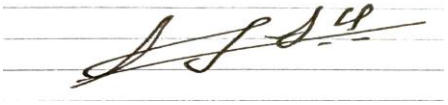
Name of Student: Sunny Satish Halkatti		
Roll Number: 17 (B)		**Lab Assignment Number: 1**
Title of Lab Assignment: To Study about the Installation Process of NS-3, NetAnim and WireShark in Linux.		
DOP: 23/03/2023		**DOS: 13/04/2023**
CO: CO1	**PO:** PO2, PO5, PSO1	**Signature:** 

NWL - Practical-1

Aim: To Study about the Installation Process of NS-3, NetAnim and WireShark in Linux.

Theory:

Linux:

Linux is a free and open-source operating system based on the Unix operating system. It was developed in the early 1990s by Linus Torvalds, a computer science student at the University of Helsinki, Finland.
Linux is known for its stability, security, and flexibility. It is widely used in servers, supercomputers, embedded systems, and mobile devices. Linux also powers many popular open-source software applications, including the Apache web server, the MySQL database, and the Python programming language. There are many different distributions of Linux, including Debian, Ubuntu, Fedora, CentOS, and Red Hat. Each distribution has its own set of features and software packages, but they all share the same core components, including the Linux kernel, which is the heart of the operating system.
Linux is often used by developers and system administrators because of its command-line interface and powerful tools for scripting and automation. However, many Linux distributions also have graphical user interfaces and are user-friendly for non-technical users.
Overall, Linux is a powerful and versatile operating system that is well-suited for a wide range of applications, from web servers and supercomputers to desktops and mobile devices.

NS-3 in Linux:

NS-3 is a popular open-source network simulation software tool that is widely used by researchers, engineers, and educators to simulate and analyze various network protocols and technologies. It is designed to simulate both wired and wireless networks and can be used to model a variety of network scenarios.
NS-3 is a discrete-event network simulator that allows you to model and simulate various network protocols and technologies in a controlled environment. It is written in C++ and provides a Python API for scripting and automation.
NS-3 includes a wide range of pre-built network models and protocols, such as TCP, IP, Ethernet, WiFi, LTE, and more. It also allows you to create your own custom network models and protocols using the C++ programming language.
NS-3 is designed to be highly customizable and configurable, allowing you to adjust various parameters and settings to simulate a wide range of network scenarios. It also includes various built-in tools for analyzing and visualizing simulation data, such as trace file output and real-time visualization using tools like Wireshark and NetAnim.
NS-3 is a cross-platform tool and can be used on various operating systems, including Linux, macOS, and Windows. However, it is primarily developed and tested on Linux systems, and using it in Linux may offer some performance advantages due to the lower overhead and greater control over system resources.
NS-3 is licensed under the GNU General Public License (GPL), which means that it is free and open-source software. This allows anyone to use, modify, and distribute NS-3 as long as they comply with the terms of the license.
Overall, NS-3 is a powerful and flexible tool for simulating and analyzing network protocols and technologies in Linux. It requires some programming and networking knowledge to use effectively, but it is well-documented and has a large and active user community that can provide support and guidance.

WHY NS-3 ?

NS-3 is a popular choice for network simulation for several reasons:

1. Open-source: NS-3 is free and open-source software, which means that anyone can use, modify, and distribute it without restrictions. This makes it accessible to a wide range of users, including researchers, engineers, educators, and hobbyists.
2. Realistic simulations: NS-3 is designed to provide realistic network simulations, which means that it takes into account real-world network conditions such as network congestion, packet loss, and delay. This makes it useful for simulating and testing network protocols and technologies in a controlled environment before deploying them in real-world scenarios.
3. Wide range of network models and protocols: NS-3 includes a wide range of pre-built network

models and protocols, including TCP, IP, Ethernet, WiFi, LTE, and more. It also allows you to create your own custom network models and protocols using the C++ programming language. This makes it a versatile tool for simulating and testing various network scenarios.

4. Customizability: NS-3 is highly customizable and configurable, allowing you to adjust various parameters and settings to simulate a wide range of network scenarios. This makes it useful for testing and optimizing network performance under different conditions.
5. Active user community: NS-3 has a large and active user community that provides support, guidance, and contributions to the project. This makes it easier to get started with NS-3 and to get help if you encounter any issues or have questions.

