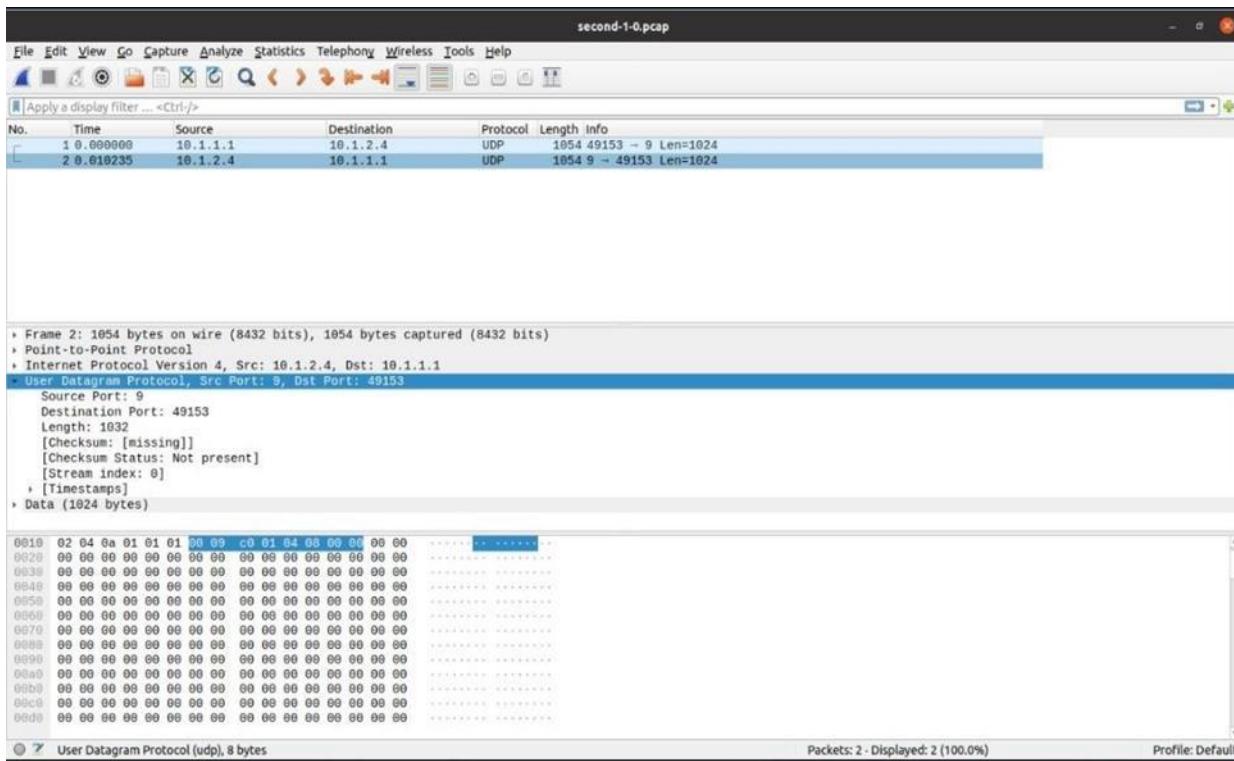
File->open-> all second pcap file->start simultion



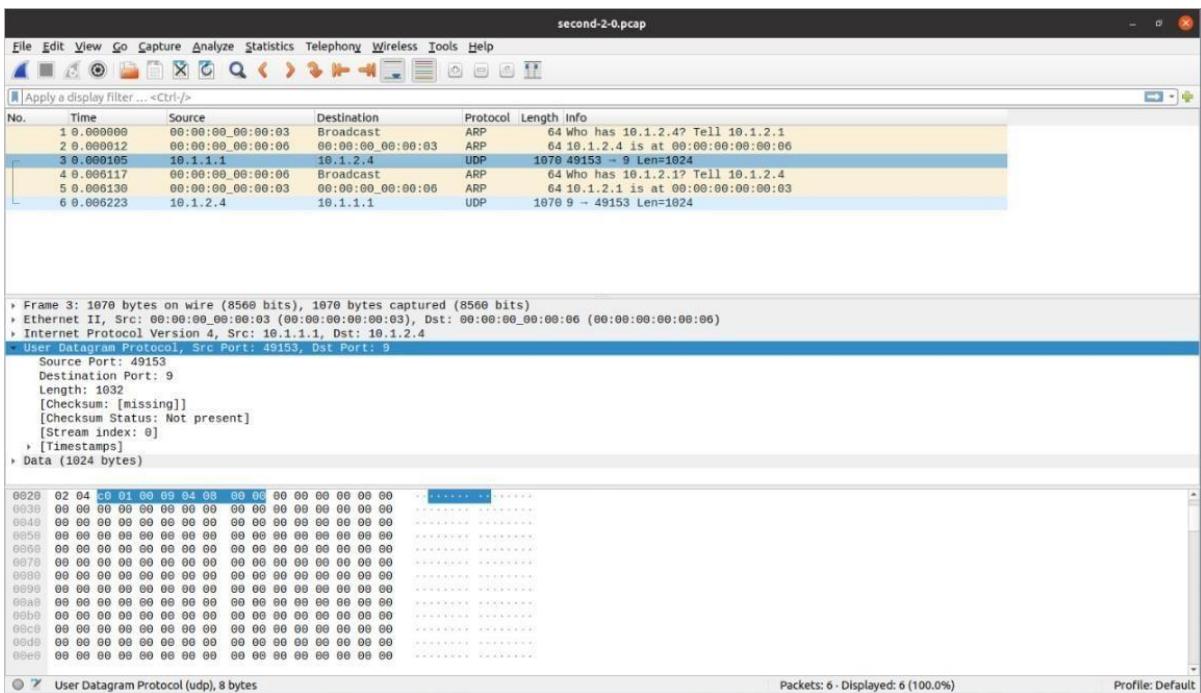
Second-0-0.pcap:



Second.1-0.pcap:



Second-2-0.pcap:



## **Practical 9** **Program to simulate star topology**

### **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/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 (_FILE_);
    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");

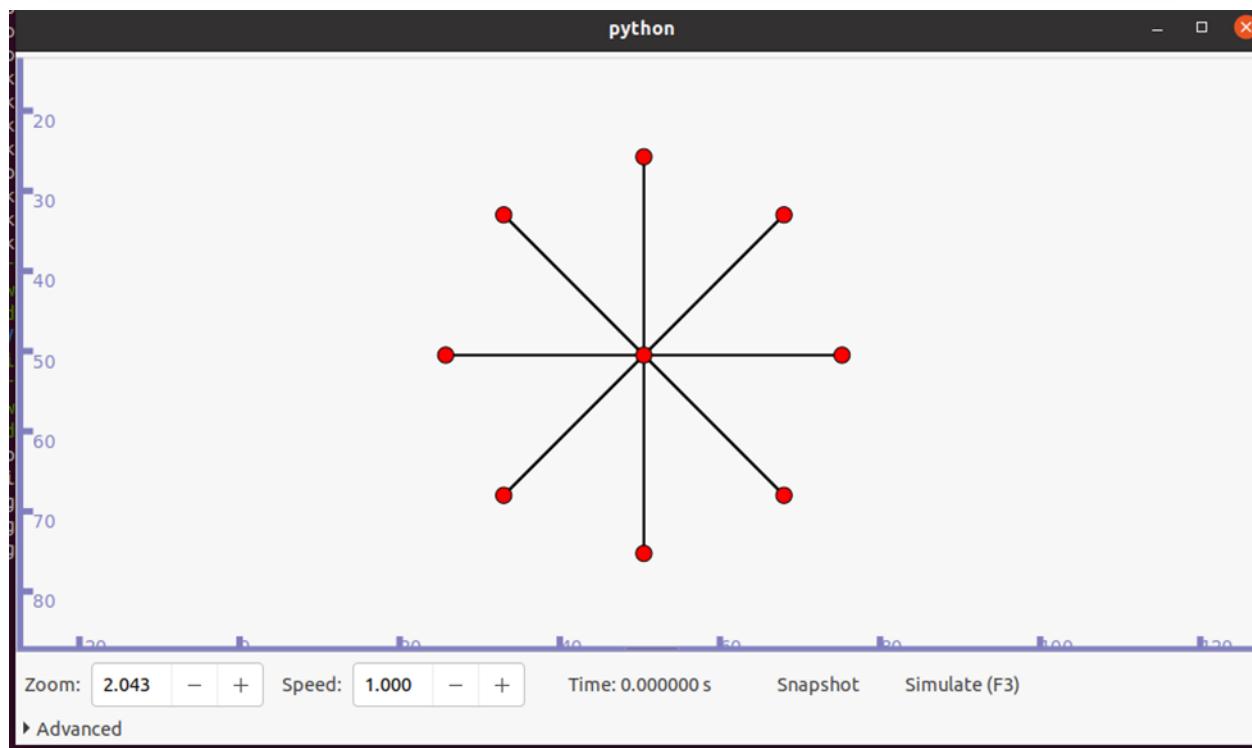
NS_LOG_INFO ("Run Simulation.");
Simulator::Run ();
Simulator::Destroy ();
NS_LOG_INFO ("Done.");

return 0;
}

```

**Output:**

```
bvimit@bvimit-Lenovo: ~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run scratch/star.cc
Waf: Entering directory `/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
[2814/2872] Compiling scratch/star.cc
[2821/2872] Linking build/scratch/second
[2822/2872] Linking build/scratch/third
[2833/2872] Linking build/scratch/star
Waf: Leaving directory `/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (16.996s)
bvimit@bvimit-Lenovo:~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run "scratch/star.cc --nSpokes=5"
Waf: Entering directory `/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
Waf: Leaving directory `/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (1.177s)
bvimit@bvimit-Lenovo:~/workspace/ns-allinone-3.32/ns-3.32$
```



## Practical 10

### Animate Star Topology in NetAnim

#### **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"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("Star");
int main (int argc, char *argv[])
{
Config::SetDefault ("ns3::OnOffApplication::PacketSize", UintegerValue (137));
Config::SetDefault ("ns3::OnOffApplication::DataRate", StringValue ("14kb/s"));

uint32_t nSpokes = 8;

CommandLine cmd ( FILE );
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.");
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));

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.");
Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

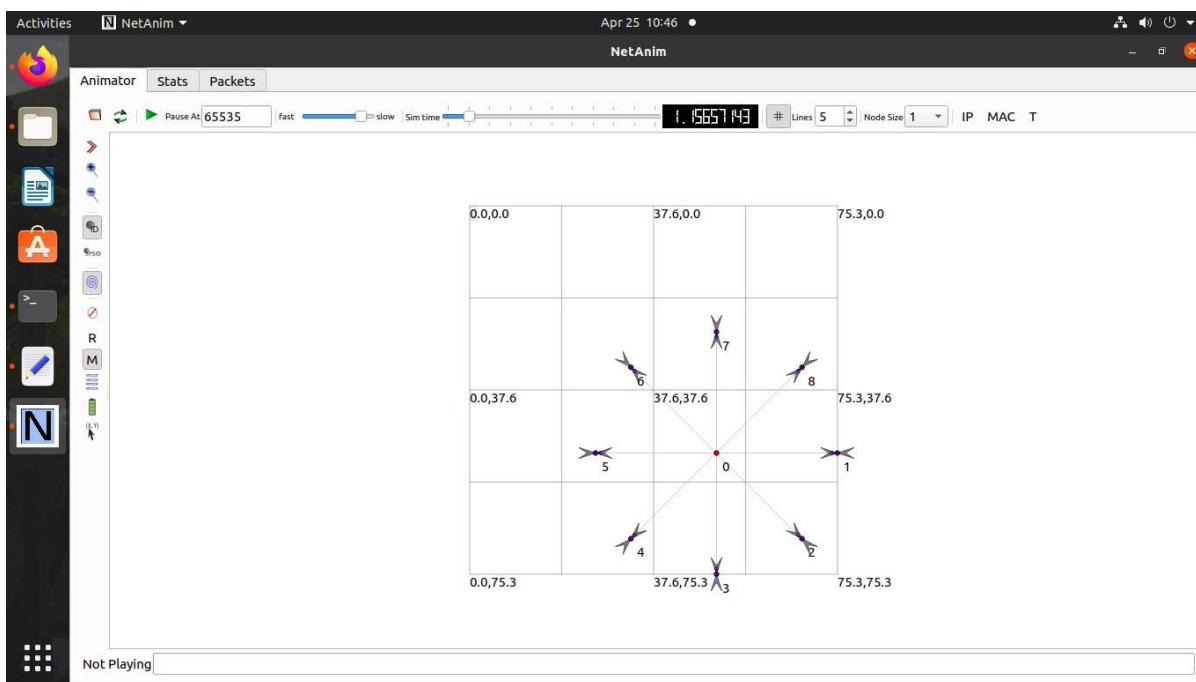
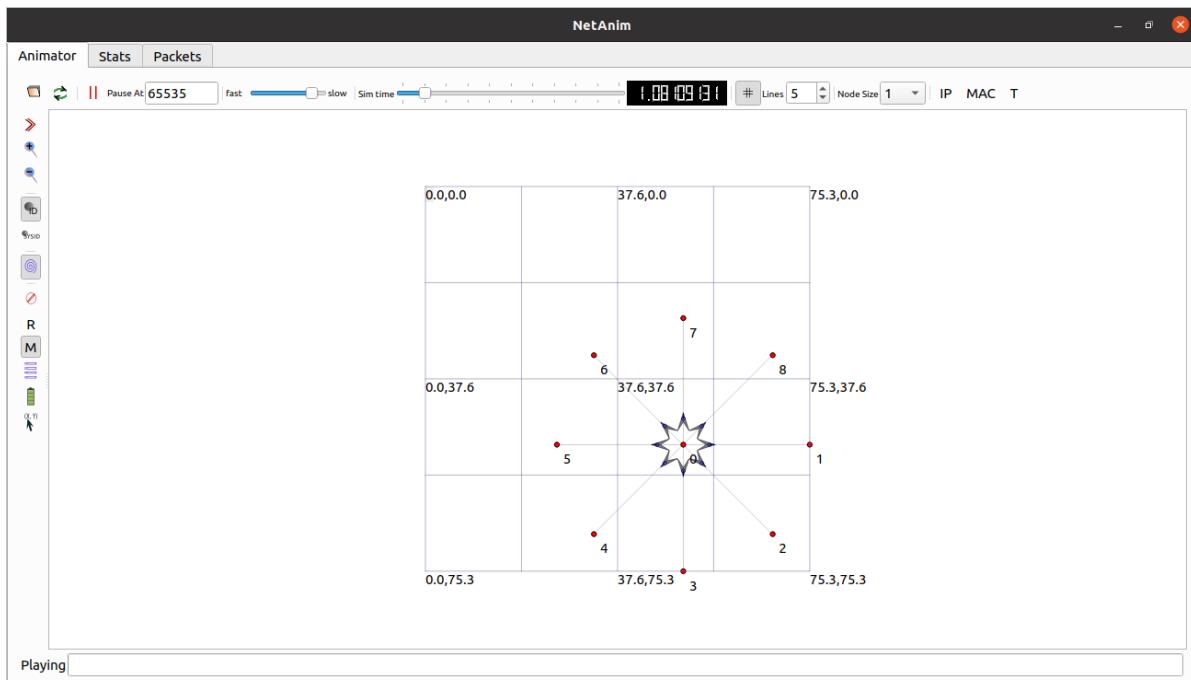
NS_LOG_INFO ("Enable pcap tracing.");
pointToPoint.EnablePcapAll ("star");

std::string animFile = "star.xml";
star.BoundingBox (1, 1, 100, 100);
AnimationInterface anim (animFile);

NS_LOG_INFO ("Run Simulation.");
Simulator::Run ();
Simulator::Destroy ();
NS_LOG_INFO ("Done.");
return 0;
}

