ECGR 6120/8120 Wireless Communications and Networking

Final Project-LTE Network Simulation using Network Simulator-3

Avadhut Naik – 801045233

Dushyant Singh Tomar – 801043306

I. OBJECTIVES

Simulation of LTE Network by constructing topology and implementation of Fading, Pathloss, AMC with BER, Frequency Reuse, MAC Schedulers and X-2 based Handover Algorithms. Also, comparing performance metrics of three MAC Schedulers which are Proportional Fair MAC Scheduler, Time domain maximize throughput MAC Scheduler and Time Domain Blind Equal Throughput scheduler through Flow Monitor.

II. NETWORK DISCRIPTION

The network consists of three EnodeB nodes and 9 User Equipment (UE) nodes. The network also supports Evolved Packet Core which is connected to 1 Remote Host. Trace Fading Loss Model and Friis Propagation Pathloss model \ has been utilized in the network to simulate the effects of fading and pathloss on LTE network. Utilizing isotropic antennas to transmit radiations equally in all directions, EnodeB nodes are also connected to each other through X2 interfaces. The Default MAC Scheduler utilized for the network is Time domain maximize throughput MAC Scheduler (TDMTMAC Scheduler).

A. EnodeB Nodes and User Equipment Nodes:

In the network, EnodeB 0 node (Node 2) supports 1 User Equipment, EnodeB 1 (Node 3) supports 5 User Equipment while EnodeB 2 (Node 4) supports 3 User Equipment. The connections of UE nodes to ENB Nodes has been undertaken in a format to demonstrate handover mechanism distinctly.

B. Evolved Packet Core (EPC) Network:

Consisting of mobility management entity (MME) along with Home Subscriber Server, the EPC node (Node 0) is the Serving and Packet Data Network Gateway.

C. *Remote Host:*

Connecting to the network through the EPC node to which connection is established through a point to point channel, the UE's can communicate with remote host and vice-a-versa through the same.

D. *Adaptive Modulation Coding (AMC):*

The network also implements Piro AMC Model with the BER of 0.0001.

E. *Pathloss Model:*

The network implements Friis Propagation Pathloss Model in which the received Power at a nodoe is given by the following equation

$$P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi d)^2}$$

F. *Fading Model:*

With the network utilizing Trace Loss fading model the parameters of the same can be changed through the source files in NS3's Fading Traces folder. Selecting a different file as a parameter can also be undertaken to demonstrate effects of fading on LTE network.

G. *Flow Monitor:*

To monitor the throughput of the network under various MAC Schedulers and Frequency Reuse algorithms, Flow Monitor has been implemented to monitor their effects on throughout of every link of the network.

III. NETWORK DESIGN (NETANIM)

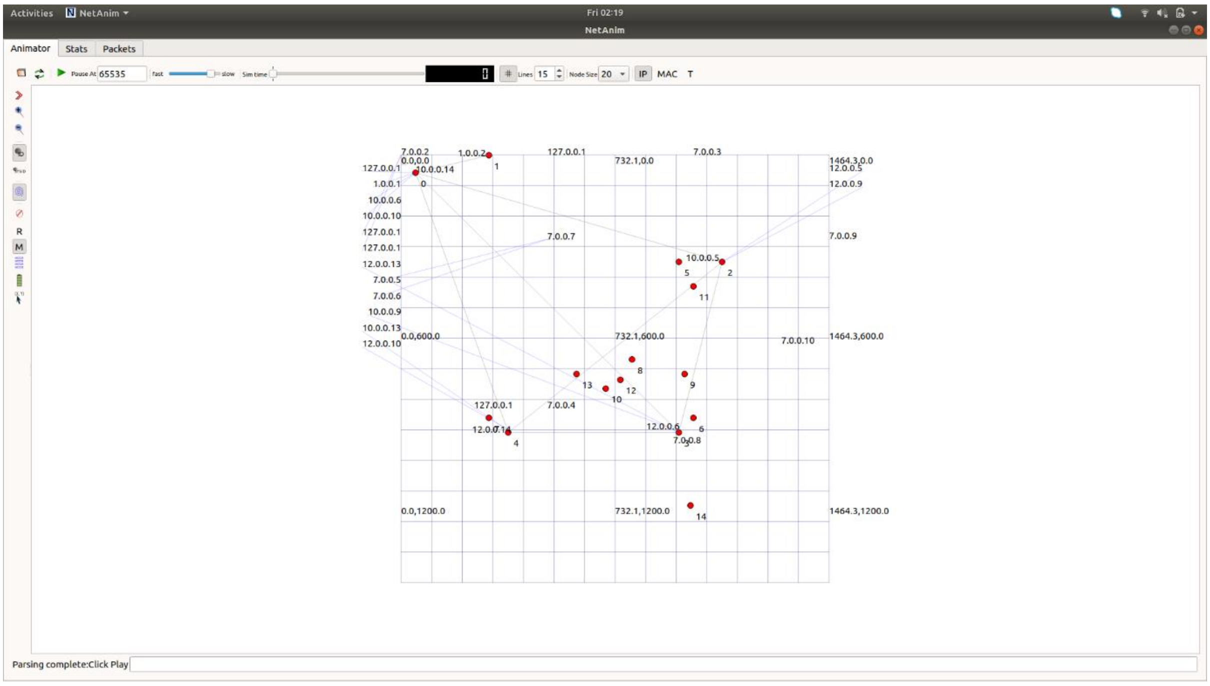


Figure 1: Network Design

Node 0 in Figure 1 above is the EPC Node while Node 1 is the Remote Host. Nodes 2, 3 and 4 are EnodeB nodes serving the User Equipment node which number from 5 to 13. Node 14 is the Spectrum Analyser Node.

IV. STARTING THE NETWORK

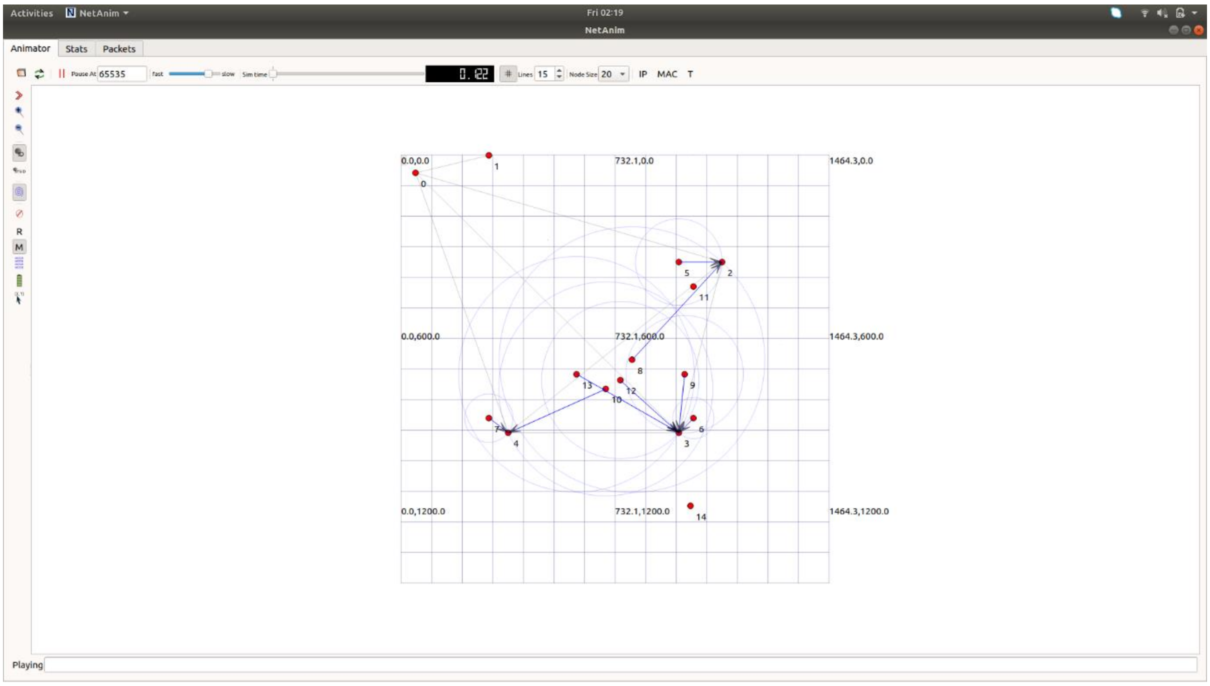


Figure 2: Network Started (Netanim Simulation)

Building i.e. compiling the program for the network through the *"waf"* tool of NS3, the network starts, as demonstrated in figures 2, with User Equipment Nodes establishing connection with EnodeB nodes. The same can be observed through the Animation Interface implemented through Netanim.

