**Experiment 1**

**Aim:** Introduction to Discrete-EventSimulation.

**Theory:**

**System:**

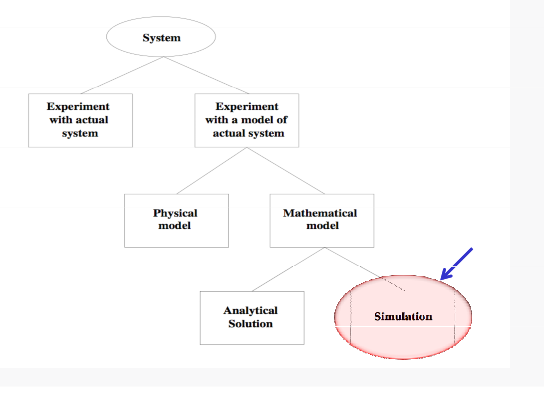
A collection of entities that act and interact together toward the accomplishment of some logical end.

**Discrete system:**

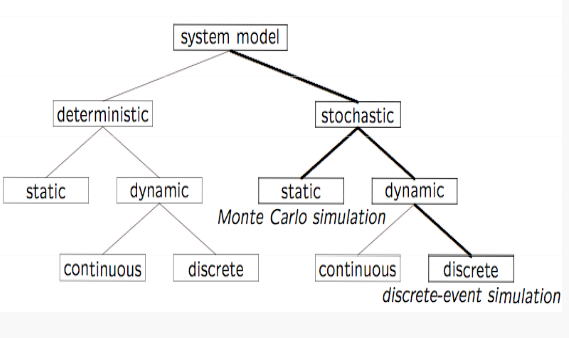
State variables change instantaneously at separated point in time, e.g., a bank, since state variables - number of customers, change only when a customer arrives or when a customer finishes being served and departs

**Continuous system:**

State variable change continuously with respect to time, e.g., airplane moving through the air, since state variables - position and velocity change continuously with respect to time



*Fig 2: Ways to study the system*



*Fig 3: Model Taxonomy*

**Why Simulation?**

ibilit f lti l lti

• Many systems are highly complex, precluding the possibility of analytical solution

•The analytical solutions are extraordinarily complex, requiring vast computing resources

•Thus, such systems should be studied by means of simulation

numerically exercising the model for inputs in question to see how they affect the output measures of performance

*“Simulation is the process of designing a model of a real system and conducting experiments with this model for the purpose either of understanding the behavior of the system or of evaluating various strategies(within the limits imposed by a criterion or set of criteria) for the operation of a system.”*

**What is Discrete-Event Simulation (DES)**

**A discrete-event simulation**

models a system whose state may change only at discrete point models a system whose state may change only at discrete point in time

**System:**

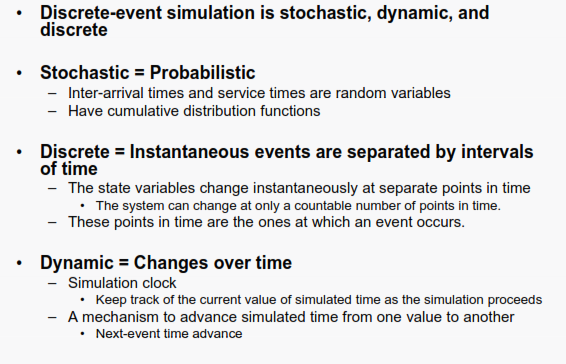
is composed of objects called entities that have certain properties called attributes.

**State:**

a collection of attributes or state variables that represent the entities of the system.

**Event:**

an instantaneous occurrence in time that may alter the state of the system



*Fig 3: Introduction to Discrete Event Simulation*

**VIVA Questions**

Q-1) What is discrete event simulation?

Ans: What do you mean by discrete event simulation?

Discrete event simulation (DES) is **a method used to model real world systems that can be decomposed into a set of logically separate processes that autonomously progress through time**. Each event occurs on a specific process, and is assigned a logical time (a timestamp).

Q-2) What is the importance and limitations of simulation?

Ans: Simulation modeling **solves real-world problems safely and efficiently**. It provides an important method of analysis which is easily verified, communicated, and understood. Across industries and disciplines, simulation modeling provides valuable solutions by giving clear insights into complex systems. Bits not atoms.