```

**Output:**





## Practical 11

### Program to simulate Mesh topology

#### Code:

```
* m_xSize * step
* |<----->|
* step
* |<-->|
* * --- * --- * <---Ping sink -
* | \ | /|           ^
* | \ | / |           |
* * --- * --- * m_ySize * step |
* | /| \ |           |
* | / | \ |           |
* * --- * --- *
#include <iostream>
#include <sstream>
#include <fstream>

#include "ns3/core-module.h"
#include "ns3/internet-module.h"
#include "ns3/network-module.h"
#include "ns3/applications-module.h"
#include "ns3/mesh-module.h"
#include "ns3/mobility-module.h"
#include "ns3/mesh-helper.h"
#include "ns3/yans-wifi-helper.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("TestMeshScript");
/***
*      \ingroup mesh
*      \brief MeshTest class
*/
class MeshTest
{
public:
    /// Init test MeshTest ();
    /***
```

```

*      Configure test from command line arguments
*
*      \param argc command line argument count
*      \param argv command line arguments
*/
void Configure (int argc, char ** argv);
/** 
*      Run test
*      \returns the test status
*/
int Run ();
private:
int m_xSize; ///< X size int
m_ySize; ///< Y size double
m_step; ///< step
double m_randomStart; ///< random start
double m_totalTime; ///< total time
double m_packetInterval; ///< packet interval
uint16_t m_packetSize; ///< packet size
uint32_t m_nIfaces; ///< number interfaces
bool m_chan; ///< channel
bool m_pcap; ///< PCAP
bool m_ascii; ///< ASCII
std::string m_stack; ///< stack
std::string m_root; ///< root

/// List of network nodes
NodeContainer nodes;

/// List of all mesh point devices
NetDeviceContainer meshDevices;

/// Addresses of interfaces:
Ipv4InterfaceContainer interfaces;

/// MeshHelper. Report is not static methods
MeshHelper mesh;

private:
/// Create nodes and setup their mobility
void CreateNodes ();

/// Install internet m_stack on nodes
void InstallInternetStack ();

/// Install applications
void InstallApplication ();

```

```

/// Print mesh devices diagnostics
void Report ();
};

MeshTest::MeshTest () :
m_xSize (3),
m_ySize (3),
m_step (100.0),
m_randomStart (0.1),
m_totalTime (100.0),
m_packetInterval (0.1),
m_packetSize (1024),
m_nIfaces (1),
m_chan (true),
m_pcap (false),
m_ascii (false),
m_stack ("ns3::Dot11sStack"),
m_root ("ff:ff:ff:ff:ff:ff")
{
}

void MeshTest::Configure (int argc, char *argv[])
{
CommandLine cmd ( FILE );
cmd.AddValue ("x-size", "Number of nodes in a row grid", m_xSize);
cmd.AddValue ("y-size", "Number of rows in a grid", m_ySize);
cmd.AddValue ("step", "Size of edge in our grid (meters)", m_step);

// Avoid starting all mesh nodes at the same time (beacons may collide)
cmd.AddValue ("start", "Maximum random start delay for beacon jitter (sec)", m_randomStart);
cmd.AddValue ("time", "Simulation time (sec)", m_totalTime);
cmd.AddValue ("packet-interval", "Interval between packets in UDP ping (sec)", m_packetInterval);
cmd.AddValue ("packet-size", "Size of packets in UDP ping (bytes)", m_packetSize); cmd.addValue
("interfaces", "Number of radio interfaces used by each mesh point", m_nIfaces); cmd.addValue ("channels",
"Use different frequency channels for different interfaces", m_chan);
cmd.addValue ("pcap", "Enable PCAP traces on interfaces", m_pcap); cmd.addValue ("ascii", "Enable Ascii
traces on interfaces", m_ascii);

cmd.addValue ("stack", "Type of protocol stack. ns3::Dot11sStack by default", m_stack); cmd.addValue
("root", "Mac address of root mesh point in HWMP", m_root);

cmd.Parse (argc, argv);
NS_LOG_DEBUG ("Grid:" << m_xSize << "*" << m_ySize);
NS_LOG_DEBUG ("Simulation time: " << m_totalTime << " s");

if (m_ascii)
{
PacketMetadata::Enable ();
}

```

```

}

void MeshTest::CreateNodes ()
{
/*
 *      Create m_ySize*m_xSize stations to form a grid topology
 */
nodes.Create (m_ySize*m_xSize);

// Configure YansWifiChannel
YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default ();
YansWifiChannelHelper wifiChannel = YansWifiChannelHelper::Default ();
wifiPhy.SetChannel (wifiChannel.Create ());

/*
 *      Create mesh helper and set stack installer to it
 *      Stack installer creates all needed protocols and install them to
 *      mesh point device
*/
mesh = MeshHelper::Default ();
if (!Mac48Address (m_root.c_str ()).IsBroadcast ())
{
mesh.SetStackInstaller (m_stack, "Root", Mac48AddressValue (Mac48Address (m_root.c_str ())));
}
else
{
//If root is not set, we do not use "Root" attribute, because it
//is specified only for 11s mesh.SetStackInstaller (m_stack);
}
if (m_chan)

{
mesh.SetSpreadInterfaceChannels (MeshHelper::SPREAD_CHANNELS);
}
else
{
mesh.SetSpreadInterfaceChannels (MeshHelper::ZERO_CHANNEL);
}
mesh.SetMacType ("RandomStart", TimeValue (Seconds (m_randomStart)));

// Set number of interfaces - default is single-interface mesh point
mesh.SetNumberOfInterfaces (m_nIfaces);

// Install protocols and return container if MeshPointDevices
meshDevices = mesh.Install (wifiPhy, nodes);

// Setup mobility - static grid topology

```

```

MobilityHelper mobility;
mobility.SetPositionAllocator ("ns3::GridPositionAllocator", "MinX", DoubleValue (0.0),
"MinY", DoubleValue (0.0),
"DeltaX", DoubleValue (m_step),
"DeltaY", DoubleValue (m_step),
"GridWidth", UintegerValue (m_xSize),
"LayoutType", StringValue ("RowFirst"));

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

if (m_pcap)
wifiPhy.EnablePcapAll (std::string ("mp-"));

if (m_ascii)
{
AsciiTraceHelper ascii;
wifiPhy.EnableAsciiAll (ascii.CreateFileStream ("mesh.tr"));
}

void MeshTest::InstallInternetStack ()
{
InternetStackHelper internetStack;
internetStack.Install (nodes);
Ipv4AddressHelper address;
address.SetBase ("10.1.1.0", "255.255.255.0");
interfaces = address.Assign (meshDevices);
}

void MeshTest::InstallApplication ()
{
UdpEchoServerHelper echoServer (9);
ApplicationContainer serverApps = echoServer.Install (nodes.Get (0));
serverApps.Start (Seconds (0.0));
serverApps.Stop (Seconds (m_totalTime));

UdpEchoClientHelper echoClient (interfaces.GetAddress (0), 9);
echoClient.SetAttribute("MaxPackets", UintegerValue((uint32_t)(m_totalTime*(1/m_packetInterval))));
echoClient.SetAttribute ("Interval", TimeValue (Seconds (m_packetInterval)));
echoClient.SetAttribute ("PacketSize", UintegerValue (m_packetSize));

ApplicationContainer clientApps = echoClient.Install (nodes.Get (m_xSize*m_ySize-1)); clientApps.Start
(Seconds (0.0));
clientApps.Stop (Seconds (m_totalTime));
}

int MeshTest::Run ()

```

```

{
CreateNodes ();
InstallInternetStack ();
InstallApplication ();
Simulator::Schedule (Seconds (m_totalTime), &MeshTest::Report, this);

Simulator::Stop (Seconds (m_totalTime));

/*//Netanim code
std::string animFile = "star.xml";

// Set the bounding box for animation
star.BoundingBox (1, 1, 100, 100);
// Create the animation object and configure for specified output AnimationInterface anim (animFile);*/

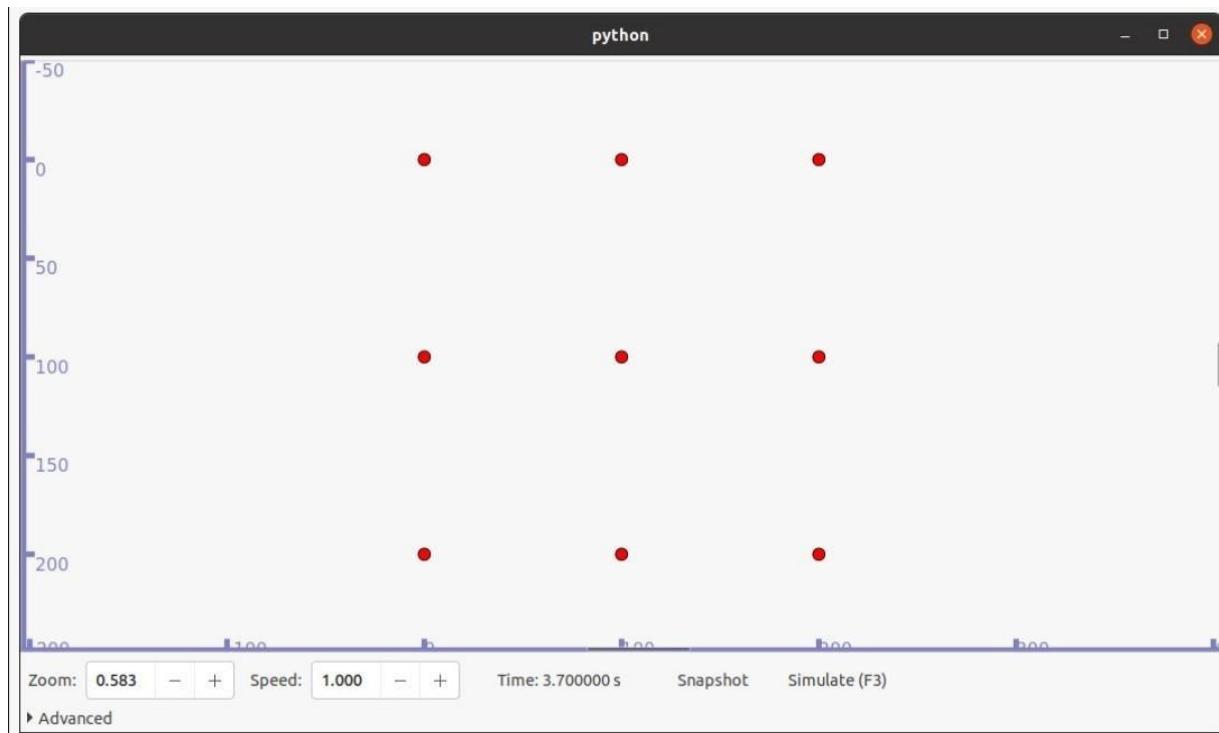
Simulator::Run ();
Simulator::Destroy ();
return 0;
}

void MeshTest::Report ()
{
unsigned n (0);
for (NetDeviceContainer::Iterator i = meshDevices.Begin (); i != meshDevices.End (); ++i, ++n)
{
std::ostringstream os;
os << "mp-report-" << n << ".xml";
std::cerr << "Printing mesh point device #" << n << " diagnostics to " << os.str () << "\n"; std::ofstream of;
of.open (os.str ().c_str ());
if (!of.is_open ())
{
std::cerr << "Error: Can't open file " << os.str () << "\n";
return;
}
mesh.Report (*i, of);
of.close ();
}
}
int
main (int argc, char *argv[])
{
MeshTest t;
t.Configure (argc, argv);
return t.Run ();
}

