# Practical - 1

**AIM: Installation of NS-3 in Linux**

**Objective:** To learn to Install NS3 in Ubuntu Linux

**Step to Install**:

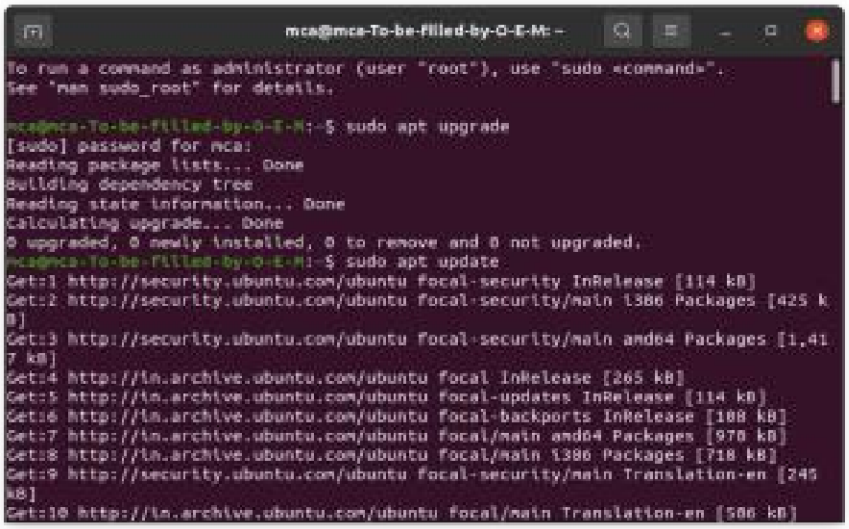
After Installing Ubuntu 20.04 LTS successfully, you can now start with installing of NS3 packages.

List of Packages for Installing ns-3 in Ubantu Systems

**Prequisite for installing NS3.32**

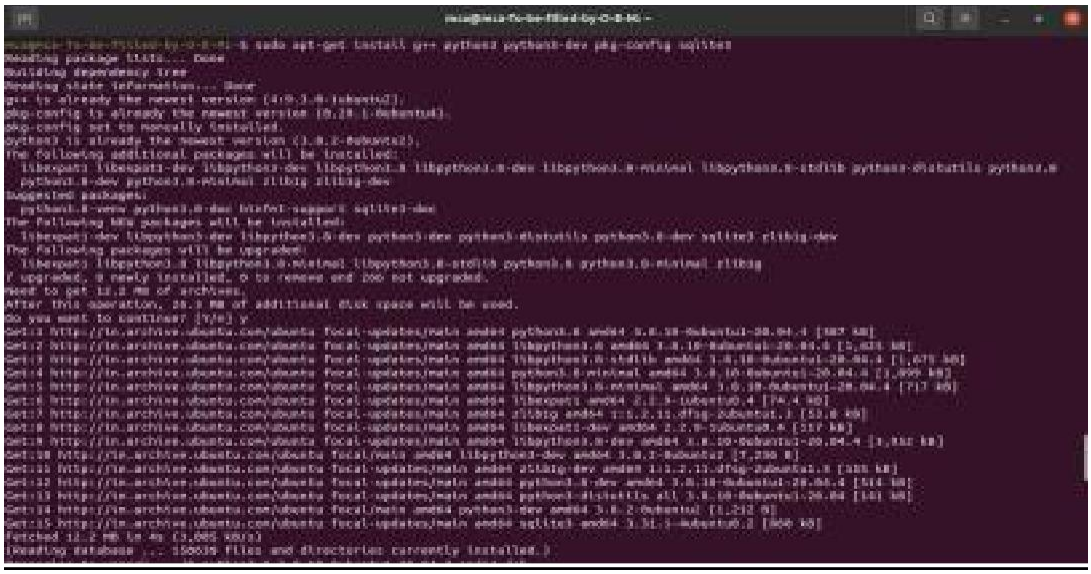
1] sudo apt upgrade

2] Sudo apt update

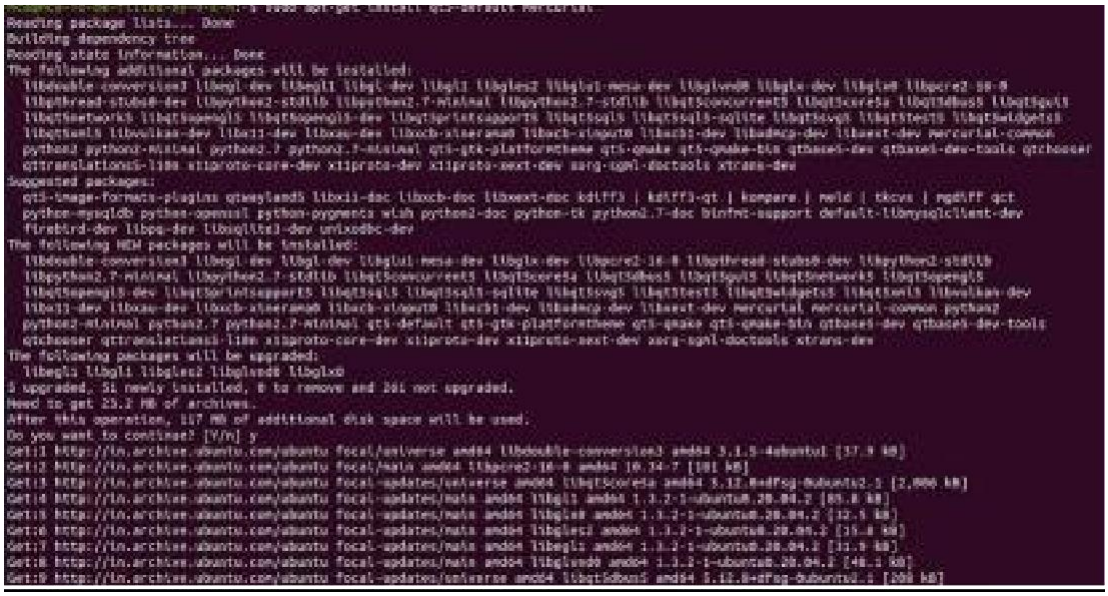


3] **Minimal requirements for C++ users** apt-get install g++ python3


4] **Minimal requirements for Python API users** apt-get install g++ python3 python3-dev pkg-config sqlite3



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

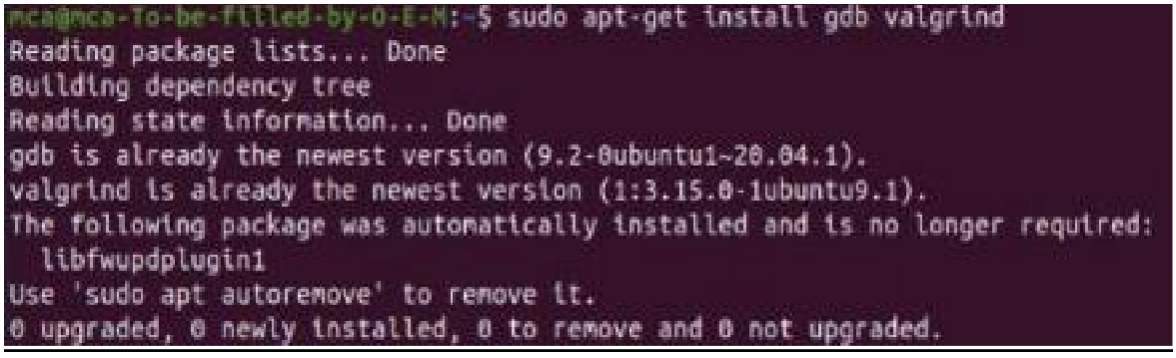


**6] ns-3-pyviz visualizer**

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

**7]Debugging:**

8]apt-get install gdb valgrind




**9] Doxygen and related inline documentation:**

apt-get install doxygen graphviz imagemagick

```
mca@mca-To-be-filled-by-O-E-M:~$ sudo apt-get install doxygen graphviz imagemagick
Reading package lists... Done
Building dependency tree
Reading state information... Done
doxygen is already the newest version (1.8.17-0ubuntu2).
graphviz is already the newest version (2.42.2-3build2).
imagemagick is already the newest version (8:6.9.10.23+dfsg-2.1ubuntu11.4).
```

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

```
mca@mca-To-be-filled-by-O-E-M:~$ 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
dvipng is already the newest version (1.15-1.1).
latexmk is already the newest version (1:4.67-0.1).
texlive is already the newest version (2019.20200218-1).
texlive-extra-utils is already the newest version (2019.20200218-1).
texlive-font-utils is already the newest version (2019.20200218-1).
texlive-latex-extra is already the newest version (2019.20200218-1).
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

**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):**

apt-get install python3-sphinx dia

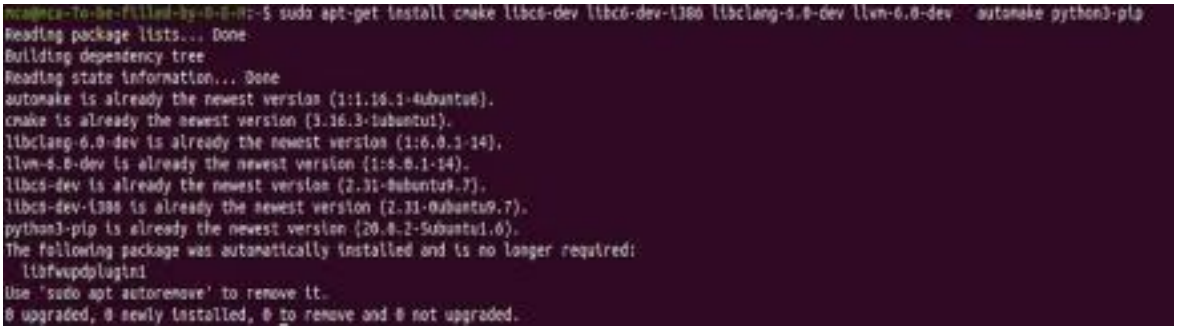
```
mca@mca-To-be-filled-by-O-E-M:~$ sudo apt-get install python3-sphinx dia
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-sphinx is already the newest version (1.8.5-7ubuntu3).
dia is already the newest version (0.97.3+git20160930-9).
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

