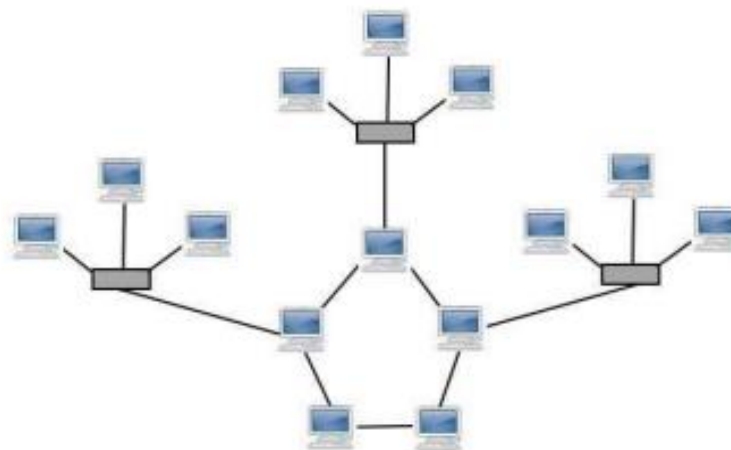


**AIM: ANIMATE A SIMPLE NETWORK USING NETANIM IN NETWORK SIMULATOR.**

**THEORY:**

**Understanding Hybrid Topology (Wireless Network Topology):**

A hybrid topology is a kind of network topology that is a combination of two or more network topologies, such as mesh topology, bus topology, and ring topology. Its usage and choice are dependent on its deployments and requirements like the performance of the desired network, and the number of computers, their location. The below figure is describing the structure of hybrid topology that contains more than one topology.



However, a variety of technologies are needed for its physical implementation, and it offers a complex structure. Also, it includes an advantage as increasing flexibility; it can increase fault tolerance, and allows new basic topologies to be added or removed easily. The hybrid topology is more useful when you need to fulfil diversity in Computer Network. In this topology, all network sections can include the configuration of different Network Topology. For instance, you can have a Hybrid network made by two different networks Star Backbone and the Ring Network. You can also use the Star Mesh Hybrid Topology in which if the main backbone gets fail, the entire network will destroy.

**SOURCE CODE:**

**1<sup>st</sup>:** search for star.cc which is located in cd workspace/ns-allinone-3.33/ns-3.33/scratch/bus

**2<sup>nd</sup>:** If its not there then create one using cp examples/tutorial/third.cc scratch/Hybrid.cc and paste the following code:

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

// Default Network Topology
//
// Wifi 10.1.3.0
//      AP
// * * * *
// | | | | 10.1.1.0
// 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 nCsmma = 3;
    uint32_t nWifi = 3;
    bool tracing = false;

    CommandLine cmd (__FILE__);
    cmd.AddValue ("nCsmma", "Number of \"extra\" CSMA nodes/devices", nCsmma);
    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);

    // The underlying restriction of 18 is due to the grid position
    // allocator's configuration; the grid layout will exceed the
    // bounding box if more than 18 nodes are provided.
    if (nWifi > 18)
    {
        std::cout << "nWifi should be 18 or less; otherwise grid layout exceeds the bounding box" <<
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;
phy.SetChannel (channel.Create ());

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

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 ();

AnimationInterface anim ("Hybrid.xml");
anim.SetConstantPosition (p2pNodes.Get(0),10,40);
```

```
anim.SetConstantPosition (p2pNodes.Get(1),30,40);  
anim.SetConstantPosition (csmaNodes.Get(1),60,40);  
anim.SetConstantPosition (csmaNodes.Get(2),90,40);  
anim.SetConstantPosition (csmaNodes.Get(3),120,40);
```