```

**Output:**

```
bwat@bwat-ubuntu:~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run "scratch/mesh_topology.cc"
Waf: Entering directory '/home/bwat/workspace/ns-allinone-3.32/ns-3.32/build'
Waf: Leaving directory '/home/bwat/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (2.584s)
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
bwat@bwat-ubuntu:~/workspace/ns-allinone-3.32/ns-3.32$
```



## Practical 12

### Program to simulate hybrid topology

#### **Code:**

##### **A] With NetAnim**

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

// Default Network Topology
// Number of wifi or csma nodes can be increased up to 250
// Wifi 10.1.3.0
//          AP
// *   *   *   *
// |   |   |   |
// 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 = true;

CommandLine cmd;
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);

// Check for valid number of csma or wifi nodes
// 250 should be enough, otherwise IP addresses
// soon become an issue

if (nWifi > 250 || nCsma > 250)
{
std::cout << "Too many wifi or csma nodes, no more than 250 each." << std::endl;
return 1;
}

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

NodeContainer p2pNodes;
p2pNodes.Create (2);

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

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

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

```

```

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

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

NodeContainer wifiStaNodes;
wifiStaNodes.Create (nWifi);
NodeContainer wifiApNode = p2pNodes.Get (0);

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

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

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

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

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

NetDeviceContainer apDevices;
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);

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 (wifiStaNodes.Get (nWifi - 1)); clientApps.Start
(Seconds (2.0));
clientApps.Stop (Seconds (10.0));

Ipv4GlobalRoutingHelper::PopulateRoutingTables ();
Simulator::Stop (Seconds (10.0));

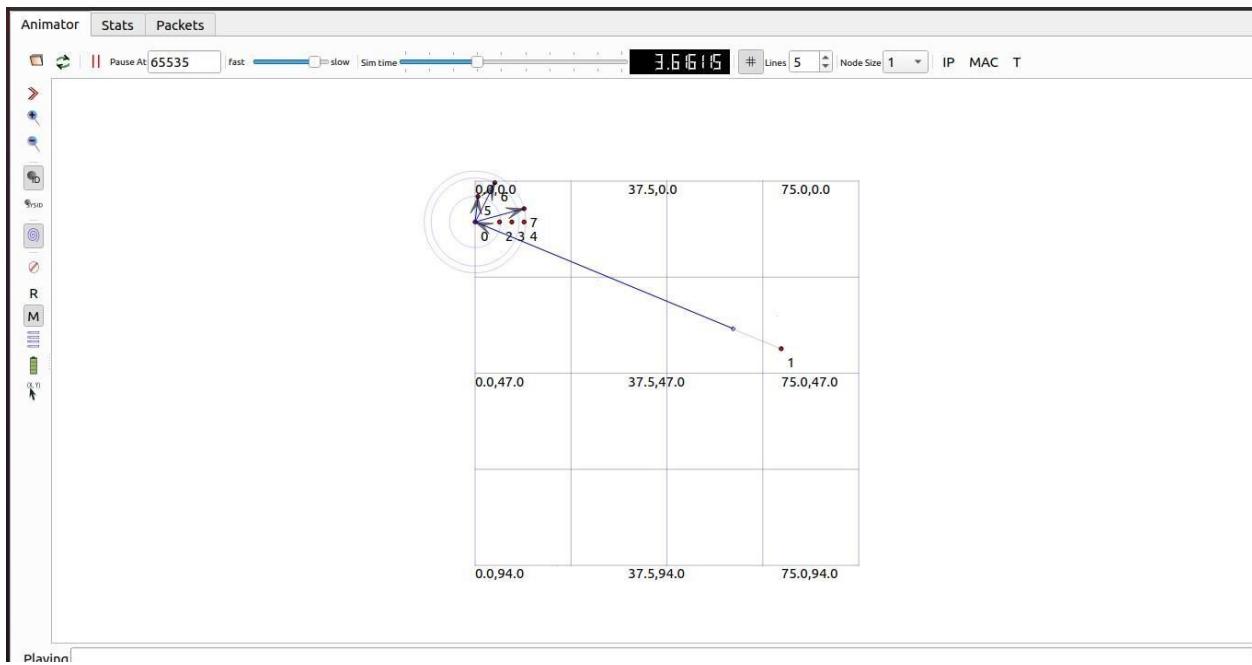
if (tracing == true)
{
pointToPoint.EnablePcapAll ("third");
phy.EnablePcap ("third", apDevices.Get (0));
csma.EnablePcap ("third", csmaDevices.Get (0), true);
}

//Netanim code
AnimationInterface anim ("third.xml");
anim.SetConstantPosition (csmaNodes.Get(1), 6, 10);
anim.SetConstantPosition (csmaNodes.Get(2), 9, 10);
anim.SetConstantPosition (csmaNodes.Get(3), 12, 10);
Simulator::Run ();
Simulator::Destroy ();
return 0;

```

}

## **Output:**



## **B] Using vis**

### **Code:**

```
#include "ns3/netanim-module.h"
#include "ns3/core-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/network-module.h"
#include "ns3/applications-module.h"
#include "ns3/wifi-module.h"
#include "ns3/mobility-module.h"
#include "ns3/csma-module.h"
#include "ns3/internet-module.h"
// Default Network Topology
// Number of wifi or csma nodes can be increased up to 250
// Wifi 10.1.3.0
//          AP
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 = true;  
    CommandLine cmd(_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);  
    // Check for valid number of csma or wifi nodes  
    // 250 should be enough, otherwise IP addresses  
    // soon become an issue  
    if (nWifi>18)  
    {  
        std::cout << "Too many wifi or csma nodes, no more than 250 each." << std::endl;  
        return 1;  
    }  
    if (verbose)  
    {  
        LogComponentEnable ("UdpEchoClientApplication", LOG_LEVEL_INFO);  
        LogComponentEnable ("UdpEchoServerApplication", LOG_LEVEL_INFO);  
    }  
}
```

```

        }

NodeContainer p2pNodes;
p2pNodes.Create (2);
PointToPointHelper pointToPoint;
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));
NetDeviceContainer p2pDevices;
p2pDevices = pointToPoint.Install (p2pNodes);
NodeContainer csmaNodes;
csmaNodes.Add (p2pNodes.Get (1));
csmaNodes.Create (nCsma);
CsmaHelper csma;
csma.SetChannelAttribute ("DataRate", StringValue ("100Mbps"));
csma.SetChannelAttribute ("Delay", TimeValue (NanoSeconds (6560)));

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

NodeContainer wifiStaNodes;
wifiStaNodes.Create (nWifi);
NodeContainer wifiApNode = p2pNodes.Get (0);
YansWifiChannelHelper channel = YansWifiChannelHelper::Default ();
YansWifiPhyHelper phy = YansWifiPhyHelper::Default ();
phy.SetChannel (channel.Create ());
WifiHelper wifi;
wifi.SetRemoteStationManager ("ns3::AarfWifiManager");
WifiMacHelper mac;
Ssid ssid = Ssid ("ns-3-ssid");
mac.SetType ("ns3::StaWifiMac",
    "Ssid", SsidValue (ssid),
    "ActiveProbing", BooleanValue (false));
NetDeviceContainer staDevices;
staDevices = wifi.Install (phy, mac, wifiStaNodes);
mac.SetType ("ns3::ApWifiMac",
    "Ssid", SsidValue (ssid));
NetDeviceContainer apDevices;
apDevices = wifi.Install (phy, mac, wifiApNode);
MobilityHelper mobility;
mobility.SetPositionAllocator ("ns3::GridPositionAllocator",
    "MinX", DoubleValue (0.0),
    "MinY", DoubleValue (0.0),
    "MaxX", DoubleValue (100.0),
    "MaxY", DoubleValue (100.0),
    "CellSize", DoubleValue (5.0));
mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
mobility.Install (staDevices);
mobility.Install (apDevices);

```

```

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

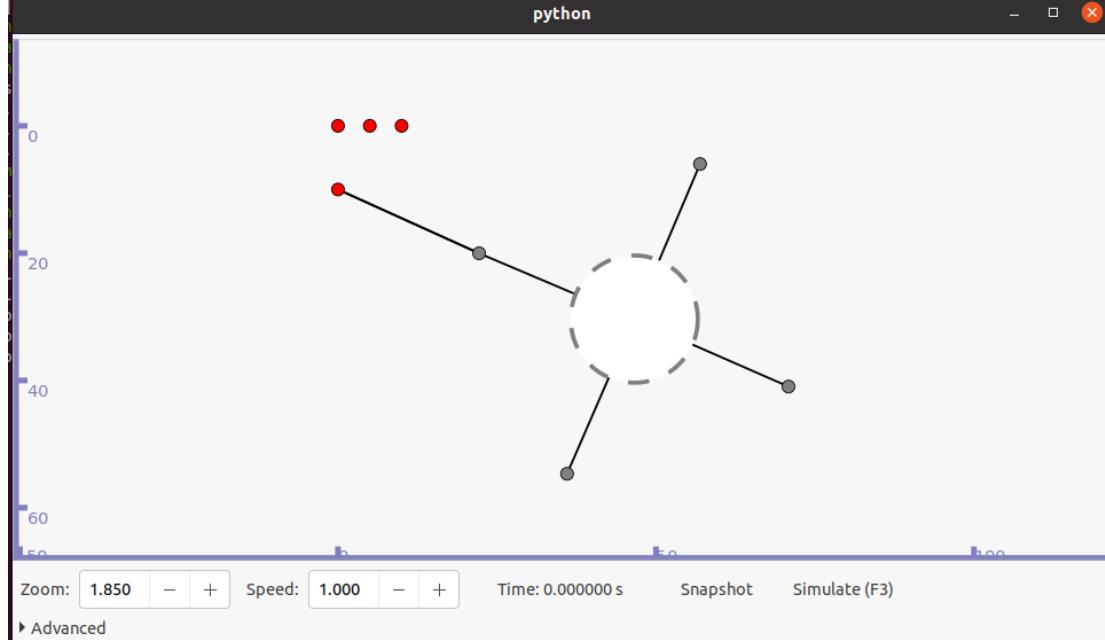
```

## Output:

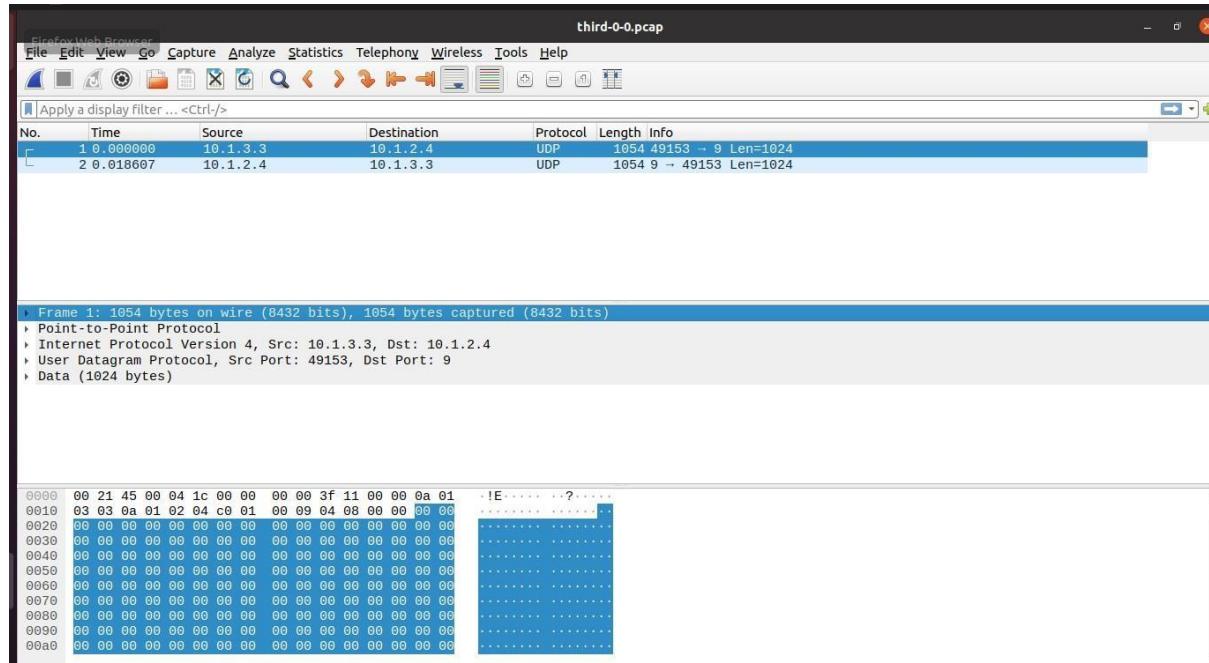
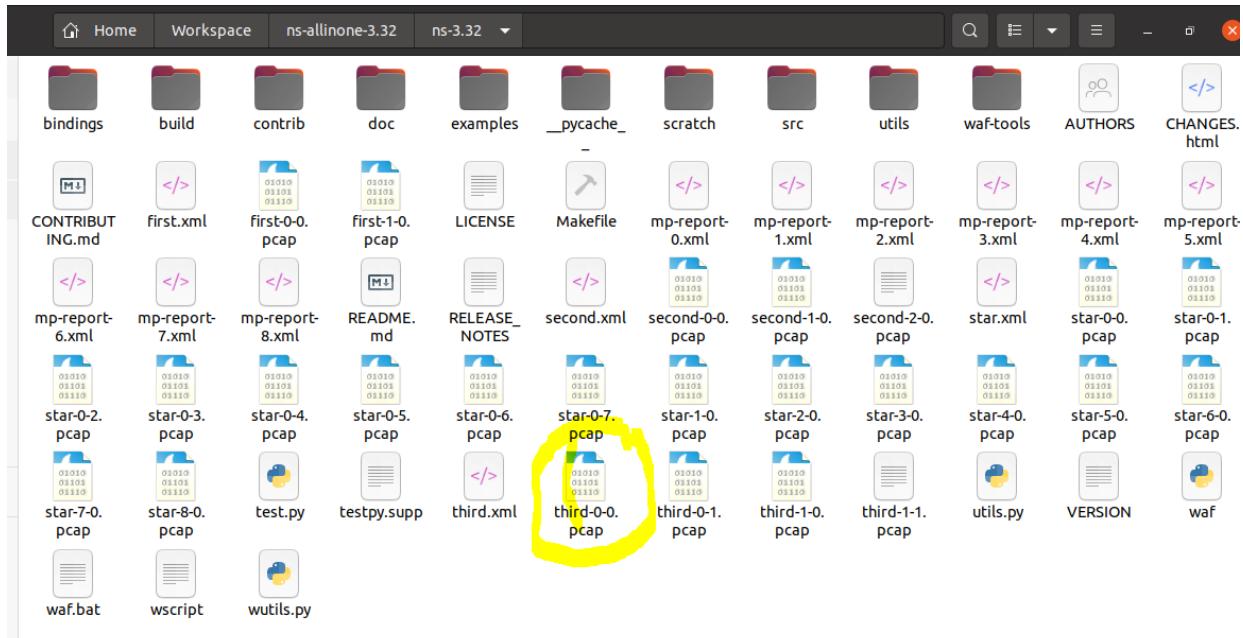
```

