Name of Student : Sunny Satish Halkatti		
Roll Number : 17		**LAB Assignment Number: 8**
Title of LAB Assignment: Animate a simple network using NetAnim in Network Simulator.		
DOP : 11-05-2023		**DOS :** 22-05-2023
CO Mapped : CO2,CO4	**PO Mapped:** PO1, PO2, PO3,PO4, PO5, PO7, PSO1,PSO2	**Signature :**

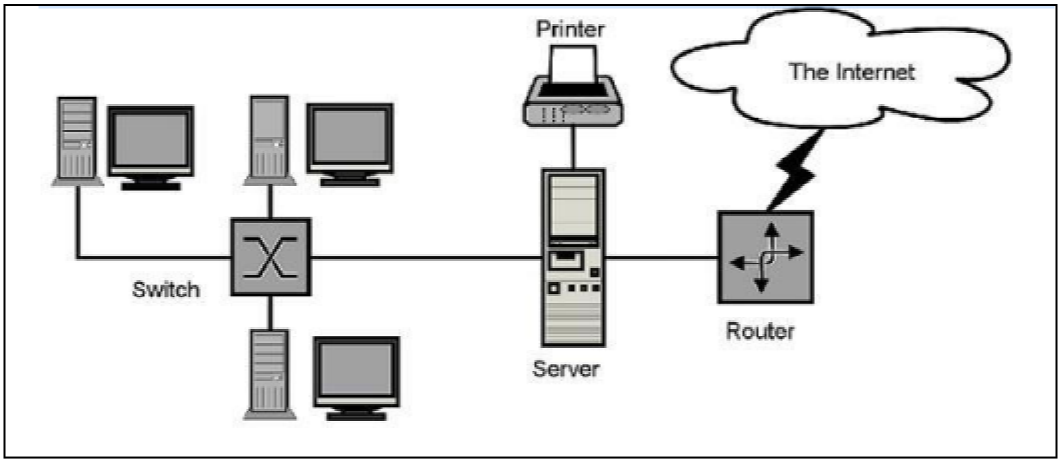
NWL - Practical - 8

Aim: Animate a simple network using NetAnim in Network Simulator.

Theory:

What is Network Simulation?

Definition: Network simulation is one kind of method in the research of a computer network where a software program forms the performance of a network by analyzing the relations between the various network entities such as links, Nswitched, routers, nodes, access points. The network performance, different applications, services & supports can be monitored in an analysis lab. Different features of the surroundings can also be changed in a controlled way to evaluate how the network or protocols would perform beneath different conditions.



Different Network Simulations:

The different types of network simulators/ network simulation tools are open source and commercial:

Network Simulator Version 2 (NS-2) -

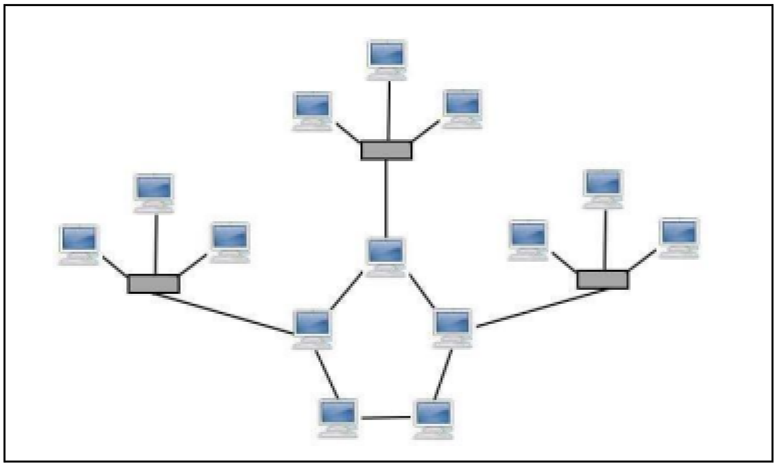
It is an object-oriented kind of simulator mainly used for simulating the protocols of networking as well as routing protocols for the networks like wired & wireless. These can be implemented through OTCl & C++.

Ns3 -

This type of simulator is mainly designed for the purpose of education as well as research. When compared with the Ns2 type, it uses Python to work in a better way because of the low-level of abstraction. The modules of Ns3 Include protocols and network devices, written in the languages of C++, Python.