**11] To read pcap packet traces**

apt-get install tcpdump

```
mca@mca-To-be-filled-by-O-E-M:~$ 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).
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

12] **Support for generating modified python bindings** apt-get install cmake libc6-dev libc6-dev-i386 libclang 6.0-dev llvm-6.0-dev automake python3-pip




**After installing the required packages**,

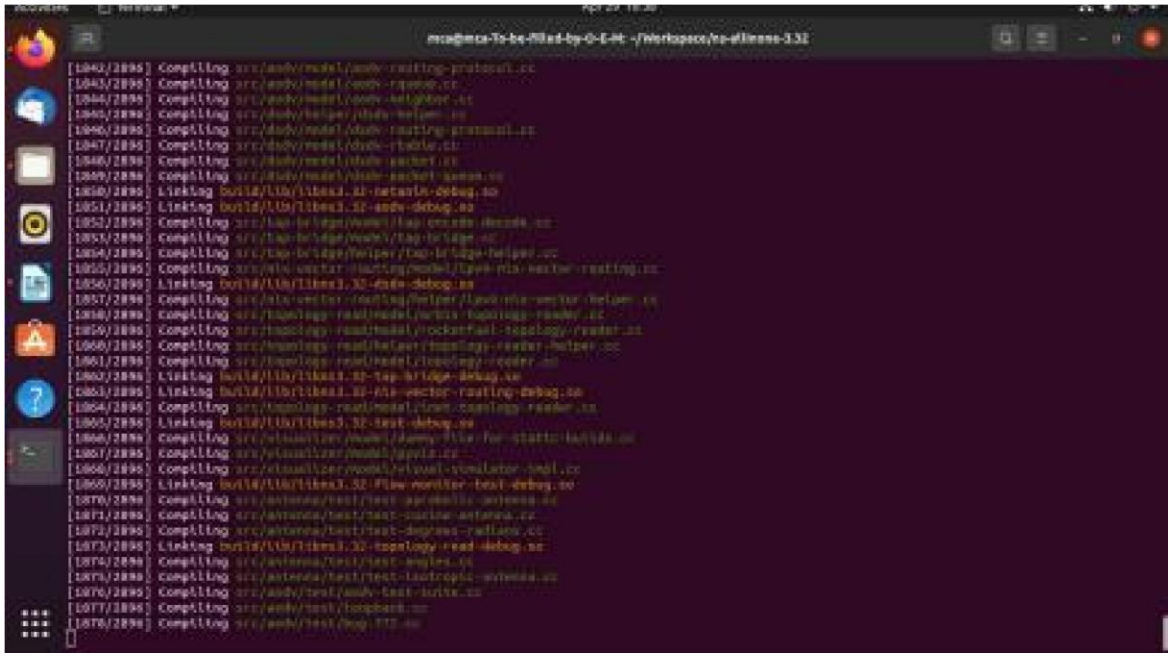
1. create a folder named workspace in the home directory and then put the NS3 tar package into the workspace.

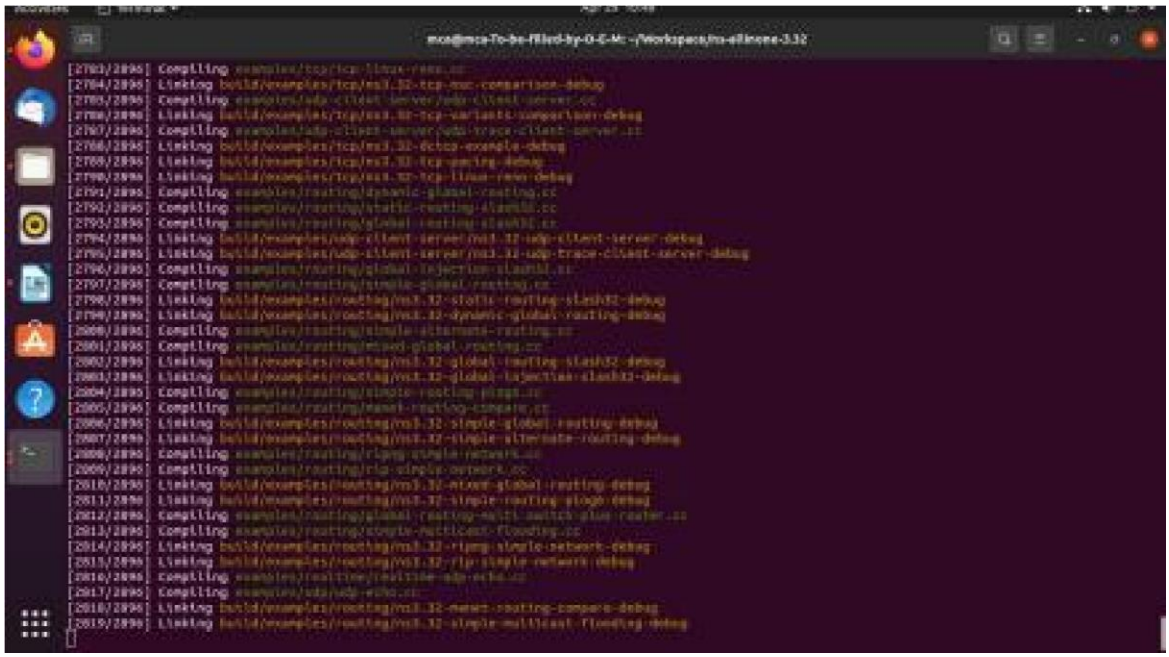
2. Go to terminal and input these commands consecutively after each command finishes executing:

2.1. cd
2.2. cd workspace tar xjf <name of NS3 downloaded file  name>
2.3. cd <name of extracted NS3>
2.4. ./build.py --enable-examples --enable-tests
2.5. It takes time be patient !!

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

3.1. cd ns-<version number>
3.2. ./test.py





# Practical No - 2

**AIM: Installation of Net anim in Linux**

**Objective:** To learn to Install Net anim in Ubuntu Linux

**Step to Install:**

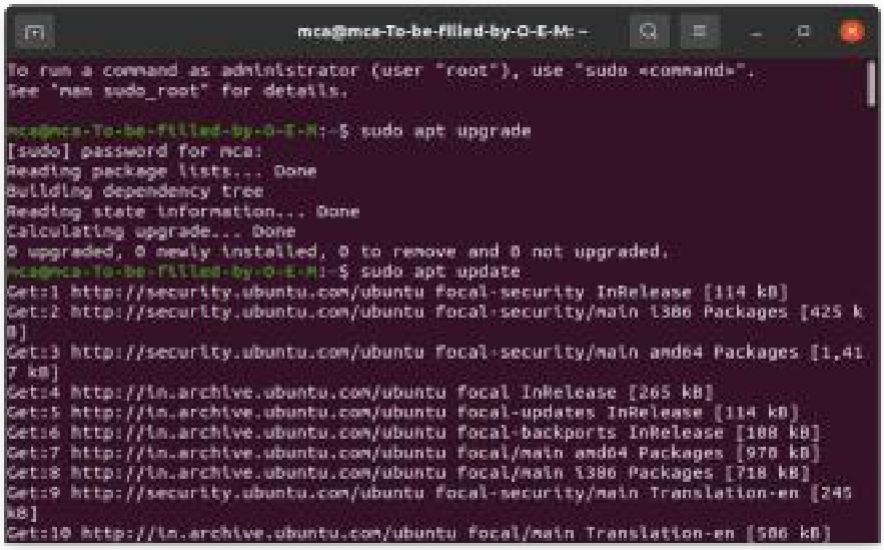
After Installing Ubuntu 20.04 LTS successfully, you can now start with installing of Net anim packages.

List of Packages for Installing ns-3 in Ubantu Systems

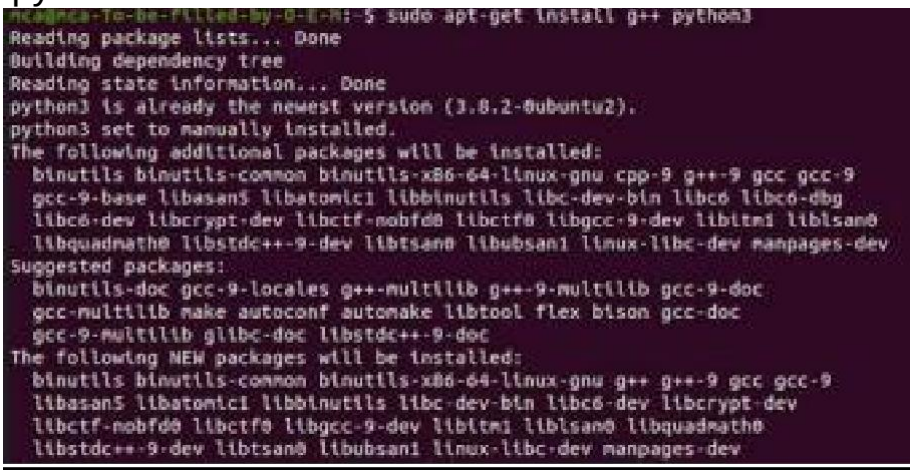
**Prequisite for installing NS3.32**

1] sudo apt upgrade

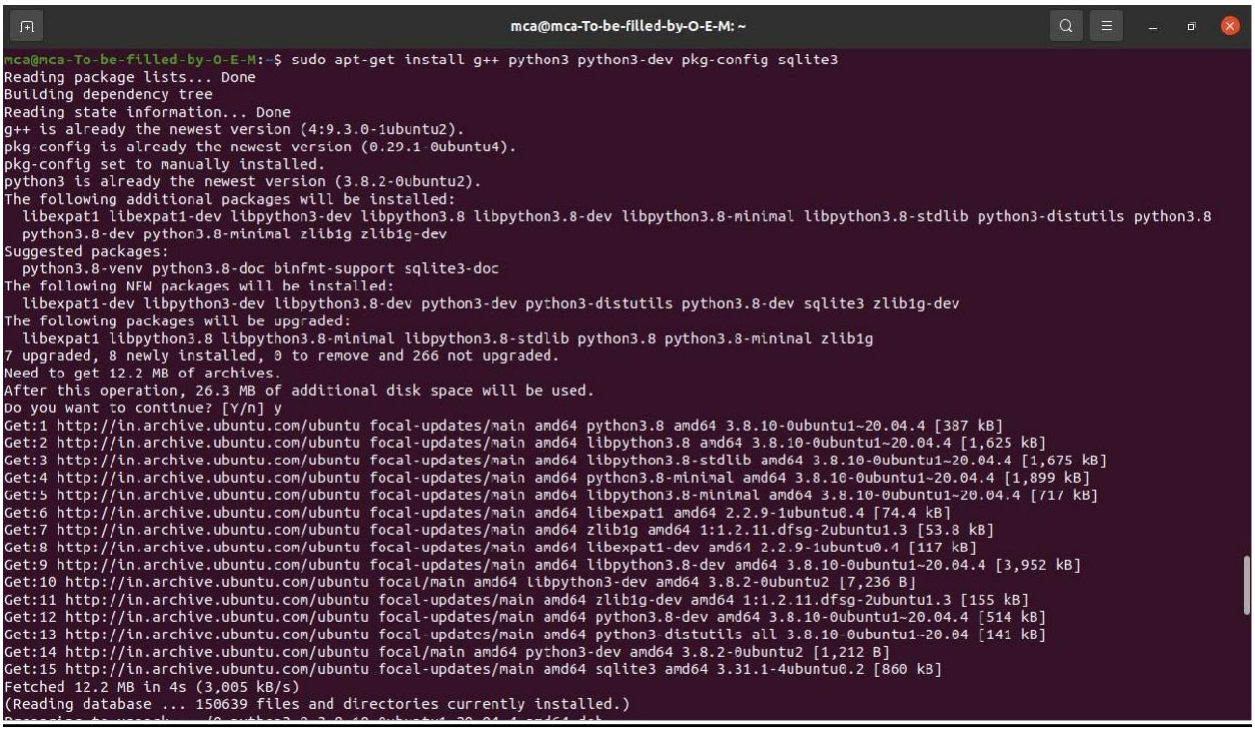
2] Sudo apt update



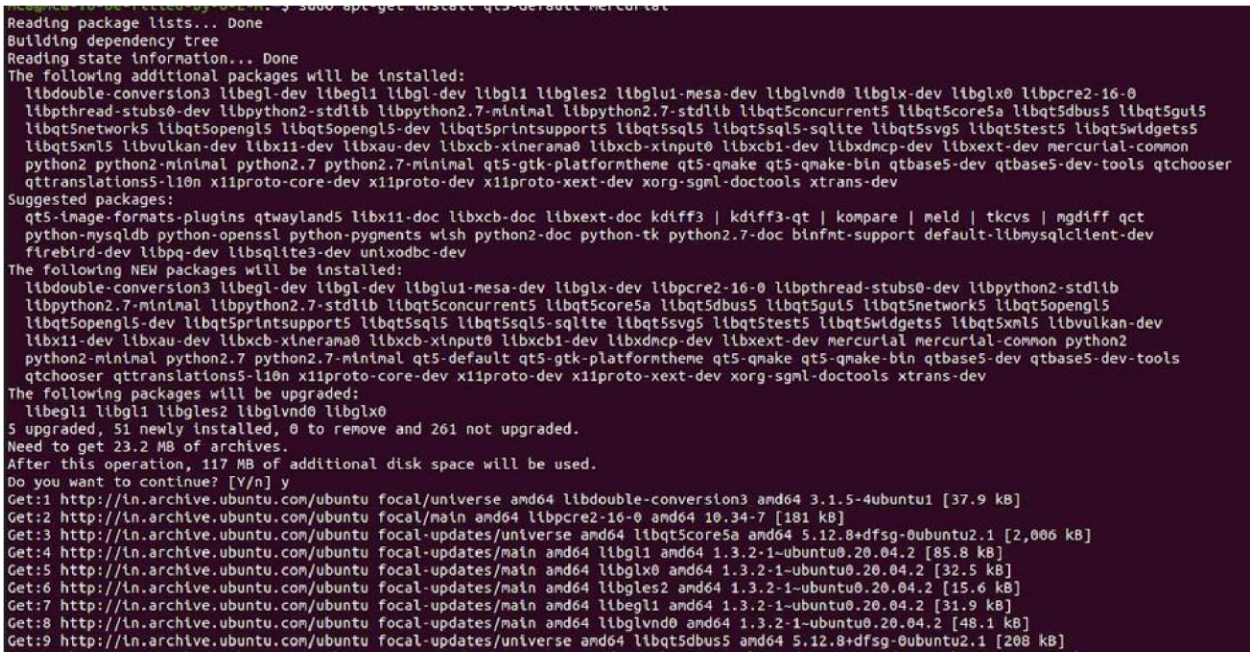

**3] Minimal requirements for C++ users** apt-get install g++ python3




**4] Minimal requirements for Python API users**apt-get install g++ python3 python3-dev pkg-config sqlite3




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

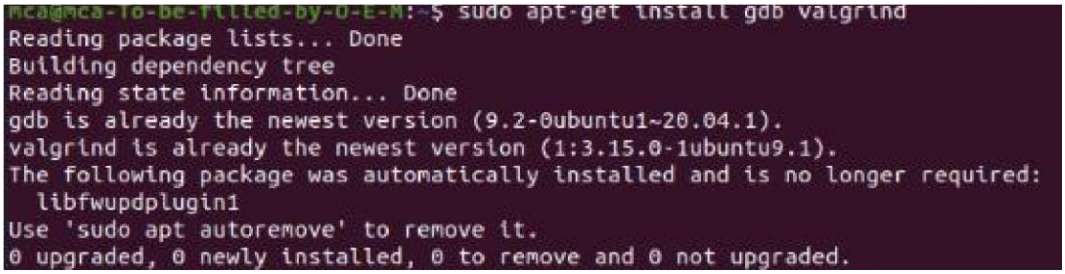


**6] ns-3-pyviz visualizer**

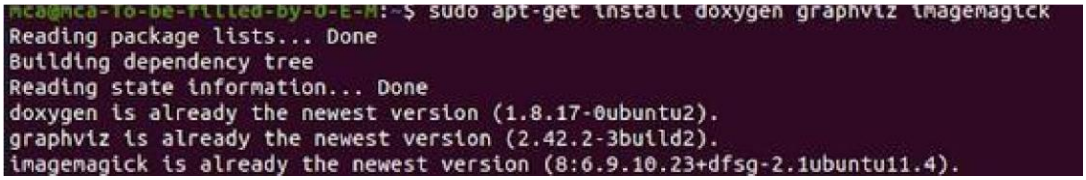
apt-get install gir1.2-goocanvas-2.0 python-gi python-gi- cairo python-pygraphviz python3-gi python3-gi-cairo python3-pygraphviz gir1.2-gtk-3.0 ipython ipython3


**7] Debugging:**

**8]** apt-get install gdb valgrind




**9] Doxygen and related inline documentation:** apt-get install doxygen graphviz imagemagick




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):**apt-get install python3-sphinx dia