bvimit@bvimit:~/Workspace/ns-allinone-3.32/ns-3.32$ ./waf --run scratch/hybrid
Waf: Entering directory `/home/bvimit/Workspace/ns-allinone-3.32/ns-3.32/build'
[2844/2914] Compiling scratch/hybrid.cc
[2874/2914] Linking build/scratch/hybrid
Waf: Leaving directory `/home/bvimit/Workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (8.696s)
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 +2.03371s client received 1024 bytes from 10.1.2.4 port 9
bvimit@bvimit:~/Workspace/ns-allinone-3.32/ns-3.32$ ./waf --run scratch/hybrid --vis
Waf: Entering directory `/home/bvimit/Workspace/ns-allinone-3.32/ns-3.32/build'
Waf: Leaving directory `/home/bvimit/Workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (2.130s)
Could not load plugin 'show_last_packets.py': No module named 'kiwi.ui'
Could not load icon applets-screenshooter due to missing gnomedesktop Python module
scanning topology: 8 nodes...
scanning topology: calling graphviz layout
scanning topology: all done.

```



## With Wireshark



## Practical 13

### Program to simulate UDP server client

#### **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
 */
// Network topology
//
//      n0      n1
//      |        |
//      =====
//      LAN
//
// - UDP flows from n0 to n1 #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 ("UdpClientServerExample");

Int main (int argc, char *argv[])
{
//
// Enable logging for UdpClient and
//

LogComponentEnable ("UdpClient", LOG_LEVEL_INFO);
LogComponentEnable ("UdpServer", LOG_LEVEL_INFO);

bool useV6 = false;
Address serverAddress;
```

```

CommandLine cmd ( FILE );
cmd.AddValue ("useV6", "Use Ipv6", useV6);
cmd.Parse (argc, argv);
// Explicitly create the nodes required by the topology (shown above).
NS_LOG_INFO ("Create nodes.");

NodeContainer n;
n.Create (2);

InternetStackHelper internet;
internet.Install (n);

NS_LOG_INFO ("Create channels.");
// Explicitly create the channels required by the topology (shown above).
CsmaHelper csma;
csma.SetChannelAttribute ("DataRate", DataRateValue (DataRate (5000000)));
csma.SetChannelAttribute ("Delay", TimeValue (MilliSeconds (2)));
csma.SetDeviceAttribute ("Mtu", UintegerValue (1400));

NetDeviceContainer d = csma.Install (n);
// We've got the "hardware" in place. Now we need to add IP addresses. NS_LOG_INFO ("Assign IP
Addresses.");

if (useV6 == false)
{
Ipv4AddressHelper ipv4;
ipv4.SetBase ("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer i = ipv4.Assign (d);
serverAddress = 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 ("Create Applications.");
// Create one udpServer applications on node one.

uint16_t port = 4000;
UdpServerHelper server (port);
ApplicationContainer apps = server.Install (n.Get (1));
apps.Start (Seconds (1.0));
apps.Stop (Seconds (10.0));

// Create one UdpClient application to send UDP datagrams from node zero to node one.
uint32_t MaxPacketSize = 1024;

```

```

Time interPacketInterval = Seconds (0.05);
uint32_t maxPacketCount = 320;

UdpClientHelper client (serverAddress, port);
client.SetAttribute ("MaxPackets", UintegerValue (maxPacketCount));
client.SetAttribute ("Interval", TimeValue (interPacketInterval));
client.SetAttribute ("PacketSize", UintegerValue (MaxPacketSize));

apps = client.Install (n.Get (0));
apps.Start (Seconds (2.0));
apps.Stop (Seconds (10.0));

// Now, do the actual simulation.
//
NS_LOG_INFO ("Run Simulation.");
Simulator::Run ();
Simulator::Destroy ();
NS_LOG_INFO ("Done.");
}

```

## Output:

```

bvinit@bvinit-VirtualBox:~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run scratch/udp
Waf: Entering directory '/home/bvinit/workspace/ns-allinone-3.32/ns-3.32/build'
Waf: Leaving directory '/home/bvinit/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (2.058s)
TraceDelay TX 1024 bytes to 10.1.1.2 Uid: 0 Time: +2s
TraceDelay: RX 1012 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 1012 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
TraceDelay: RX 1012 bytes from 10.1.1.1 Sequence Number: 2 Uid: 14 TXtime: +2.1e+09ns RXtime: +2.10371e+09ns Delay: +3.712e+06ns
TraceDelay TX 1024 bytes to 10.1.1.2 Uid: 17 Time: +2.15s
TraceDelay: RX 1012 bytes from 10.1.1.1 Sequence Number: 3 Uid: 17 TXtime: +2.15e+09ns RXtime: +2.15371e+09ns Delay: +3.712e+06ns
TraceDelay TX 1024 bytes to 10.1.1.2 Uid: 20 Time: +2.2s
TraceDelay: RX 1012 bytes from 10.1.1.1 Sequence Number: 4 Uid: 20 TXtime: +2.2e+09ns RXtime: +2.20371e+09ns Delay: +3.712e+06ns
TraceDelay TX 1024 bytes to 10.1.1.2 Uid: 23 Time: +2.25s
TraceDelay: RX 1012 bytes from 10.1.1.1 Sequence Number: 5 Uid: 23 TXtime: +2.25e+09ns RXtime: +2.25371e+09ns Delay: +3.712e+06ns
TraceDelay TX 1024 bytes to 10.1.1.2 Uid: 26 Time: +2.3s
TraceDelay: RX 1012 bytes from 10.1.1.1 Sequence Number: 6 Uid: 26 TXtime: +2.3e+09ns RXtime: +2.30371e+09ns Delay: +3.712e+06ns
TraceDelay TX 1024 bytes to 10.1.1.2 Uid: 29 Time: +2.35s
TraceDelay: RX 1012 bytes from 10.1.1.1 Sequence Number: 7 Uid: 29 TXtime: +2.35e+09ns RXtime: +2.35371e+09ns Delay: +3.712e+06ns
TraceDelay TX 1024 bytes to 10.1.1.2 Uid: 32 Time: +2.4s
TraceDelay: RX 1012 bytes from 10.1.1.1 Sequence Number: 8 Uid: 32 TXtime: +2.4e+09ns RXtime: +2.40371e+09ns Delay: +3.712e+06ns
TraceDelay TX 1024 bytes to 10.1.1.2 Uid: 35 Time: +2.45s
TraceDelay: RX 1012 bytes from 10.1.1.1 Sequence Number: 9 Uid: 35 TXtime: +2.45e+09ns RXtime: +2.45371e+09ns Delay: +3.712e+06ns
TraceDelay TX 1024 bytes to 10.1.1.2 Uid: 38 Time: +2.5s
TraceDelay: RX 1012 bytes from 10.1.1.1 Sequence Number: 10 Uid: 38 TXtime: +2.5e+09ns RXtime: +2.50371e+09ns Delay: +3.712e+06ns
TraceDelay TX 1024 bytes to 10.1.1.2 Uid: 41 Time: +2.55s
TraceDelay: RX 1012 bytes from 10.1.1.1 Sequence Number: 11 Uid: 41 TXtime: +2.55e+09ns RXtime: +2.55371e+09ns Delay: +3.712e+06ns
TraceDelay TX 1024 bytes to 10.1.1.2 Uid: 44 Time: +2.6s
TraceDelay: RX 1012 bytes from 10.1.1.1 Sequence Number: 12 Uid: 44 TXtime: +2.6e+09ns RXtime: +2.60371e+09ns Delay: +3.712e+06ns
TraceDelay TX 1024 bytes to 10.1.1.2 Uid: 47 Time: +2.65s
TraceDelay: RX 1012 bytes from 10.1.1.1 Sequence Number: 13 Uid: 47 TXtime: +2.65e+09ns RXtime: +2.65371e+09ns Delay: +3.712e+06ns
TraceDelay TX 1024 bytes to 10.1.1.2 Uid: 50 Time: +2.7s
TraceDelay: RX 1012 bytes from 10.1.1.1 Sequence Number: 14 Uid: 50 TXtime: +2.7e+09ns RXtime: +2.70371e+09ns Delay: +3.712e+06ns
TraceDelay TX 1024 bytes to 10.1.1.2 Uid: 53 Time: +2.75s
TraceDelay: RX 1012 bytes from 10.1.1.1 Sequence Number: 15 Uid: 53 TXtime: +2.75e+09ns RXtime: +2.75371e+09ns Delay: +3.712e+06ns
TraceDelay TX 1024 bytes to 10.1.1.2 Uid: 56 Time: +2.8s
TraceDelay: RX 1012 bytes from 10.1.1.1 Sequence Number: 16 Uid: 56 TXtime: +2.8e+09ns RXtime: +2.80371e+09ns Delay: +3.712e+06ns
TraceDelay TX 1024 bytes to 10.1.1.2 Uid: 59 Time: +2.85s

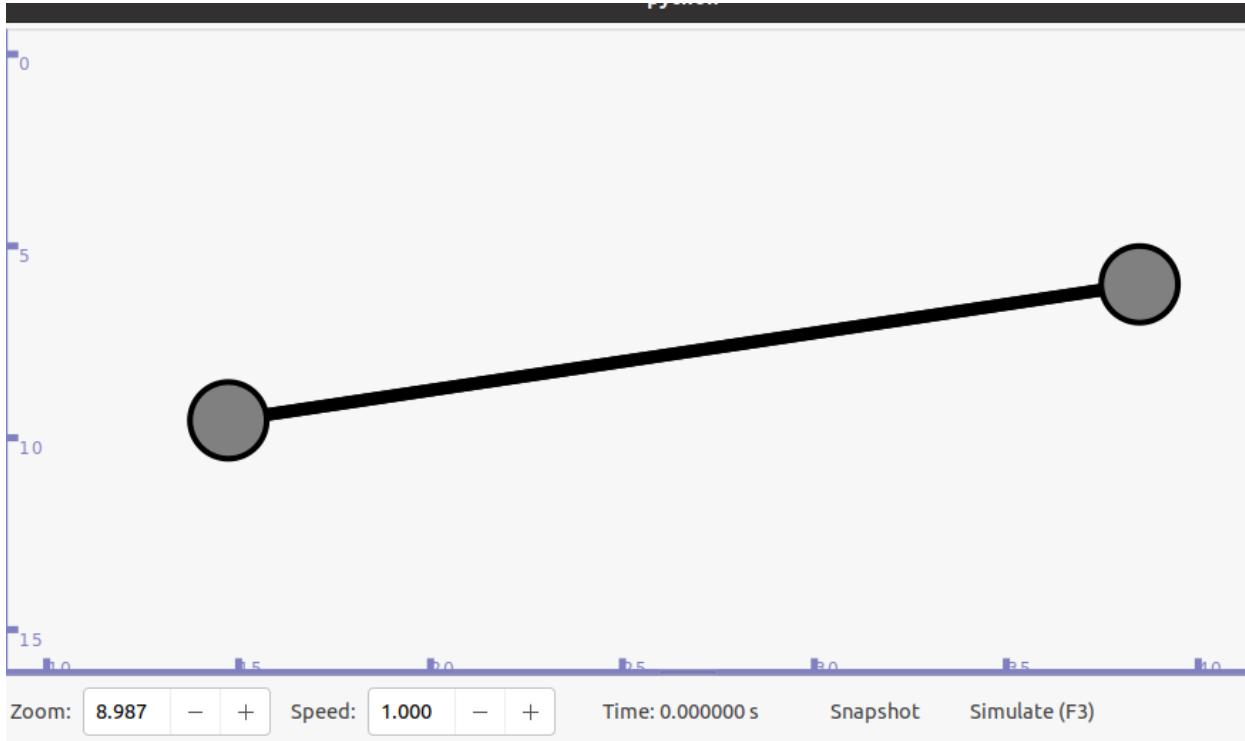
```