Java-based Simulation (JSIM) -
It is used mainly in web-based simulation to build by using the event package otherwise Process the package. This Is used to design quantitative numeric models & estimate them with respect to the data from the experiment.

OPNET -
It is used in research & development to provide whole flexibility to study regarding communication networks, protocols, and applications. When it includes both programming environment and GUI, then it provides a platform for the user to form the network when they require it.



Code:

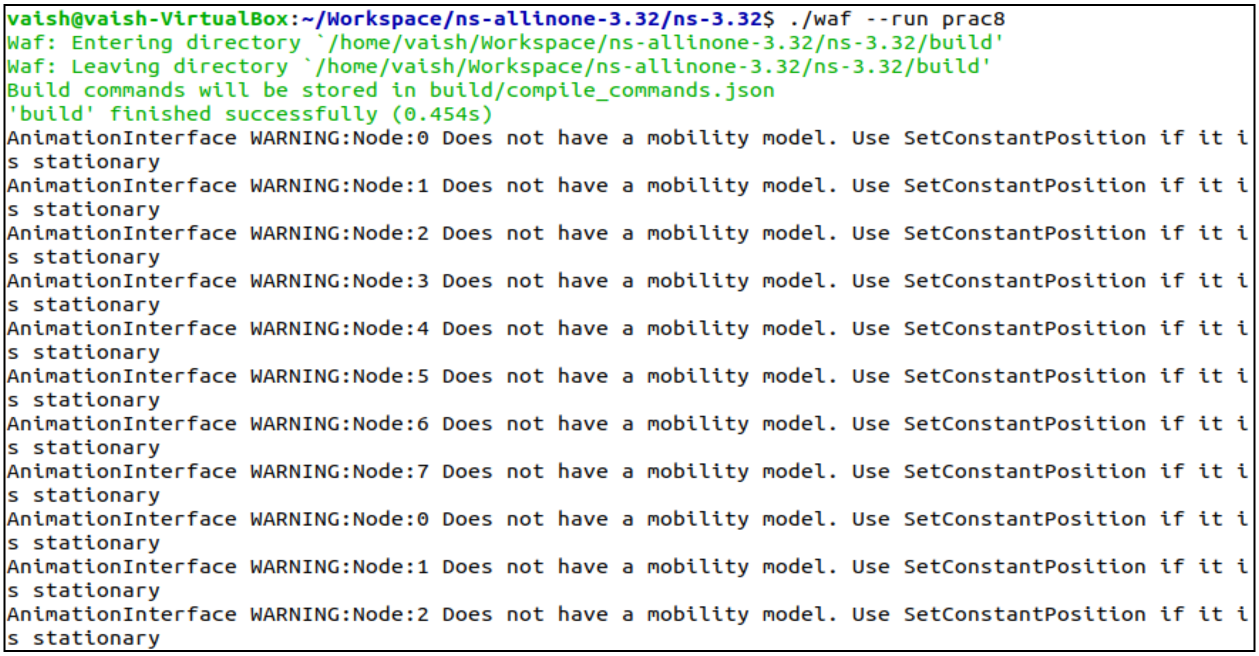
```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/csma-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
#include "ns3/ipv4-global-routing-helper.h"
#include "ns3/netanim-module.h"
#include "ns3/mobility-module.h"
using namespace ns3;
NS_LOG_COMPONENT_DEFINE("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;
   NodeContainer routers;
   nodes.Create(1);
   routers.Create(4);
   NodeContainer csmaNodes;
   csmaNodes.Create(3);
   InternetStackHelper stack;
   stack.Install(nodes);
   stack.Install(routers);
   stack.Install(csmaNodes);
   // subnet1

   PointToPointHelper pointToPoint;
   pointToPoint.SetDeviceAttribute("DataRate", StringValue("5Mbps"));
   pointToPoint.SetChannelAttribute("Delay", StringValue("2ms"));
   NodeContainer subnet1;
subnet1.Add(nodes.Get(0));
   subnet1.Add(routers.Get(0));
   NetDeviceContainer Subnet1devices;
   Subnet1devices = pointToPoint.Install(subnet1);
   Ipv4AddressHelper subnet1address;
   subnet1address.SetBase("10.1.1.0", "255.255.255.0");
   Ipv4InterfaceContainer p2pInterfaces1;
   p2pInterfaces1 = subnet1address.Assign(Subnet1devices);
   // subnet2
   NodeContainer subnet2;
   subnet2.Add(routers.Get(0));
   subnet2.Add(routers.Get(1));
   NetDeviceContainer Subnet2devices;
   Subnet2devices = pointToPoint.Install(subnet2);
   Ipv4AddressHelper subnet2address;
   subnet2address.SetBase("10.1.2.0", "255.255.255.0");
   Ipv4InterfaceContainer p2pInterfaces2;
   p2pInterfaces2 = subnet2address.Assign(Subnet2devices);
   // subnet3
   NodeContainer subnet3;
   subnet3.Add(routers.Get(1));
   subnet3.Add(routers.Get(2));
```