```
mca@mca-To-be-filled-by-O-E-M:~$ sudo apt-get install python3-sphinx dia
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-sphinx is already the newest version (1.8.5-7ubuntu3).
dia is already the newest version (0.97.3+git20160930-9).
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

**11] To read pcap packet traces** apt-get install tcpdump

```
mca@mca-To-be-filled-by-O-E-M:~$ 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).
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

**12] Support for generating modified python bindings** apt-get install cmake libc6-dev libc6-dev-i386 libclang- 6.0-dev llvm-6.0-dev automake python3-pip

```
mca@mca-To-be-filled-by-O-E-M:~$ 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
automake is already the newest version (1:1.16.1-4ubuntu6).
cmake is already the newest version (3.16.3-1ubuntu1).
libclang-6.0-dev is already the newest version (1:6.0.1-14).
llvm-6.0-dev is already the newest version (1:6.0.1-14).
libc6-dev is already the newest version (2.31-0ubuntu9.7).
libc6-dev-i386 is already the newest version (2.31-0ubuntu9.7).
python3-pip is already the newest version (20.0.2-5ubuntu1.6).
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

**After installing the required packages,**

4. create a folder named workspace in the home directory and then put the NS3 tar package into the workspace.

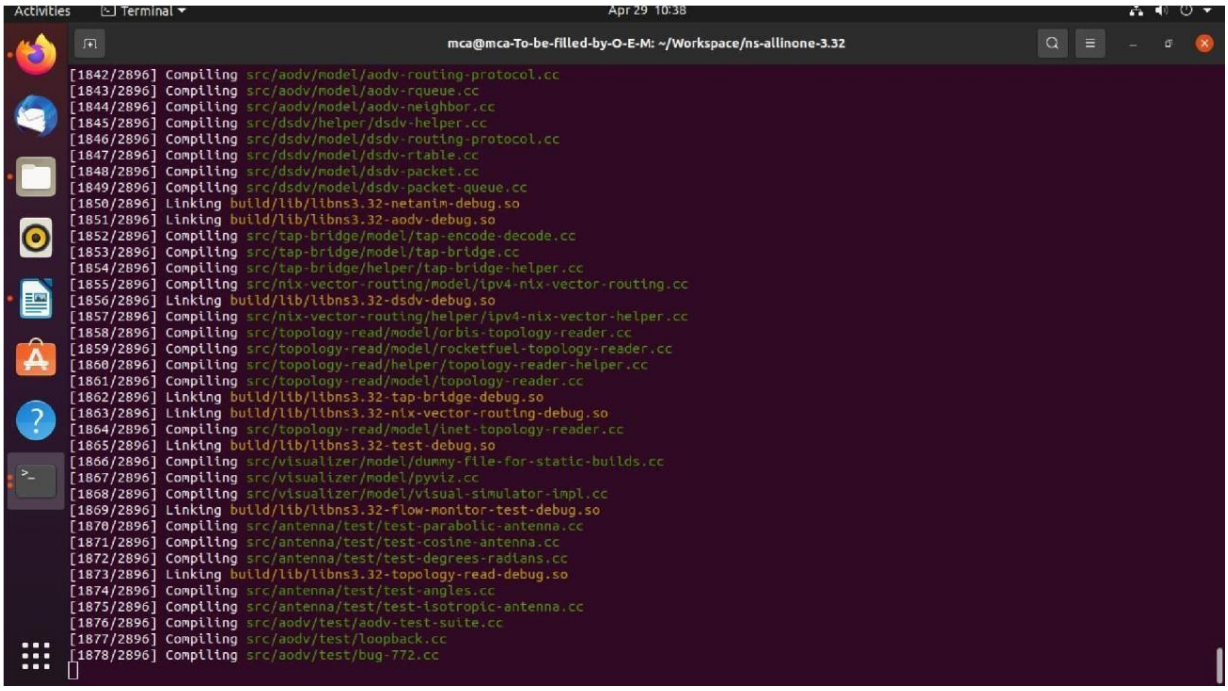
5. Go to terminal and input these commands consecutively after each command finishes executing:

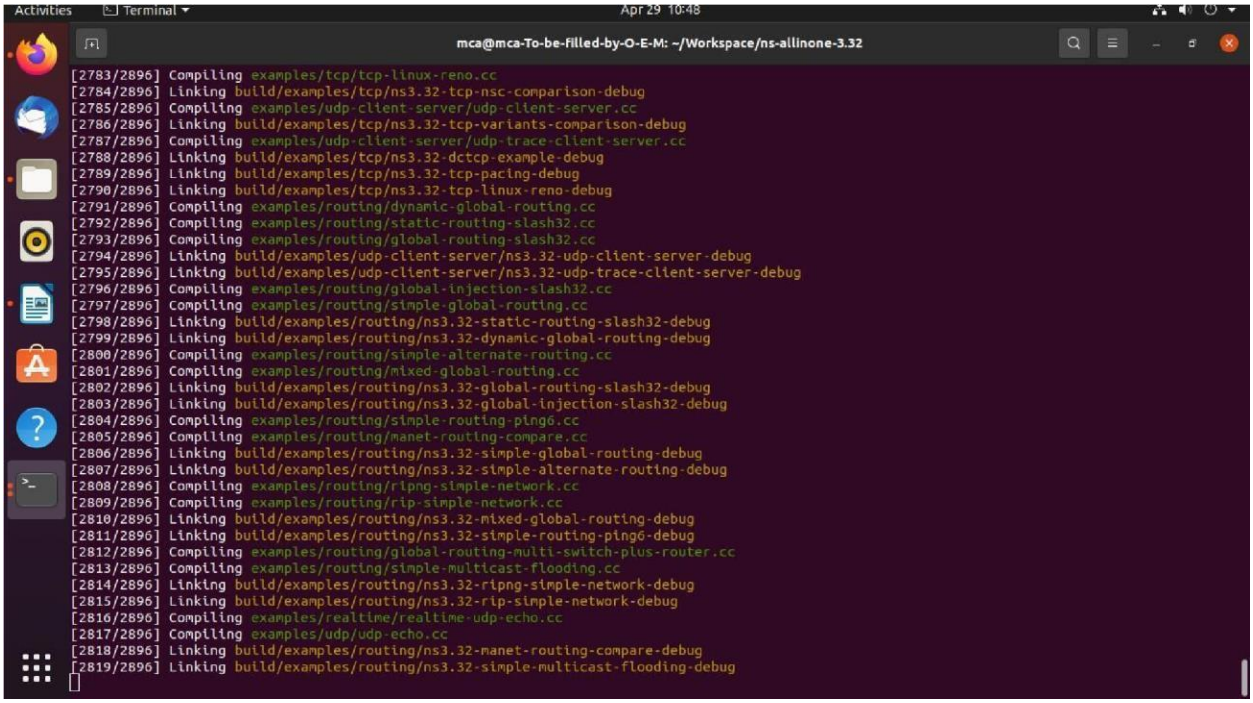
5.1. cd

5.2. cd workspace tar xjf <name of NS3 downloaded file name>

5.3. cd <name of extracted NS3>
5.4. ./build.py --enable-examples --enable-tests
5.5. It takes time be patient !!
6. Test the NS3 build and installation success by running test.py in the ns directory using the following commands:

6.1. cd ns-<version number>
6.2. ./test.py




Activities
Terminal
Apr 29 10:48
mca@mca-To-be-filled-by-O-E-M: ~/Workspace/ns-allinone-3.32
[2783/2896] Compiling examples/tcp/tcp-linux-reno.cc
[2784/2896] Linking build/examples/tcp/ns3.32-tcp-nsc-comparison-debug
[2785/2896] Compiling examples/udp-client-server/udp-client-server.cc
[2786/2896] Linking build/examples/tcp/ns3.32-tcp-variants-comparison-debug
[2787/2896] Compiling examples/udp-client-server/udp-trace-client-server.cc
[2788/2896] Linking build/examples/tcp/ns3.32-dctcp-example-debug
[2789/2896] Linking build/examples/tcp/ns3.32-tcp-pacing-debug
[2790/2896] Linking build/examples/tcp/ns3.32-tcp-linux-reno-debug
[2791/2896] Compiling examples/routing/dynamic-global-routing.cc
[2792/2896] Compiling examples/routing/static-routing-slash32.cc
[2793/2896] Compiling examples/routing/global-routing-slash32.cc
[2794/2896] Linking build/examples/udp-client-server/ns3.32-udp-client-server-debug
[2795/2896] Linking build/examples/udp-client-server/ns3.32-udp-trace-client-server-debug
[2796/2896] Compiling examples/routing/global-injection-slash32.cc
[2797/2896] Compiling examples/routing/simple-global-routing.cc
[2798/2896] Linking build/examples/routing/ns3.32-static-routing-slash32-debug
[2799/2896] Linking build/examples/routing/ns3.32-dynamic-global-routing-debug
[2800/2896] Compiling examples/routing/simple-alternate-routing.cc
[2801/2896] Compiling examples/routing/mixed-global-routing.cc
[2802/2896] Linking build/examples/routing/ns3.32-global-routing-slash32-debug
[2803/2896] Linking build/examples/routing/ns3.32-global-injection-slash32-debug
[2804/2896] Compiling examples/routing/simple-routing-ping6.cc
[2805/2896] Compiling examples/routing/manet-routing-compare.cc
[2806/2896] Linking build/examples/routing/ns3.32-simple-global-routing-debug
[2807/2896] Linking build/examples/routing/ns3.32-simple-alternate-routing-debug
[2808/2896] Compiling examples/routing/ripng-simple-network.cc
[2809/2896] Compiling examples/routing/rip-simple-network.cc
[2810/2896] Linking build/examples/routing/ns3.32-mixed-global-routing-debug
[2811/2896] Linking build/examples/routing/ns3.32-simple-routing-ping6-debug
[2812/2896] Compiling examples/routing/global-routing-multi-switch-plus-router.cc
[2813/2896] Compiling examples/routing/simple-multicast-flooding.cc
[2814/2896] Linking build/examples/routing/ns3.32-ripng-simple-network-debug
[2815/2896] Linking build/examples/routing/ns3.32-rip-simple-network-debug
[2816/2896] Compiling examples/realtime/realtime-udp-echo.cc
[2817/2896] Compiling examples/udp/udp-echo.cc
[2818/2896] Linking build/examples/routing/ns3.32-manet-routing-compare-debug
[2819/2896] Linking build/examples/routing/ns3.32-simple-multicast-flooding-debug