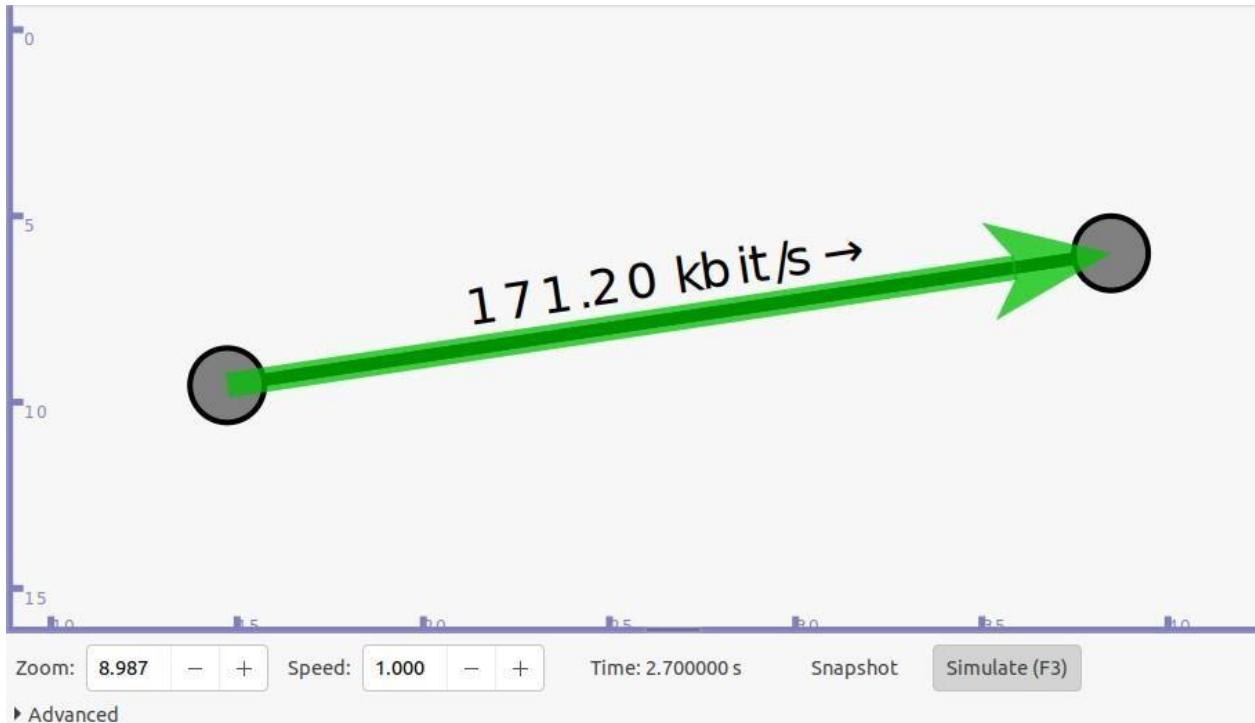
Using --vis

```

$ ./waf --run scratch/udp --vis

```





For command line input:

`./waf --run "scratch/udp --use V6=true"`

```
bvimit@bvimit-VirtualBox:~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run "scratch/udp --useV6=true"
Waf: Entering directory '/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
Waf: Leaving directory '/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (2.049s)
TraceDelay TX 1024 bytes to 2001:0:f00d:cafe:200:ff:fe00:2 Uid: 18 Time: +2s
TraceDelay: RX 1012 bytes from 2001:0:f00d:cafe:200:ff:fe00:1 Sequence Number: 0 Uid: 18 TXtime: +2e+09ns RXtime: +2.00803e+09ns Delay: +8.034e+06ns
TraceDelay TX 1024 bytes to 2001:0:f00d:cafe:200:ff:fe00:2 Uid: 27 Time: +2.05s
TraceDelay: RX 1012 bytes from 2001:0:f00d:cafe:200:ff:fe00:1 Sequence Number: 1 Uid: 27 TXtime: +2.05e+09ns RXtime: +2.05374e+09ns Delay: +3.744e+06ns
TraceDelay TX 1024 bytes to 2001:0:f00d:cafe:200:ff:fe00:2 Uid: 30 Time: +2.1s
TraceDelay: RX 1012 bytes from 2001:0:f00d:cafe:200:ff:fe00:1 Sequence Number: 2 Uid: 30 TXtime: +2.1e+09ns RXtime: +2.10374e+09ns Delay: +3.744e+06ns
TraceDelay TX 1024 bytes to 2001:0:f00d:cafe:200:ff:fe00:2 Uid: 33 Time: +2.15s
TraceDelay: RX 1012 bytes from 2001:0:f00d:cafe:200:ff:fe00:1 Sequence Number: 3 Uid: 33 TXtime: +2.15e+09ns RXtime: +2.15374e+09ns Delay: +3.744e+06ns
TraceDelay TX 1024 bytes to 2001:0:f00d:cafe:200:ff:fe00:2 Uid: 36 Time: +2.2s
TraceDelay: RX 1012 bytes from 2001:0:f00d:cafe:200:ff:fe00:1 Sequence Number: 4 Uid: 36 TXtime: +2.2e+09ns RXtime: +2.20374e+09ns Delay: +3.744e+06ns
TraceDelay TX 1024 bytes to 2001:0:f00d:cafe:200:ff:fe00:2 Uid: 39 Time: +2.25s
TraceDelay: RX 1012 bytes from 2001:0:f00d:cafe:200:ff:fe00:1 Sequence Number: 5 Uid: 39 TXtime: +2.25e+09ns RXtime: +2.25374e+09ns Delay: +3.744e+06ns
TraceDelay TX 1024 bytes to 2001:0:f00d:cafe:200:ff:fe00:2 Uid: 42 Time: +2.3s
TraceDelay: RX 1012 bytes from 2001:0:f00d:cafe:200:ff:fe00:1 Sequence Number: 6 Uid: 42 TXtime: +2.3e+09ns RXtime: +2.30374e+09ns Delay: +3.744e+06ns
TraceDelay TX 1024 bytes to 2001:0:f00d:cafe:200:ff:fe00:2 Uid: 45 Time: +2.35s
TraceDelay: RX 1012 bytes from 2001:0:f00d:cafe:200:ff:fe00:1 Sequence Number: 7 Uid: 45 TXtime: +2.35e+09ns RXtime: +2.35374e+09ns Delay: +3.744e+06ns
```

## **Practical 14**

### **Program to simulate DHCP server and n clients**

#### **Code:**

```
#include "ns3/core-module.h"
#include "ns3/network-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[])
{
    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"),
Ipv4Address ("172.30.0.0"), Ipv4Mask ("/24"),
Ipv4Address ("172.30.0.10"), Ipv4Address ("172.30.0.15"),
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);
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. ");
}

```

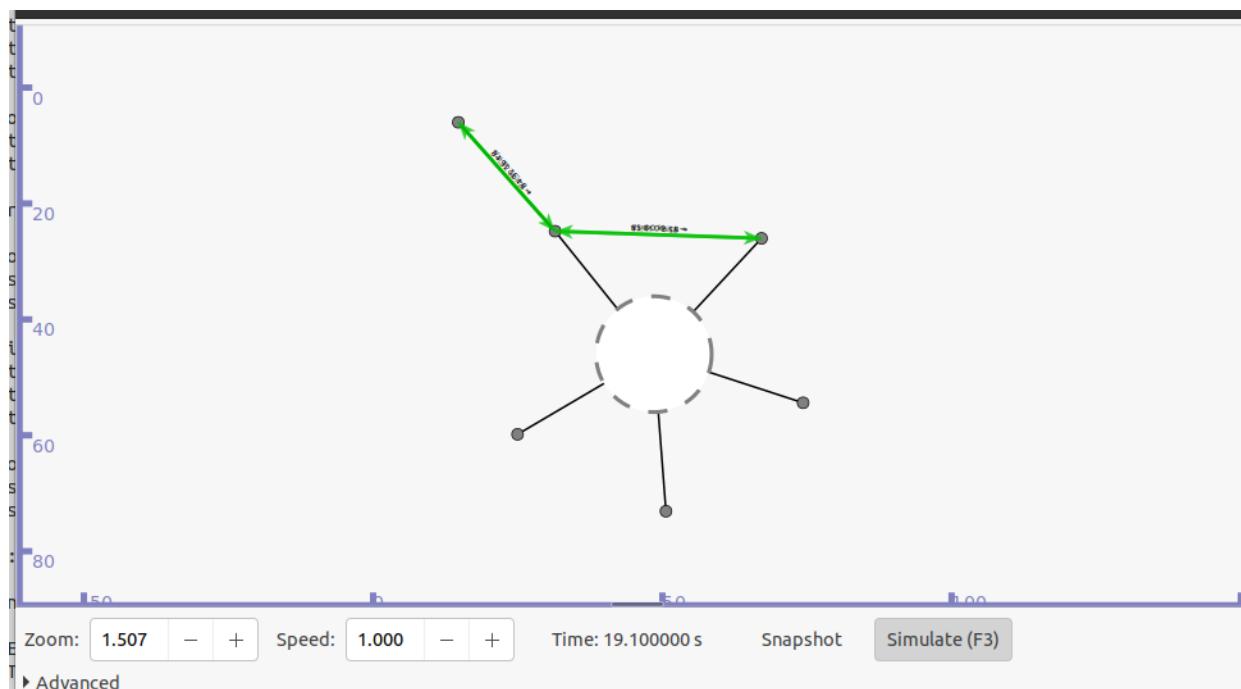
**Output:**

```

bvimit@bvimit-VirtualBox:~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run scratch/dhcp
Waf: Entering directory `/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
[2843/2928] Compiling scratch/dhcp.cc
[2844/2928] Compiling scratch/animation.cc
[2845/2928] Compiling scratch/second.cc
[2846/2928] Compiling scratch/scratch-simulator.cc
[2877/2928] Compiling scratch/mesh1.cc
[2878/2928] Linking build/scratch/scratch-simulator
[2879/2928] Compiling scratch/bus.cc
[2880/2928] Linking build/scratch/second
[2881/2928] Linking build/scratch/dhcp
[2882/2928] Compiling scratch/abc.cc
[2883/2928] Compiling scratch/star.cc
[2884/2928] Linking build/scratch/animation
[2885/2928] Linking build/scratch/mesh1
[2886/2928] Linking build/scratch/bus
[2887/2928] Linking build/scratch/abc
[2888/2928] Linking build/scratch/star
Waf: Leaving directory `/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (14.340s)
bvimit@bvimit-VirtualBox:~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run scratch/dhcp --vis

```

## Using --vis



## **Practical 15** **Program to simulate FTP using TCP protocol**

### **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/csma-module.h"
#include "ns3/ipv4-global-routing-helper.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("FTP_Simulation");

int main(int argc, char *argv[])
{
    Time::SetResolution (Time::NS);
    LogComponentEnable ("OnOffApplication", LOG_LEVEL_INFO);
    LogComponentEnable ("OnOffApplication", LOG_LEVEL_INFO);

    NodeContainer csmaNodes;
    csmaNodes.Create(2);

    // using csma helper to configure attributes
    // and creating connectivity between two nodes
    CsmaHelper csma;

    // set channel attributes for device
    csma.SetChannelAttribute ("DataRate",StringValue("1Mbps"));
    csma.SetChannelAttribute ("Delay",StringValue("20ms"));

    // installing nodes to create, configure, install devices of required type
    NetDeviceContainer csmaDevices;
    csmaDevices = csma.Install(csmaNodes);
    // installing protocol stack in nodes; once executed will install TCP, UDP, IP etc. on each node of the
    // nodeContainer
    InternetStackHelper stack;
    stack.Install(csmaNodes);

    // assigning IP address to the devices, setting base address here
    Ipv4AddressHelper address;
    address.SetBase ("10.10.10.0", "255.255.255.0");
    Ipv4InterfaceContainer csmaInterfaces = address.Assign(csmaDevices);

    // assigning ip address to the devices, since devices contain nodes, while ip stack is created within a
    // node
```

```

// uint32_t port_num=21; // port number for Rx

Address RxAddress(InetSocketAddress(csmaInterfaces.GetAddress(1), 21)); //Rx obtaining a IPv4
//address, and the port number to listen

PacketSinkHelper sinkHelper ("ns3::TcpSocketFactory", RxAddress); // PacketSinkHelper is a class
//reference for creating socket references and passing address at which Rx is instantiated

ApplicationContainer sinkApp = sinkHelper.Install (csmaNodes.Get(1)); // ApplicationContainer,
//installs application on the node, in our case it is packet sink helper ; node here is the server node
sinkApp.Start (Seconds (1.0));
sinkApp.Stop (Seconds (10.0));

//creating a TCP transmitter
// set up address and port number to send TCP traffic to the server
Address TxAddress(InetSocketAddress(csmaInterfaces.GetAddress(1),21));

OnOffHelper clientHelper ("ns3::TcpSocketFactory", TxAddress);
// setting up attributes for onoff application helper

clientHelper.SetAttribute("DataRate",StringValue("1Mbps"));
clientHelper.SetAttribute("PacketSize",UintegerValue(1280));

ApplicationContainer Tx = clientHelper.Install (csmaNodes.Get (0));
Tx.Start (Seconds (1.0));
Tx.Stop (Seconds (10.0));

csma.EnablePcap("ftpeg",csmaDevices.Get(1),true);

AsciiTraceHelper ascii;
csma.EnableAsciiAll(ascii.CreateFileStream ("ftp_trace.tr"));

NS_LOG_INFO ("Run Simulation.");
Simulator::Run ();
NS_LOG_INFO ("Done");
return(0);
}

```

**Output:**

