**Automatic plotting of 5G parameters in NetSim**

**Software:** NetSim Standard v13.1 (64-bit), Visual Studio 2019.

**Project Download Link:**

<https://github.com/NetSim-TETCOS/5G_Radio_Measurements_v13.1/archive/refs/heads/main.zip>

Follow the instructions specified in the following link to download and setup the Project in NetSim:

https://support.tetcos.com/en/support/solutions/articles/14000128666-downloading-and-setting-up-netsim-file-exchange-projects

**Features**

Using this workspace:

1. Users can plot Pathloss, Shadow Fading Loss, Total Loss, Rx\_Power, SNR, Beam Forming Gain, MCS Index, and CQI Index vs. time using NetSim Plot.
2. Users can log Pathloss, Shadow Fading Loss, Total Loss, Rx\_Power, SNR, Beam Forming Gain, MCS Index, and CQI Index with time stamps, to a CSV log file.
3. Users need to provide a file-based input (per a certain format) at the start of simulation for the parameters to be plotted or logged.
4. The plots are unique to
5. Each gNB-UE pair
6. Carrier ID
7. DL or UL
8. The log entries are unique to
9. Each gNB-UE pair
10. Carrier ID
11. DL or UL
12. Each layer
13. The output parameters for different MIMO layers ( are stacked in a single plot
14. Parameters are logged every slot time (1ms) and plotted.
15. There is no restriction in NetSim on the number of gNBs / UE in the network.

**Example**

In the below scenario

* The RAN portion has a MIMO layer count of 2, and both FastFading and ShadowFadingLoss are enabled.
* UE-10 moves in a straight line away from the gNB.
* The network is simulated for 60 s.

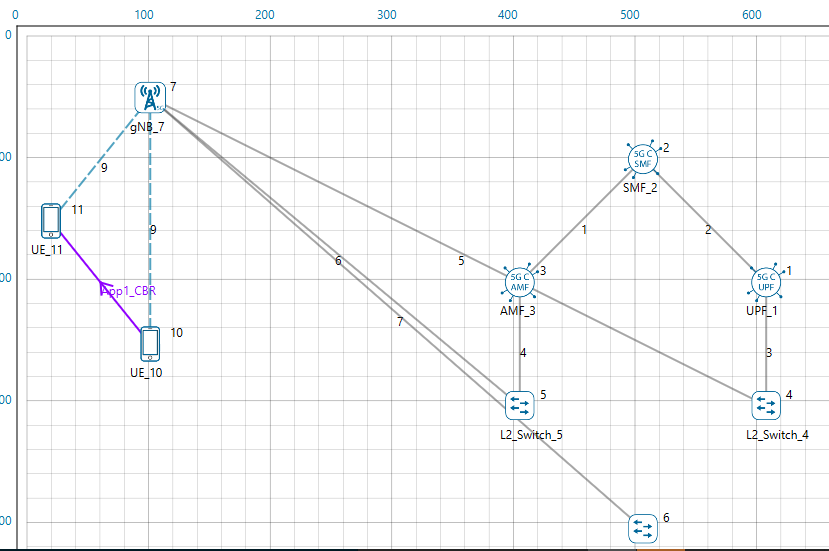


Figure 1: Network Topology in this experiment

* Upon running the simulation, a text file will open for the user to input the parameters and devices (tx-rx pair) for which parameters need to be logged and plotted. The input is per the format of <parameter>,<device1>,<device2> in the text file. To log gNB – UE (DL) data flow the gNB would be the 1st device while the UE would be the 2nd device. In the reverse direction (UL, UE to gNB) the UE would be the 1st device and the gNB the 2nd devices. In case of multiple gNBs, this input can be given for various gNB-UE pairs. Inputs are not case sensitive.
* For the above example, the input text file is as follows.

TOTALLOSS,gNB\_7,UE\_10

PATHLOSS,gNB\_7,UE\_10

SHADOWFADINGLOSS, gNB\_7,UE\_11

SHADOWFADINGLOSS,gNB\_7,UE\_10

RX\_POWER,gNB\_7,UE\_10

SNR,gNB\_7,UE\_10

BEAMFORMINGGAIN,gNB\_7,UE\_10

CQI,gNB\_7,UE\_10

MCS,gNB\_7,UE\_10

SNR,gNB\_7,UE\_11

* Once the simulation starts, In the command prompt window it will show a message as “**Please update, Save and close the file and press any key to continue**”.
* Add the parameters to be logged, close the input text file and press any key.
* Simulation starts running.