PyViz:

PyViz is a suite of open-source Python visualization tools for creating interactive visualizations and dashboards. It is designed to be easy to use, flexible, and customizable, and it provides a range of options for creating and deploying interactive visualizations for data analysis and exploration.
Some of the key features of PyViz include:

1. Holoviews: Holoviews is a Python library for creating interactive visualizations that can be easily shared and deployed. It provides a simple and intuitive interface for creating visualizations and supports a wide range of data types and sources.
2. Bokeh: Bokeh is a Python library for creating interactive web-based visualizations. It provides a range of tools for creating charts, graphs, and other visualizations that can be easily customized and embedded in web applications.
3. Datashader: Datashader is a Python library for visualizing large datasets. It uses intelligent data sampling and aggregation to create visualizations that are fast, responsive, and easy to interpret.
4. Panel: Panel is a Python library for creating interactive dashboards. It provides a range of tools for building custom dashboards that can be easily shared and deployed.
5. GeoViews: GeoViews is a Python library for creating interactive geospatial visualizations. It provides a range of tools for creating maps, overlays, and other visualizations that are responsive and easy to customize.

NetAmin:

A simple animator is presented here, called NetAnim. Written by George Riley, and based on the multi-platform QT4 GUI toolkit, it is capable of animating simulations with PointToPoint links in NS-3. It depicts frames on the links as they travel from node to node.

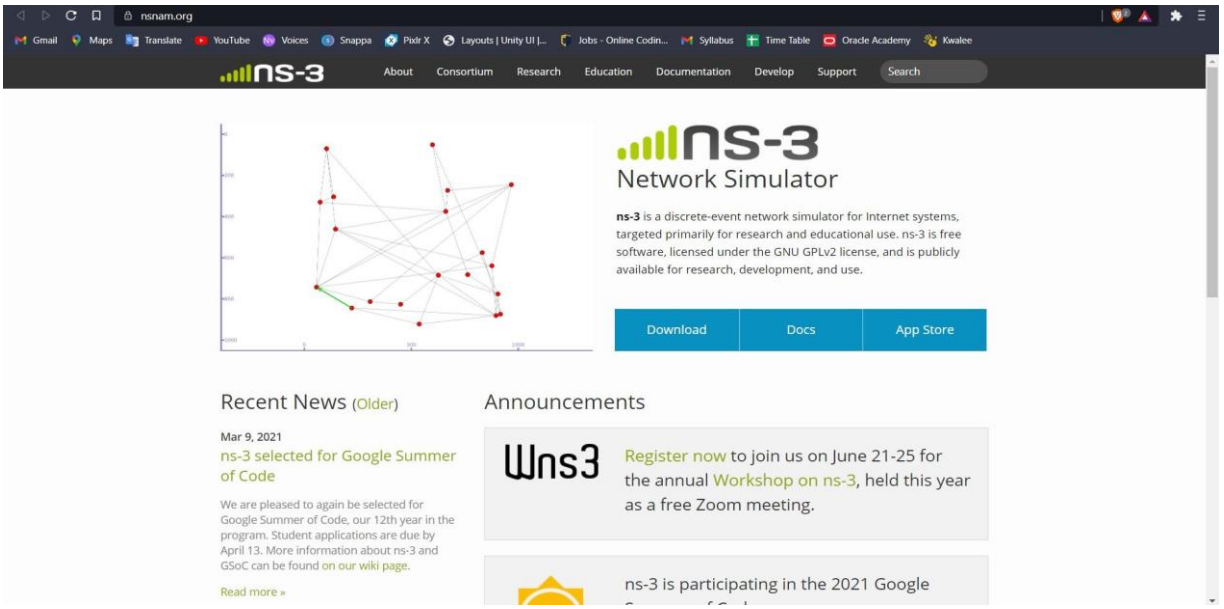
NetAnim requires a custom trace file for animation. This trace file is generated by an animation interface and is included in the current version of NS-3.