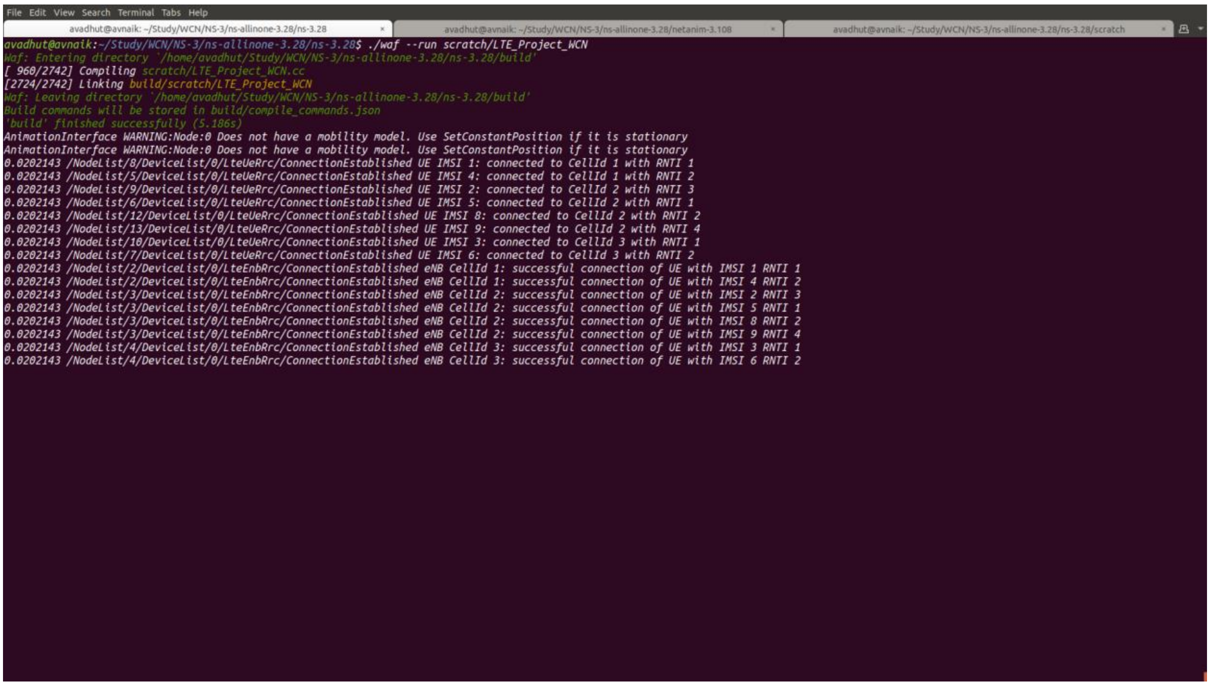
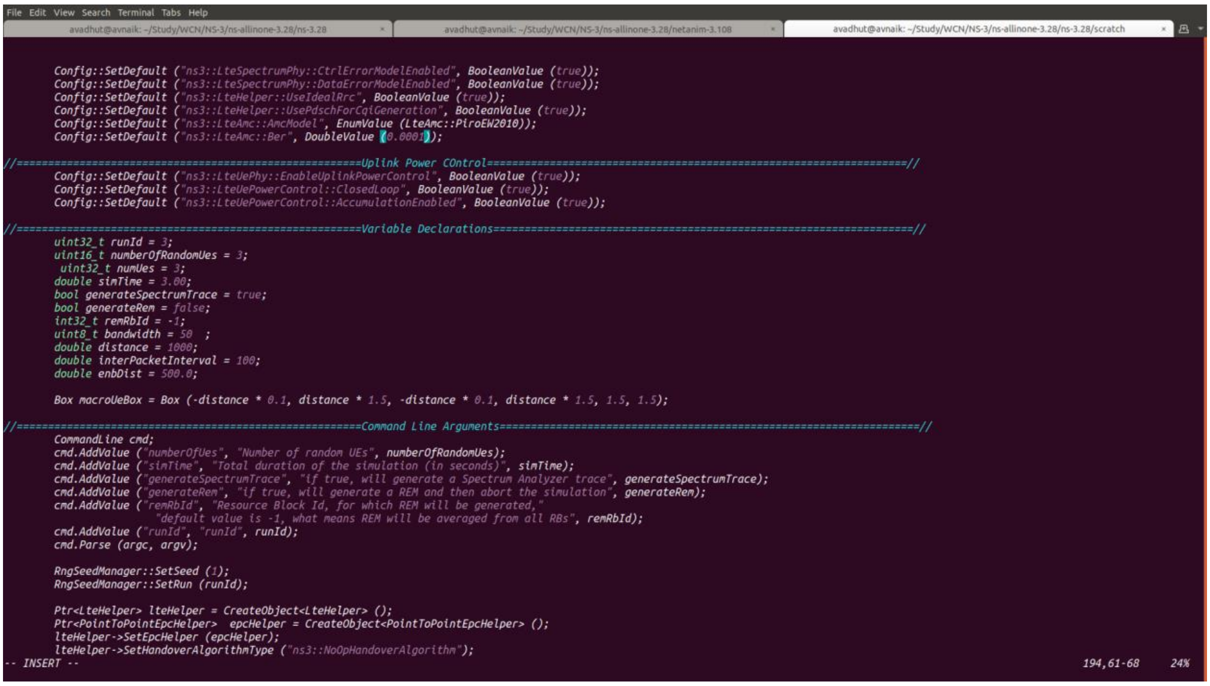


Figure 3: Network Connections Established (Terminal Output).

