Project Title: LTE simulation in NS-3

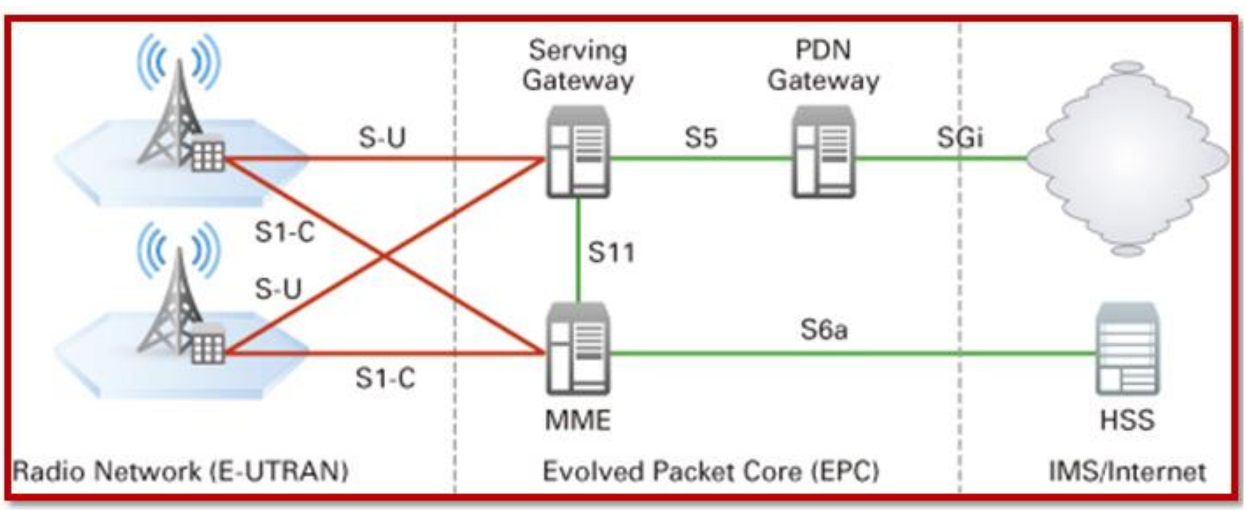
INTRODUCTION:

A basic LTE network is composed of a number of cells, wherein each cell has a base station node (eNodeB) and many User Equipment (UE) nodes. The simulation then proceeds to show us how two UE nodes communicate with each other, as well as with a base station node or an eNodeB. A simulation of the uplink communication shows us how the UE nodes send messages/information to the base station, while a simulation of the downlink communication illustrates how the base stations transmit information to the corresponding UE nodes that fall under their jurisdiction. The project also simulates a handover scenario wherein if a UE node happens to get a stronger signal from a base station other than the one by which it was being served previously, then it successfully gets 'handed over' to the base station from which it is receiving a stronger signal at present.

OBJECTIVE/PURPOSE:

NS3 LTE Simulation is also the latest high-speed cellular transmission network. LTE is a 4G technology also with download speeds that run the gamut from 3 to 28 Mbps worldwide.4G LTE is also one of several competing 4G standards along with Ultra Mobile Broadband (UMB) andWiMax (IEEE 802.16).Ns3 is also the best choice among network simulator for simulating LTE framework.

Architecture of NS3 LTE Simulation:



LTE parameters:

- Transmission bandwidth.
- Mobility.
- Frequency range.
- Duplexing.
- Channel bandwidth.
- Channel coding.
- MIMO.
- Multi-antenna technology.

Long Term Evolution (LTE) technology is very famous nowadays. This is because of the increasing usage of 4G smartphones that use this technology. LTE is the project name given to development of a high-performance air interface for cellular mobile communication systems. It is used for advancing series of mobile telecommunications systems. It is a high-speed data network. It is the fastest wireless network for smartphones and mobile devices.

Advantages:

1. Data as well as voice can be exchanged between participants. This is because LTE supports packet switching.
2. High amounts of data can be transferred between the sender and the receiver.
3. All data exchange occurs with very less power consumption. This leads to better life of the smartphone batteries.
4. It has high speed of the file upload and download.
5. It releases the network usage faster. This decreases the load on the network.
6. This decreases the traffic and moves towards lesser crashes in the service.
7. Live shows, matches and events can be easily watched using LTE.