```
bvimit@bvimit-VirtualBox:~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run scratch/ftp
Waf: Entering directory `/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
Waf: Leaving directory `/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (2.040s)
At time +2.01024s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 1280 bytes
At time +2.02048s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 2560 bytes
At time +2.03072s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 3840 bytes
At time +2.04096s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 5120 bytes
At time +2.0512s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 6400 bytes
At time +2.06144s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 7680 bytes
At time +2.07168s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 8960 bytes
At time +2.08192s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 10240 bytes
At time +2.09216s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 11520 bytes
At time +2.1024s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 12800 bytes
At time +2.11264s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 14080 bytes
At time +2.12288s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 15360 bytes
At time +2.13312s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 16640 bytes
At time +2.14336s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 17920 bytes
At time +2.1536s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 19200 bytes
At time +2.16384s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 20480 bytes
At time +2.17408s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 21760 bytes
At time +2.18432s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 23040 bytes
At time +2.19456s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 24320 bytes
At time +2.2048s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 25600 bytes
At time +2.21504s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 26880 bytes
At time +2.22528s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 28160 bytes
At time +2.23552s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 29440 bytes
At time +2.24576s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 30720 bytes
At time +2.256s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 32000 bytes
At time +2.26624s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 33280 bytes
At time +2.27648s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 34560 bytes
At time +2.28672s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 35840 bytes
At time +2.29696s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 37120 bytes
At time +2.3072s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 38400 bytes
At time +2.31744s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 39680 bytes
At time +2.32768s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 40960 bytes
At time +2.33792s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 42240 bytes
```

```
bvimit@bvimit-VirtualBox:~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run scratch/ftp --vis
Waf: Entering directory `/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
Waf: Leaving directory `/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (2.057s)
At time +2.01024s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 1280 bytes
At time +2.02048s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 2560 bytes
At time +2.03072s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 3840 bytes
At time +2.04096s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 5120 bytes
At time +2.0512s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 6400 bytes
At time +2.06144s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 7680 bytes
At time +2.07168s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 8960 bytes
At time +2.08192s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 10240 bytes
At time +2.09216s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 11520 bytes
At time +2.1024s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 12800 bytes
At time +2.11264s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 14080 bytes
At time +2.12288s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 15360 bytes
At time +2.13312s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 16640 bytes
At time +2.14336s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 17920 bytes
At time +2.1536s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 19200 bytes
At time +2.16384s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 20480 bytes
At time +2.17408s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 21760 bytes
At time +2.18432s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 23040 bytes
At time +2.19456s on-off application sent 1280 bytes to 10.10.10.2 port 21 total Tx 24320 bytes
```

## Practical 16

### Program to simulate Three-way handshake for TCP connection using NetAnim

#### **Code:**

```
#include <iostream>
#include <fstream>
#include <string>
#include <cassert>

#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/ipv4-global-routing-helper.h"
#include "ns3/netanim-module.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("TcpServer");

int main (int argc, char *argv[])
{
// Users may find it convenient to turn on explicit debugging
// for selected modules; the below lines suggest how to do this
//LogComponentEnable ("TcpServer", LOG_LEVEL_INFO);
//LogComponentEnable ("TcpL4Protocol", LOG_LEVEL_ALL);
//LogComponentEnable ("TcpSocketImpl", LOG_LEVEL_ALL);
//LogComponentEnable ("PacketSink", LOG_LEVEL_ALL);

// Set up some default values for the simulation.
Config::SetDefault ("ns3::OnOffApplication::PacketSize", UintegerValue (250));
Config::SetDefault ("ns3::OnOffApplication::DataRate", StringValue ("5kb/s"));
uint32_t N = 5; //number of nodes in the star
// Allow the user to override any of the defaults and the above
// Config::SetDefault()s at run-time, via command-line arguments CommandLine cmd;
cmd.AddValue ("nNodes", "Number of nodes to place in the star", N); cmd.Parse (argc, argv);
// Here, we will create N nodes in a star.

NS_LOG_INFO ("Create nodes.");

NodeContainer serverNode;
NodeContainer clientNodes;
serverNode.Create (1);
clientNodes.Create (N-1);
NodeContainer allNodes = NodeContainer (serverNode, clientNodes);
// Install network stacks on the nodes
```

```

InternetStackHelper internet;
internet.Install (allNodes);

//Collect an adjacency list of nodes for the p2p topology
std::vector<NodeContainer> nodeAdjacencyList (N-1);

for(uint32_t i=0; i<nodeAdjacencyList.size (); ++i)
{
nodeAdjacencyList[i] = NodeContainer (serverNode, clientNodes.Get (i));
}

// We create the channels first without any IP addressing information
NS_LOG_INFO ("Create channels.");

PointToPointHelper p2p;
p2p.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
p2p.SetChannelAttribute ("Delay", StringValue ("2ms"));

std::vector<NetDeviceContainer> deviceAdjacencyList (N-1);

for(uint32_t i=0; i<deviceAdjacencyList.size (); ++i)
{
deviceAdjacencyList[i] = p2p.Install (nodeAdjacencyList[i]);
}

// Later, we add IP addresses.
NS_LOG_INFO ("Assign IP Addresses.");
Ipv4AddressHelper ipv4;

std::vector<Ipv4InterfaceContainer> interfaceAdjacencyList (N-1);

for(uint32_t i=0; i<interfaceAdjacencyList.size (); ++i)
{
std::ostringstream subnet; subnet<<"10.1."<<i+1<<".0";
ipv4.SetBase (subnet.str (), "255.255.255.0");
interfaceAdjacencyList[i] = ipv4.Assign (deviceAdjacencyList[i]);
}

//Turn on global static routing Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

// Create a packet sink on the star "hub" to receive these packets
uint16_t port = 50000;

Address sinkLocalAddress (InetSocketAddress (Ipv4Address::GetAny (), port));
PacketSinkHelper sinkHelper ("ns3::TcpSocketFactory", sinkLocalAddress);
ApplicationContainer sinkApp = sinkHelper.Install (serverNode);
sinkApp.Start (Seconds (1.0));
sinkApp.Stop (Seconds (10.0));

```

```

// Create the OnOff applications to send TCP to the server OnOffHelper clientHelper
("ns3::TcpSocketFactory", Address ());
clientHelper.SetAttribute ("OnTime", StringValue ("ns3::ConstantRandomVariable[Constant=1]"));
clientHelper.SetAttribute ("OffTime", StringValue ("ns3::ConstantRandomVariable[Constant=0]"));

//normally wouldn't need a loop here but the server IP address is different
//on each p2p subnet
ApplicationContainer clientApps;

for(uint32_t i=0; i<clientNodes.GetN (); ++i)
{
    AddressValue remoteAddress
    (InetSocketAddress (interfaceAdjacencyList[i].GetAddress (0), port));
    clientHelper.SetAttribute ("Remote", remoteAddress);
    clientApps.Add (clientHelper.Install (clientNodes.Get (i)));
}

clientApps.Start (Seconds (1.0));
clientApps.Stop (Seconds (10.0));

//configure tracing AsciiTraceHelper ascii;
p2p.EnableAsciiAll (ascii.CreateFileStream ("tcp-star-server.tr"));
p2p.EnablePcapAll ("tcp-star-server");

NS_LOG_INFO ("Run Simulation.");
AnimationInterface anim ("Anim.xml");
anim.EnablePacketMetadata (true);
Simulator::Run ();
Simulator::Destroy ();
NS_LOG_INFO ("Done.");
return 0;
}

```

## Output:

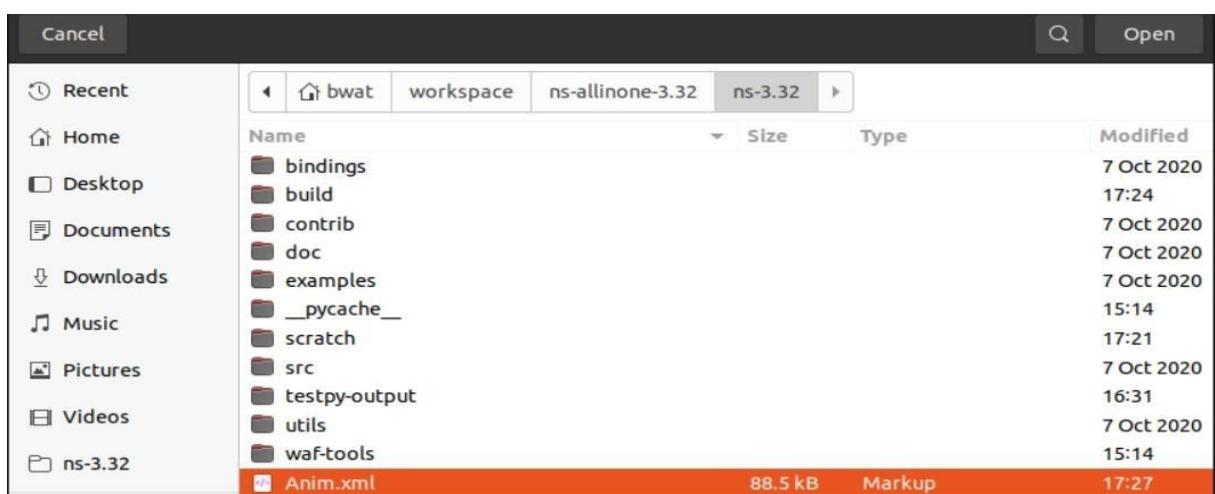
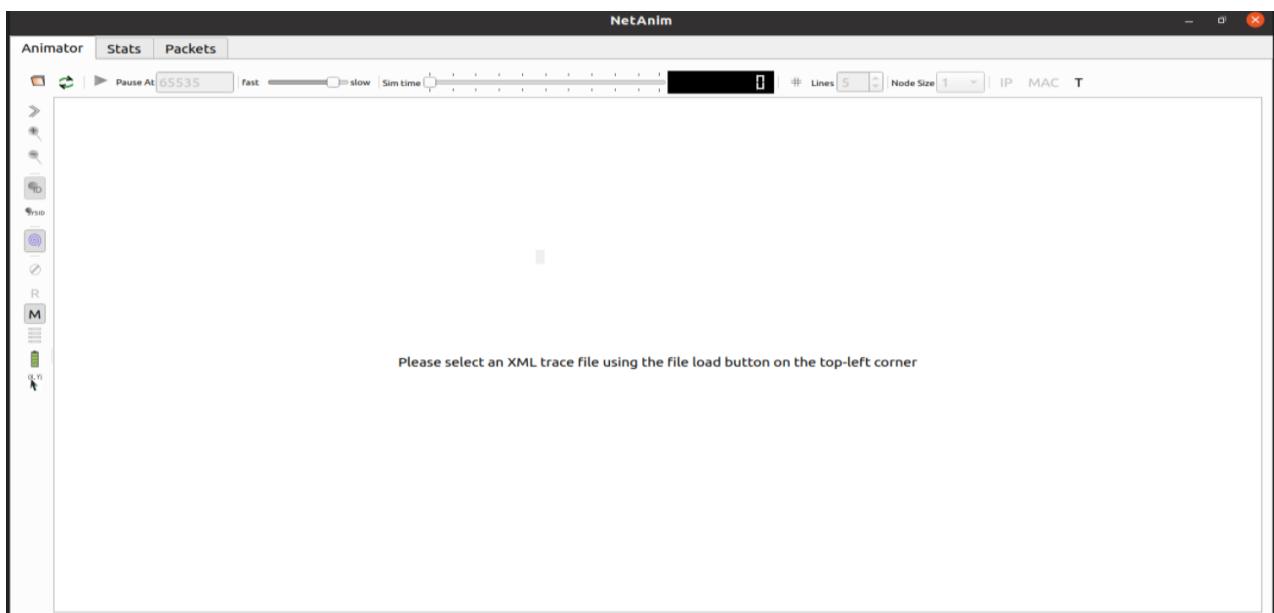
```