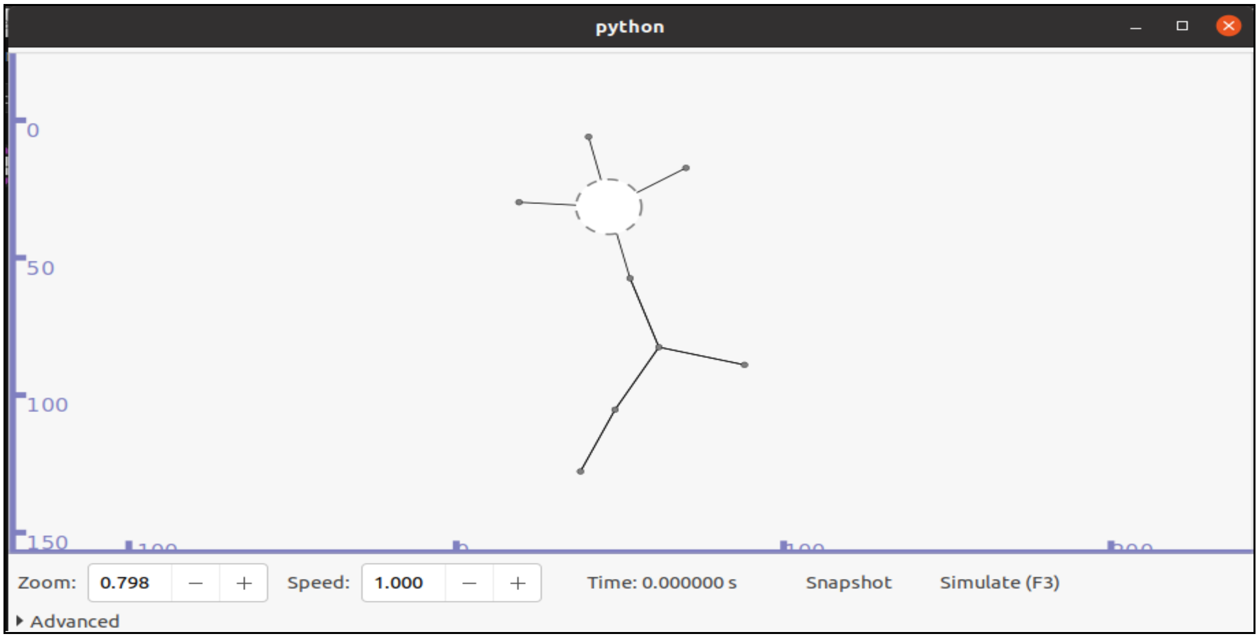
```
  NetDeviceContainer Subnet3devices;
  Subnet3devices = pointToPoint.Install(subnet3);
  Ipv4AddressHelper subnet3address;
  subnet3address.SetBase("10.1.3.0", "255.255.255.0");

  Ipv4InterfaceContainer p2pInterfaces3;
  p2pInterfaces3 = subnet3address.Assign(Subnet3devices);
  // subnet4
  NodeContainer subnet4;
  subnet4.Add(routers.Get(1));
  subnet4.Add(routers.Get(3));
  NetDeviceContainer Subnet4devices;
  Subnet4devices = pointToPoint.Install(subnet4);
  Ipv4AddressHelper subnet4address;
  subnet4address.SetBase("10.1.4.0", "255.255.255.0");
  Ipv4InterfaceContainer p2pInterfaces4;
  p2pInterfaces4 = subnet4address.Assign(Subnet4devices);
  // subnet5
  CsmaHelper csma;
  csma.SetChannelAttribute("DataRate", StringValue("100Mbps"));
  csma.SetChannelAttribute("Delay", TimeValue(NanoSeconds(6560)));
  NodeContainer subnet5;
  subnet5.Add(csmaNodes.Get(0));
  subnet5.Add(csmaNodes.Get(1));
subnet5.Add(csmaNodes.Get(2));
  subnet5.Add(routers.Get(2));
  NetDeviceContainer csmaDevices;
  csmaDevices = csma.Install(subnet5);
  Ipv4AddressHelper subnet5address;
  subnet5address.SetBase("10.1.5.0", "255.255.255.0");
  Ipv4InterfaceContainer p2pInterfaces5;
  p2pInterfaces5 = subnet5address.Assign(csmaDevices);
  UdpEchoServerHelper echoServer(9);
  ApplicationContainer serverApps = echoServer.Install(csmaNodes.Get(0));
  serverApps.Start(Seconds(1.0));
  serverApps.Stop(Seconds(10.0));
  UdpEchoClientHelper echoClient(p2pInterfaces5.GetAddress(0), 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));
    Ipv4GlobalRoutingHelper::PopulateRoutingTables();
    pointToPoint.EnablePcap("p2pSubnet", Subnet1devices.Get(0), true);
    csma.EnablePcap("csmaSubnet", csmaDevices.Get(0), true);
    AnimationInterface anim("Subnet.xml");
    Simulator::Run();
    Simulator::Destroy();
    return 0;
}
```