**Results and discussion**

Upon completion of simulation in the result window users can view the various plots.

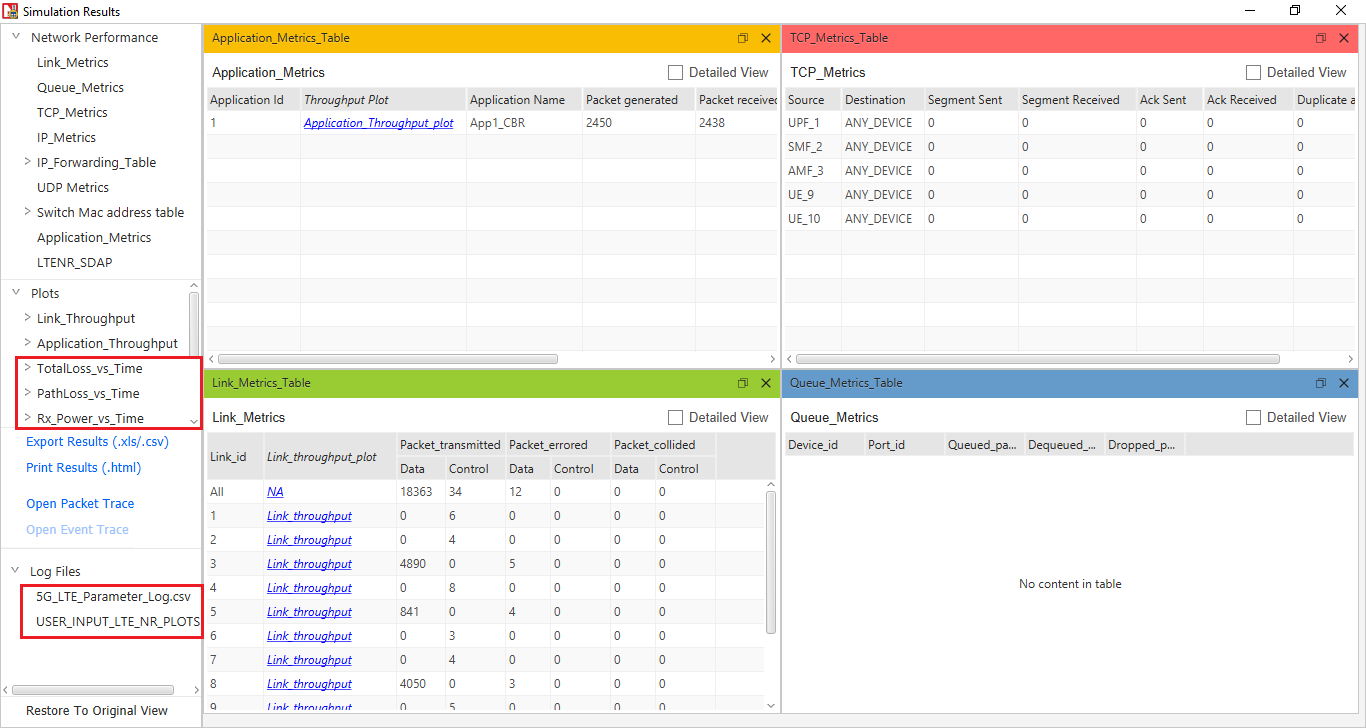


Figure 2: NetSim results dashboard with throughput highlighted

For each carrier, a separate plot is plotted with all the MIMO layers stacked in a single plot.

The pathloss, shadow fading loss, and total loss remains same across the layers. Hence, for these parameters there is a single plot for all layers.