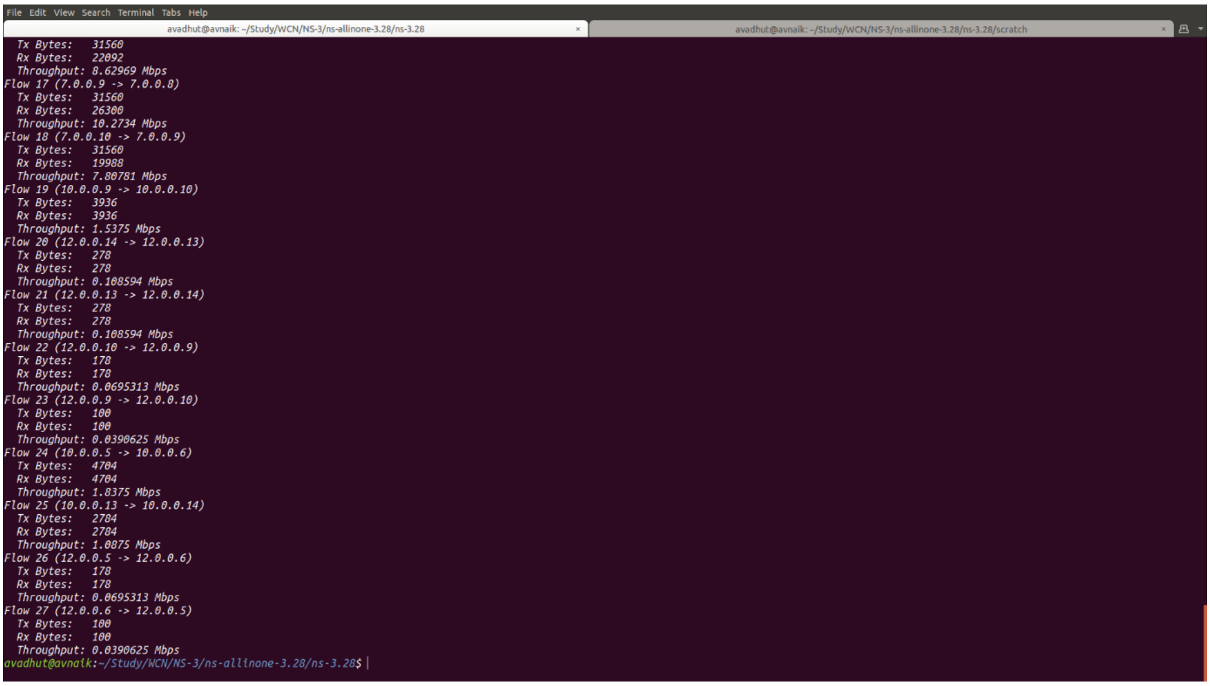
V. ADAPTIVE MODULATION AND CODING

The network like LTE has Adaptive Modulation Coding i.e. Piro AMC Model implemented within so as to enable modulation with the scheme. The Bit Error Rate (BER) has been set to 0.0001 and can set any value desired to observe the appropriate changes.



```
      Config::SetDefault ("ns3::LteSpectrumPhy::CtrlErrorModelEnabled", BooleanValue (true));
      Config::SetDefault ("ns3::LteSpectrumPhy::DataErrorModelEnabled", BooleanValue (true));
      Config::SetDefault ("ns3::LteHelper::UseIdealRrc", BooleanValue (true));
      Config::SetDefault ("ns3::LteHelper::UsePdschForCqiGeneration", BooleanValue (true));
      Config::SetDefault ("ns3::LteAmc::AmcModel", EnumValue (LteAmc::PiroEW2010));
      Config::SetDefault ("ns3::LteAmc::Ber", DoubleValue (0.0001));

//=====================================================Uplink Power COntrol=====================================================//
      Config::SetDefault ("ns3::LteUePhy::EnableUplinkPowerControl", BooleanValue (true));
      Config::SetDefault ("ns3::LteUePowerControl::ClosedLoop", BooleanValue (true));
      Config::SetDefault ("ns3::LteUePowerControl::AccumulationEnabled", BooleanValue (true));

//=====================================================Variable Declarations=====================================================//
      uint32_t runId = 3;
      uint16_t numberOfRandomUes = 3;
       uint32_t numUes = 3;
      double simTime = 3.00;
      bool generateSpectrumTrace = true;
      bool generateRem = false;
      int32_t remRbId = -1;
      uint8_t bandwidth = 50  ;
      double distance = 1000;
      double interPacketInterval = 100;
      double enbDist = 500.0;

      Box macroUeBox = Box (-distance * 0.1, distance * 1.5, -distance * 0.1, distance * 1.5, 1.5, 1.5);

//=====================================================Command Line Arguments=====================================================//
      CommandLine cmd;
      cmd.AddValue ("numberOfUes", "Number of random UEs", numberOfRandomUes);
      cmd.AddValue ("simTime", "Total duration of the simulation (in seconds)", simTime);
      cmd.AddValue ("generateSpectrumTrace", "if true, will generate a Spectrum Analyzer trace", generateSpectrumTrace);
      cmd.AddValue ("generateRem", "if true, will generate a REM and then abort the simulation", generateRem);
      cmd.AddValue ("remRbId", "Resource Block Id, for which REM will be generated,"
                    "default value is -1, what means REM will be averaged from all RBs", remRbId);
      cmd.AddValue ("runId", "runId", runId);
      cmd.Parse (argc, argv);

      RngSeedManager::SetSeed (1);
      RngSeedManager::SetRun (runId);

      Ptr<LteHelper> lteHelper = CreateObject<LteHelper> ();
      Ptr<PointToPointEpcHelper>  epcHelper = CreateObject<PointToPointEpcHelper> ();
      lteHelper->SetEpcHelper (epcHelper);
      lteHelper->SetHandoverAlgorithmType ("ns3::NoOpHandoverAlgorithm");
-- INSERT --                                                                              194,61-68     24%
```

Figure 4: AMC Implemented with BER set to 0.0001



```
  Tx Bytes:   31560
  Rx Bytes:   22092
  Throughput: 8.62969 Mbps
Flow 17 (7.0.0.9 -> 7.0.0.8)
  Tx Bytes:   31560
  Rx Bytes:   26300
  Throughput: 10.2734 Mbps
Flow 18 (7.0.0.10 -> 7.0.0.9)
  Tx Bytes:   31560
  Rx Bytes:   19988
  Throughput: 7.80781 Mbps
Flow 19 (10.0.0.9 -> 10.0.0.10)
  Tx Bytes:   3936
  Rx Bytes:   3936
  Throughput: 1.5375 Mbps
Flow 20 (12.0.0.14 -> 12.0.0.13)
  Tx Bytes:   278
  Rx Bytes:   278
  Throughput: 0.108594 Mbps
Flow 21 (12.0.0.13 -> 12.0.0.14)
  Tx Bytes:   278
  Rx Bytes:   278
  Throughput: 0.108594 Mbps
Flow 22 (12.0.0.10 -> 12.0.0.9)
  Tx Bytes:   178
  Rx Bytes:   178
  Throughput: 0.0695313 Mbps
Flow 23 (12.0.0.9 -> 12.0.0.10)
  Tx Bytes:   100
  Rx Bytes:   100
  Throughput: 0.0390625 Mbps
Flow 24 (10.0.0.5 -> 10.0.0.6)
  Tx Bytes:   4704
  Rx Bytes:   4704
  Throughput: 1.8375 Mbps
Flow 25 (10.0.0.13 -> 10.0.0.14)
  Tx Bytes:   2784
  Rx Bytes:   2784
  Throughput: 1.0875 Mbps
Flow 26 (12.0.0.5 -> 12.0.0.6)
  Tx Bytes:   178
  Rx Bytes:   178
  Throughput: 0.0695313 Mbps
Flow 27 (12.0.0.6 -> 12.0.0.5)
  Tx Bytes:   100
  Rx Bytes:   100
  Throughput: 0.0390625 Mbps
avadhut@avnaik:~/Study/WCN/NS-3/ns-allinone-3.28/ns-3.28$
```

Figure 5: Throughput of the network with AMC implemented.

VI. FADING and PATHLOSS MODEL

In the network, Trace Fading Loss model and Friis Propagation Pathloss Model has been implemented to simulate the effects of fading on LTE network.