**Limitations of Simulation :**

* Simulation does not produce optimum results. ...
* Quantification of the variables is another difficulty. ...
* In very large and complex problems, the large number of variables and the inter-relationships between them make the problem very unwieldy and hard to program.

Q-3) What is the relationship between state , system and event?

Ans: The event field describes a system action that has been triggered, either automatically or by an operator. The state field by itself is the overall reported state of the transaction and should be the only field that third parties take any action on.

Q-4) What is the difference between deterministic and stochastic model?

Ans: Unlike deterministic models that produce the same exact results for a particular set of inputs, stochastic models are the opposite; the model presents data and predicts outcomes that account for certain levels of unpredictability or randomness.

Q-5) What is the proof of the correctness? Why mathematical models are more reliable than simulations?

Ans: A proof of correctness is **a mathematical proof that a computer program or a part thereof will, when executed, yield correct results, i.e. results fulfilling specific requirements**. Before proving a program correct, the theorem to be proved must, of course, be formulated. A mathematical model is a description of a system using mathematics and mathematical concepts (often pde's).

(Computer)Simulation is the simulation run (on computers) to reproduce the behaviour of a system (that has been modelled as above).

**Experiment 2**

**Aim: Introduction to NS3 and its comparison with NS2**

**Theory:**

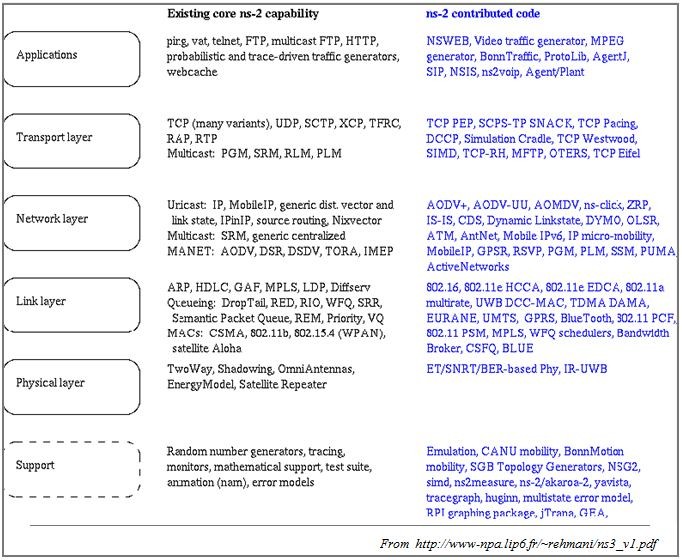
In this lab, we will be using the Network Simulator, NS3, available from [www.nsnam.org](http://www.cis.gvsu.edu/%7Ekalafuta/cis457/w13/labs/www.nsnam.org). NS3 is a powerful program, however we will only be looking at some basic features. NS3 simulations are built in C++.