**Result Plots**

1. **Pathloss Plot**

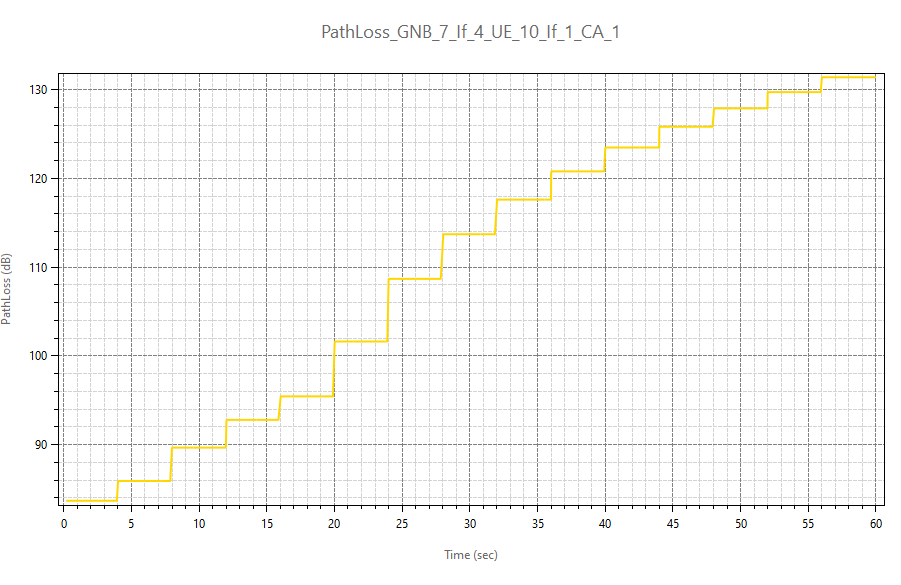


Figure 3: Pathloss Plot in NetSim

1. **Total Loss (Shadow Fading loss plus Path loss)**

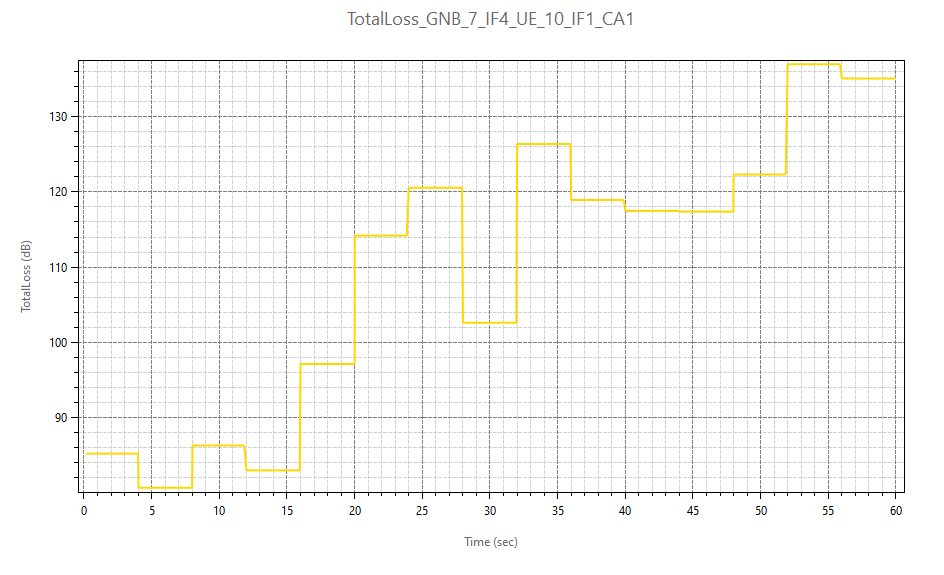


Figure 4: Total Loss (Shadow Fading loss plus Path loss) in NetSim

1. **Shadow Fading Loss**

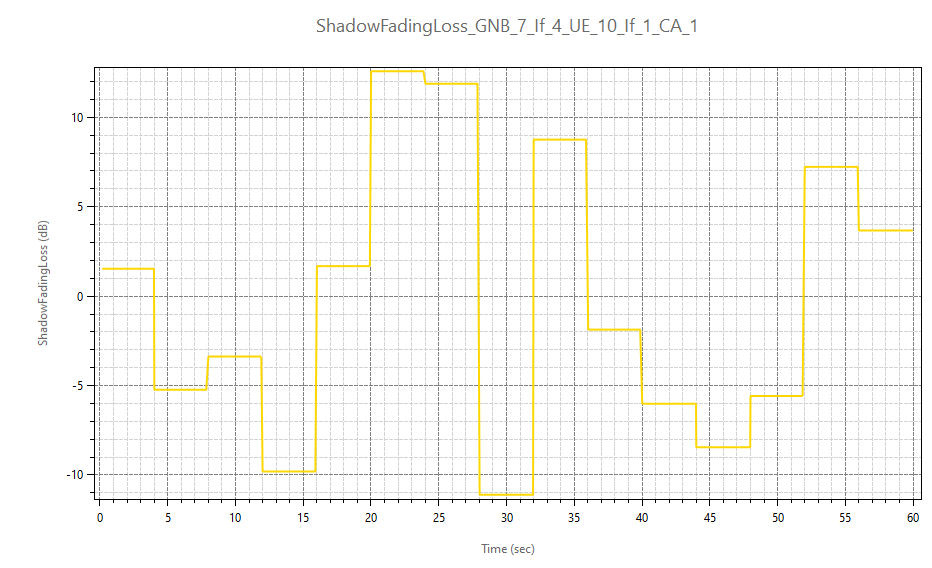
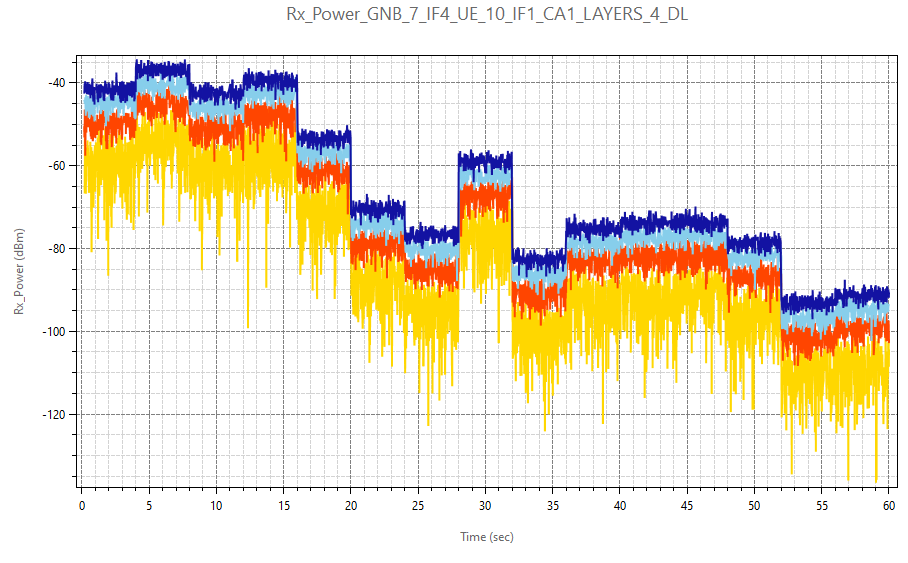


Figure 5: Shadow Fading Loss in NetSim