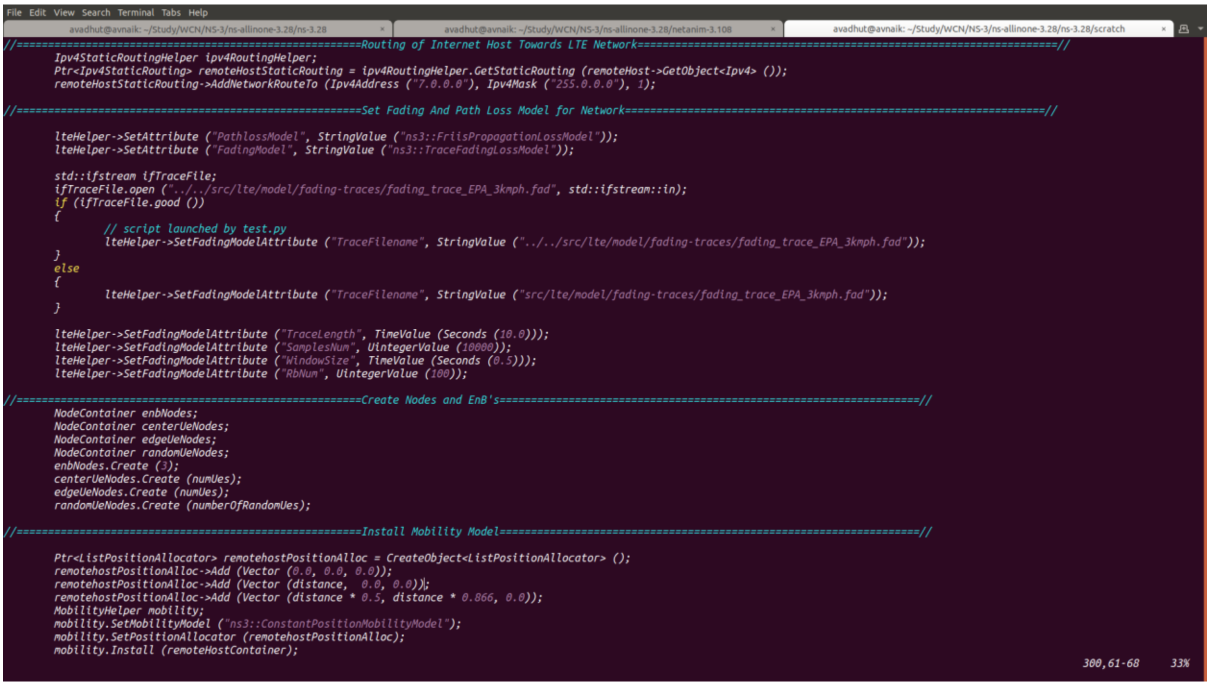


Figure 6: Fading and Pathloss Model

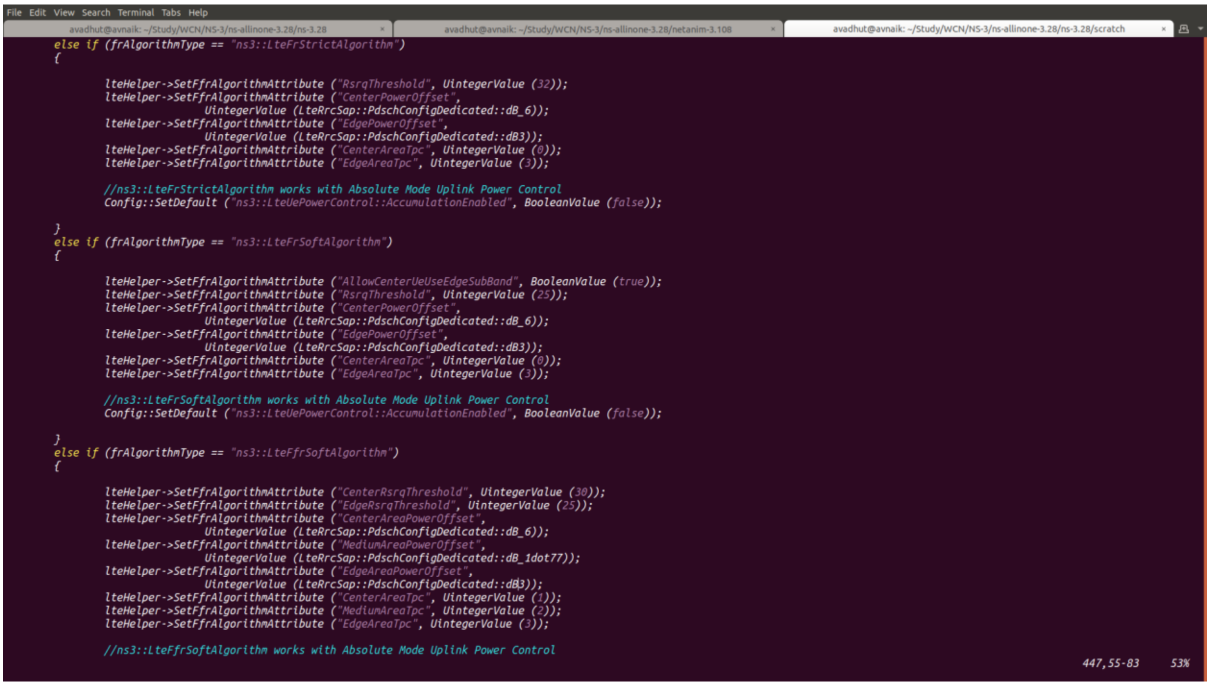
The parameters for the Fading model can be modified through source files of NS3's Fading Traces Folder. Selecting a different file as a parameter can also be undertaken (through the Trace File Open Command) to demonstrate different types and effects of fading on LTE network.

Similarly, for the pathloss model, other models can be implemented in the network by replacing the Friis Propagation loss model in the code by desired appropriate pathloss model which shall be instantiated by the LTE Helper.

VII. FREQUENCY REUSE

A. ALGORITHMS IMPLEMENTED:

Soft and Strict Algorithms of Frequency Reuse have been implemented in the network with Strict Algorithm being the default One.



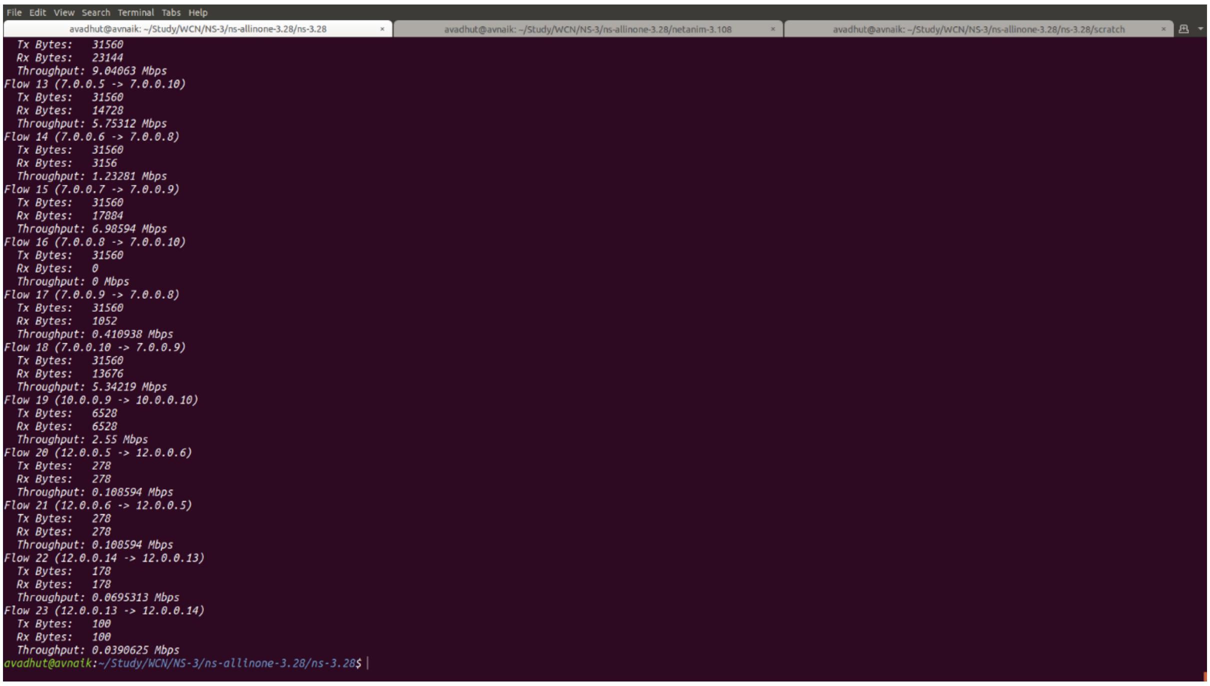
```
    else if (frAlgorithmType == "ns3::LteFrStrictAlgorithm")
    {

            lteHelper->SetFfrAlgorithmAttribute ("RsrqThreshold", UintegerValue (32));
            lteHelper->SetFfrAlgorithmAttribute ("CenterPowerOffset",
                            UintegerValue (LteRrcSap::PdschConfigDedicated::dB_6));
            lteHelper->SetFfrAlgorithmAttribute ("EdgePowerOffset",
                            UintegerValue (LteRrcSap::PdschConfigDedicated::dB3));
            lteHelper->SetFfrAlgorithmAttribute ("CenterAreaTpc", UintegerValue (0));
            lteHelper->SetFfrAlgorithmAttribute ("EdgeAreaTpc", UintegerValue (3));

            //ns3::LteFrStrictAlgorithm works with Absolute Mode Uplink Power Control
            Config::SetDefault ("ns3::LteUePowerControl::AccumulationEnabled", BooleanValue (false));

    }
    else if (frAlgorithmType == "ns3::LteFrSoftAlgorithm")
    {

            lteHelper->SetFfrAlgorithmAttribute ("AllowCenterUeUseEdgeSubBand", BooleanValue (true));
            lteHelper->SetFfrAlgorithmAttribute ("RsrqThreshold", UintegerValue (25));
            lteHelper->SetFfrAlgorithmAttribute ("CenterPowerOffset",
                            UintegerValue (LteRrcSap::PdschConfigDedicated::dB_6));
            lteHelper->SetFfrAlgorithmAttribute ("EdgePowerOffset",
                            UintegerValue (LteRrcSap::PdschConfigDedicated::dB3));
            lteHelper->SetFfrAlgorithmAttribute ("CenterAreaTpc", UintegerValue (0));
            lteHelper->SetFfrAlgorithmAttribute ("EdgeAreaTpc", UintegerValue (3));

            //ns3::LteFrSoftAlgorithm works with Absolute Mode Uplink Power Control
            Config::SetDefault ("ns3::LteUePowerControl::AccumulationEnabled", BooleanValue (false));

    }
    else if (frAlgorithmType == "ns3::LteFfrSoftAlgorithm")
    {

            lteHelper->SetFfrAlgorithmAttribute ("CenterRsrqThreshold", UintegerValue (30));
            lteHelper->SetFfrAlgorithmAttribute ("EdgeRsrqThreshold", UintegerValue (25));
            lteHelper->SetFfrAlgorithmAttribute ("CenterAreaPowerOffset",
                            UintegerValue (LteRrcSap::PdschConfigDedicated::dB_6));
            lteHelper->SetFfrAlgorithmAttribute ("MediumAreaPowerOffset",
                            UintegerValue (LteRrcSap::PdschConfigDedicated::dB_1dot77));
            lteHelper->SetFfrAlgorithmAttribute ("EdgeAreaPowerOffset",
                            UintegerValue (LteRrcSap::PdschConfigDedicated::dB3));
            lteHelper->SetFfrAlgorithmAttribute ("CenterAreaTpc", UintegerValue (1));
            lteHelper->SetFfrAlgorithmAttribute ("MediumAreaTpc", UintegerValue (2));
            lteHelper->SetFfrAlgorithmAttribute ("EdgeAreaTpc", UintegerValue (3));

            //ns3::LteFfrSoftAlgorithm works with Absolute Mode Uplink Power Control
```

Figure 7: Code for Frequency Reuse Algorithms

The switch from Strict to Soft Frequency Reuse algorithm can be undertaken by commenting header file of Strict Algorithm and uncommenting header file of Soft Algorithm.