**NS2 versus NS3**

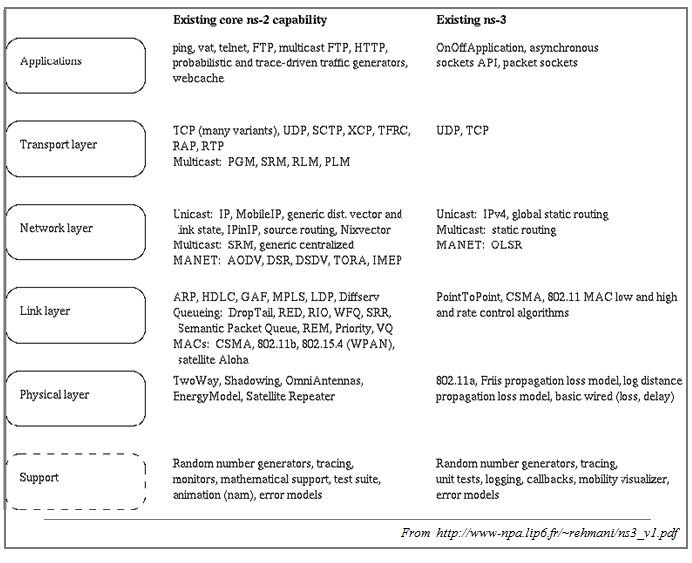
|  |  |  |
| --- | --- | --- |
|  | **NS2** | **NS3** |
| **Programming languages** | NS2 is implemented using a combination of oTCL (for scripts describing the network topology) and C++(The core of the simulator).  This system was chosen in the early  1990s to avoid the recompilation of C++ as it was very time consuming using the hardware available at that time, oTCL recompilation takes less time than C++.  oTCL disadvantage: there is overhead introduced with large simulations.  . | NS3isimplemented using C++  With modern hardware capabilities, compilation time was not an issue like for NS2, NS3 can be developed with C++entirely.  A simulation script can be written as a C++ program, which is not possible in NS2.  There is a limited support for Python in scripting and visualization. |
| **MemoryManagement** | NS2requiresbasicmanualC++  Memory management functions. | Because NS3 is implemented in C++, all normal C++ memory management functions such as new, delete, malloc, and free are still available.  Automatic de-allocation of objects is supported using reference counting (track number of pointers to an object);this is useful when dealing with Packet objects. |
| **Packets** | A packet consists of 2 distinct regions; one for headers, and the second stores payload data.  NS2 never frees memory used to store packets until the simulation  terminates, it just reuses the allocated packets repeatedly, as a result, the header region of any packet includes all headers defined as part of the used protocol even if that particular packet won't use that particular header  ,but just to be available when this  Packet allocation is reused. | A packet consists of a single buffer of bytes, and optionally a collection of small tags containing meta-data.  The buffer corresponds exactly to the stream of bits that would be sent over are al network.  Information is added to the packet by using subclasses; Header, which adds  Information to the beginning of the  buffer, Trailer, which adds to the end.  Unlike NS2, there is generally easy way to determine if a specific header is attached. |
| **Performance** | The total computation time required to run a simulation scales better in NS3 than NS2.  This is due to the removal of the overhead associated with interfacing oTcl with C++,and the overhead associated with the oTcl interpreter. | NS3 performs better than NS2 in terms of memory management.  The aggregation system prevents unneeded parameters from being stored, and packets don't contain unused reserved header space. |
| **Simulation output** | NS2 comes with a package called NAM (Network Animator), it's aTcl based animation system that produces a visual representation of the network described. | NS3 employs a package known as PyViz, which is a python based real- time visualization package |

Some Important Points about NS3:

1. NS3isnotbackwardcompatiblewithNS2;it's builtfromthescratchtoreplaceNS2.
2. NS3 is written in C++, Python Programming Language can be optionally used as an interface.
3. NS3 is trying to solve problems present in NS2.
4. There is very limited number of contributed codes made with NS3 compared to NS2
5. In NS2, bi-language system make debugging complex (C++/Tcl), but for NS3 only knowledge of C++ is enough (single-language architecture is more robust in the long term).
6. NS3 has an emulation mode, which allows for the integration with real networks.

**

***Figure4:*** *NS2contributedcode*



**Figure3:** NS2 and NS3 existing core capabilities (in July 2010)

VIVA Questions

Q-1) How the scripting language is different from programming language?

Ans: The basic difference between a scripting language and a programming language is that scripting languages do not need an additional step of compilation and rather they are interpreted, whereas programming languages are compiled and hence need a compilation step to convert the high-level language to machine code.

Q-2) State the importance of NS3 over NS2?

Ans: Ns 2 is used for wired and wireless simulation wheras **ns3 is used for internet simulation**. NS3 is not backward compatible with NS2; it's built from the scratch to replace NS2. • NS3 is written in C++, Python Programming Language can be optionally used as an interface. • NS3 is trying to solve problems present in NS2.

Q-3) Why it is debated to work on NS2 for research work instead of NS3 for research works?

Ans: NS2 is better for research because **Complex scenarios can be easily tested**. Results can be quickly obtained-more ideas can be tested in a smaller time frame.

Q-4) What are the parameters on which NS2 can be differentiated with the NS3?

Ans: Supporting tools, Supporting languages, Modules, Supporting domains etc.