The plot title is ShadowFadingLoss\_GNB\_7\_IF4\_UE\_10\_IF1\_CA1. And the naming convention is <ParameterType>\_GNB\_<ID>\_IF<InterfaceID>\_UE\_<ID>\_IF<InterfaceID>\_CA<Carrier\_ID>

1. **Rx\_Power Plot**



Layer\_4

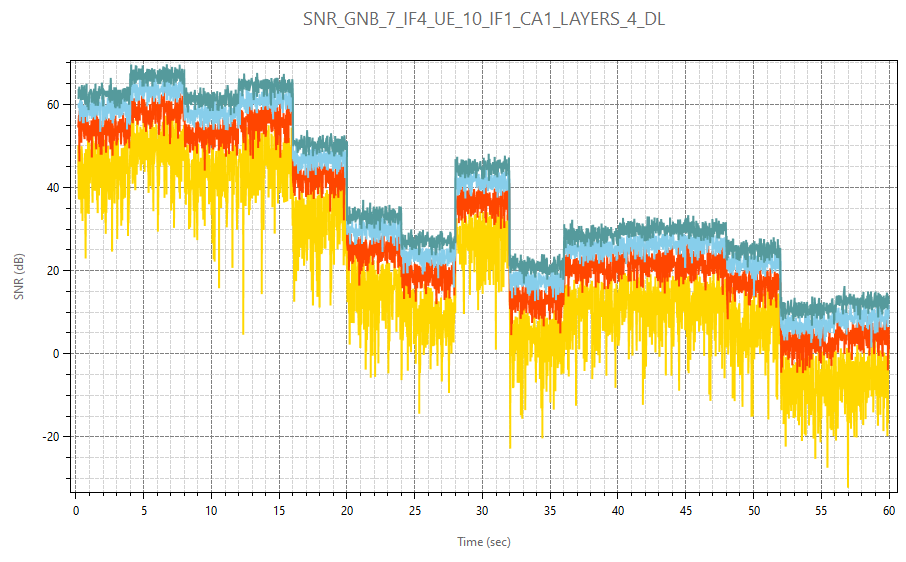
Layer\_1

Layer\_2

Layer\_3

Figure 6: Rx\_Power Plot in NetSim

1. **SNR Plot**



Layer\_4

Layer\_3

Layer\_2

Layer\_1

Figure 7: SNR Plot in NetSim

1. **Beam Forming Gain** (if, the Fast-Fading Model is set to Rayleigh with Eigen Beamforming)