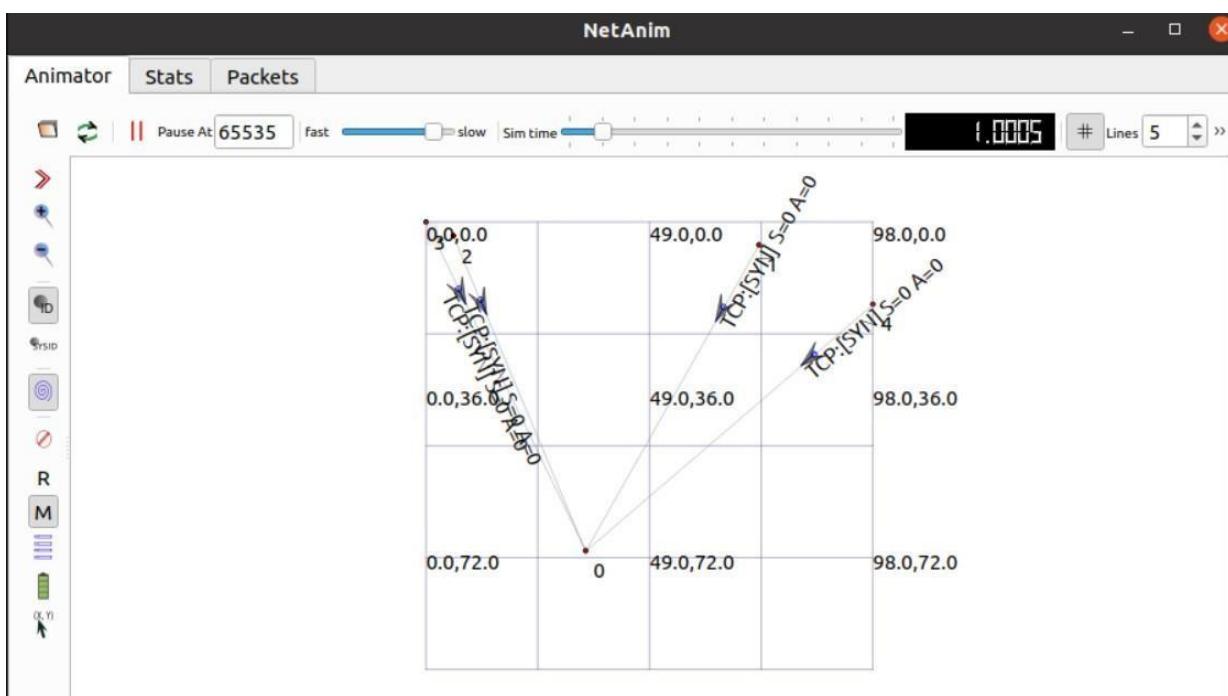
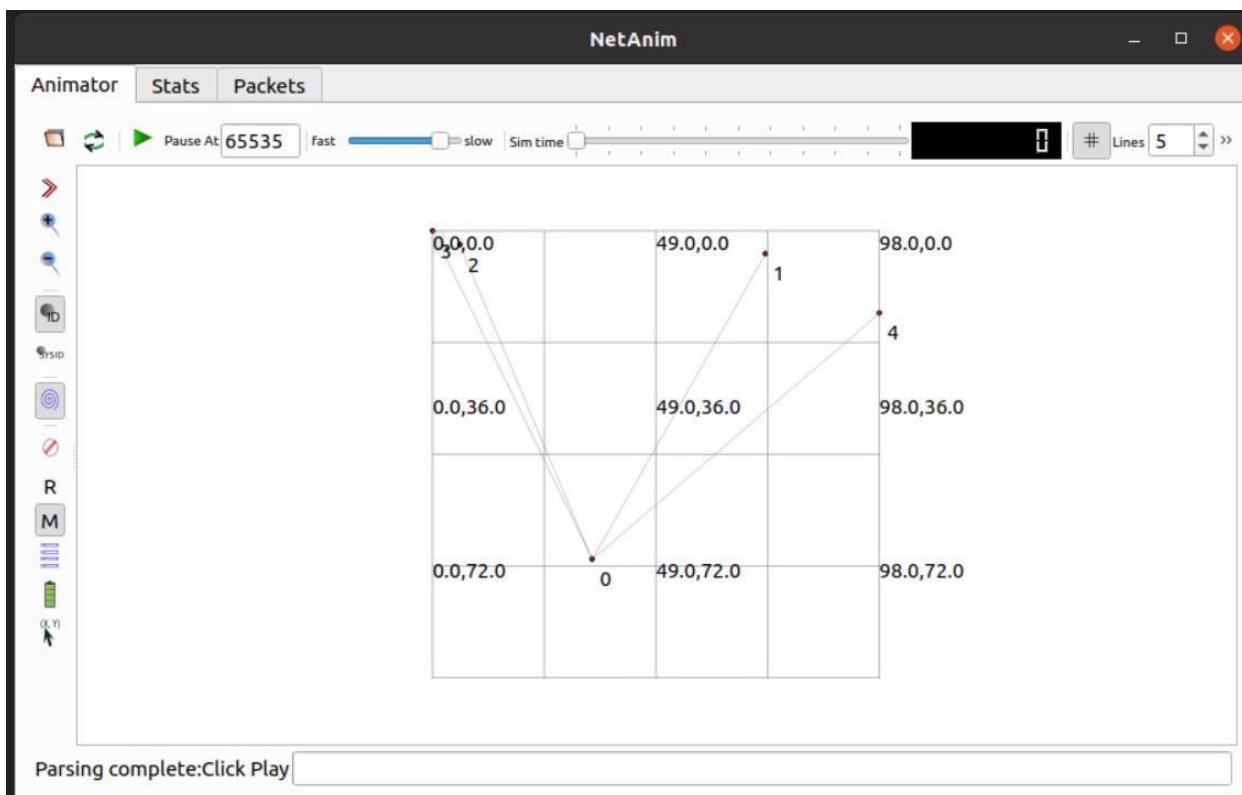
bwat@bwat-ubuntu: ~/workspace/ns-allinone-3.32/ns-3.32
bwat@bwat-ubuntu: ~/workspace/ns-allinone-3.32/ns-3.32
bwat@bwat-ubuntu: ~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run "scratch/threewayhandshake.cc"
Waf: Entering directory '/home/bwat/workspace/ns-allinone-3.32/ns-3.32/build'
Waf: Leaving directory '/home/bwat/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (3.01s)

AnimationInterface WARNING:Node:0 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:1 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:2 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:3 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:4 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:0 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:1 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:2 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:3 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:4 Does not have a mobility model. Use SetConstantPosition if it is stationary

```

## ./NetAnim

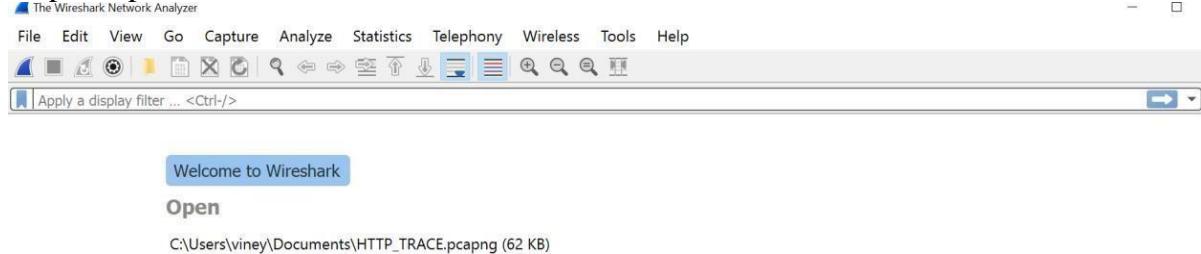




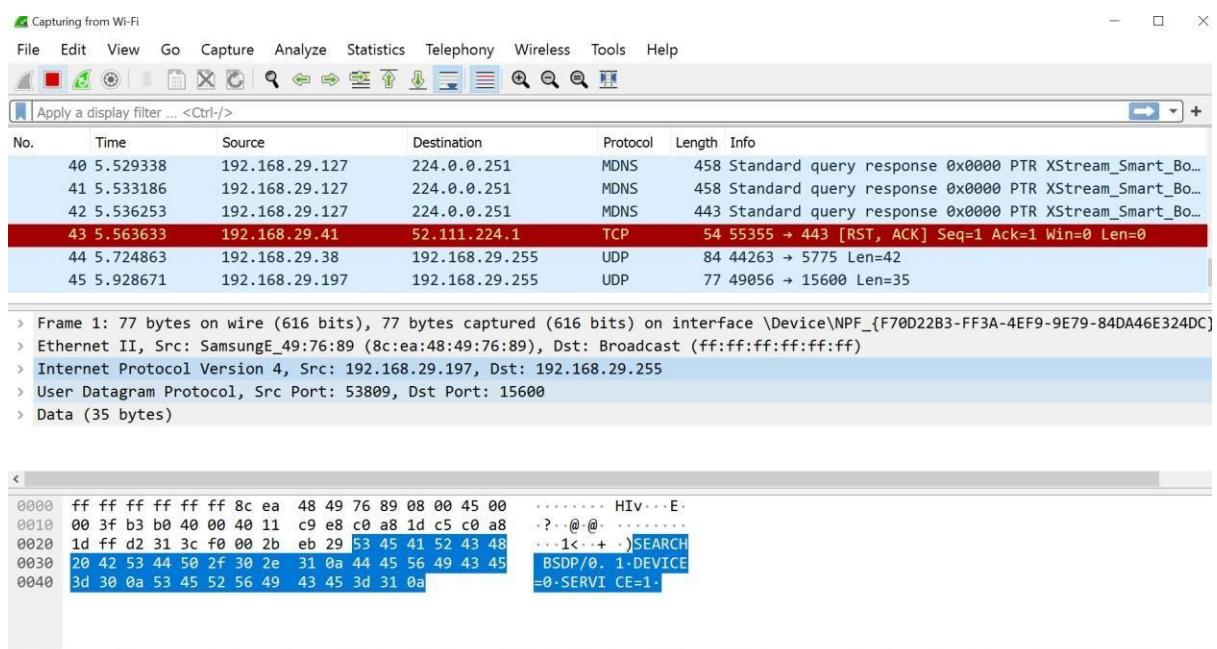
## Practical 17

### Analyze the network traffic using Wireshark

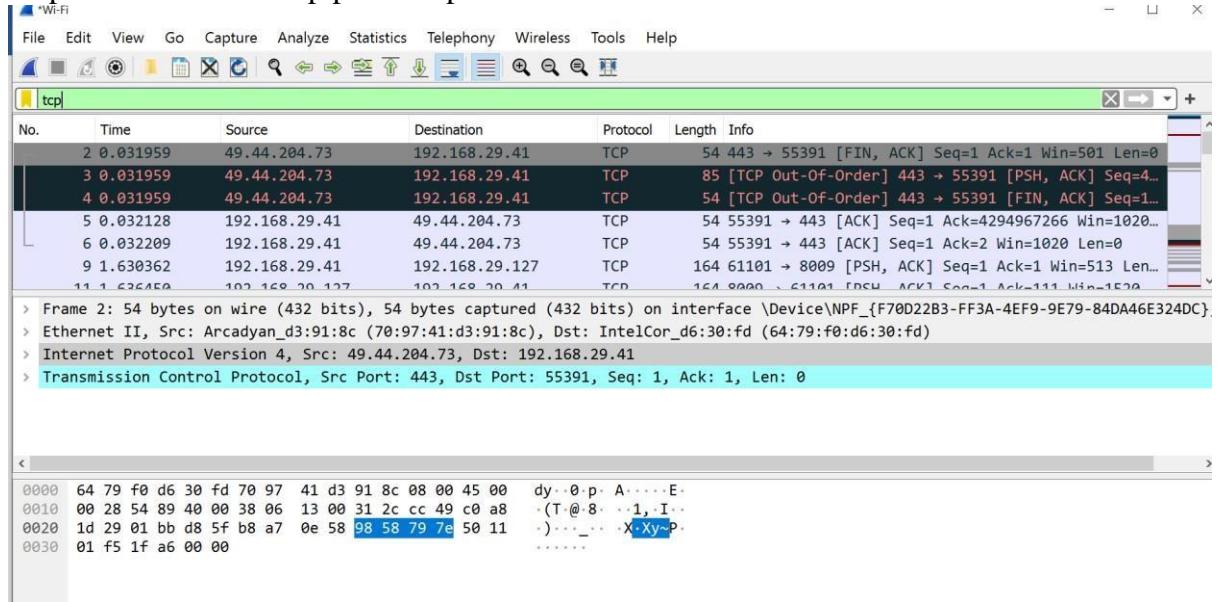
Step 1: Open Wireshark and select the network connection which is Wifi in this case.



Step 2: Start capturing packets which will record the browsing history. And then stop capturing.



### Step 3: Search for all tcp protocol packets.



You can select any packets to get the information about that.

```
> Frame 413: 289 bytes on wire (2312 bits), 289 bytes captured (2312 bits) on interface \Device\NPF_{F70D22B3-FF3A-4EF9-9E79-84DA46E324DC}
> Ethernet II, Src: IntelCor_d6:30:fd (64:79:f0:d6:30:fd), Dst: SamsungE_49:76:89 (8c:ea:48:49:76:89)
> Internet Protocol Version 4, Src: 192.168.29.41, Dst: 192.168.29.197
> Transmission Control Protocol, Src Port: 55438, Dst Port: 7678, Seq: 1, Ack: 1, Len: 235
> Hypertext Transfer Protocol
```

Then expand Ethernet column to get additional details.

```
> Frame 2: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{F70D22B3-FF3A-4EF9-9E79-84DA46E324DC}
  ↓
Ethernet II, Src: Arcadyan_d3:91:8c (70:97:41:d3:91:8c), Dst: IntelCor_d6:30:fd (64:79:f0:d6:30:fd)
  > Destination: IntelCor_d6:30:fd (64:79:f0:d6:30:fd)
  > Source: Arcadyan_d3:91:8c (70:97:41:d3:91:8c)
  Type: IPv4 (0x0800)
```

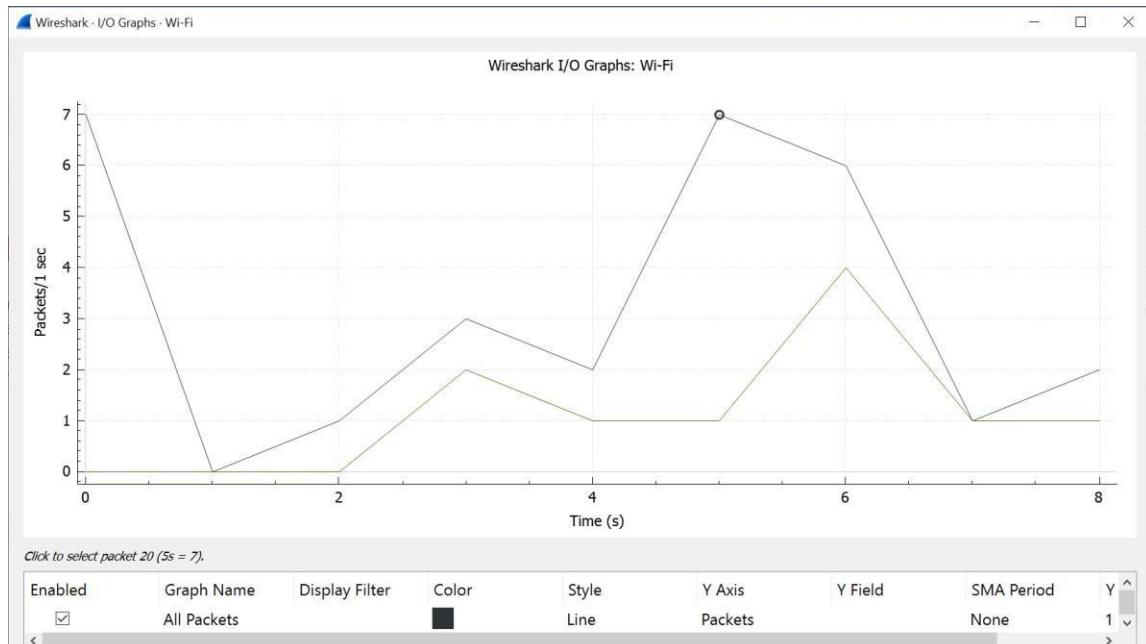
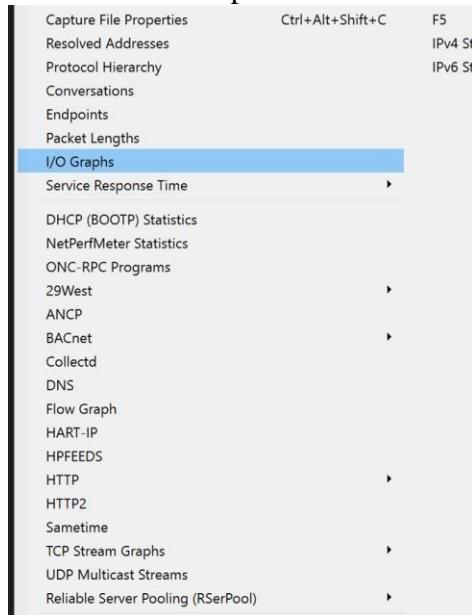
Then expand Transmission Control Protocol column to get additional details.