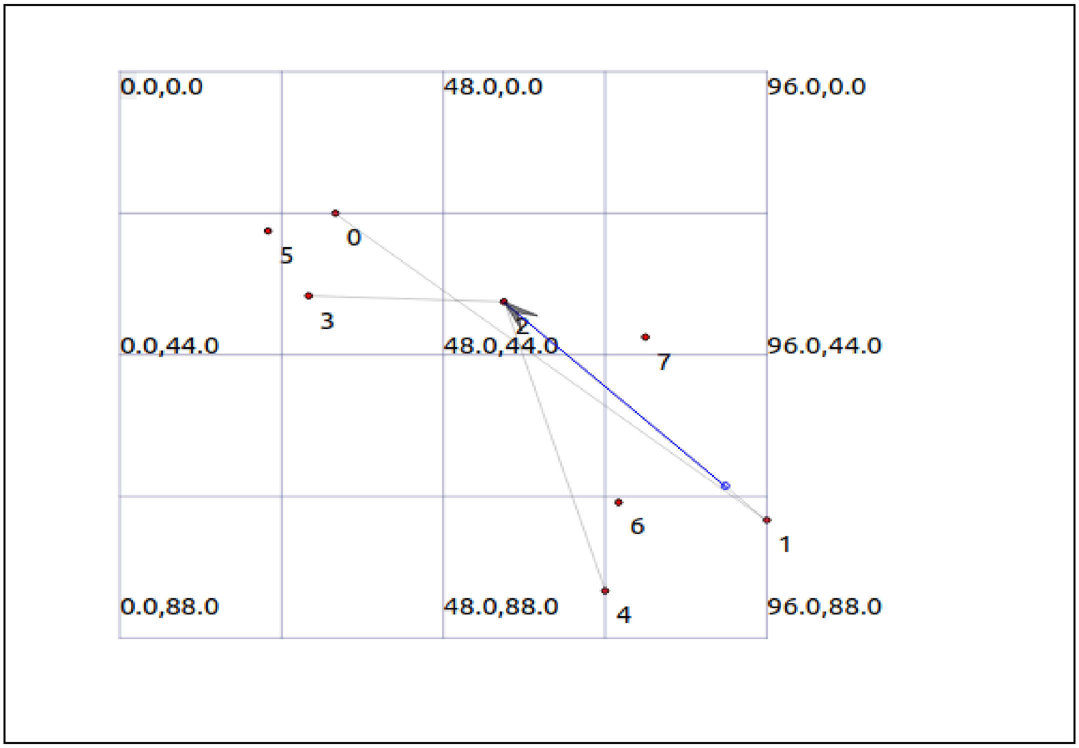
Output:

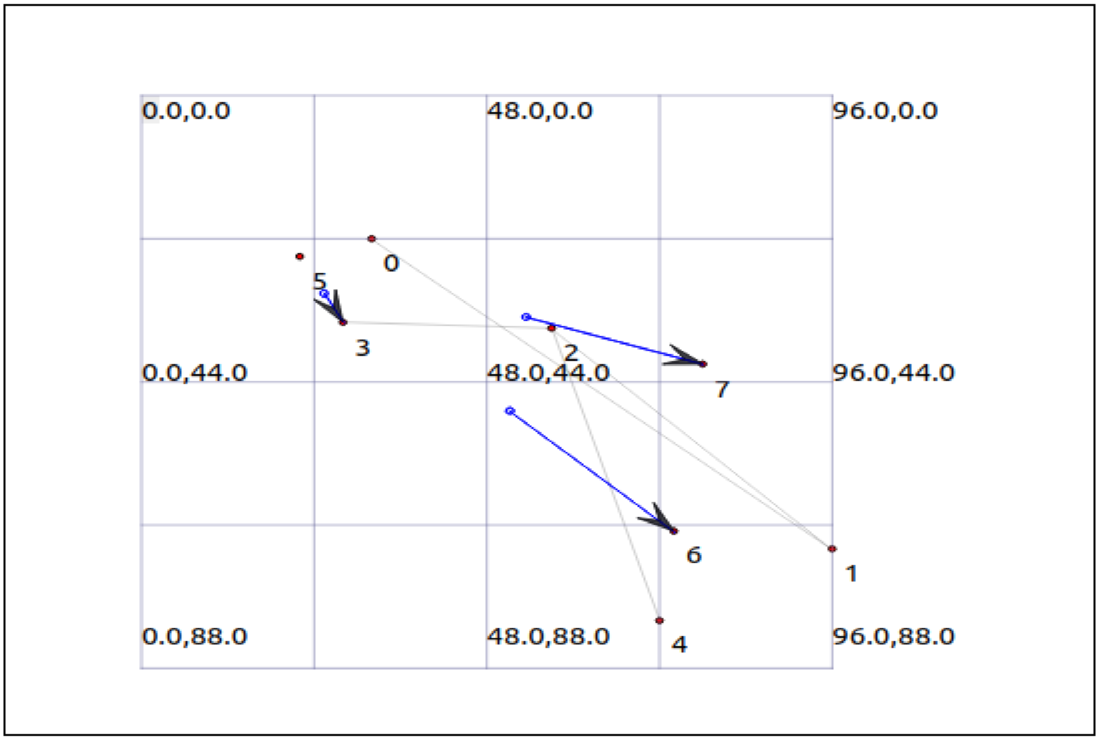


```
vaish@vaish-VirtualBox:~/Workspace/ns-allinone-3.32/ns-3.32$ ./waf --run prac8
Waf: Entering directory `/home/vaish/Workspace/ns-allinone-3.32/ns-3.32/build'
Waf: Leaving directory `/home/vaish/Workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.454s)
AnimationInterface WARNING:Node:0 Does not have a mobility model. Use SetConstantPosition if it i
s stationary
AnimationInterface WARNING:Node:1 Does not have a mobility model. Use SetConstantPosition if it i
s stationary
AnimationInterface WARNING:Node:2 Does not have a mobility model. Use SetConstantPosition if it i
s stationary
AnimationInterface WARNING:Node:3 Does not have a mobility model. Use SetConstantPosition if it i
s stationary
AnimationInterface WARNING:Node:4 Does not have a mobility model. Use SetConstantPosition if it i
s stationary
AnimationInterface WARNING:Node:5 Does not have a mobility model. Use SetConstantPosition if it i
s stationary
AnimationInterface WARNING:Node:6 Does not have a mobility model. Use SetConstantPosition if it i
s stationary
AnimationInterface WARNING:Node:7 Does not have a mobility model. Use SetConstantPosition if it i
s stationary
AnimationInterface WARNING:Node:0 Does not have a mobility model. Use SetConstantPosition if it i
s stationary
AnimationInterface WARNING:Node:1 Does not have a mobility model. Use SetConstantPosition if it i
s stationary
AnimationInterface WARNING:Node:2 Does not have a mobility model. Use SetConstantPosition if it i
s stationary
```



Zoom: 0.798 Speed: 1.000 Time: 0.000000 s





Conclusions:

We have successfully implemented a simple network using NetAnim in Network Simulator in this practical.