Layer\_4

Layer\_3

Layer\_2

Layer\_1

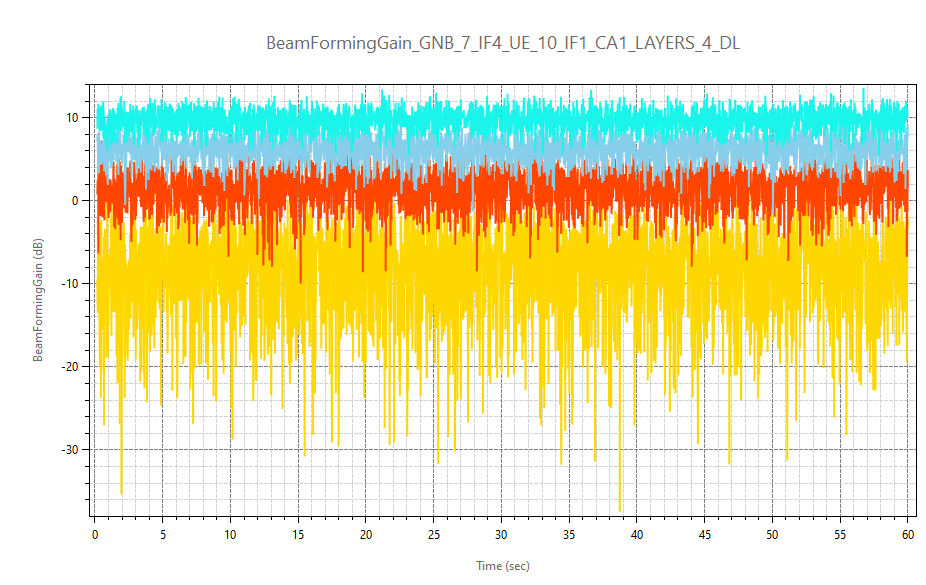


Figure 8: Beam Forming Gain in NetSim

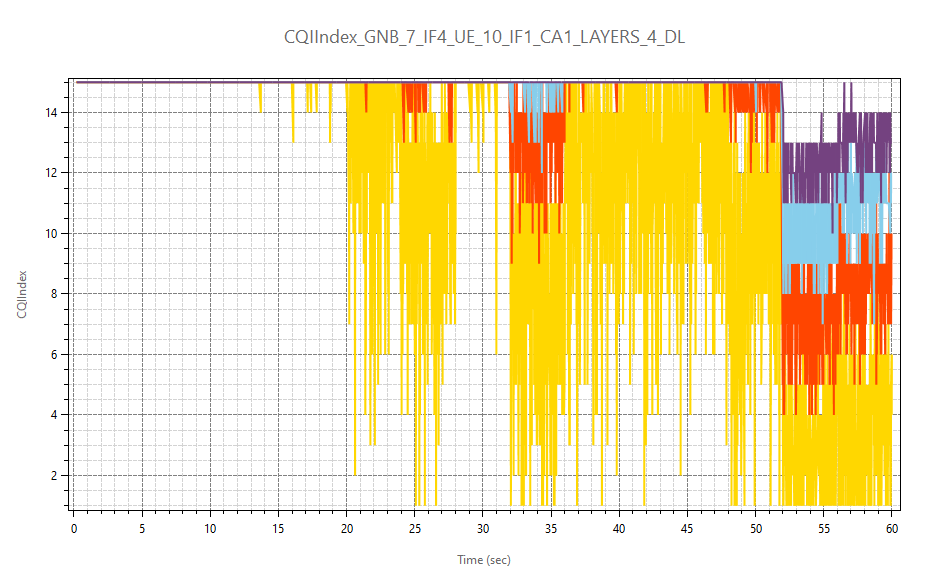
1. **Array Gain** (if, the Fast-Fading Model is set to No Fading MIMO Array Gain)