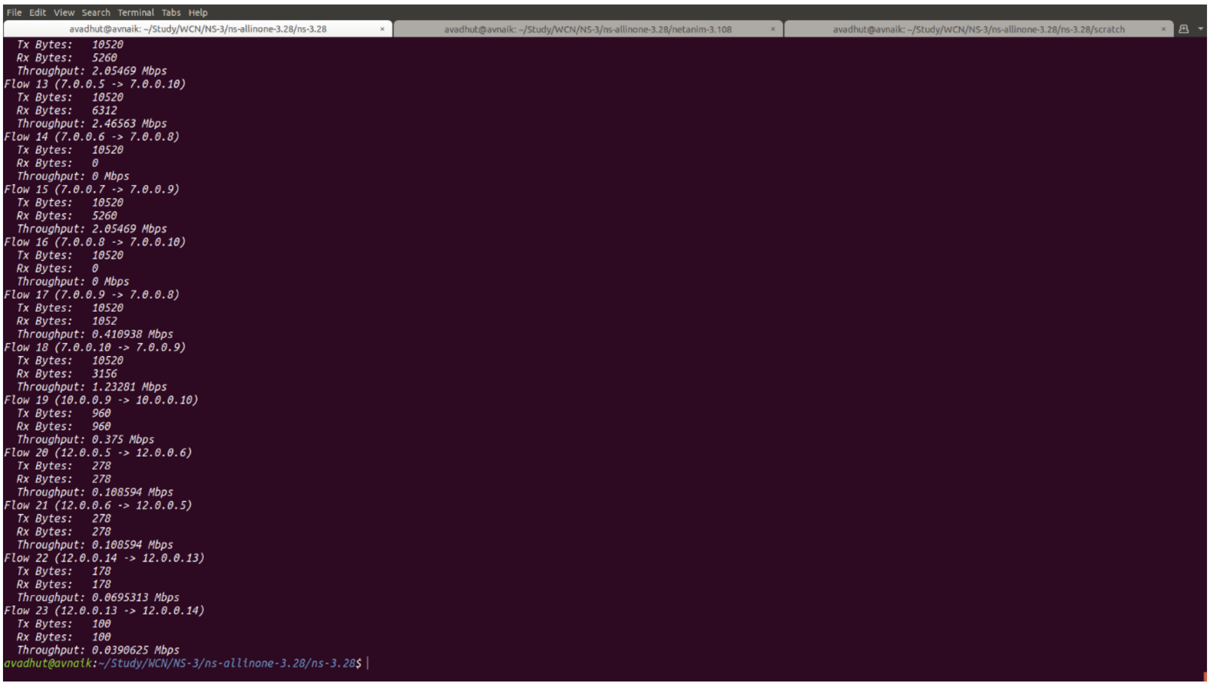
The network was run using both algorithms, one at a time to monitor their effects on throughput.

B. THROUGHPUT OF THE NETWORK:



```
File Edit View Search Terminal Tabs Help
avadhut@avnaik: ~/Study/WCN/NS-3/ns-allinone-3.28/ns-3.28 | avadhut@avnaik: ~/Study/WCN/NS-3/ns-allinone-3.28/netanim-3.108 | avadhut@avnaik: ~/Study/WCN/NS-3/ns-allinone-3.28/ns-3.28/scratch
  Tx Bytes:   31560
  Rx Bytes:   23144
  Throughput: 9.04063 Mbps
Flow 13 (7.0.0.5 -> 7.0.0.10)
  Tx Bytes:   31560
  Rx Bytes:   14728
  Throughput: 5.75312 Mbps
Flow 14 (7.0.0.6 -> 7.0.0.8)
  Tx Bytes:   31560
  Rx Bytes:   3156
  Throughput: 1.23281 Mbps
Flow 15 (7.0.0.7 -> 7.0.0.9)
  Tx Bytes:   31560
  Rx Bytes:   17884
  Throughput: 6.98594 Mbps
Flow 16 (7.0.0.8 -> 7.0.0.10)
  Tx Bytes:   31560
  Rx Bytes:   0
  Throughput: 0 Mbps
Flow 17 (7.0.0.9 -> 7.0.0.8)
  Tx Bytes:   31560
  Rx Bytes:   1052
  Throughput: 0.410938 Mbps
Flow 18 (7.0.0.10 -> 7.0.0.9)
  Tx Bytes:   31560
  Rx Bytes:   13676
  Throughput: 5.34219 Mbps
Flow 19 (10.0.0.9 -> 10.0.0.10)
  Tx Bytes:   6528
  Rx Bytes:   6528
  Throughput: 2.55 Mbps
Flow 20 (12.0.0.5 -> 12.0.0.6)
  Tx Bytes:   278
  Rx Bytes:   278
  Throughput: 0.108594 Mbps
Flow 21 (12.0.0.6 -> 12.0.0.5)
  Tx Bytes:   278
  Rx Bytes:   278
  Throughput: 0.108594 Mbps
Flow 22 (12.0.0.14 -> 12.0.0.13)
  Tx Bytes:   178
  Rx Bytes:   178
  Throughput: 0.0695313 Mbps
Flow 23 (12.0.0.13 -> 12.0.0.14)
  Tx Bytes:   100
  Rx Bytes:   100
  Throughput: 0.0390625 Mbps
avadhut@avnaik:~/Study/WCN/NS-3/ns-allinone-3.28/ns-3.28$
```

Figure 8: Throughput with Strict FR Algorithm.



```
File Edit View Search Terminal Tabs Help
avadhut@avnaik: ~/Study/WCN/NS-3/ns-allinone-3.28/ns-3.28 | avadhut@avnaik: ~/Study/WCN/NS-3/ns-allinone-3.28/netanim-3.108 | avadhut@avnaik: ~/Study/WCN/NS-3/ns-allinone-3.28/ns-3.28/scratch
  Tx Bytes:   10520
  Rx Bytes:   5260
  Throughput: 2.05469 Mbps
Flow 13 (7.0.0.5 -> 7.0.0.10)
  Tx Bytes:   10520
  Rx Bytes:   6312
  Throughput: 2.46563 Mbps
Flow 14 (7.0.0.6 -> 7.0.0.8)
  Tx Bytes:   10520
  Rx Bytes:   0
  Throughput: 0 Mbps
Flow 15 (7.0.0.7 -> 7.0.0.9)
  Tx Bytes:   10520
  Rx Bytes:   5260
  Throughput: 2.05469 Mbps
Flow 16 (7.0.0.8 -> 7.0.0.10)
  Tx Bytes:   10520
  Rx Bytes:   0
  Throughput: 0 Mbps
Flow 17 (7.0.0.9 -> 7.0.0.8)
  Tx Bytes:   10520
  Rx Bytes:   1052
  Throughput: 0.410938 Mbps
Flow 18 (7.0.0.10 -> 7.0.0.9)
  Tx Bytes:   10520
  Rx Bytes:   3156
  Throughput: 1.23281 Mbps
Flow 19 (10.0.0.9 -> 10.0.0.10)
  Tx Bytes:   960
  Rx Bytes:   960
  Throughput: 0.375 Mbps
Flow 20 (12.0.0.5 -> 12.0.0.6)
  Tx Bytes:   278
  Rx Bytes:   278
  Throughput: 0.108594 Mbps
Flow 21 (12.0.0.6 -> 12.0.0.5)
  Tx Bytes:   278
  Rx Bytes:   278
  Throughput: 0.108594 Mbps
Flow 22 (12.0.0.14 -> 12.0.0.13)
  Tx Bytes:   178
  Rx Bytes:   178
  Throughput: 0.0695313 Mbps
Flow 23 (12.0.0.13 -> 12.0.0.14)
  Tx Bytes:   100
  Rx Bytes:   100
  Throughput: 0.0390625 Mbps
avadhut@avnaik:~/Study/WCN/NS-3/ns-allinone-3.28/ns-3.28$
```

Figure 9: Throughput with Soft FR Algorithm.

VIII. MAC SCHEDULER

A. ALGORITHMS IMPLEMENTED:

Three MAC Schedulers have been implemented in the network with the default one being Time Domain Maximize Throughput MAC Scheduler.