**Experiment 3**

**Aim:** Installation of NS3

**Theory**: Following are the basic steps which must be followed for installing NS3

1. Install prerequisite packages
2. Download ns3 codes
3. Build ns3
4. Validate ns3

**Prerequisite packages for Linux are as follows:**

1.Minimal requirements for Python: gcc g++ python

2. Debugging and GNU Scientific Library (GSL) support: gdbpython-dev

3. valgrind gsl-bin libgsl0-dev libgsl0ldbl Network Simulation Cradle (nsc): flex bison

Reading pcap packet traces: tcpdump

4.Database support for statistics framework: sqlite sqlite3

5. Xml-based version of the config store: libxml2

6.A GTK-based configuration system: libgtk2.0-0

7. Experimental with virtual machines and ns-3: vtun lxc

**Detail steps are as follows:**

1.sudo apt-get update / dnf update

2.sudo apt-get upgrade / dnf upgrade

3 Once ubuntu/fedora is installed run following command opening the terminal(ctrl+alt+T) window.

4.To install prerequisites dependancy packages- Type the following command in terminal window.

sudo apt-get/ dnf install gcc g++ python python-dev mercurial bzr gdb valgrind gsl-bin libgsl0-dev libgsl0ldbl flex bison tcpdump sqlite sqlite3 libsqlite3-dev libxml2 libxml2-dev libgtk2.0-0 libgtk2.0-dev uncrustify doxygen graphviz imagemagick texlive texlive-latex-extra texlive-generic-extra texlive-generic-recommended texinfo dia texlive texlive-latex-extra texlive-extra-utils texlive-generic-recommended texi2html python-pygraphviz python-kiwi python-pygoocanvas libgoocanvas-dev python-pygccxml

5.After downloading NS3 on the drive, extract all the files in the NS3 folder, which you have created.

6.Then you can find build.py along with other files in NS3 folder.

Then to build the examples in ns-3 run :

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

If the build is successful then it will give output

 "Build finished successfully".

7. Now run the following command on the terminal window,to configure with waf(build tool)

  ./waf -d debug --enable-examples --enable-tests configure

To build with waf(optional)

 ./waf

8.To test everything all right run the following command on the terminal window,

 ./test.py

 If the tests are ok the installation is done

9.Now after installing ns3 and testing it run some programs first to be ns3 user:

make sure you are in directory where waf script is available then run

VIVA Questions

Q1 What protocols does ns support?

Ans: It provides **support** for OSI or TCP/IP **protocols** stack and many standard routing and application **protocols** for wire and wireless networks.

Q2. How should one can start doing something (like implementing a new protocol or trying an experiment)?

## Ans: The components of an experiment protocol

### ****Purpose****

This is a concise statement of what question you are trying to answer and what hypothesis you will test.

### ****Materials****

List all important items needed to carry out the experiment. This list need not be long or exhaustive, but it should include the essentials. Use bullet points or a table to make the list clear.

### ****Methods****

* How will you set up your experiment?
* How many experimental groups will you have?
* How will you measure the effect you wish to study?
* How long will the experiment last?

These and any other methods should be stated or referenced so that a reader has all the information they need to be able to replicate your experiment and verify your results.

### ****Controls****

Specify the control treatment. Think about the variables to be manipulated. Your control needs to be held under conditions which are not affected by the tested variable.

### ****Data interpretation****

What will be done with the data once it is collected?

Data must be organised and summarised so that the scientist and other researchers can determine if the hypothesis has been proved true or false. Results are usually shown in figures such as tables and graphs. Statistical analyses are often carried out to compare manipulated and controlled groups.

### ****References****

Any published works such as journals, books, and websites cited in your protocol should be listed in the reference section so that anyone can refer to that work.

Keep your protocol less than 2 pages long.

Q3. What is waf and its importance in the simulation process?

Ans: A WAF **protects your web apps by filtering, monitoring, and blocking any malicious HTTP/S traffic traveling to the web application, and prevents any unauthorized data from leaving the app**. It does this by adhering to a set of policies that help determine what traffic is malicious and what traffic is safe.

Q4. How we can test that NS3 has been installed properly?

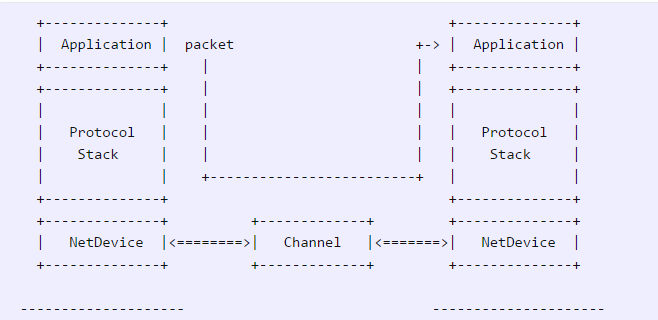
Ans: To check your version, **type "sphinx-build"**. To fetch this package alone, outside of the Ubuntu package system, try "sudo easy\_install -U Sphinx".