```
Transmission Control Protocol, Src Port: 443, Dst Port: 55391, Seq: 1, Ack: 1, Len: 0
  Source Port: 443
  Destination Port: 55391
  [Stream index: 0]
  [Conversation completeness: Incomplete (28)]
  [TCP Segment Len: 0]
  Sequence Number: 1 (relative sequence number)
  Sequence Number (raw): 3097955928
  [Next Sequence Number: 2 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 2555935102
  0101 .... = Header Length: 20 bytes (5)
```

You can analyze the statistics graphically. Click on the statistics option.

Capture Analyze Statistics Telephony Wireless Tools Help

Click on I/O Graphs.



#### Step 4: Search for all http protocol packets.

The screenshot shows the Wireshark interface with the following details:

- Protocol:** http
- Packets:** 123 (total)
- Selected Packet:** 413 (Time: 9.470605, Source: 192.168.29.41, Destination: 192.168.29.197, Protocol: HTTP, Length: 289, Info: GET /nservice/ HTTP/1.1)
- Details Section:** Shows the selected packet's details, including destination, source, and protocol information.
- Bytes Section:** Shows the raw hex and ASCII representation of the selected packet's bytes.

#### Step 5: Search for all http request protocol packets with GET method.

The screenshot shows the Wireshark interface with the following details:

- Protocol:** http.request.method==GET
- Packets:** 123 (total)
- Selected Packet:** 4278 (Time: 21.565457, Source: 192.168.29.41, Destination: 103.235.105.107, Protocol: HTTP, Length: 483, Info: GET /bvimit/images/placedStudents2019/snehal\_sh...)
- Details Section:** Shows the selected packet's details, including destination, source, and protocol information.
- Bytes Section:** Shows the raw hex and ASCII representation of the selected packet's bytes.

## Step 6: Search for all http response protocol packets.

\*Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http.response

No.	Time	Source	Destination	Protocol	Length	Info
4285	21.581655	103.235.105.107	192.168.29.41	HTTP	767	HTTP/1.1 200 OK (text/html)
4325	21.614663	103.235.105.107	192.168.29.41	HTTP	203	HTTP/1.1 200 OK (PNG)
4397	21.689086	103.235.105.107	192.168.29.41	HTTP	1445	HTTP/1.1 200 OK (JPEG JFIF image)
4411	21.694812	103.235.105.107	192.168.29.41	HTTP	974	HTTP/1.1 200 OK (JPEG JFIF image)
4522	21.781536	103.235.105.107	192.168.29.41	HTTP	471	HTTP/1.1 200 OK (JPEG JFIF image)
4552	21.839024	103.235.105.107	192.168.29.41	HTTP	820	HTTP/1.1 200 OK (GIF89a)
4587	21.891280	103.235.105.107	192.168.29.41	HTTP	92	HTTP/1.1 200 OK (JPEG JFIF image)
4610	21.915844	103.235.105.107	192.168.29.41	HTTP	1094	HTTP/1.1 200 OK (JPEG JFIF image)
4975	22.232526	103.235.105.107	192.168.29.41	HTTP	600	HTTP/1.1 200 OK (JPEG JFIF image)

```

> Destination: IntelCor_d6:30:fd (64:79:f0:d6:30:fd)
> Source: EardaTec_99:4a:94 (48:7e:48:99:4a:94)
Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 192.168.29.38, Dst: 192.168.29.41
> User Datagram Protocol, Src Port: 1900, Dst Port: 60884
<
0000  64 79 f0 d6 30 fd 48 7e 48 99 4a 94 08 00 45 00  dy..0 H~ H.J...E.
0010  01 04 c8 5b 40 00 40 11 b5 ed c0 a8 1d 26 c0 a8  ...[@ @. ....&..
0020  1d 29 07 6c ed d4 00 f0 6e c2 48 54 54 50 2f 31  .).l.... n. HTTP/1
0030  2e 31 20 32 30 30 20 4f 4b 0d 0a 4c 4f 43 41 54  .1 200 O K.. LOCAT
0040  49 4f 4e 3a 20 68 74 74 70 3a 2f 2f 31 39 32 2e  ION: htt p://192.
0050  31 36 38 2e 32 39 2e 33 38 3a 35 32 32 33 35 2f  168.29.3 8:52235/
0060  64 64 2e 78 6d 6c 0d 0a 43 41 43 48 45 2d 43 4f  dd.xml.. CACHE-CO

```

## Practical 18 – Mini Project

### Mixed Routing

**Aim:** Mixed routing refers to a networking approach that combines different routing protocols or techniques within a single network infrastructure

**Objective:** The motivation behind mixed routing is to leverage the strengths of different routing protocols and adapt them to specific requirements within a network

#### Code:

```
// This script exercises global routing code in a mixed point-to-point
// and csma/cd environment
//
// Network topology
//
// n0
//   \ p-p
//   \   (shared csma/cd)
//   n2 ----- n3
//   /     |     |
//   / p-p   n4   n5 ----- n6
// n1           p-p
//
// - CBR/UDP flows from n0 to n6
// - Tracing of queues and packet receptions to file "mixed-global-routing.tr"

#include <iostream>
#include <fstream>
#include <string>
#include <cassert>

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

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("MixedGlobalRoutingExample");

int
main (int argc, char *argv[])
{
    Config::SetDefault ("ns3::OnOffApplication::PacketSize", UintegerValue (210));
    Config::SetDefault ("ns3::OnOffApplication::DataRate", StringValue ("448kb/s"));

    // Allow the user to override any of the defaults and the above
    // Bind ()s at run-time, via command-line arguments
    CommandLine cmd (__FILE__);
    cmd.Parse (argc, argv);
```

```

NS_LOG_INFO ("Create nodes.");
NodeContainer c;
c.Create (7);
NodeContainer n0n2 = NodeContainer (c.Get (0), c.Get (2));
NodeContainer n1n2 = NodeContainer (c.Get (1), c.Get (2));
NodeContainer n5n6 = NodeContainer (c.Get (5), c.Get (6));
NodeContainer n2345 = NodeContainer (c.Get (2), c.Get (3), c.Get (4), c.Get (5));

InternetStackHelper internet;
internet.Install (c);

// We create the channels first without any IP addressing information
NS_LOG_INFO ("Create channels.");
PointToPointHelper p2p;
p2p.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
p2p.SetChannelAttribute ("Delay", StringValue ("2ms"));
NetDeviceContainer d0d2 = p2p.Install (n0n2);

NetDeviceContainer d1d2 = p2p.Install (n1n2);

p2p.SetDeviceAttribute ("DataRate", StringValue ("1500kbps"));
p2p.SetChannelAttribute ("Delay", StringValue ("10ms"));
NetDeviceContainer d5d6 = p2p.Install (n5n6);

// We create the channels first without any IP addressing information
CsmaHelper csma;
csma.SetChannelAttribute ("DataRate", StringValue ("5Mbps"));
csma.SetChannelAttribute ("Delay", StringValue ("2ms"));
NetDeviceContainer d2345 = csma.Install (n2345);

// Later, we add IP addresses.
NS_LOG_INFO ("Assign IP Addresses.");
Ipv4AddressHelper ipv4;
ipv4.SetBase ("10.1.1.0", "255.255.255.0");
ipv4.Assign (d0d2);

ipv4.SetBase ("10.1.2.0", "255.255.255.0");
ipv4.Assign (d1d2);

ipv4.SetBase ("10.1.3.0", "255.255.255.0");
Ipv4InterfaceContainer i5i6 = ipv4.Assign (d5d6);

ipv4.SetBase ("10.250.1.0", "255.255.255.0");
ipv4.Assign (d2345);

// Create router nodes, initialize routing database and set up the routing
// tables in the nodes.
Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

// Create the OnOff application to send UDP datagrams of size
// 210 bytes at a rate of 448 Kb/s
NS_LOG_INFO ("Create Applications.");
uint16_t port = 9; // Discard port (RFC 863)
OnOffHelper onoff ("ns3::UdpSocketFactory",
InetSocketAddress (i5i6.GetAddress (1), port)),

```

```

onoff.SetConstantRate (DataRate ("300bps"));
onoff.SetAttribute ("PacketSize", UintegerValue (50));

ApplicationContainer apps = onoff.Install (c.Get (0));
apps.Start (Seconds (1.0));
apps.Stop (Seconds (10.0));

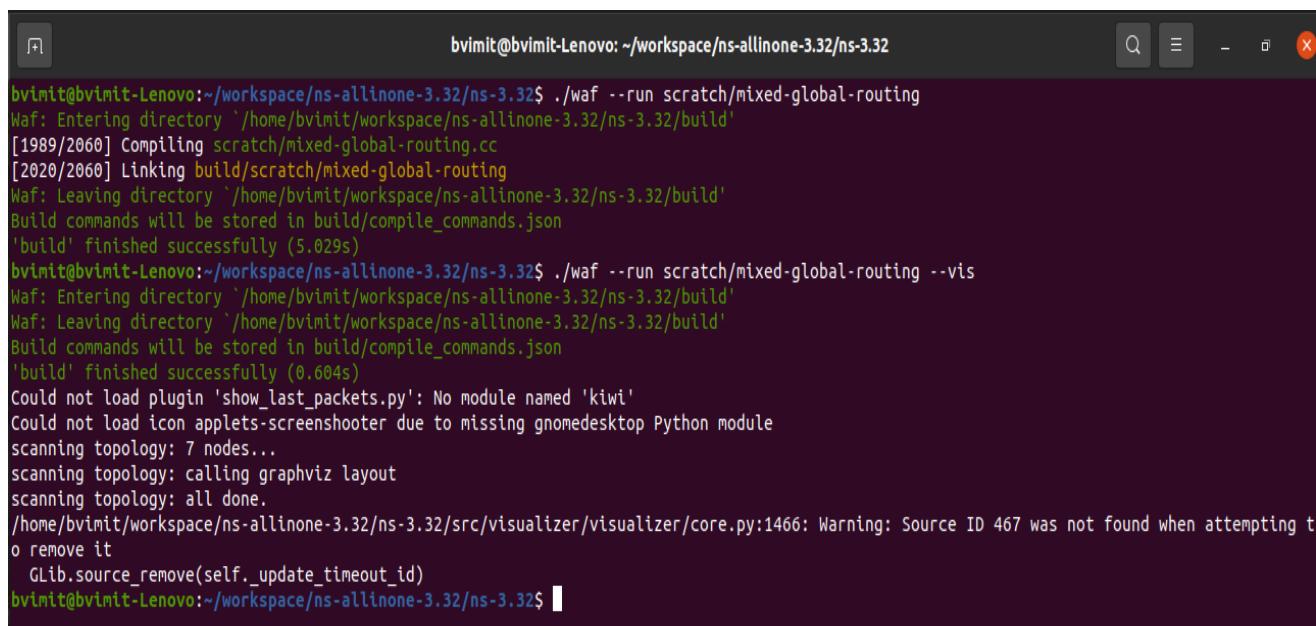
AsciiTraceHelper ascii;
Ptr<OutputStreamWrapper> stream = ascii.CreateFileStream ("mixed-global-routing.tr");
p2p.EnableAsciiAll (stream);
csma.EnableAsciiAll (stream);

p2p.EnablePcapAll ("mixed-global-routing");
csma.EnablePcapAll ("mixed-global-routing", false);

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

```

## **Output:**



```

bvimit@bvimit-Lenovo:~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run scratch/mixed-global-routing
Waf: Entering directory `/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
[1989/2060] Compiling scratch/mixed-global-routing.cc
[2020/2060] Linking build/scratch/mixed-global-routing
Waf: Leaving directory `/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (5.029s)
bvimit@bvimit-Lenovo:~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run scratch/mixed-global-routing --vis
Waf: Entering directory `/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
Waf: Leaving directory `/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.604s)
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: 7 nodes...
scanning topology: calling graphviz layout
scanning topology: all done.
/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/src/visualizer/visualizer/core.py:1466: Warning: Source ID 467 was not found when attempting to remove it
    GLib.source_remove(self._update_timeout_id)
bvimit@bvimit-Lenovo:~/workspace/ns-allinone-3.32/ns-3.32$ 

```