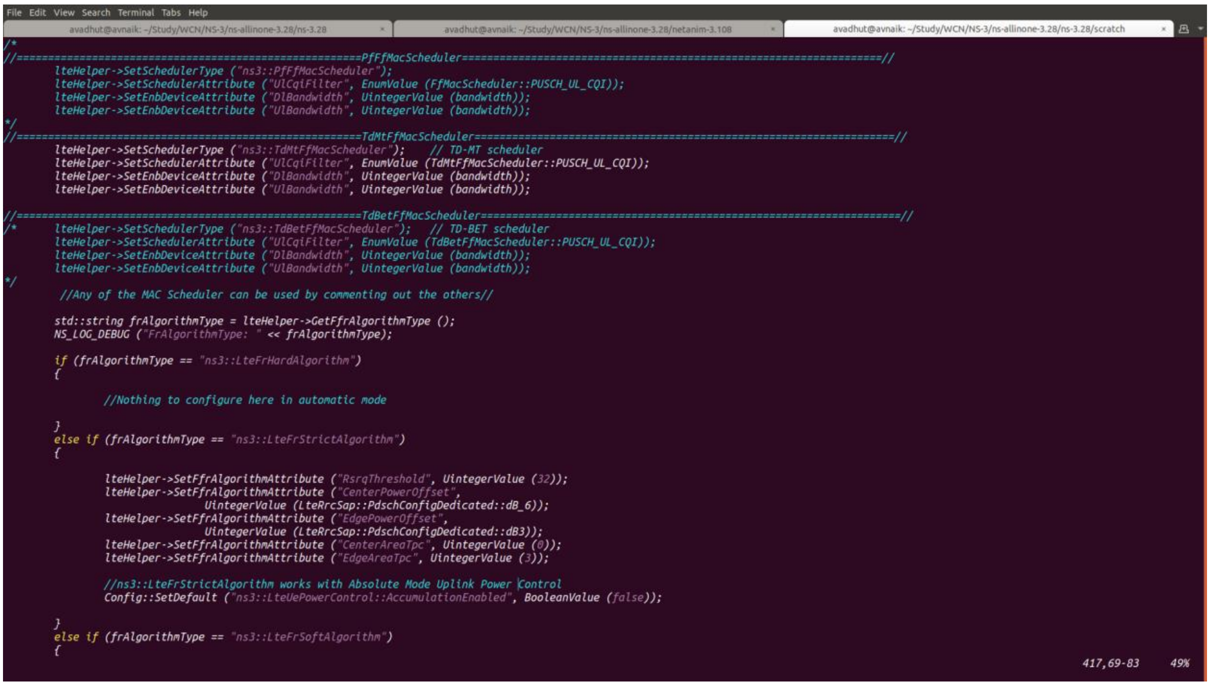


Figure 10: MAC Schedulers.

Any of the MAC Schedulers can be selected by commenting the other two schedulers. For testing, we have run the network using all of the implemented MAC Schedulers.

B. THROUGHPUT OF THE NETWORK:

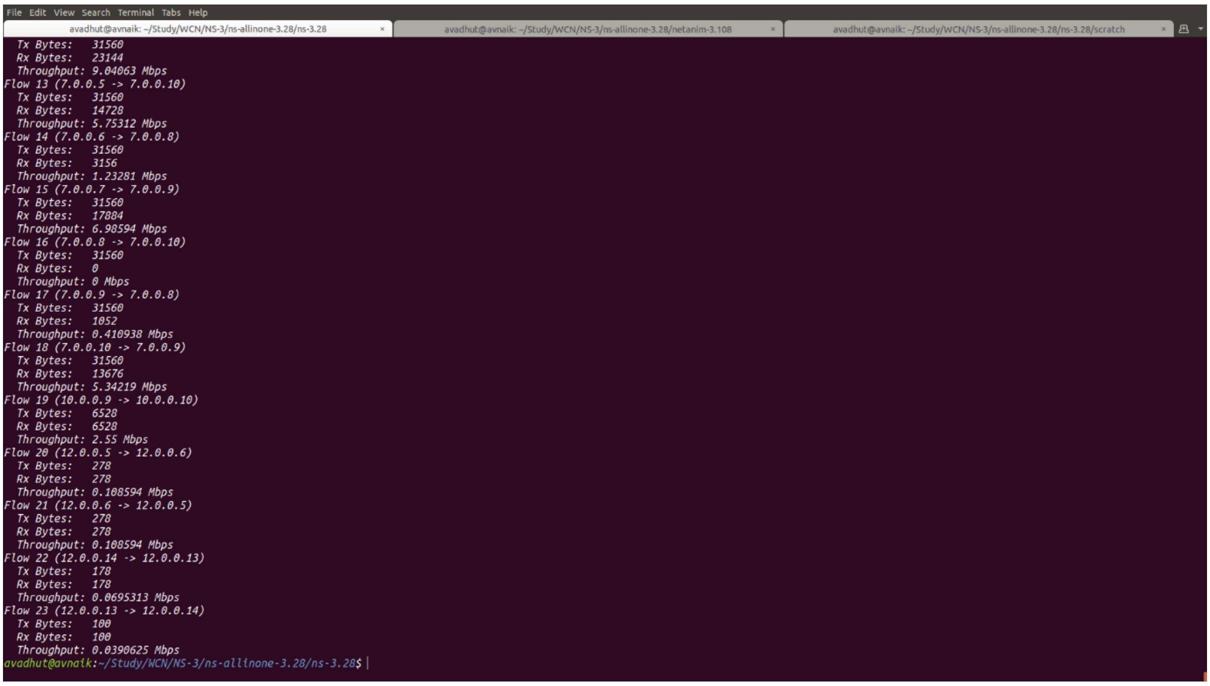


Figure 11: Throughput with TDMT MAC Scheduler.

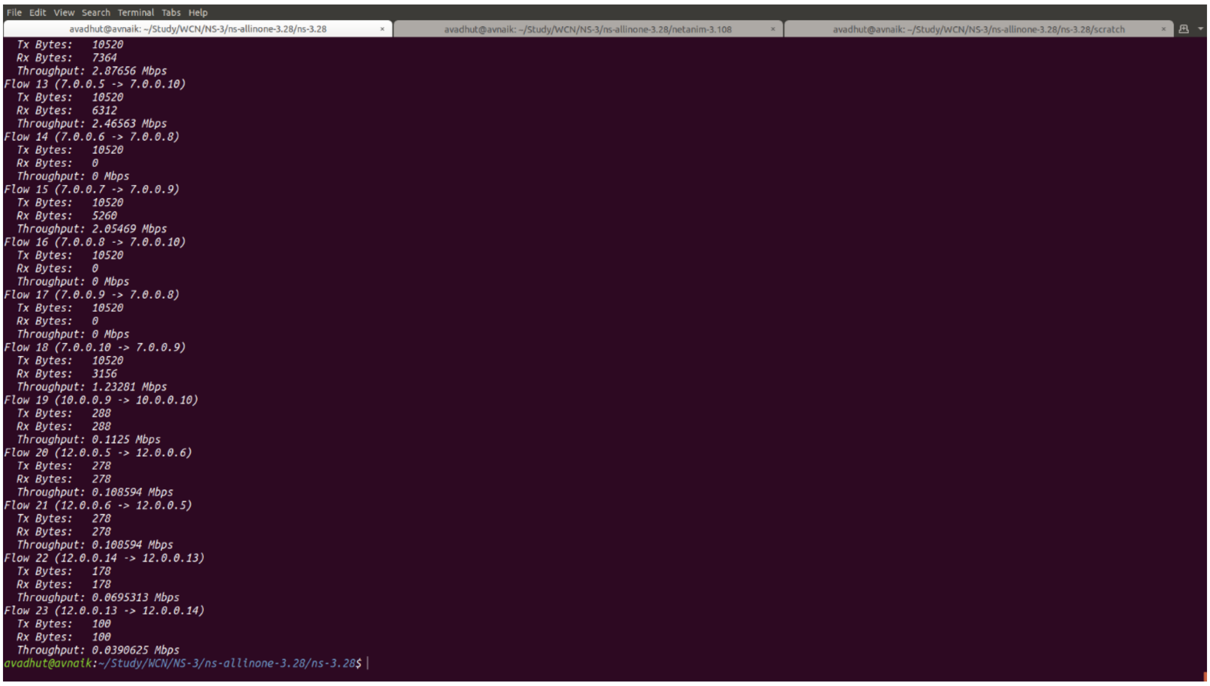


Figure 12: Throughput with PF MAC Scheduler.