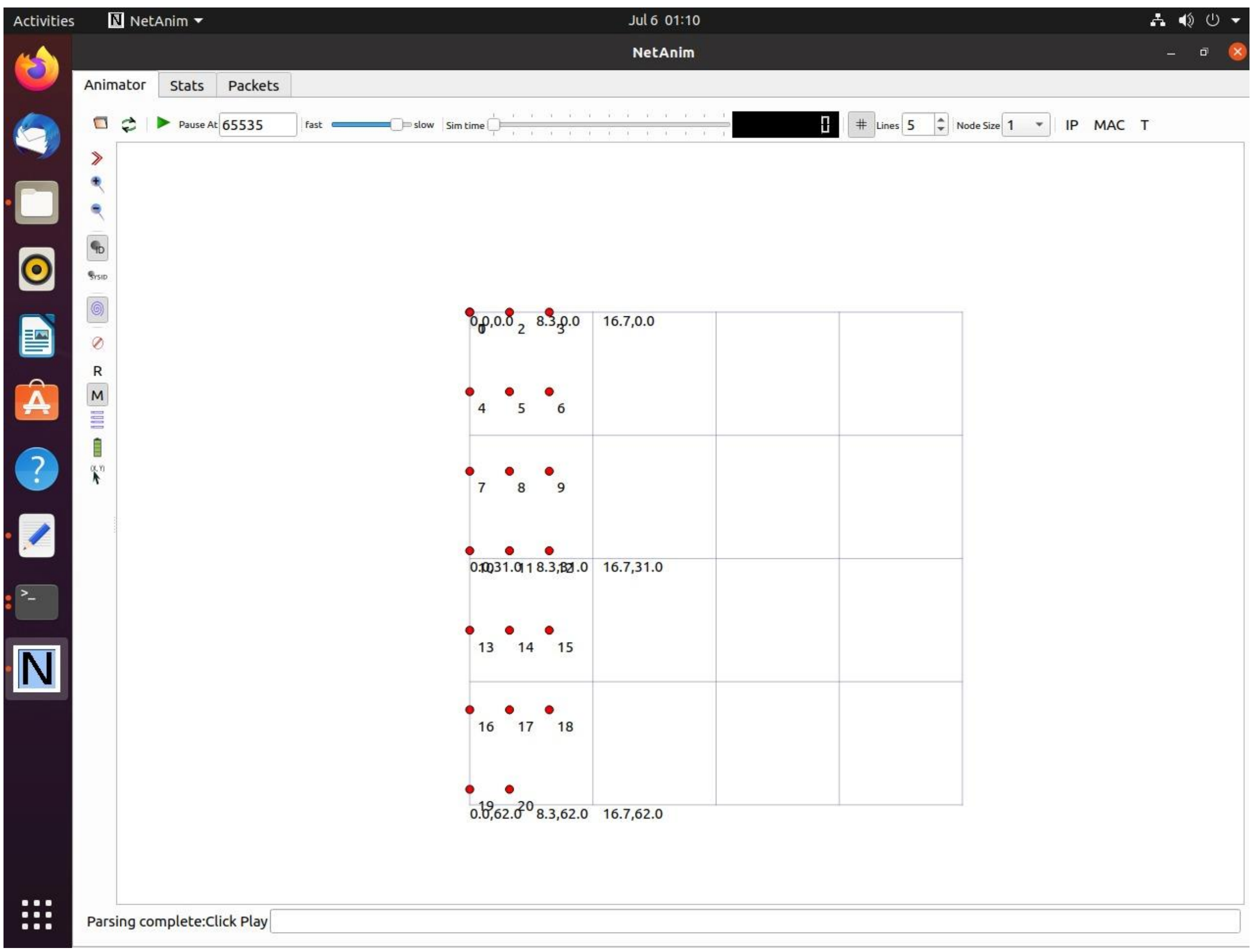
Disadvantages:

1. This service is not currently available in all cities.
2. More towers and fresh technologies need to be developed for better signals while in transit e.g., in buses and trains.
3. LTE being complex needs only skilled people to manage the system. They even need to be paid a higher salary.
4. This technology cannot be used in old versions of smartphones.
5. Buying new smartphones for LTE is a costly affair.

SYSTEM REQUIREMENTS:

- Ubuntu Linux,
- NS3,
- NetAnim,
- Wireshark
- Python
- Ram: 8 GB
- I3 Processor

NETWORK DESIGN:



STEPS FOR BUILDING NETWORK:

1) The first step in setting up a basic LTE network is to create a number of node objects and then provide them with the methods to add eNodeBs and UEs before finally configuring them.
2) Now that we have configured a set of nodes as eNodeBs and another set of nodes as UEs, we proceed to install an LTE protocol stack on top of them, as they are just empty nodes as of now. We then proceed to configure the mobility model for all the nodes.
3) The positions of the nodes can be set by using the 'SetPositionAllocator' function of the mobility module. Hardcoding the positions of all the nodes can be a tedious task since we have many nodes in our case. Hence, we assigned locations to the nodes in such a way that they formed a grid inside the cell by declaring the number of required rows and columns in the grid, and by declaring the gap between any two consecutive nodes.
4) The next step is to attach all the UE nodes to one of the eNodeBs. In this way, we can configure each UE node in line with the configuration of the eNodeB. Consequently, a Radio Resource Control (RRC) connection is established between each UE node and its corresponding eNodeB. This means that we have all are channels firmly in place so that we can start the required communications between the UEs and the eNodeBs.