Q5. Why prerequisite packages are needed before the installation of NS3?

Ans: ns-3 can be built and used with support for a few common (often installed by default) components: a C++ compiler, an installation of Python, a source code editor (such as vim, emacs, or Eclipse) and, if using the development repositories, an installation of Git source code control system. Most beginning users need not concern themselves if their configuration reports some missing optional features of ns-3, but for those wishing a full installation.

**EXPERIMENT 4**

**Aim**: Program in NS3 to connect two nodes

**Theory:**

### Node:

Because in any network simulation, we will need nodes. So ns-3 comes with [NodeContainer](https://www.nsnam.org/doxygen/classns3_1_1_node_container.html) that you can use to manage all the nodes (Add, Create, Iterate, etc.).

// Create two nodes to hold.

NodeContainer nodes;

nodes.Create (2);

### Channel and NetDevice

In the real world, they correspond to network cables (or wireless media) and peripheral cards (NIC). Typically these two things are intimately tied together. In the first example, we are usingPointToPointHelper that wraps the Channel and NetDevice.

// Channel: PointToPoint, a direct link with `DataRate` and `Delay` specified.

PointToPointHelper pointToPoint;

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

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

Then we need to install the devices. The internal of Install is actually more complicated, but for now, let’s just skip the magic behind the scene.

// NetDevice: installed onto the channel

NetDeviceContainer devices;

devices = pointToPoint.Install (nodes);

### Protocols

Internet and IPv4. Since Internet is the current largest network to study, ns-3 has a particular focus on it. The InternetStackHelper will install an Internet Stack (TCP, UDP, IP, etc.) on each of the nodes in the node container.

// Protocol Stack: Internet Stack

InternetStackHelper stack;

stack.Install (nodes);

To assign IP addresses, use a helper and set the base. The low level ns-3 system actually remembers all of the IP addresses allocated and will generate a fatal error if you accidentally cause the same address to be generated twice.

// Since IP Address assignment is so common, the helper does the dirty work!

// You only need to set the base.

Ipv4AddressHelper address;

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

// Assign the address to devices we created above

Ipv4InterfaceContainer interfaces = address.Assign (devices);

### Applications

Every application needs to have Start and Stop function so that the simulator knows how to schedule it. Other functions are application-specific. We will use UdpEchoServer and UdpEchoClientfor now.

// Application layer: UDP Echo Server and Client

// 1, Server:

UdpEchoServerHelper echoServer (9);

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

serverApps.Start (Seconds (1.0));

serverApps.Stop (Seconds (10.0));

// 2, 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));

### Simulation

// Start Simulation

Simulator::Run ();

Simulator::Destroy ();

return 0;

**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/netanim-module.h"

using namespace ns3;

NS\_LOG\_COMPONENT\_DEFINE ("FirstScriptExample");

int

main (int argc, char \*argv[])

{

Time::SetResolution (Time::NS);

LogComponentEnable ("UdpEchoClientApplication", LOG\_LEVEL\_INFO);

LogComponentEnable ("UdpEchoServerApplication", LOG\_LEVEL\_INFO);

NodeContainer nodes;

nodes.Create (2);

PointToPointHelper pointToPoint;

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

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

NetDeviceContainer devices;

devices = pointToPoint.Install (nodes);

InternetStackHelper stack;

stack.Install (nodes);

Ipv4AddressHelper address;

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

Ipv4InterfaceContainer interfaces = address.Assign (devices);

UdpEchoServerHelper echoServer (9);

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

serverApps.Start (Seconds (1.0));

serverApps.Stop (Seconds (10.0));

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

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

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

echoClient.SetAttribute ("PacketSize", UintegerValue (1024));

ApplicationContainer clientApps = echoClient.Install (nodes.Get (0));

clientApps.Start (Seconds (2.0));

clientApps.Stop (Seconds (10.0));

AnimationInterface anim(“myfirst.xml”);

Anim.SetConstantPosition(nodes.Get(0),1.0,1.0);

Anim.SetConstantPosition(nodes.Get(1),10.0,10.0);