**Chart, table

Description automatically generated**

Figure 9: Array Gain in NetSim

1. **CQI Index Plot**



Layer\_4

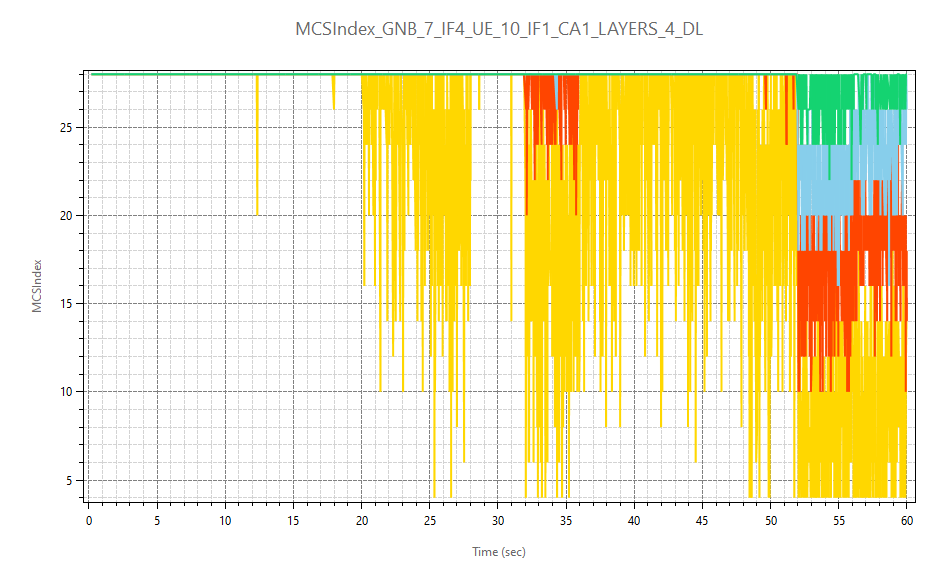
Layer\_3

Layer\_2

Layer\_1

Figure 10: CQI Index Plot in NetSim

1. **MCS Index Plot**



Layer\_4

Layer\_3

Layer\_2

Layer\_1

Figure 11: MCS Index Plot in NetSim

The SNR, Rx\_Power, Beam Forming Gain, CQI Index, MCS Index plots are plotted for all MIMO layers for a Carrier 1. In the chart title layer count and application direction (DL/UL) are also present.

**Parameter log file**

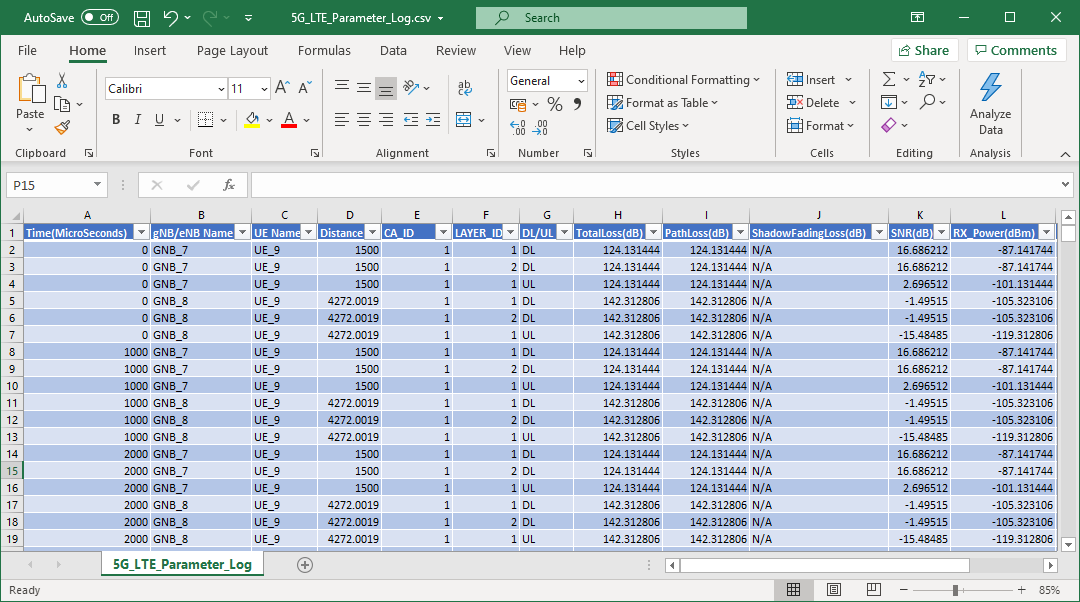


Figure 12: 5G Log file parameter

The 5G\_LTE\_Parameter\_Log.csv file logs the details of parameters specified in the input file with respect to time.