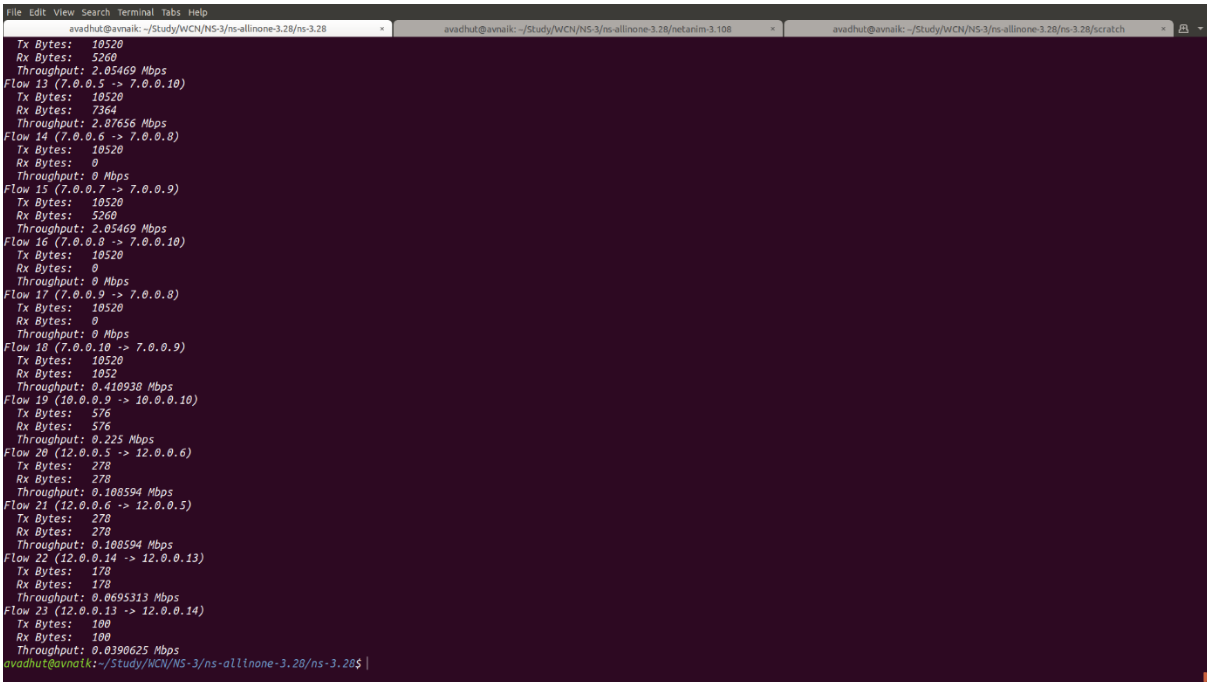


Figure 13: Throughput with TDBET MAC Scheduler.

IX. HANDOVER ALGORITHM

A. ALGORITHM IMPLEMENTED:

The LTE network simulated also has X2 based handover algorithm implemented within so as to facilitate nodes to initiate and complete handover to their nearest EnodeB nodes. The positions of some of the nodes namely node 8, node 10, node 11 and node 13 along with their assignment to EnodeB nodes has been implemented in a manner to make handover a prerequisite for these User Equipment Nodes.

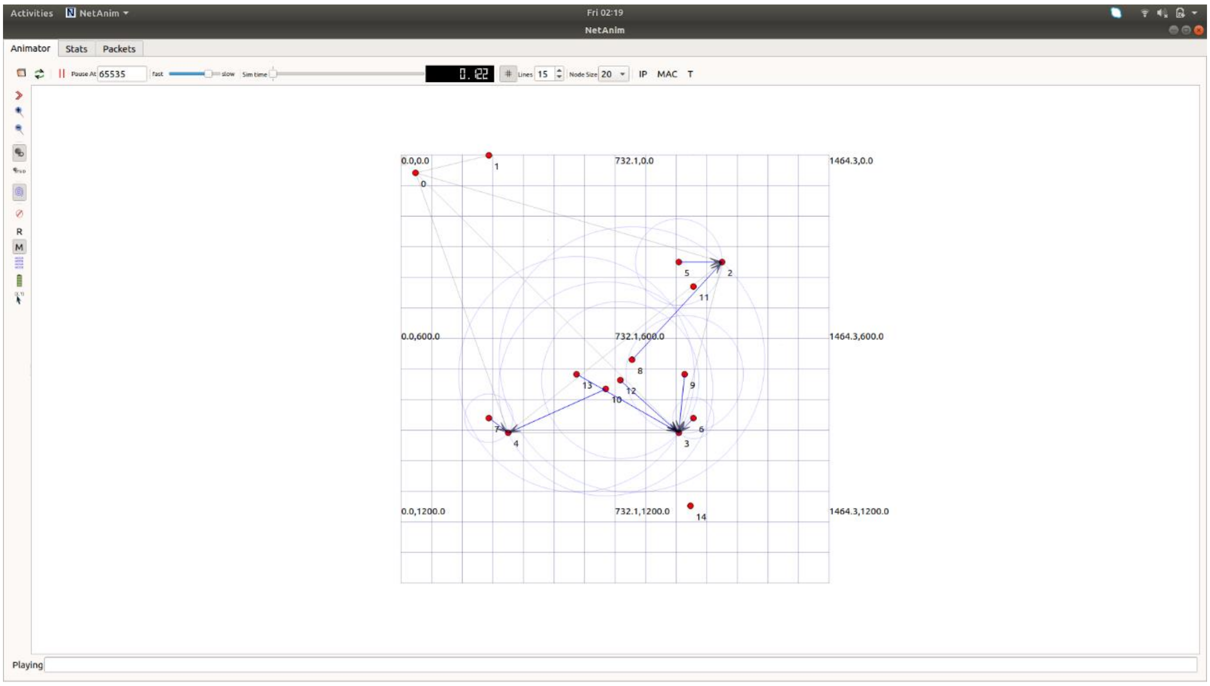


Figure 14: Network Before Handover (NETANIM Simulation)

B. EVENTS OCCURING IN THE HANDOVER:

With Node 8 connected to EnodeB 0 (Node 2), Node 10, 11 connected to EnodeB 2 (Node 4), Node 13 connected to EnodeB 1 (Node 3) which are distant EnodeB nodes in consideration of positions of the nodes, they initiate handover to their nearest EnodeB nodes. While node 8 completes handover to EnodeB 3 and Nodes 10, 11 complete handovers to EnodeB nodes 1 (Node 3), EnodeB node 0 (node 2) respectively, Node 13 completes handover to EnodeB node 2 (Node 4).

C. EXECUTION:

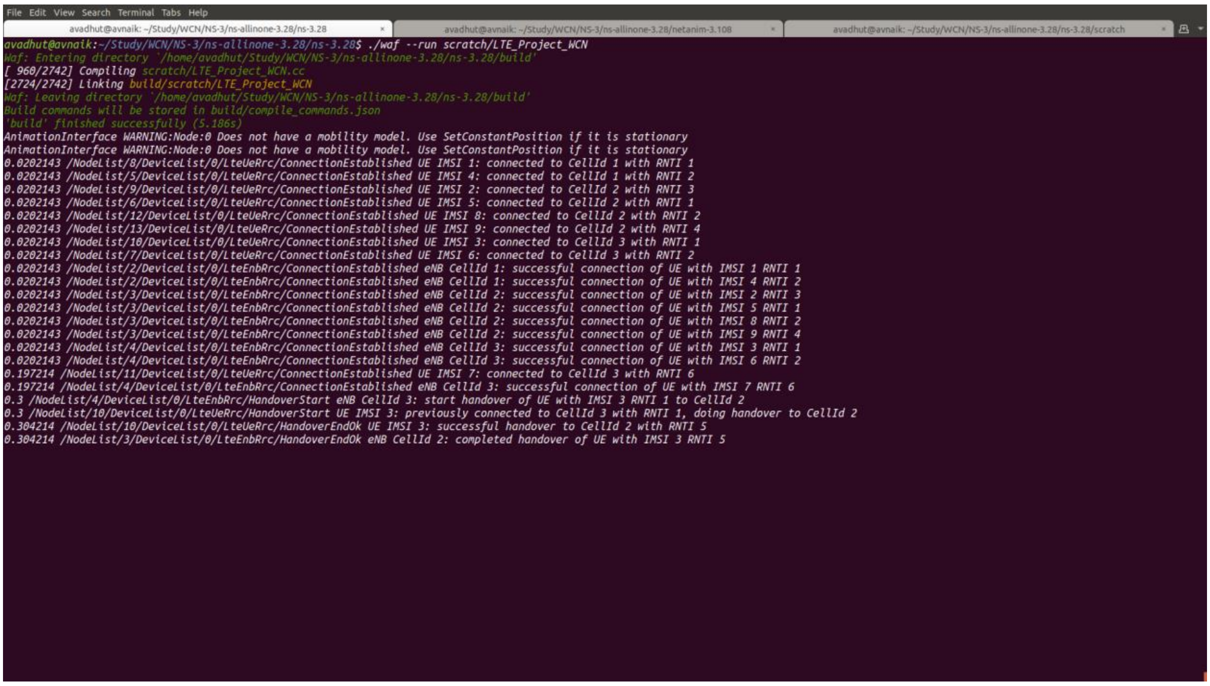


Figure 15: Handover Initiated.