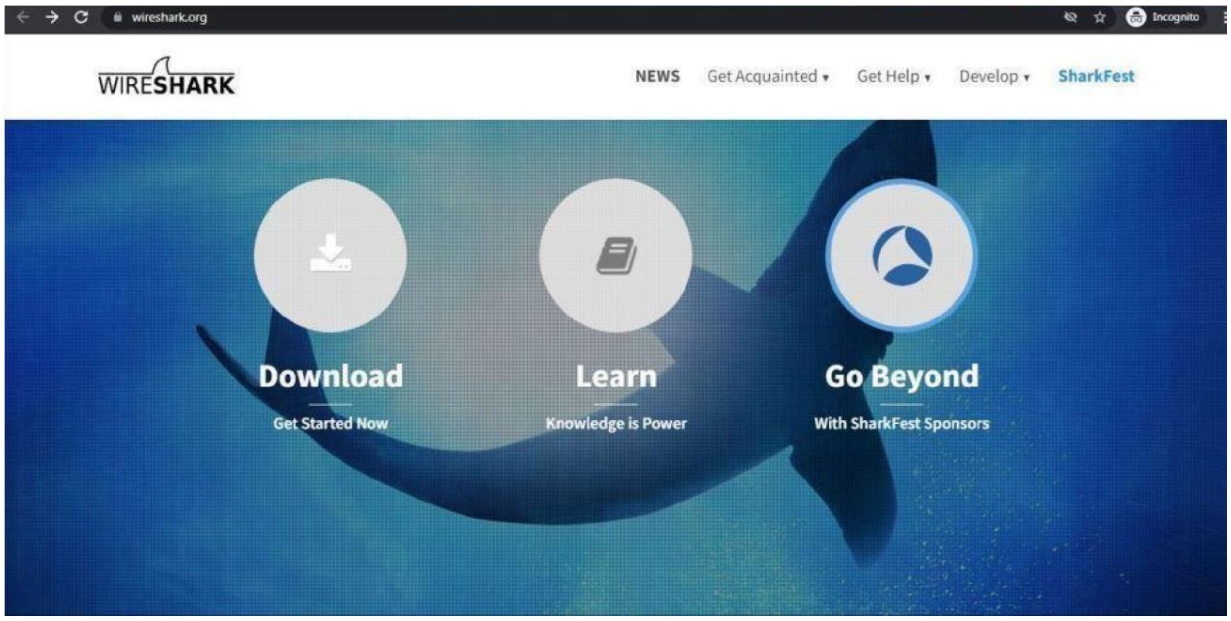
# Practical No – 3

**AIM: Installation of Wire Shark**

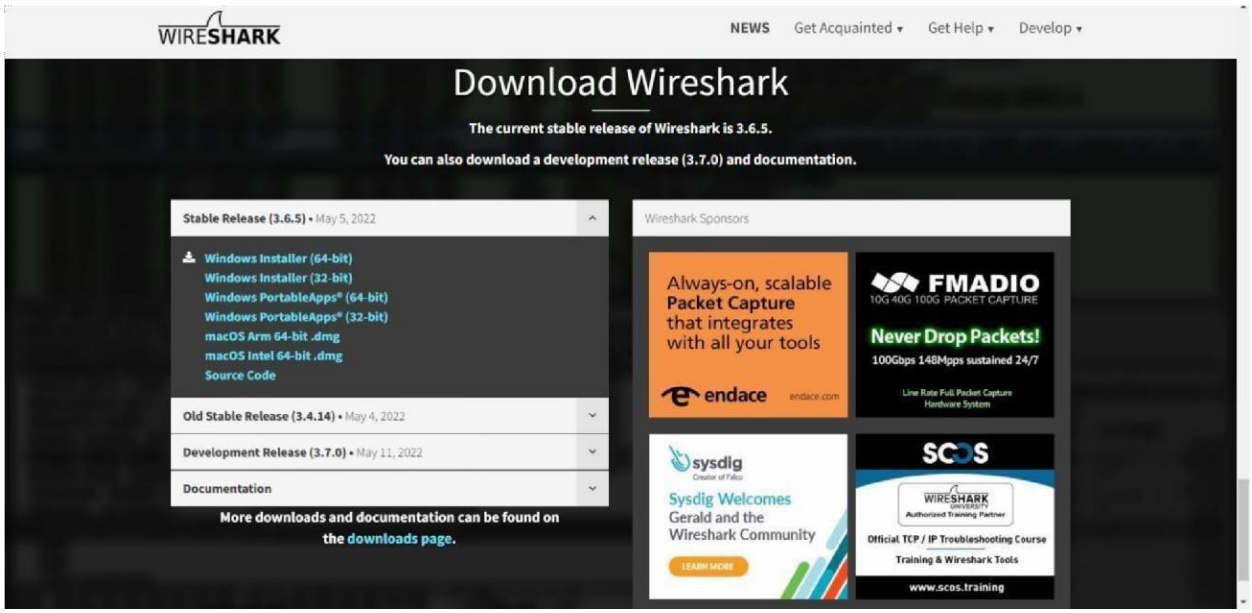
**Objective:** To learn to Install Wire Shark in Ubuntu Linux

**Step to Install:**

1] Visit the official Wireshark website using any web browser.




2] Click on Download, a new webpage will open with different installers of Wireshark



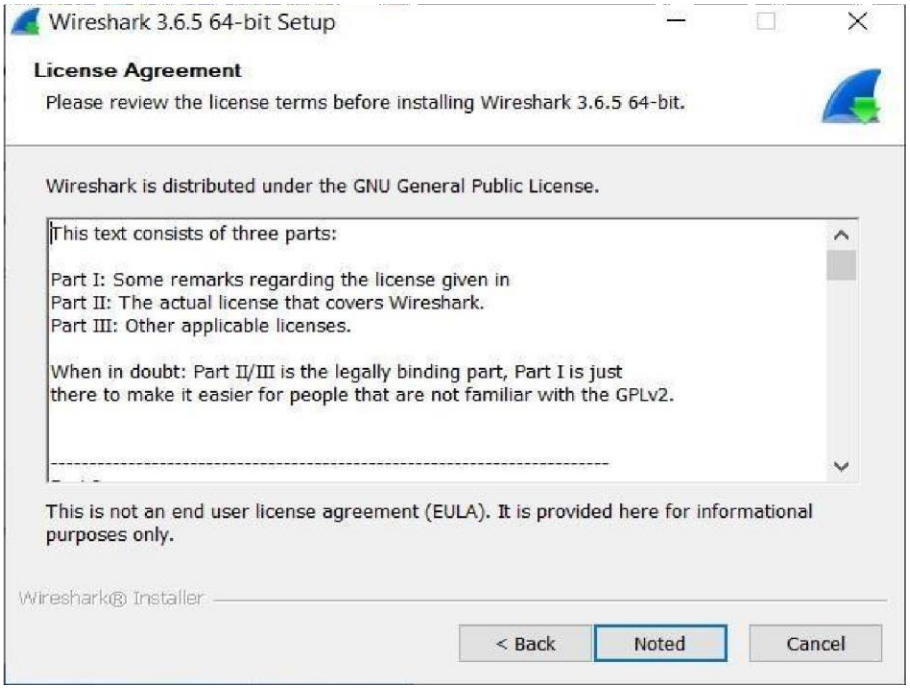
3] Downloading of the executable file will start shortly. It is a small 73.69 MB file that will take some time

4] Now check for the executable file in downloads in your system and run i

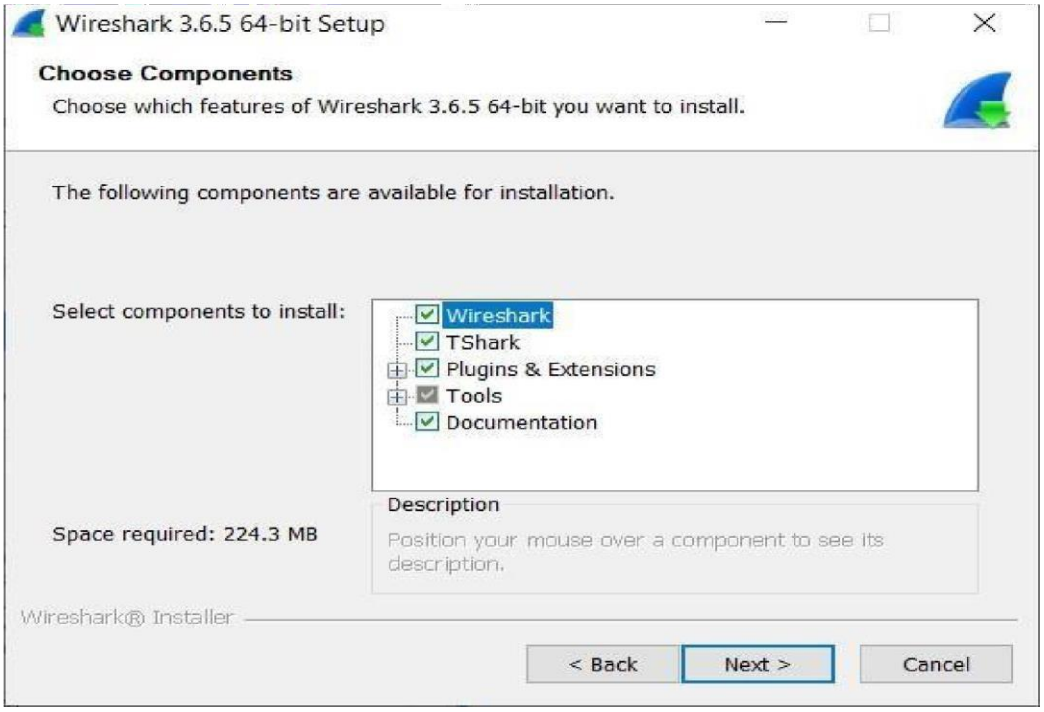
5] It will prompt confirmation to make changes to your system. Click on Ye

6] Setup screen will appear, click on Next.


7] The next screen will be of License Agreement, click on Noted.

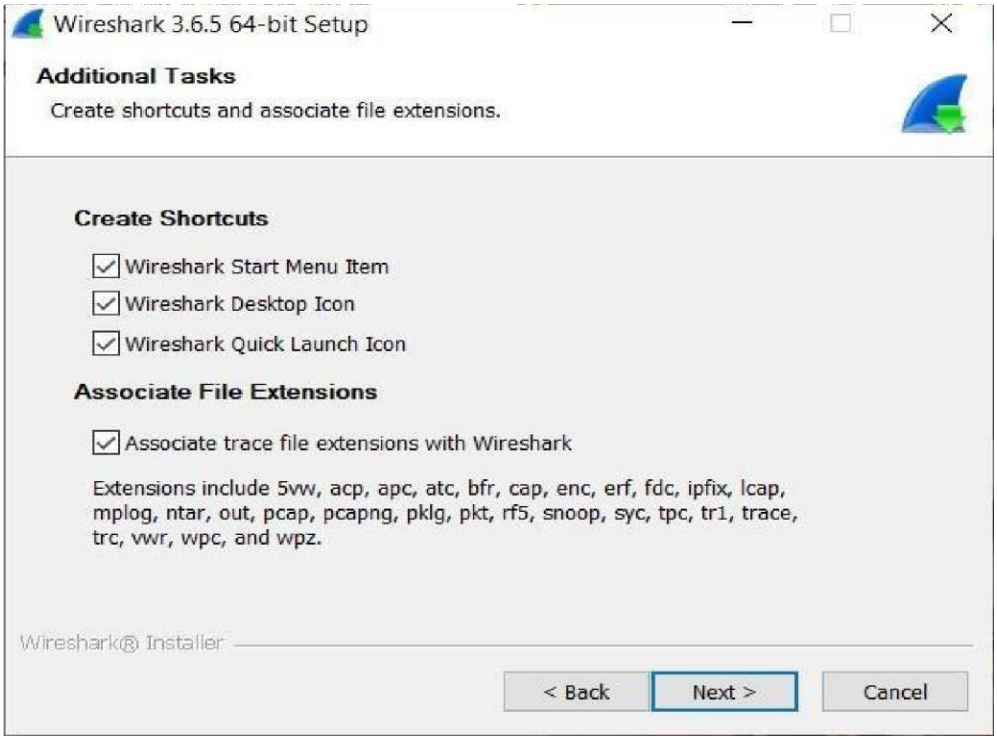


8] This screen is for choosing components, all components are already marked so don't change