```
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;  
}
```

3<sup>rd</sup> : `cd workspace/ns-allinone-3.33/ns-3.33`

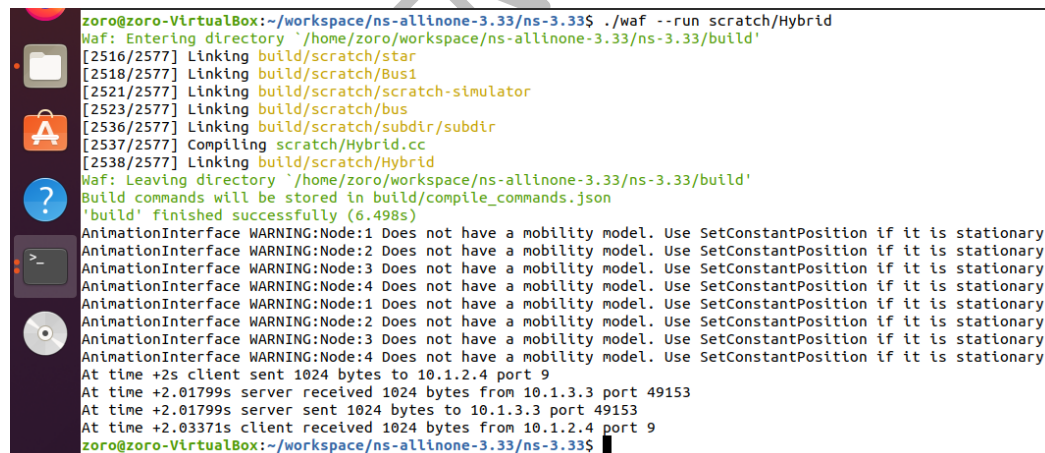
`./waf --run scratch/Hybrid`

4<sup>th</sup> : `cd`

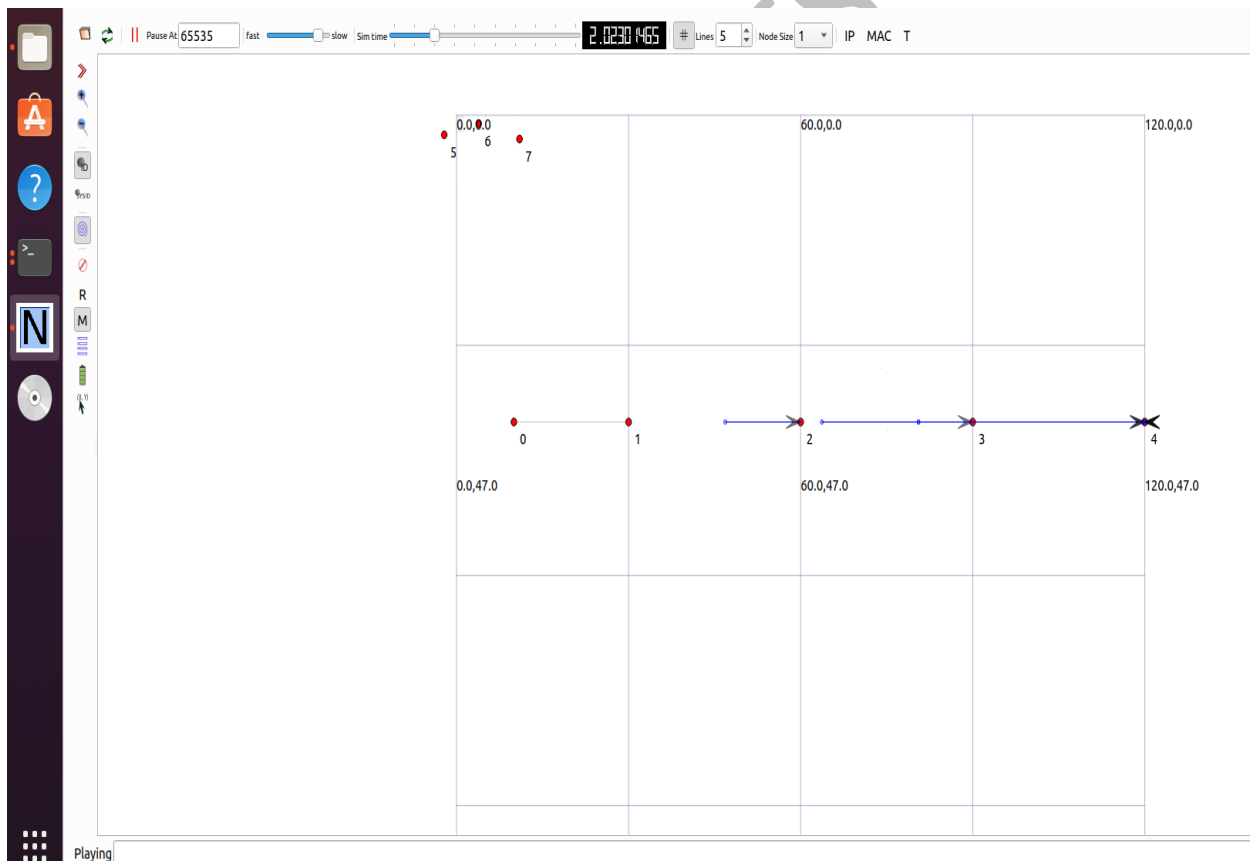
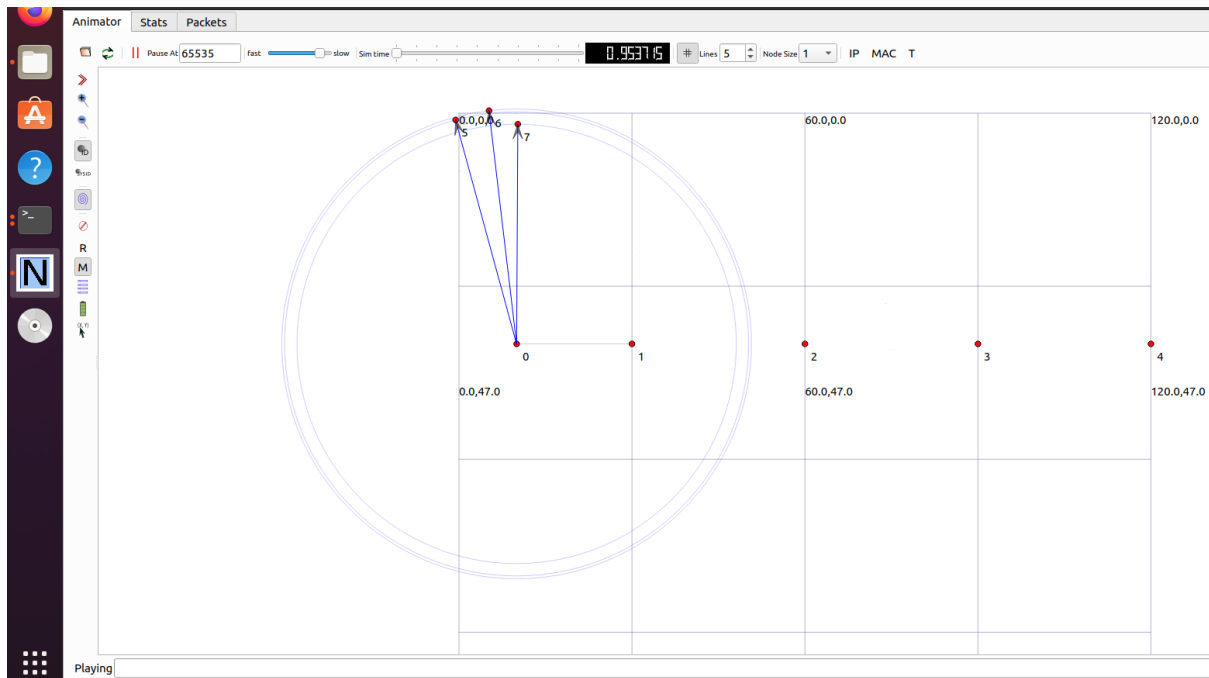
`cd workspace/ns-allinone-3.33/netanim-3.108`

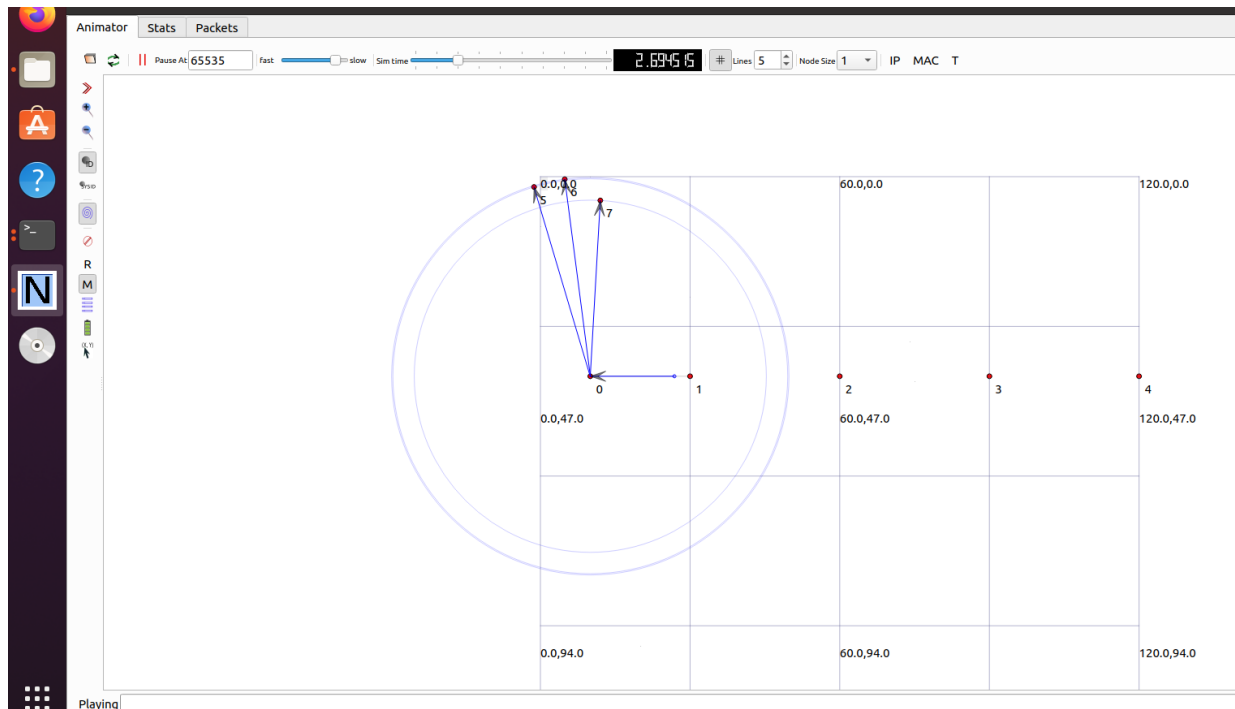
`./NetAnim`

#### OUTPUT:



```
zoro@zoro-VirtualBox:~/workspace/ns-allinone-3.33/ns-3.33$ ./waf --run scratch/Hybrid  
Waf: Entering directory `/home/zoro/workspace/ns-allinone-3.33/ns-3.33/build'  
[2516/2577] Linking build/scratch/star  
[2518/2577] Linking build/scratch/Bus1  
[2521/2577] Linking build/scratch/scratch-simulator  
[2523/2577] Linking build/scratch/bus  
[2536/2577] Linking build/scratch/subdir/subdir  
[2537/2577] Compiling scratch/Hybrid.cc  
[2538/2577] Linking build/scratch/Hybrid  
Waf: Leaving directory `/home/zoro/workspace/ns-allinone-3.33/ns-3.33/build'  
Build commands will be stored in build/compile_commands.json  
'build' finished successfully (6.498s)  
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: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  
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  
zoro@zoro-VirtualBox:~/workspace/ns-allinone-3.33/ns-3.33$
```





### CONCLUSION:

From this practical, I have learned how to implement a simple network (hybrid) using NetAnim in Network Simulator executed successfully.