NetAnim GUI Features of NetAnim:

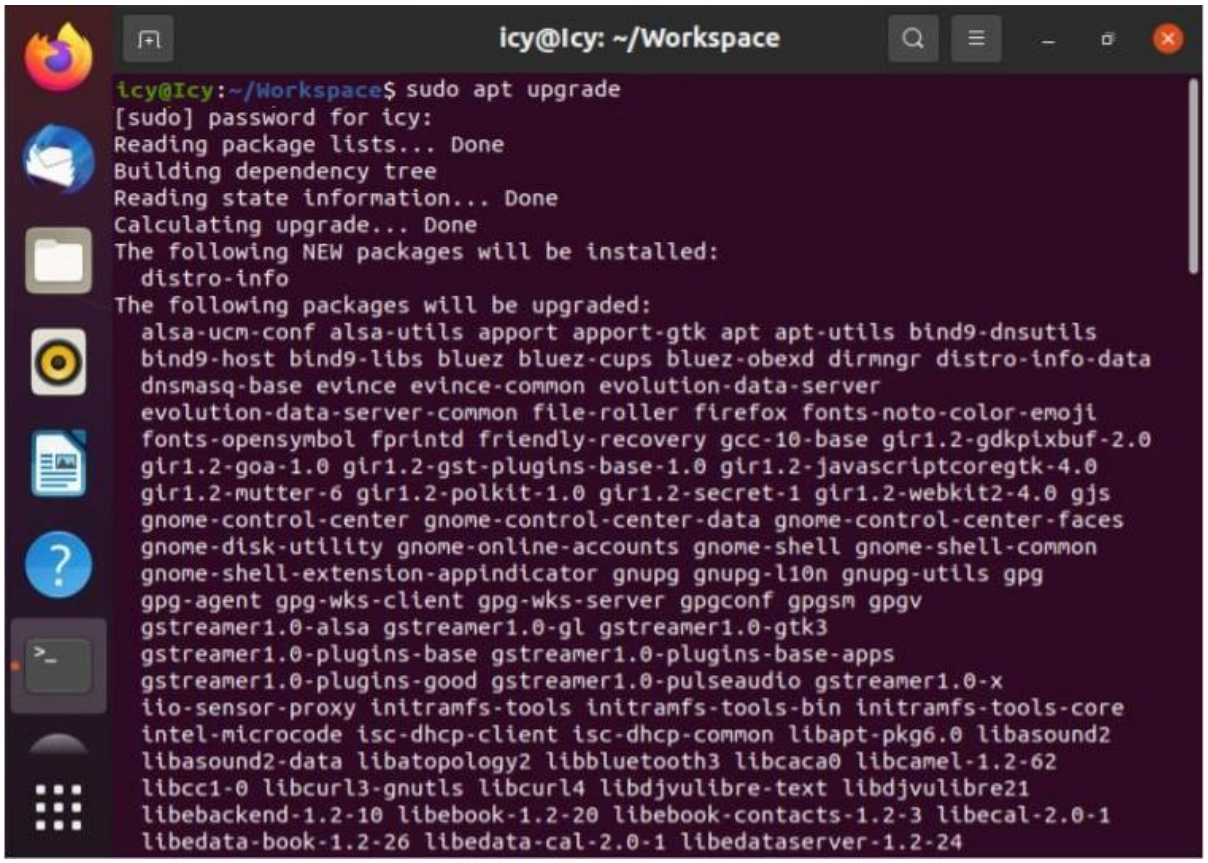
1. Animate packets over wired-links and wireless-links (Limited support for LTE traces. No support for Ipv6)
2. Packet timeline with regex filter on packet metadata.
3. Node position statistics with node trajectory plotting(path of a mobile node).
4. Print brief packet-meta data on packets
5. Parse flow-monitor XML files and display statistics for each flow.
6. Show IP and MAC information, including peer IP & MAC for point-to-point links.
7. Display double or uint32 valued counters vs time for multiple nodes in a chart or a table.
8. Print the routing table at nodes at various points in time

Installation:

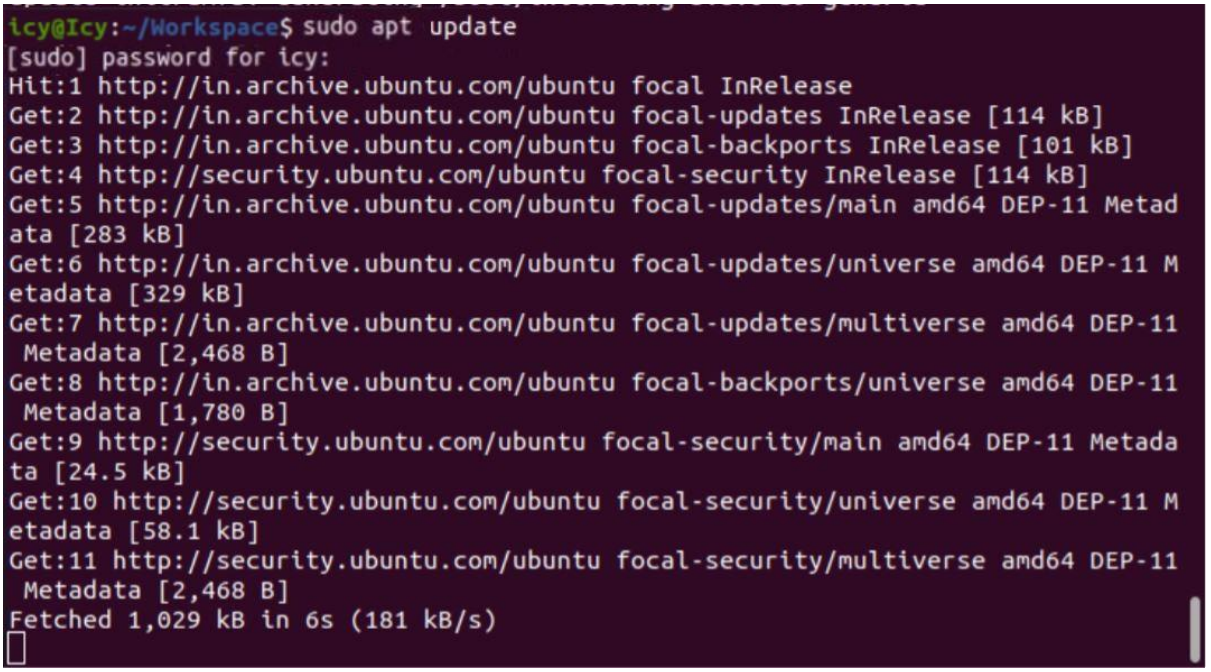
1. Download ns-allinone-3.32 zip file from https://www.nsnam.org/ and extract it.