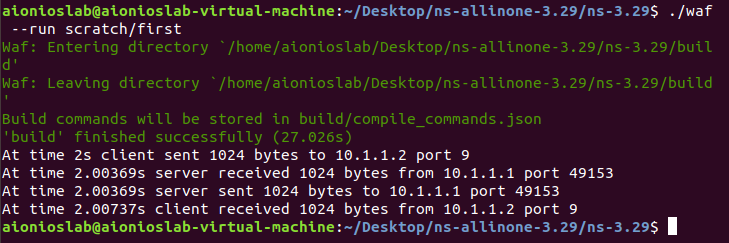
Simulator::Run ();

Simulator::Destroy ();

return 0;

}

**OUTPUT**



VIVA Questions

Q1 which probable protocols can come into play when 2 nodes are connected?

Ans: Bus,Ring,Start,Tree and Mesh .

Q2. What is the difference between UDP and TCP?

Ans: **TCP is a connection-oriented protocol, whereas UDP is a connectionless protocol**. A key difference between TCP and UDP is speed, as TCP is comparatively slower than UDP. Overall, UDP is a much faster, simpler, and efficient protocol, however, retransmission of lost data packets is only possible with TCP.

Q3. What is a Point to Point Connection? What are its characteristics?

Ans: A point-to-point connection is a permanent direct communication link between two parties.

**PPP has the following three main components:**

* a way to encapsulate multiprotocol datagrams;
* Link Control Protocol to establish, configure and test the data link connection; and.
* a group of separate network control protocols that establish and configure different types of network layer protocols.

Q4. What will happen if the client starts first then the server?

Ans: The client will not be able to access the web app hosted on the server.

Q5. What is the difference between IPv4 and IPv6?

Ans: The main difference between IPv4 and IPv6 is the **address size of IP addresses**. The IPv4 is a 32-bit address, whereas IPv6 is a 128-bit hexadecimal address. IPv6 provides a large address space, and it contains a simple header as compared to IPv4.

**EXPERIMENT 5**

**Aim:** Program in NS3 for connecting three nodes considering one node as a central node.

**Theory:**

***Comment : Similar to Two Node program we can perform an experiment to simulate a 3 node topology considering one node as Server (Central Node) and two nodes as client nodes.***

**CODE:-**

#include "ns3/core-module.h"

# include "ns3/core-netanim.h"

#include "ns3/network-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 ("FirstScriptExample");

int main (int argc, char \*argv[])

{

Time::SetResolution (Time::NS);

LogComponentEnable ("UdpEchoClientApplication", LOG\_LEVEL\_INFO);

LogComponentEnable ("UdpEchoServerApplication", LOG\_LEVEL\_INFO);

NodeContainer nodes;

nodes.Create (3);

PointToPointHelper pointToPoint;

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

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

NetDeviceContainer devices, devices1;

devices = pointToPoint.Install (nodes.Get(0),nodes.Get(1));

devices1 = pointToPoint.Install (nodes.Get(2),nodes.Get(1));

InternetStackHelper stack;

stack.Install (nodes);

Ipv4AddressHelper address;

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

Ipv4InterfaceContainer interfaces = address.Assign (devices);

Ipv4InterfaceContainer interfaces1 = address.Assign (devices1);

UdpEchoServerHelper echoServer (90);

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

serverApps.Start (Seconds (1.0));

serverApps.Stop (Seconds (10.0));

UdpEchoClientHelper echoClient (interfaces.GetAddress (1), 90);

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));

UdpEchoClientHelper echoClient1 (interfaces1.GetAddress (1), 90);

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

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

echoClient.SetAttribute ("PacketSize", UintegerValue (1024));

ApplicationContainer clientApps1 = echoClient1.Install (nodes.Get (2));

clientApps.Start (Seconds (2.0));

clientApps.Stop (Seconds (10.0));