**Appendix: NetSim source code modifications**

Open the Source codes in Visual Studio by going to Your work-> Workspace Options and Clicking on Open code button.

To the in LTE\_NR project, files LTE\_NR\_Plot.c and LTE\_NR\_Parameter\_Log.c has been added. These files contain the definitions of the functions that responsible for plotting and logging parameters associated with 5G/LTE networks in NetSim.

The function fn\_NetSim\_LTE\_NR\_Init\_Plots and fn\_NetSim\_LTE\_NR\_init\_Parameter\_Log has been called in LTENR.c file for initializing the plot.

static bool isplotinit= false;

//Function prototype

int fn\_NetSim\_LTE\_NR\_Init\_F();

int fn\_NetSim\_LTE\_NR\_Configure\_F(void\*\* var);

int fn\_NetSim\_LTE\_NR\_Finish\_F();

#pragma endregion

#pragma region LTENR\_INIT

\_declspec(dllexport) int fn\_NetSim\_LTE\_NR\_Init()

{

if(!isplotinit)

{

fn\_NetSim\_LTE\_NR\_Init\_Plots();

fn\_NetSim\_LTE\_NR\_init\_Parameter\_Log();

isplotinit= true;

}

return fn\_NetSim\_LTE\_NR\_Init\_F();

}

The initialization of functions and functions to update the logs for plotting and logging to CSV file has been made as follows in LTENR\_handleStartSlotEvent function.

void LTENR\_handleStartSlotEvent()

{

NETSIM\_ID gnbId = pstruEventDetails->nDeviceId;

NETSIM\_ID gnbIf = pstruEventDetails->nInterfaceId;

ptrLTENR\_GNBPHY phy = LTENR\_GNBPHY\_GET(gnbId, gnbIf);

#pragma warning (disable : 4047)

int CA\_ID = pstruEventDetails->szOtherDetails;

#pragma warning (default : 4047)

ptrLTENR\_CA ca = phy->spectrumConfig->CA[CA\_ID];

LTENR\_resetSlot(phy, CA\_ID);

print\_ltenr\_log("Starting new slot for gNB %d:%d\n", gnbId, gnbIf);

print\_ltenr\_log("CA\_ID for Slot = %d\n", CA\_ID);

print\_ltenr\_log("\tFrame Id = %d\n", phy->frameInfo[CA\_ID]->frameId);

print\_ltenr\_log("\tSubFrame Id = %d\n", phy->frameInfo[CA\_ID]->subFrameId);

print\_ltenr\_log("\tSlot Id = %d\n", phy->frameInfo[CA\_ID]->slotId);

print\_ltenr\_log("\tSlot start time (us) = %lf\n", phy->frameInfo[CA\_ID]->slotStartTime);

print\_ltenr\_log("\tslot end time (us) = %lf\n", phy->frameInfo[CA\_ID]->slotEndTime);

print\_ltenr\_log("\tSlot type = %s\n", strLTENR\_SLOTTYPE[phy->frameInfo[CA\_ID]->slotType]);

phy->currentFrameInfo = phy->frameInfo[CA\_ID];

phy->currentFrameInfo->Current\_CA\_ID = CA\_ID;

if (phy->frameInfo[CA\_ID]->slotId != ca->slotPerSubframe)

LTENR\_addStartSlotEvent(gnbId, gnbIf,

phy->frameInfo[CA\_ID]->slotEndTime, CA\_ID);

ptrLTENR\_ASSOCIATEDUEPHYINFO info = phy->associatedUEPhyInfo;

//

if (pstruEventDetails->dEventTime ==0 || pstruEventDetails->dEventTime==200000)

{

for (NETSIM\_ID d = 1; d <= NETWORK->nDeviceCount; d++)

{

for (NETSIM\_ID in = 1; in <= DEVICE(d)->nNumOfInterface; in++)