2. sudo apt upgrade

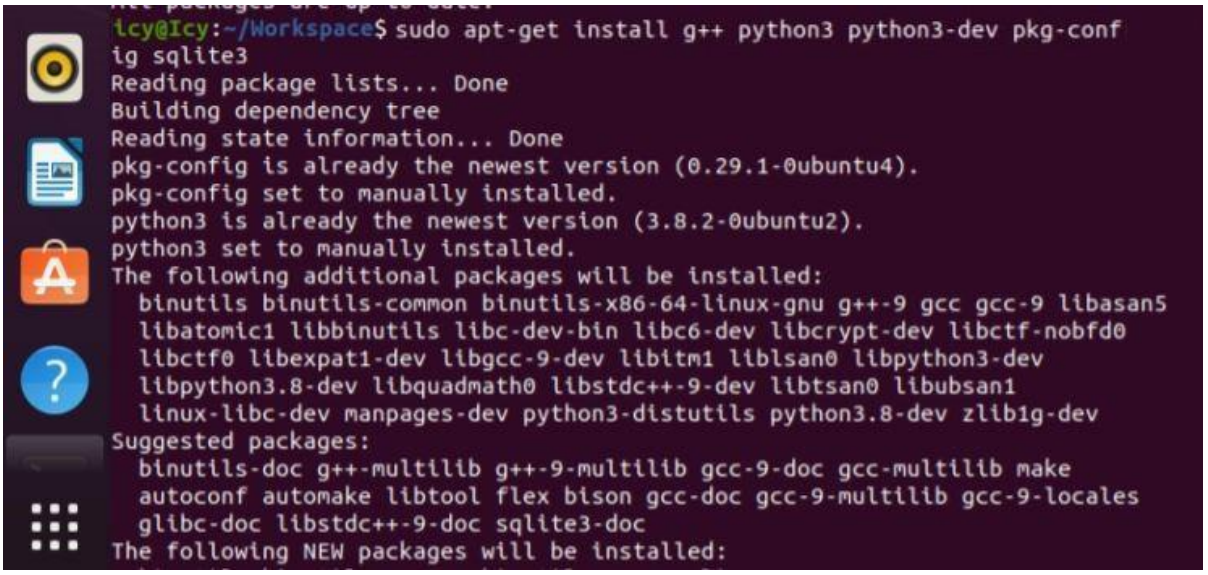


3. sudo apt update

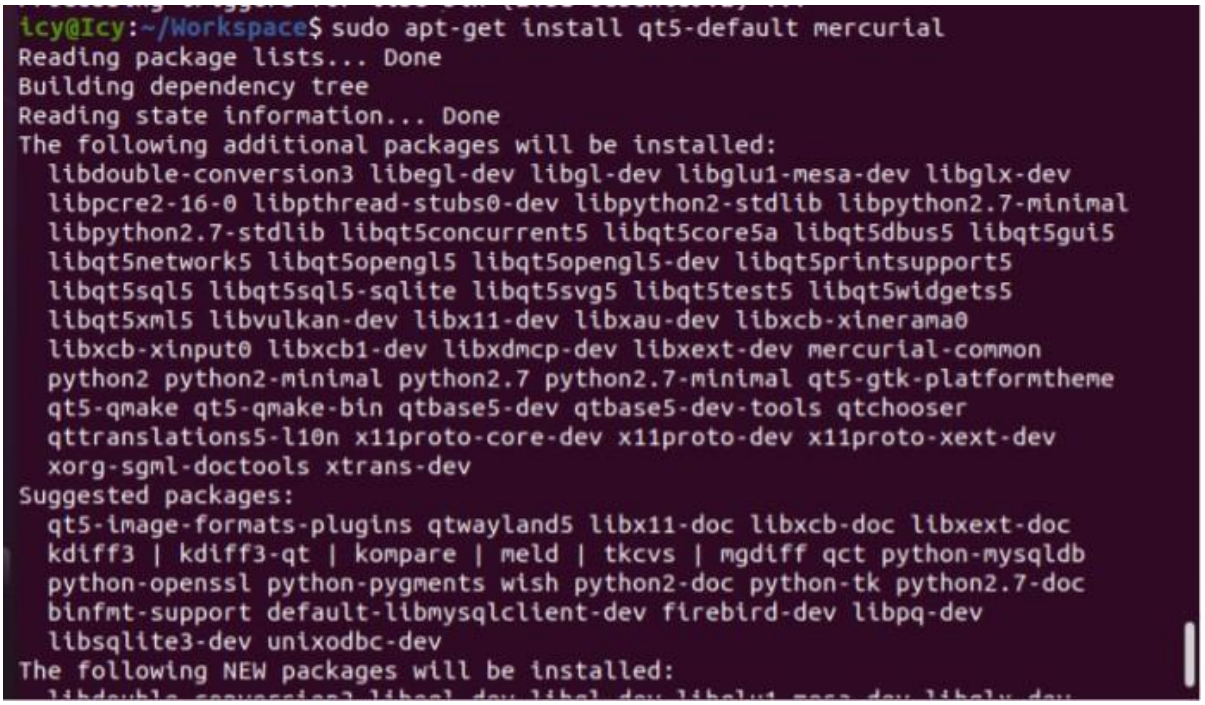


Minimal requirements for Python API users

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

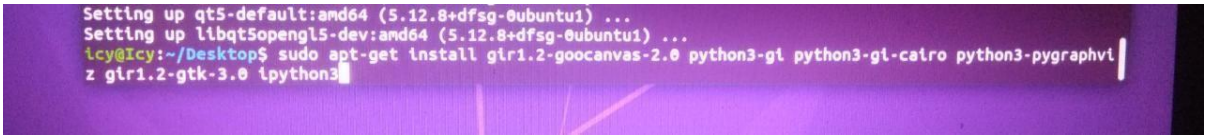


5. Netanim animator: qt5 development tools are needed for Netanim animator;
sudo apt-get install qt5-default mercurial



6. ns-3-pyviz visualizer

sudo apt-get install gir1.2-goocanvas-2.0 python3-gi python3-gi-cairo python3-pygraphviz gir1.2-gtk-3.0 ipython3




```
icy@Icy:~/Workspace$ sudo apt-get install gir1.2-goocanvas-2.0 python3-gi
python3-gi-cairo python3-pygraphviz gir1.2-gtk-3.0 ipython3
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-gi is already the newest version (3.36.0-1).
python3-gi set to manually installed.
python3-gi-cairo is already the newest version (3.36.0-1).
python3-gi-cairo set to manually installed.
gir1.2-gtk-3.0 is already the newest version (3.24.20-0ubuntu1).
gir1.2-gtk-3.0 set to manually installed.
The following additional packages will be installed:
  graphviz libann0 libcdt5 libcgraph6 libgoocanvas-2.0-9
  libgoocanvas-2.0-common libgts-0.7-5 libgts-bin libgvc6 libgvpr2
  liblab-gamut1 libpathplan4 python3-backcall python3-decorator
  python3-ipython python3-ipython-genutils python3-jedi python3-parso
  python3-pickleshare python3-prompt-toolkit python3-pygments
  python3-traitlets python3-wcwidth
Suggested packages:
  gsfonts graphviz-doc python-ipython-doc python-pygments-doc
  ttf-bitstream-vera python-pygraphviz-doc
```

7. Debugging

8. sudo apt-get install gdb valgrind

```
icy@Icy:~/Workspace$ 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).
gdb set to manually installed.
Suggested packages:
  valgrind-dbg valgrind-mpi kcachegrind alleyoop valkyrie
The following NEW packages will be installed:
  valgrind
0 upgraded, 1 newly installed, 0 to remove and 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 amd
64 1:3.15.0-1ubuntu9.1 [20.3 MB]
0% [1 valgrind 0 B/20.3 MB 0%]
```