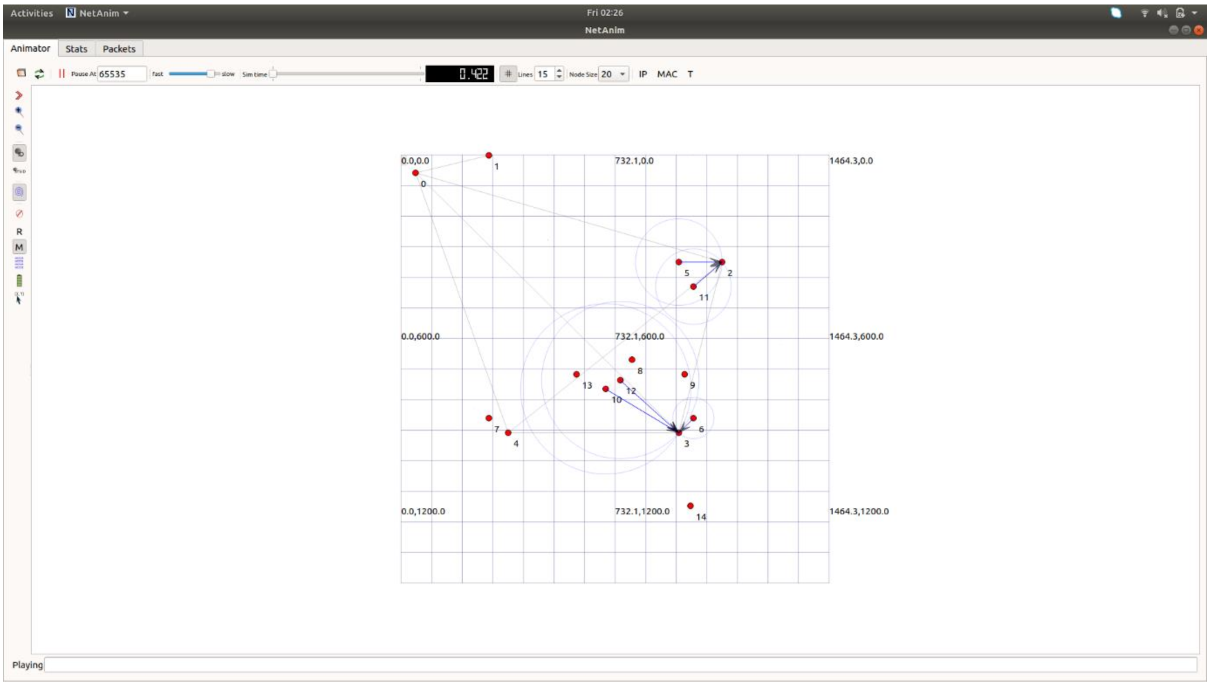


Figure 16: Handover Initiated (NETANIM Simulation).

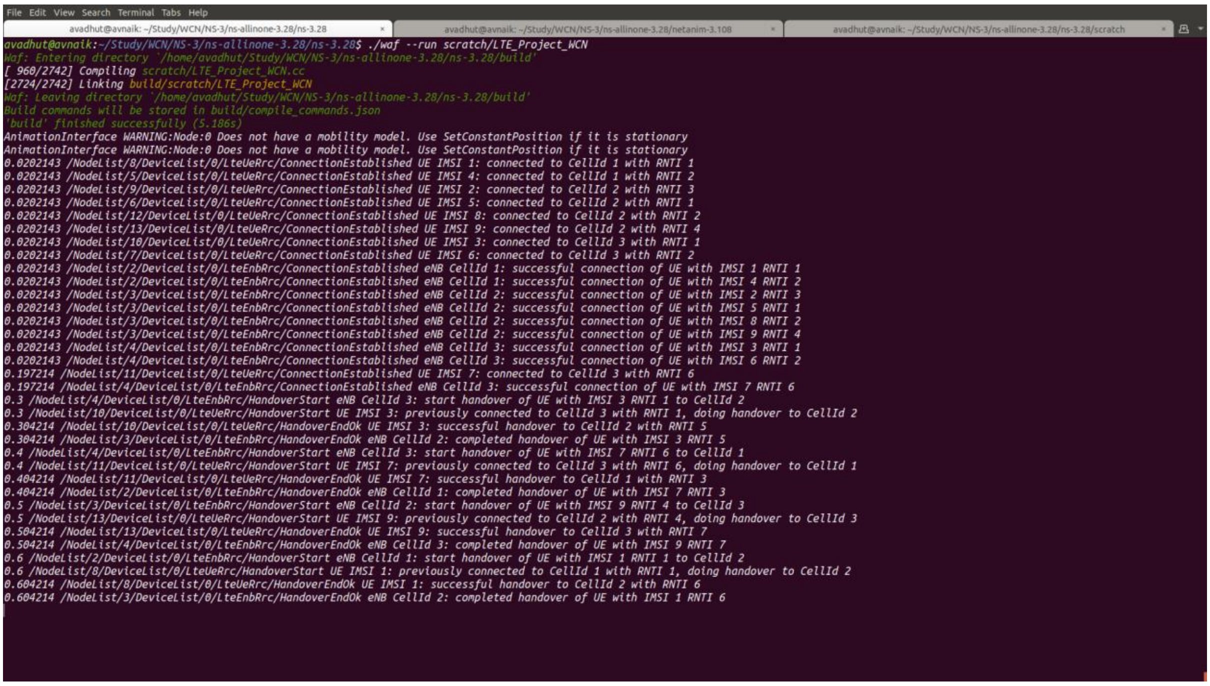


Figure 17: Handover Completed.

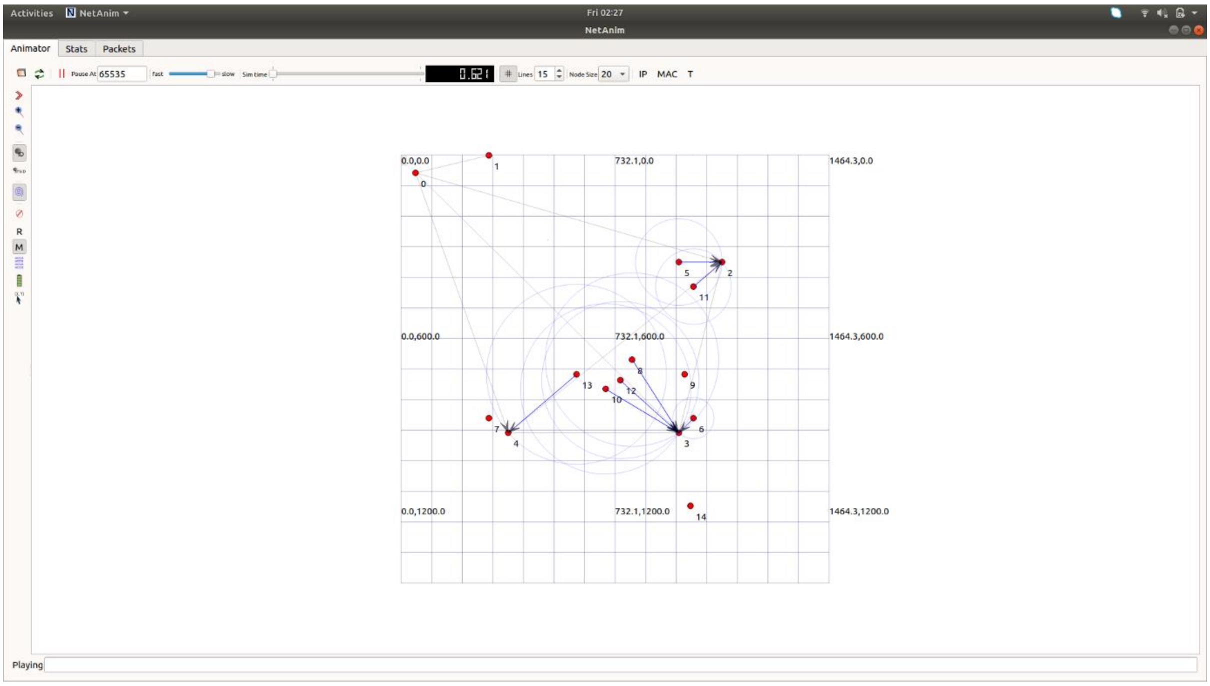


Figure 18: Handover Completed.

Thus, UE nodes (node 8, 10, 11, 13) have successfully completed handover to their nearest EnodeB nodes.

X. CONTRIBUTION

Avadhut Naik	Implemented Frequency Reuse, Topology, Fading and MAC Schedulers
Dushyant Tomar	Implemented AMC, Pathloss and X2 based Handover

XI. CONCLUSION

The LTE Network was thus, successfully implemented with algorithms of AMC, Fading, Pathloss, Frequency Reuse, MAC Scheduler and X2 based Handover. With Strict and Soft Frequency Reuse algorithms implemented, throughputs using each was successfully calculated to compare and contrast them. Also, with Proportional Fair (PF), Time domain Maximize Throughput (TDMT) and Time Domain Blind Equal Throughput (TDBET) MAC Schedulers implemented, throughput utilizing each in the network was successfully calculated. TDMT, it was noted provides the higher throughput than the two which compensate some portion of it to achieve fairness. Handover too, was successfully implemented with nodes completing handover to their nearest EnodeB nodes.