{

if (!isLTE\_NRInterface(d, in))

continue;

if (!isGNB(d, in))

continue;

ptrLTENR\_GNBPHY phy\_ = LTENR\_GNBPHY\_GET(d, in);

ptrLTENR\_ASSOCIATEDUEPHYINFO info\_ = phy\_->associatedUEPhyInfo;

while (info\_)

{

fn\_NetSim\_LTE\_NR\_init\_PropagationInfo\_Plots(phy\_, info\_);

fn\_NetSim\_LTE\_NR\_init\_Power\_Plots(phy\_, info\_);

LTENR\_ASSOCIATEDUEPHYINFO\_NEXT(info\_);

}

}

}

}

while (info)

{

if (info->isAssociated)

{

for (NETSIM\_ID i = 0; i < phy->ca\_count; i++)

LTENR\_PHY\_setAMCInfo(phy, info, i);

}

fn\_NetSim\_LTE\_NR\_add\_PropagationInfo\_Plot\_data(info, CA\_ID);

fn\_NetSim\_LTE\_NR\_add\_Power\_Plot\_data(info, CA\_ID);

ptrINFO param\_info = parameter\_log\_info;

if (param\_info->isParameterlog)

fn\_NetSim\_LTE\_NR\_Log\_Parameters(phy, CA\_ID, info);

info = LTENR\_ASSOCIATEDUEPHYINFO\_NEXT(info);

}

LTENR\_NotifyMACForStartingSlot();

phy->frameInfo[CA\_ID]->prevSlotType = phy->frameInfo[CA\_ID]->slotType;

}

For adding plot data at every slot time (1 ms) the below highlighted function has been used in LTENR\_phy.c file.

static void LTENR\_PHY\_setAMCInfo(ptrLTENR\_GNBPHY phy, ptrLTENR\_ASSOCIATEDUEPHYINFO info, int CA\_ID)

{

UINT layerCount;

ptrLTENR\_UEPHY uePhy = LTENR\_UEPHY\_GET(info->ueId, info->ueIf);

//Downlink

layerCount = LTENR\_PHY\_GET\_DLLAYER\_COUNT(uePhy);

for (UINT i = 0; i < layerCount; i++)

{

print\_ltenr\_log("\tAMC info between gNB %d:%d and UE %d:%d, Carrier Id = %d, Layer Id = %d for downlink-\n",

phy->gnbId, phy->gnbIf,

info->ueId, info->ueIf,

CA\_ID, i);

info->downlinkAMCInfo[CA\_ID][i]->SpectralEfficiency = LTENR\_PHY\_GetDownlinkSpectralEfficiency(info->propagationInfo[CA\_ID], i);

setAMCInfo(phy, info->downlinkAMCInfo[CA\_ID][i]);

}

//Uplink

layerCount = LTENR\_PHY\_GET\_ULLAYER\_COUNT(uePhy);

for (UINT i = 0; i < layerCount; i++)

{

print\_ltenr\_log("\tAMC info between gNB %d:%d and UE %d:%d, Carrier Id = %d, Layer Id = %d for uplink-\n",

phy->gnbId, phy->gnbIf,

info->ueId, info->ueIf,

CA\_ID, i);

info->uplinkAMCInfo[CA\_ID][i]->SpectralEfficiency = LTENR\_PHY\_GetUplinkSpectralEfficiency(info->propagationInfo[CA\_ID], i);

setAMCInfo(phy, info->uplinkAMCInfo[CA\_ID][i]);

}

fn\_NetSim\_LTE\_NR\_add\_AMCInfo\_Plot\_data(info, CA\_ID);

}

**Disabling Plotting/Logging**

Generation of plots or the parameter log can be disabled by commenting the function calls in the fn\_NetSim\_LTE\_NR\_Init() function. The function call fn\_NetSim\_LTE\_NR\_Init\_Plots can be commented to disable plots and the function call fn\_NetSim\_LTE\_NR\_init\_Parameter\_Log can be commented to disable generation of a parameter log CSV file.

\_declspec(dllexport) int fn\_NetSim\_LTE\_NR\_Init()

{

if(!isplotinit)

{

fn\_NetSim\_LTE\_NR\_Init\_Plots(); //comment line to disable plots

fn\_NetSim\_LTE\_NR\_init\_Parameter\_Log(); //comment line to disable parameter log

isplotinit = true;

}

return fn\_NetSim\_LTE\_NR\_Init\_F();

}