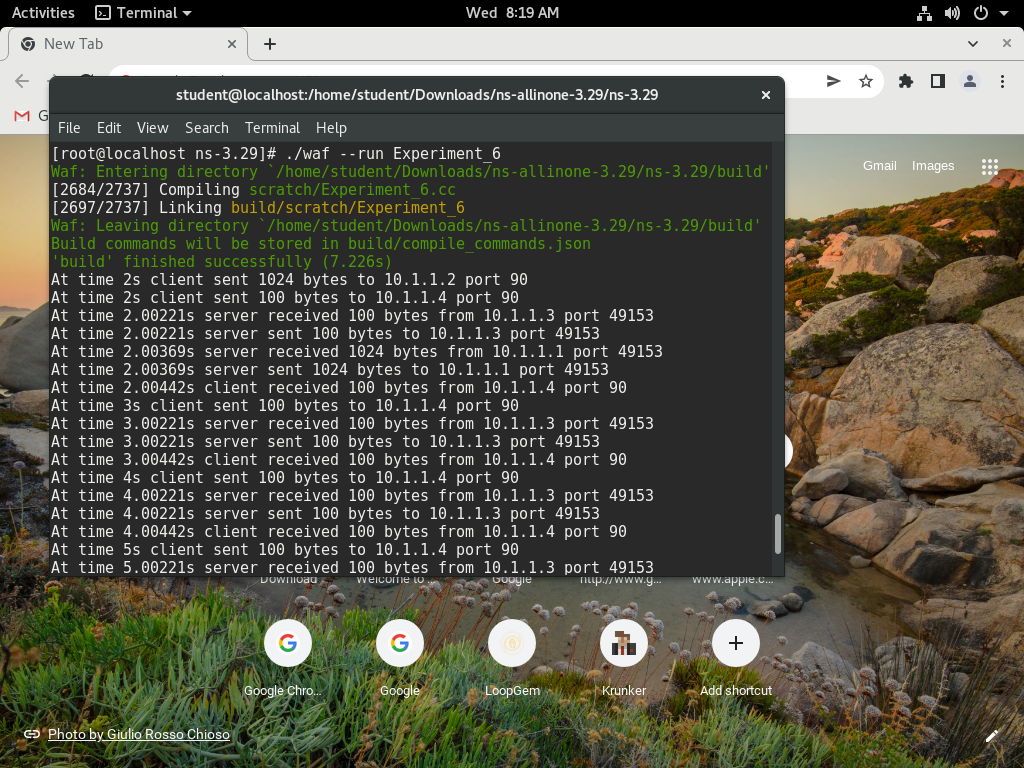
AnimationInterface anim(“threenode.xml”);

Simulator::Run ();

Simulator::Destroy ();

return 0;

}

**OUTPUT**

VIVA Questions

Q1. What if all the 3 nodes are the part of the same network and each interface is assigned the IP address of the same range?

Ans: If all the 3 nodes are the part of the same network and each interface is assigned with same IP address then the receiver will not be able to identify where from the data is sent with what IP address.

Q2. What is the port number? What is its significance?

Ans: A port number is **a way to identify a specific process to which an internet or other network message is to be forwarded when it arrives at a server**. All network-connected devices come equipped with standardized ports that have an assigned number.

Q3. What is socket address? What is the role of IP address in it?

Ans: **TCP/IP creates the socket address as an identifier that is unique throughout all Internet networks**. TCP/IP concatenates the Internet address of the local host interface with the port number to devise the Internet socket address. With TCP/IP, sockets are not tied to a destination address.

Q4. How a NS3 program is simulated? Explain various stages.

Ans:

**Install the NS-3.26 Tool**

Initially , install the NS-3.26 tool by using the ns-allinone-3.26.tar.bz2 package. For make install use and execute the command ./build.py

**Open the terminal**

Initially,Next open the terminal by press ctrl+alt+T buttons or search from the installed software list.

**Configure the package**

Change the ns-allinone installation location in the terminal, using the command cd ns-allinone-3.26/ns-3.26. And execute the command sudo ./waf configure to binding the python

**Build the package**

For build the configured package by using the sudo ./waf build command in the ns-allinone-3.26/ns-3.26 . Package and view the build packages.

**Create a main file**

Next create new program file in the scratch folder. The program file stored with the file extension .cc.

**Construct the network**

Construct the network by using the nodecontainer , with create fuction.

**Node Configuration**

In the create function we specify the no of nodes. And lte configuration

**Execute the main file**

Next, execute the main file by using the command sudo ./waf –run handover –vis

**Run the Simulation Result**

Execute the main file by using the command sudo ./waf –run handover –vis and get the  
simulation result successfully.

Q5. What is the difference between IP address and physical address?

Ans: The primary use of a MAC address is to ensure the physical address of a given device/ computer. **The IP address, on the other hand, defines a computer's logical address**. The MAC address primarily operates on the data link layer. The IP address primarily operates on the network layer.