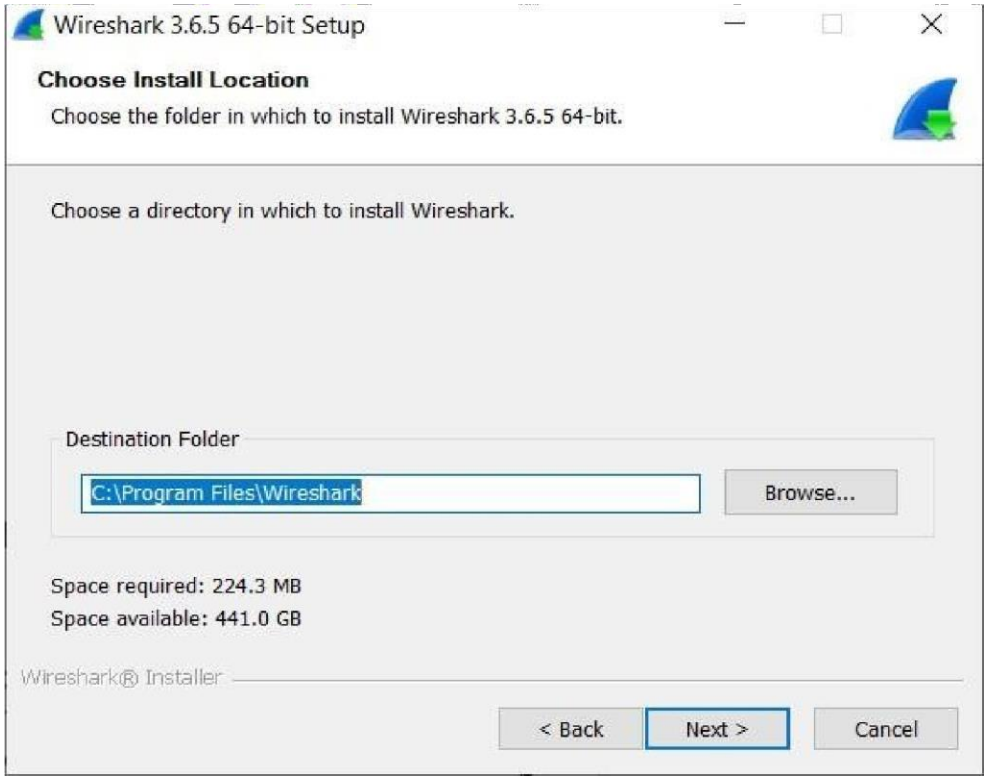



anything just click on the Next button.

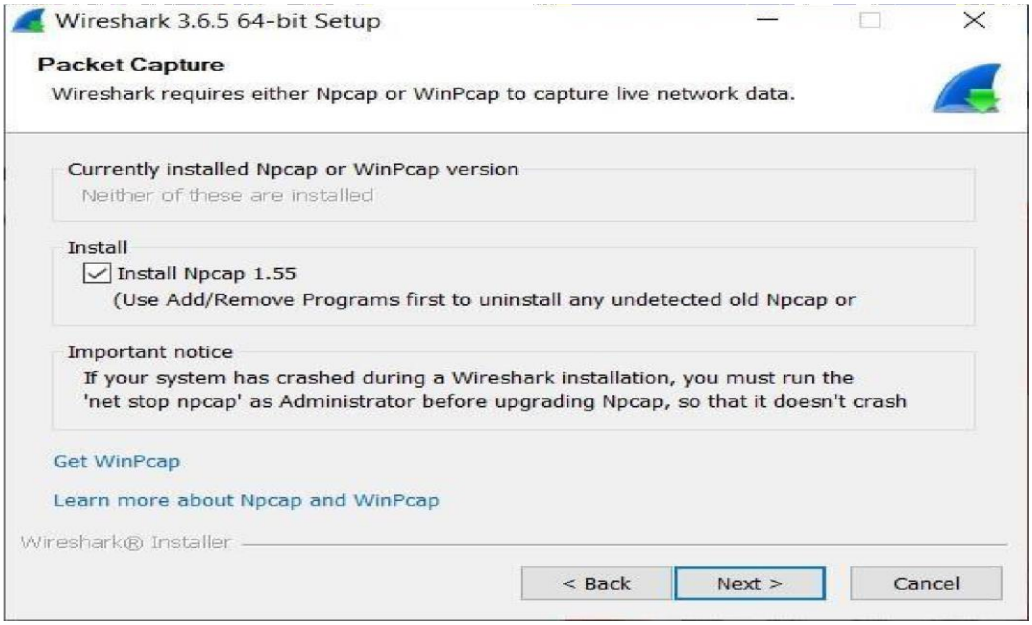
9] This screen is of choosing shortcuts like start menu or desktop icon along with file extensions which can be intercepted by Wireshark, tick all boxes and click on Next button.



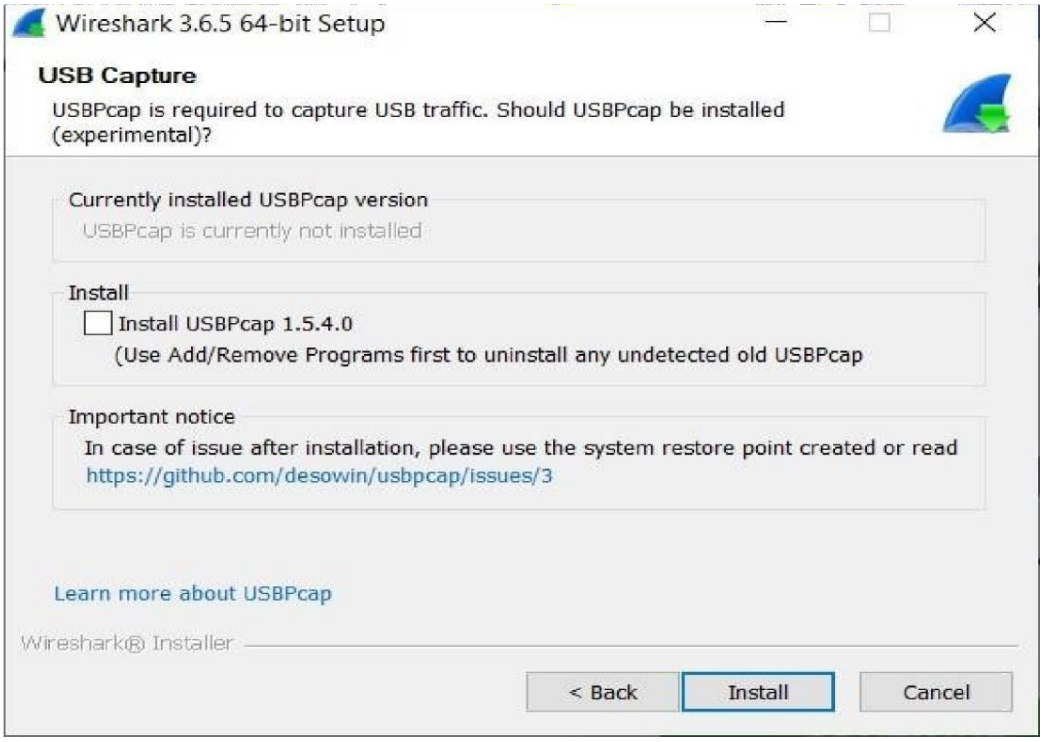
6] The next screen will be of installing location so choose the drive which will have sufficient memory space for installation.It needed only a memory space of 224.3 MB



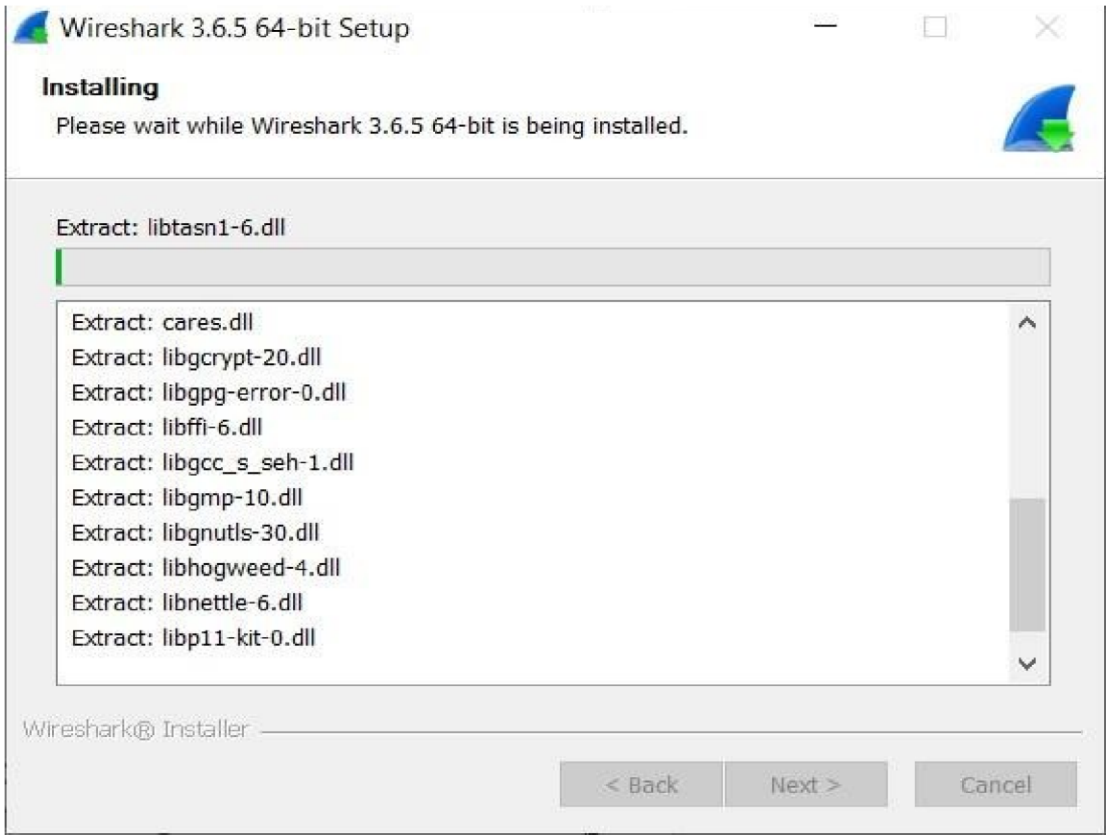

7] Next screen has an option to install Npcap which is used with Wireshark to capture packets pcap means packet capture so the install option is already checked don't change anything and click the next button.