9. Doxygen and related inline documentation:
sudo apt-get install doxygen graphviz imagemagick

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

10. 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

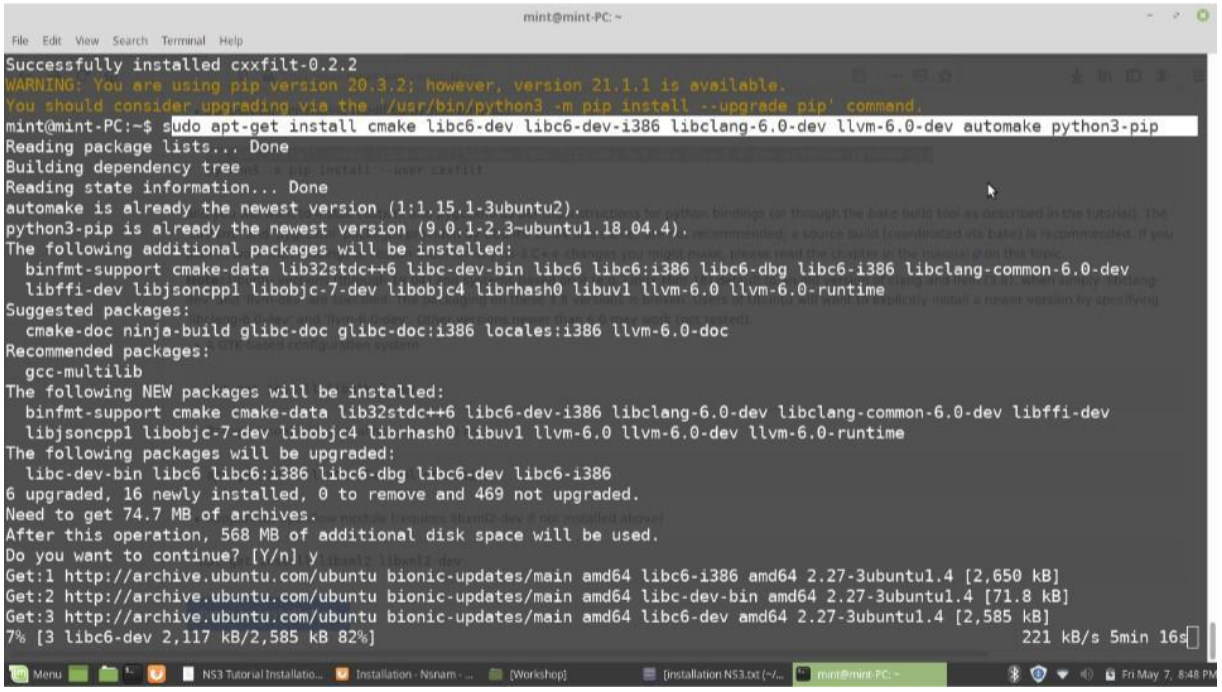
```
icy@Icy:~/Workspace$ sudo apt-get install python3-sphinx dia
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  dia-common dia-shapes docutils-common gsfonts-x11 javascript-common
  libart-2.0-2 libjs-jquery libjs-sphinxdoc libjs-underscore make
  python-babel-localedata python3-alabaster python3-babel python3-docutils
  python3-imagesize python3-jinja2 python3-packaging python3-pyparsing
  python3-roman sphinx-common
Suggested packages:
  apache2 | lighttpd | httpd make-doc docutils-doc fonts-linuxlibertine
  | ttf-linux-libertine texlive-lang-french texlive-latex-base
  texlive-latex-recommended python-jinja2-doc python-pyparsing-doc dvipng
  latexmk libjs-mathjax python3-sphinx-rtd-theme python3-stemmer sphinx-doc
  texlive-fonts-recommended texlive-latex-extra texlive-plain-generic
The following NEW packages will be installed:
  dia dia-common dia-shapes docutils-common gsfonts-x11 javascript-common
  libart-2.0-2 libjs-jquery libjs-sphinxdoc libjs-underscore make
  python-babel-localedata python3-alabaster python3-babel python3-docutils
  python3-imagesize python3-jinja2 python3-packaging python3-pyparsing
  python3-roman python3-sphinx sphinx-common
0 upgraded, 22 newly installed, 0 to remove and 0 not upgraded.
```

11. To read pcap packet traces Sudo apt-
get install tcpdump

12. 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

python3 -m pip install --user cxxfilt



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:

cd

cd workspace

tar xjf <name of NS3 downloaded file name>

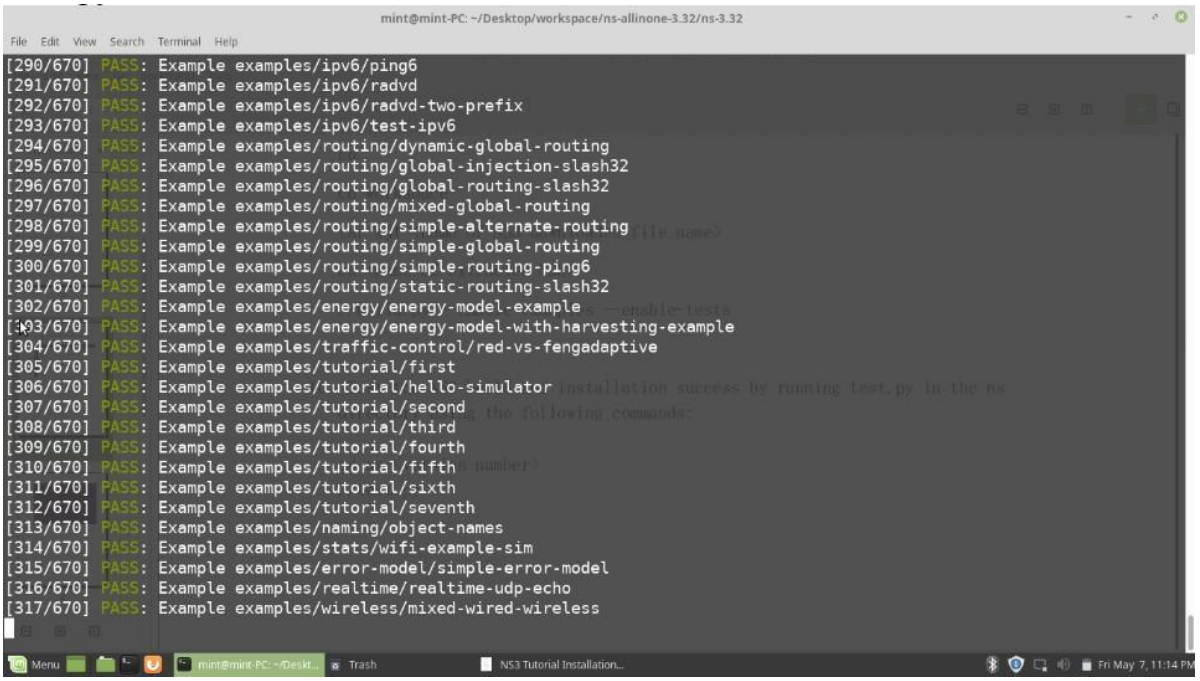
cd <name of extracted NS3>

./build.py --enable-examples --enable-tests It will take some time.

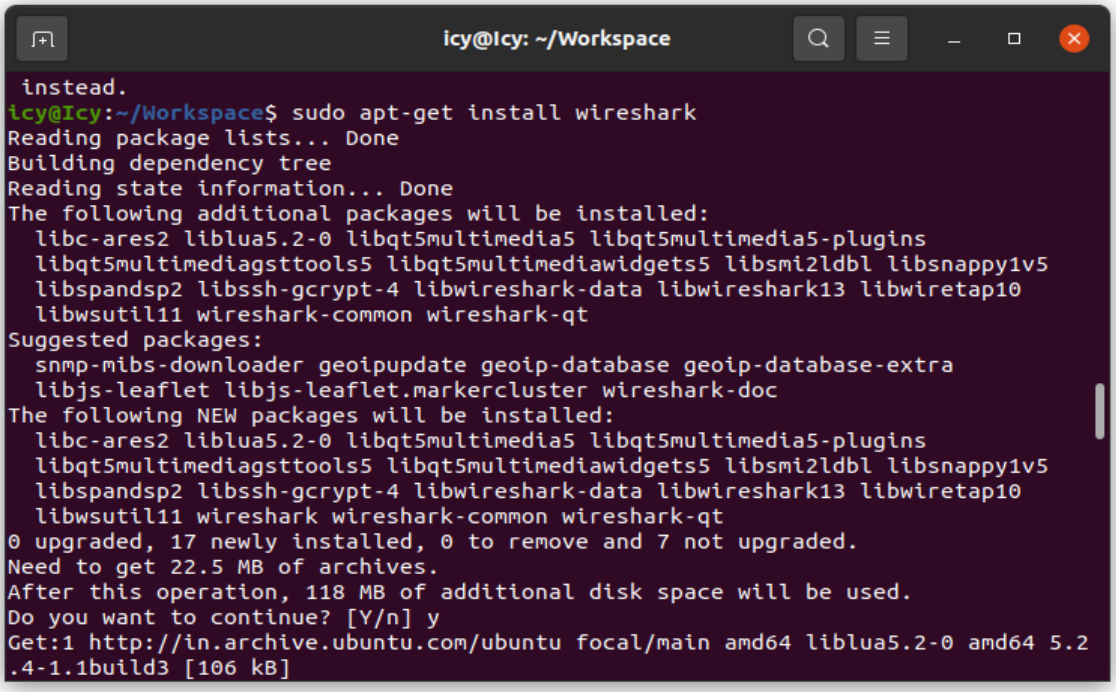
Test the NS3 build and installation success by running test.py in the ns directory using the following commands:

cd ns-<version number>

./test.py



Installation of Wireshark:



Conclusion: Installation Process of NS-3, NetAnim and WireShark in Linux is studied.