SOURCE CODE:

```
#include <ns3/core-module.h>
#include <ns3/network-module.h>
#include <ns3/mobility-module.h>
#include <ns3/lte-module.h>
#include "ns3/netanim-module.h"
#include "ns3/flow-monitor-module.h"
using namespace ns3;

int main (int argc, char *argv[])
{
 CommandLine cmd;

 Ptr<LteHelper> lteHelper = CreateObject<LteHelper> ();
 //This will instantiate some common objects (e.g., the Channel object) and provide the
methods to add eNBs and UEs and configure them.
```

```
Config::SetDefault ("ns3::UdpClient::Interval", TimeValue (MilliSeconds (1000)));
 Config::SetDefault ("ns3::UdpClient::MaxPackets", UintegerValue (1000000));
 Config::SetDefault ("ns3::LteHelper::UseIdealRrc", BooleanValue (false));
//Create Node objects for the eNB(s) and the UEs:
 NodeContainer enbNodes;
 enbNodes.Create (1);
 NodeContainer ueNodes;
 ueNodes.Create (20);

 //Note that the above Node instances at this point still don't have an LTE protocol stack
installed; they're just empty nodes.

 //Configure the Mobility model for all the nodes:

 MobilityHelper mobility;
//---------Set Mobility---------------------------------
 mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
 mobility.Install (enbNodes);
 //mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
 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 (-2000, 2000, -2000, 2000)));
 mobility.Install (ueNodes);

 //Install an LTE protocol stack on the eNB(s):

 NetDeviceContainer enbDevs;
 enbDevs = lteHelper->InstallEnbDevice (enbNodes);

 //Install an LTE protocol stack on the UEs:

 NetDeviceContainer ueDevs;
 ueDevs = lteHelper->InstallUeDevice (ueNodes);
```

```
//Attach the UEs to an eNB. This will configure each UE according to the eNB
configuration, and create an RRC connection between them:

lteHelper->Attach (ueDevs, enbDevs.Get (0));

//Activate a data radio bearer between each UE and the eNB it is attached to:

enum EpsBearer::Qci q = EpsBearer::GBR_CONV_VOICE;
EpsBearer bearer (q);
lteHelper->ActivateDataRadioBearer (ueDevs, bearer);

//this method will also activate two saturation traffic generators for that bearer, one in
uplink and one in downlink.
//Set the stop time
Simulator::Stop (Seconds (20));

///This is needed otherwise the simulation will last forever, because (among others) the
start-of-subframe event is scheduled repeatedly, and the ns-3 simulator scheduler will
hence never run out of events.
  lteHelper->EnablePhyTraces ();
  lteHelper->EnableMacTraces ();
  lteHelper->EnableRlcTraces ();

//   Ptr<RadioBearerStatsCalculator> rlcStats = lteHelper->GetRlcStats ();
// rlcStats->SetAttribute ("EpochDuration", TimeValue (Seconds (0.05)));
// Ptr<RadioBearerStatsCalculator> pdcpStats = lteHelper->GetPdcpStats ();
// pdcpStats->SetAttribute ("EpochDuration", TimeValue (Seconds (0.05)));

//Run the simulation:
  AnimationInterface anim ("lte1.xml");
  anim.SetMobilityPollInterval(Seconds(1.00));
  anim.SetMaxPktsPerTraceFile (100000000000);
// anim.EnablePacketMetadata(true);
Simulator::Run ();
Simulator::Destroy ();
return 0;
}
```

OUTPUT:

Commands to write in terminal

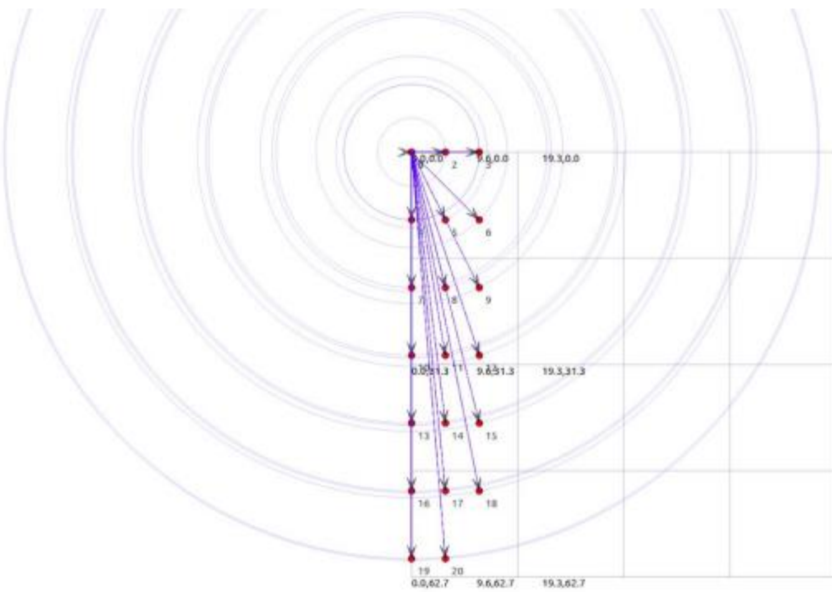
- ./waf --run scratch/lte

To run NetAnim simulation of the lte

Change directory to netanim-3.108 folder, open terminal and write

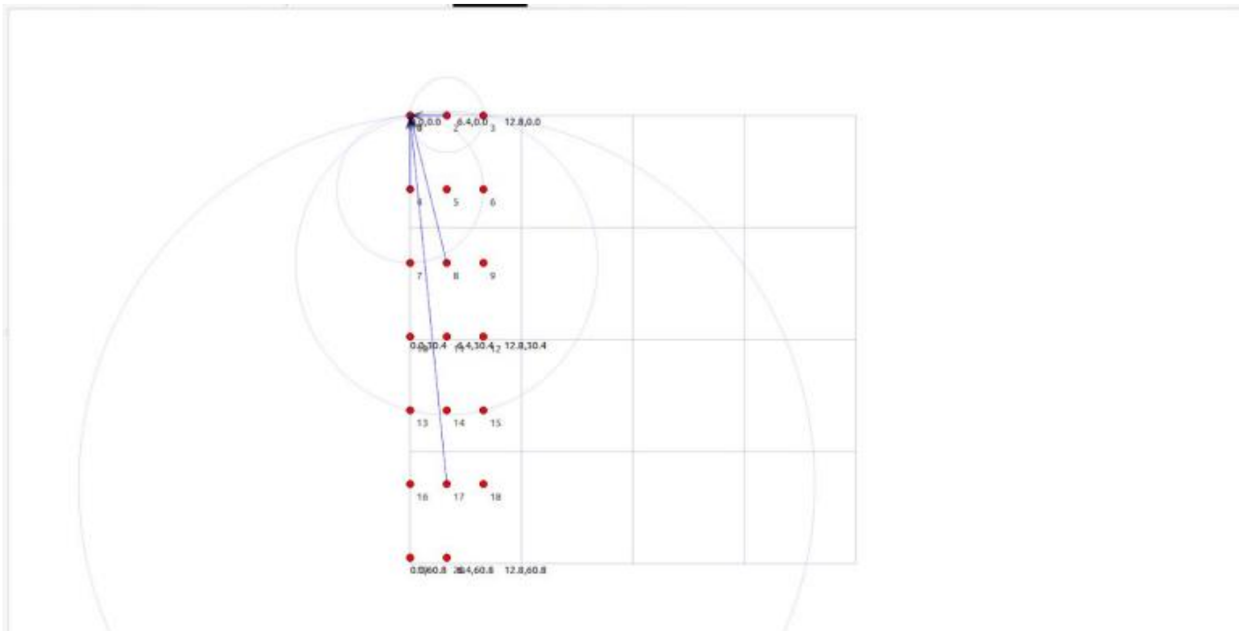
- **./NetAnim**

1) DOWNLINK COMMUNICATION



The above figure illustrates the downlink communication of the LTE network wherein the base station sends messages/information to all its corresponding UE nodes. The arrow marks pointing down indicate the direction of transmission of the messages which confirms that this is a case of downlink communication. The traffic density in a given region can be observed by virtue of the concentric circles, which we obtained by activating the data radio bearers.

2) UPLINK COMMUNICATION



The above figure illustrates the uplink communication of the LTE network wherein the UE nodes send messages/information to their corresponding base station. The arrow marks pointing upwards in the direction of the base station indicate the direction of transmission of the messages which confirms that this is a case of uplink communication. The traffic density in a given region can be observed by virtue of the concentric circles, which we obtained by activating the data radio bearers.

CONCLUSION:

In this way the simulation was performed successfully and LTE was demonstrated.

REFERENCE:

- https://github.com/RahulMotipalle/LTE-Simulation-in-NS3/blob/master/lte_basic.cc
- https://www.nsnam.org/docs/models/html/lte.html
- https://www.researchgate.net/publication/228998229_Handover_Scenario_and_Procedure_in_LTE-based_Femtocell_Networks