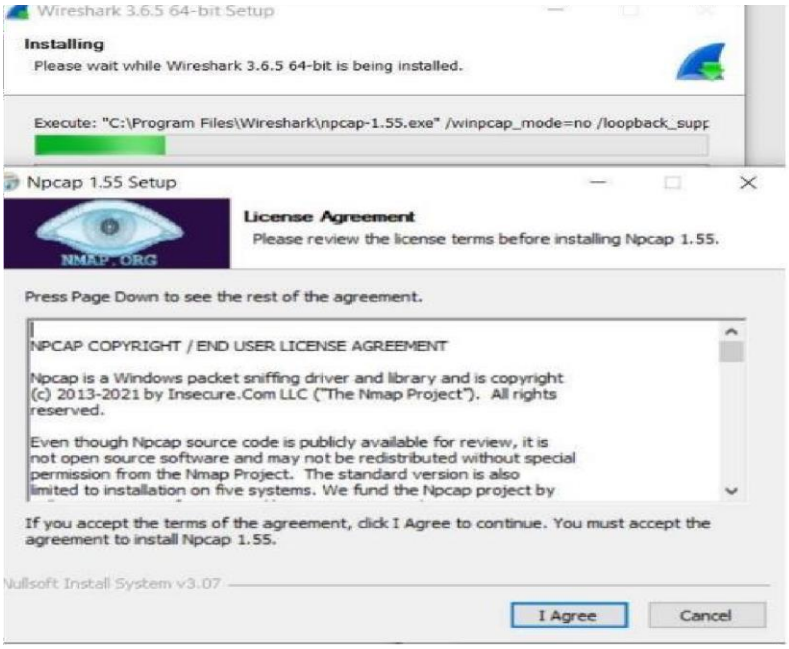
6] Next screen is about USB network capturing so it is one's choice to use or not, click on Install.



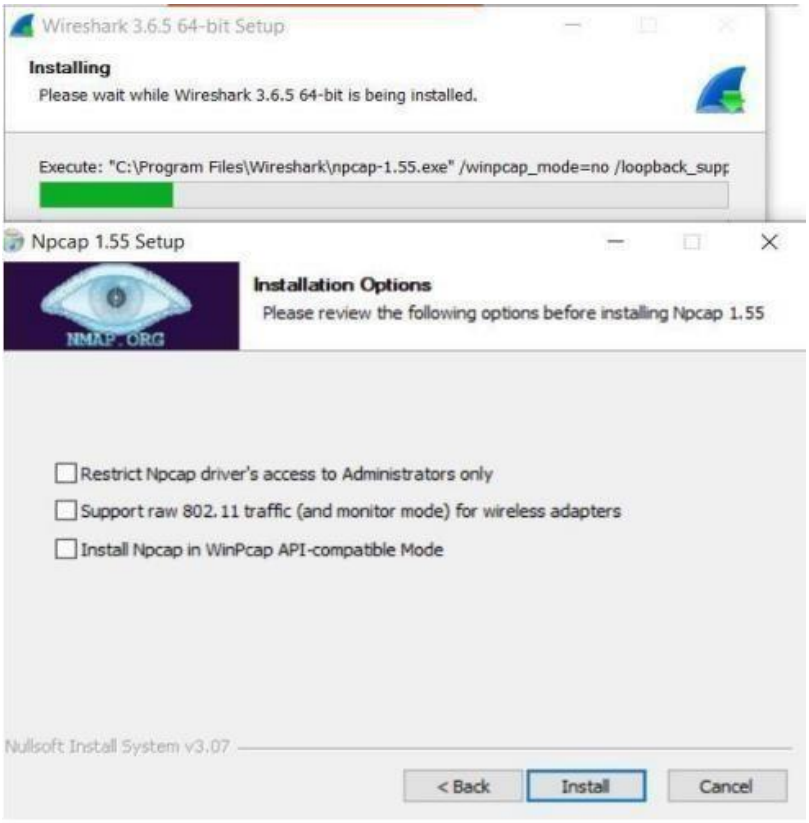

7] After this installation process will start.



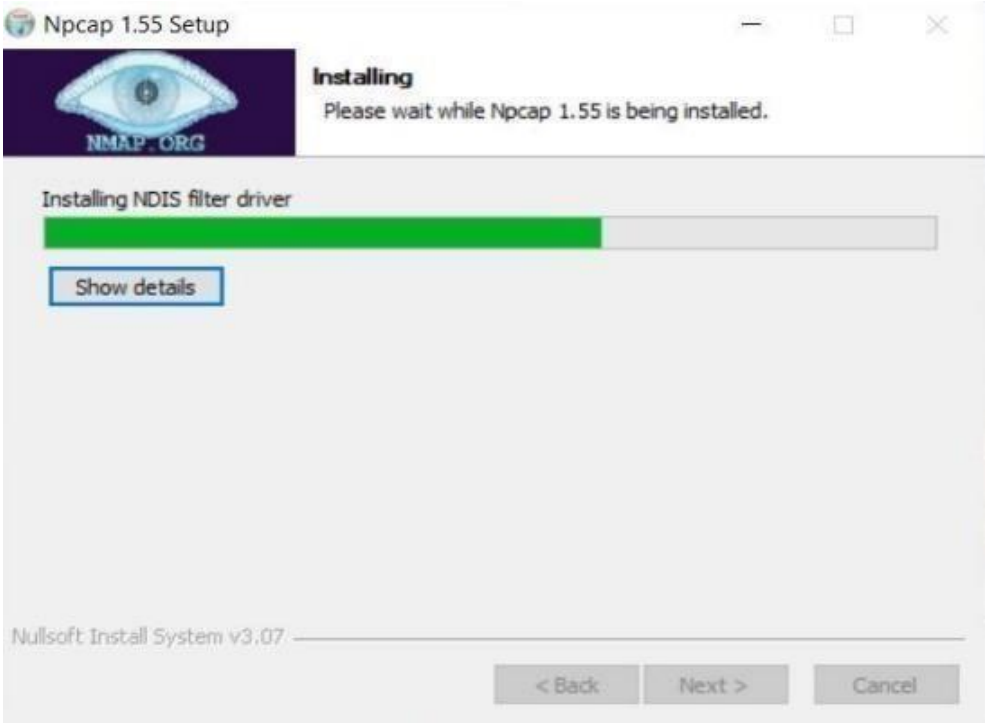
6] This installation will prompt for Npcap installation as already checked so the license agreement of Npcap will appear to click on the I Agreebutton.



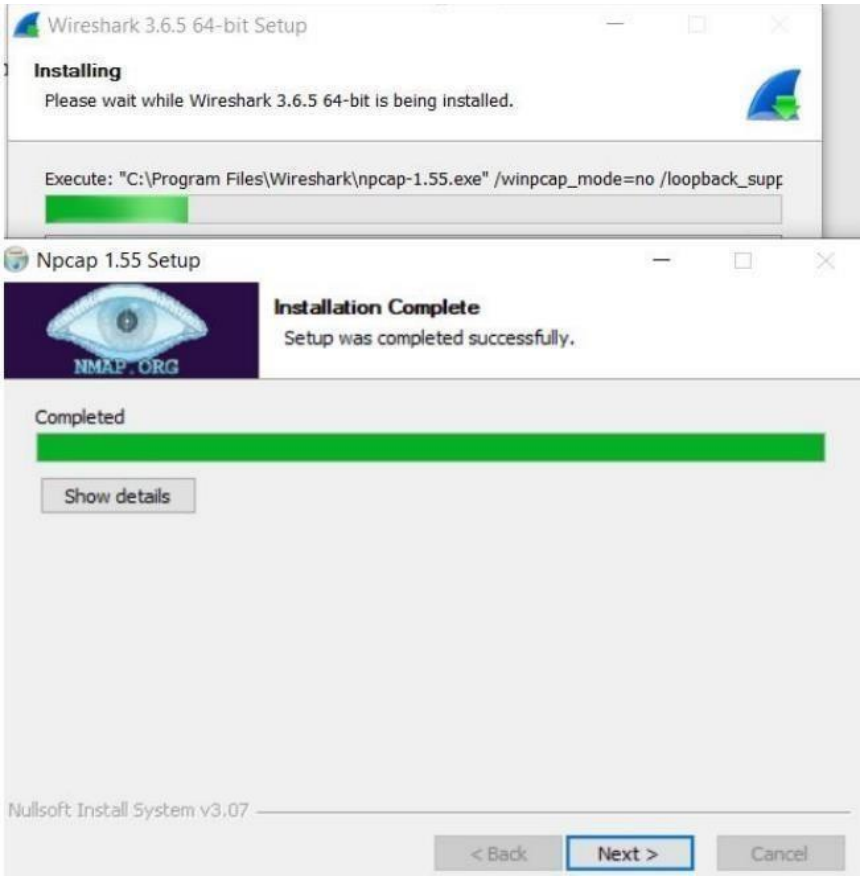

6] Next screen is about different installing options of npcap, don't do anything click on Install



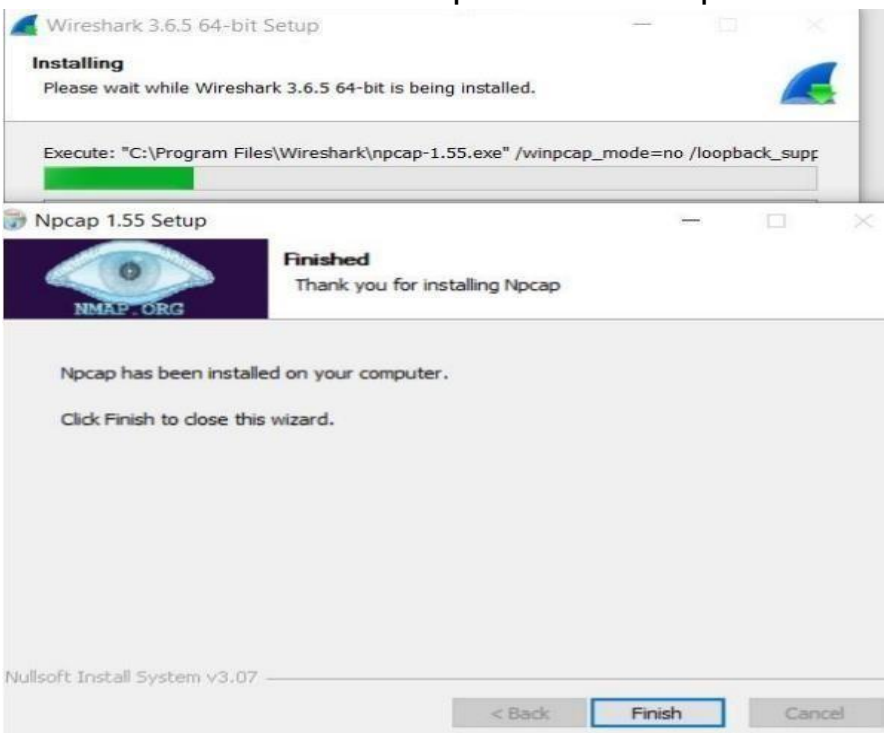
7] After this installation process will start which will take only a minute.



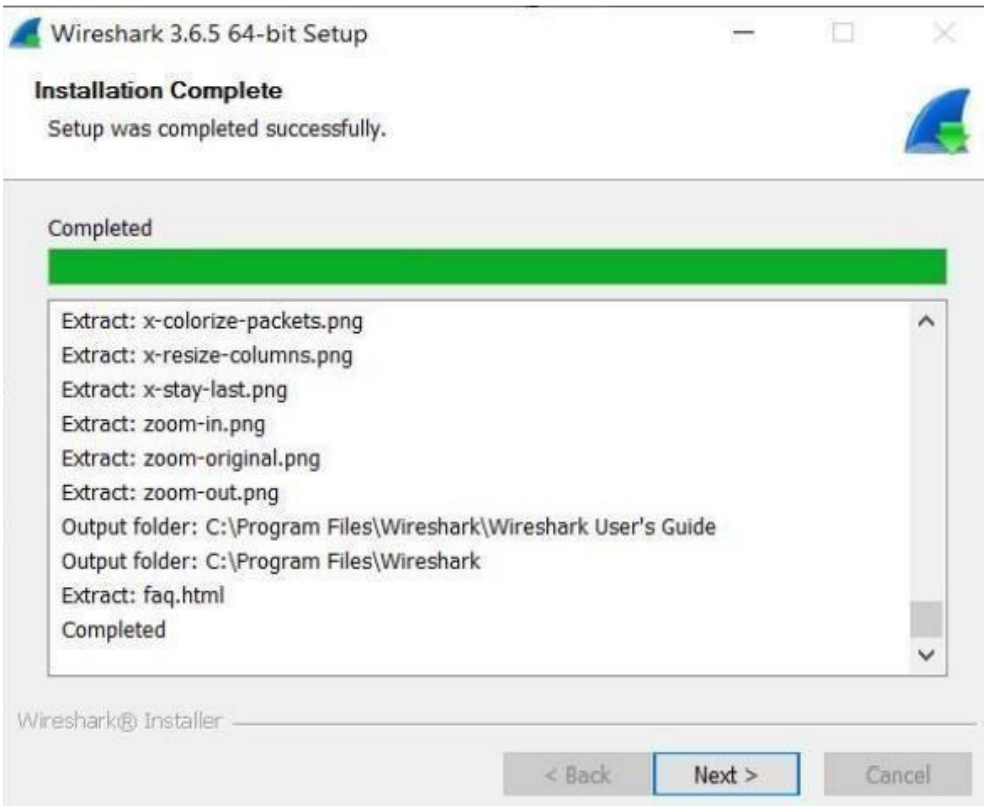

8] After this installation process will start which will take only a minute.



9] Click on Finish after the installation process is complete..




10] After this installation process of Wireshark will complete click on the Next button



11] Click on Finish after the installation process of Wireshark is complete.


12] Wireshark is successfully installed on the system and an icon is created onthe desktop

# Practical No-4

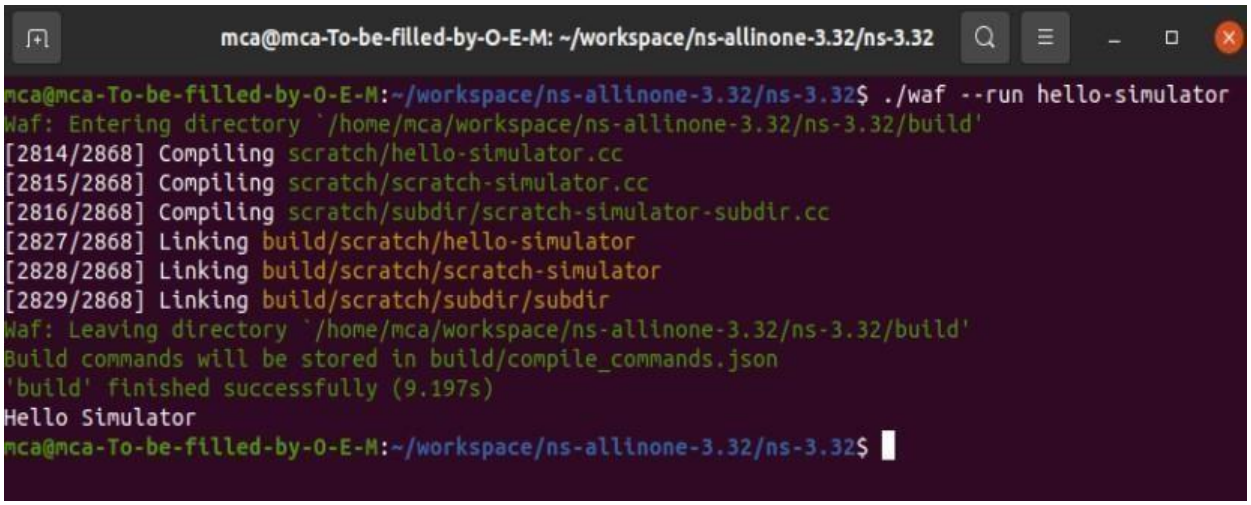
**AIM: Program to simulate traffic between two nodes**

**Objective:** To learn simulate traffic between two nodes and print String

**Code:**

Hello-Simulator.cc

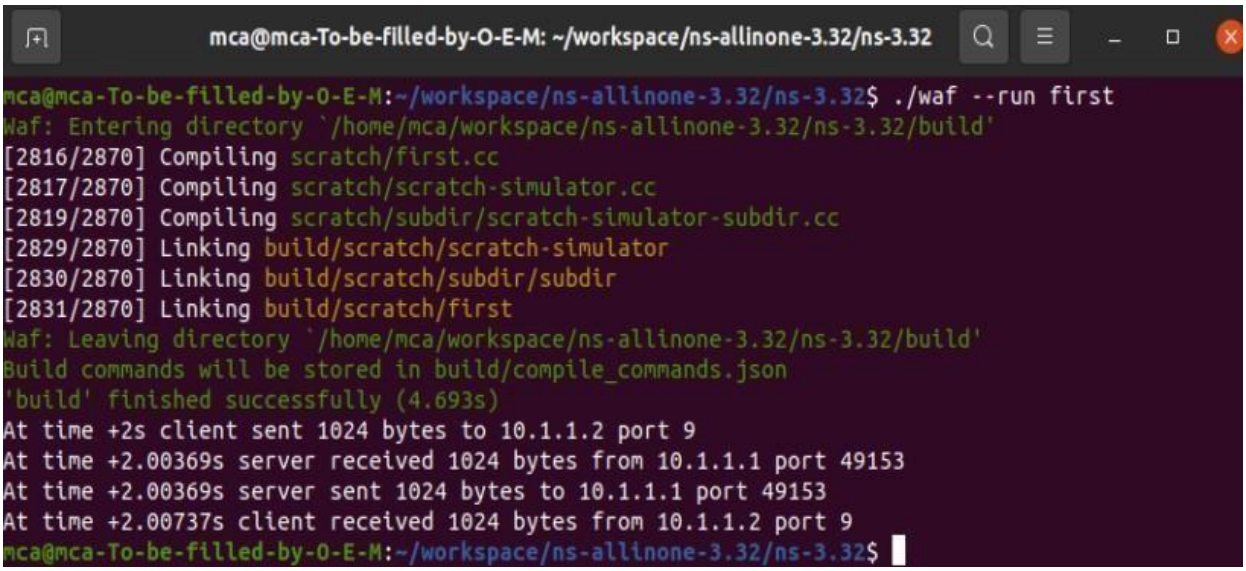
```
 /* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; - *- */
#include "ns3/core-module.h" using
namespace ns3;
NS_LOG_COMPONENT_DEFINE ("HelloSimulator");
 Int
 main (int argc, char *argv[])
 {
 NS_LOG_UNCOND ("Hello Simulator");
}
```




First.cc

```
#include  "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
// Default Network Topology
//
// 10.1.1.0
// n0 ---------- n1
// point-to-point
//
using namespace ns3;
NS_LOG_COMPONENT_DEFINE ("FirstScriptExample"); int
main (int argc, char *argv[])
{
 CommandLine cmd (__FILE__);
cmd.Parse (argc, argv);
Time::SetResolution (Time::NS);
 LogComponentEnable ("UdpEchoClientApplication",
LOG_LEVEL_INFO);
LogComponentEnable ("UdpEchoServerApplication",
LOG_LEVEL_INFO);
NodeContainer nodes;
nodes.Create (2);
 PointToPointHelper pointToPoint;
```

```
 pointToPoint.SetDeviceAttribute ("DataRate", StringValue
("5Mbps"));
 pointToPoint.SetChannelAttribute ("Delay", StringValue
("2ms"));
 NetDeviceContainer devices;  devices
= pointToPoint.Install (nodes);
InternetStackHelper stack;
stack.Install (nodes);
 Ipv4AddressHelper address; address.SetBase
("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer interfaces = address.Assign
(devices);
//server
 UdpEchoServerHelper echoServer (9);
ApplicationContainer serverApps = echoServer.Install
(nodes.Get (1));  serverApps.Start
(Seconds (1.0));
 serverApps.Stop (Seconds (10.0));
//client
 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));
Simulator::Run ();
Simulator::Destroy ();
return 0;
}
```


mca@mca-To-be-filled-by-O-E-M: ~/workspace/ns-allinone-3.32/ns-3.32
mca@mca-To-be-filled-by-O-E-M:~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run first
Waf: Entering directory `/home/mca/workspace/ns-allinone-3.32/ns-3.32/build'
[2816/2870] Compiling scratch/first.cc
[2817/2870] Compiling scratch/scratch-simulator.cc
[2819/2870] Compiling scratch/subdir/scratch-simulator-subdir.cc
[2829/2870] Linking build/scratch/scratch-simulator
[2830/2870] Linking build/scratch/subdir/subdir
[2831/2870] Linking build/scratch/first
Waf: Leaving directory `/home/mca/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (4.693s)
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
mca@mca-To-be-filled-by-O-E-M:~/workspace/ns-allinone-3.32/ns-3.32$