Modelling Obstacles between UEs and eNB in NetSim LTE

Software Recommended: NetSim Standard v13.0 (32-bit/64-bit), Visual Studio 2017/2019

Project Download Link:

https://github.com/NetSim-TETCOS/MODELLING_OBSTACLES_IN_LTE_v13.0/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

Introduction:

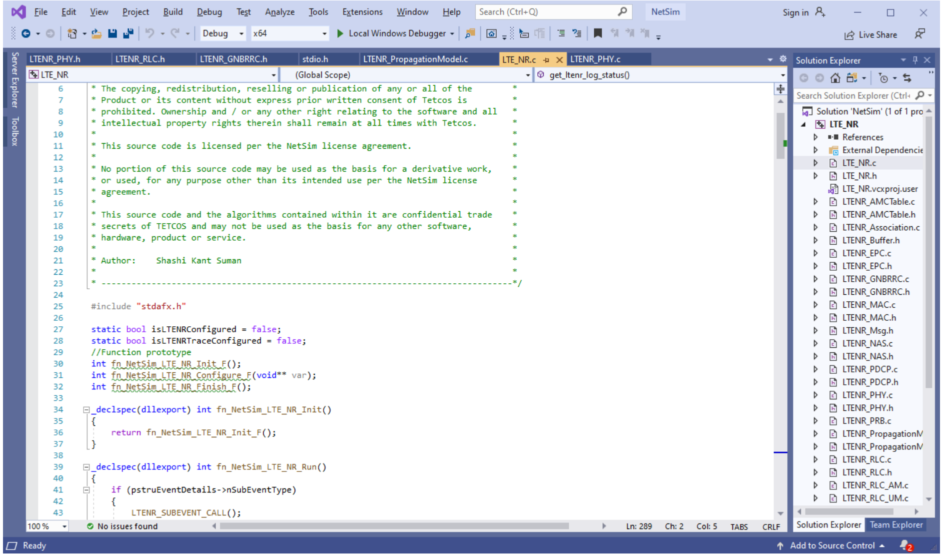
Users can model obstacles to vary the channel losses between the eNB and the connected UEs, by modifying the underlying LTE code.

This is required because, as of **NetSim v13.0**, in the GUI, the wireless link (between one eNB and all connected UEs) properties are same i.e. if we change in one of the UE-eNB links, the change will reflect in all the connect UE-eNB links.

Obstacles are modelled by adding an attenuation (dB) value. Other channel conditions can be varying the stochastic pathloss model based on 3GPPTR38.900 standard. These include environment/parameters such Rural/urban, indoor/outdoor, LOS/NLOS, O2I High-lows/Low loss etc.

Steps:

- Go to NetSim Home Page, click on Your Work->Workspace Options and click on the Open Code button.

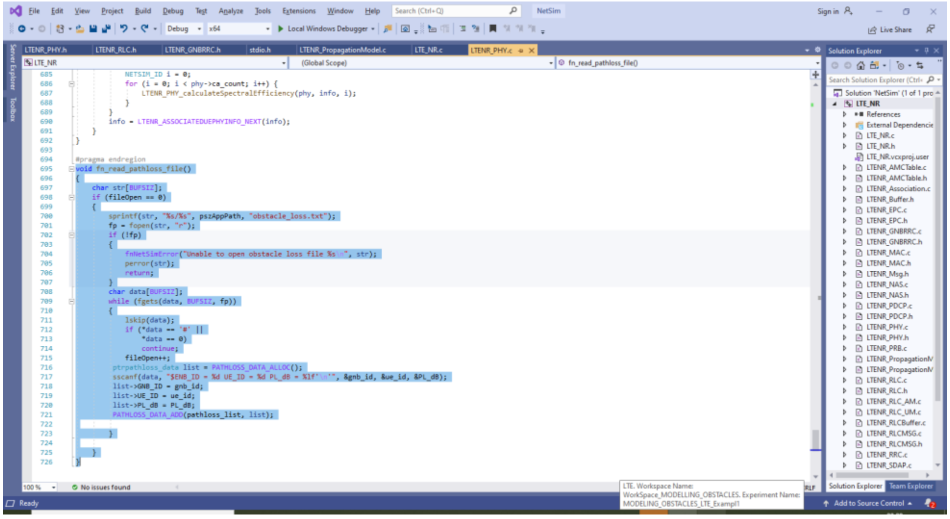


```
* The copying, redistribution, reselling or publication of any or all of the       *
* Product or its content without express prior written consent of Tetcos is       *
* prohibited. Ownership and / or any other right relating to the software and all  *
* intellectual property rights therein shall remain at all times with Tetcos.      *
*                                                                                  *
* This source code is licensed per the NetSim license agreement.                   *
*                                                                                  *
* No portion of this source code may be used as the basis for a derivative work,   *
* or used, for any purpose other than its intended use per the NetSim license      *
* agreement.                                                                       *
*                                                                                  *
* This source code and the algorithms contained within it are confidential trade   *
* secrets of TETCOS and may not be used as the basis for any other software,       *
* hardware, product or service.                                                    *
*                                                                                  *
* Author:    Shashi Kant Suman                                                     *
*                                                                                  *
* ---------------------------------------------------------------------------------*/

#include "stdafx.h"

static bool isLTENRConfigured = false;
static bool isLTENRTraceConfigured = 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();

_declspec(dllexport) int fn_NetSim_LTE_NR_Init()
{
    return fn_NetSim_LTE_NR_Init_F();
}

_declspec(dllexport) int fn_NetSim_LTE_NR_Run()
{
    if (pstruEventDetails->nSubEventType)
    {
        LTENR_SUBEVENT_CALL();
```

- Right click on Solution in Solution Explorer and select rebuild solution.
- Upon rebuilding, libLTE.dll will get created in the bin_x86/ bin_x64 folder.
- Go to NetSim home page, click on Your Work, Click on MODELING_OBSTACLES_LTE_Experiment.
- After simulation, note down the throughputs available in the metrics window.

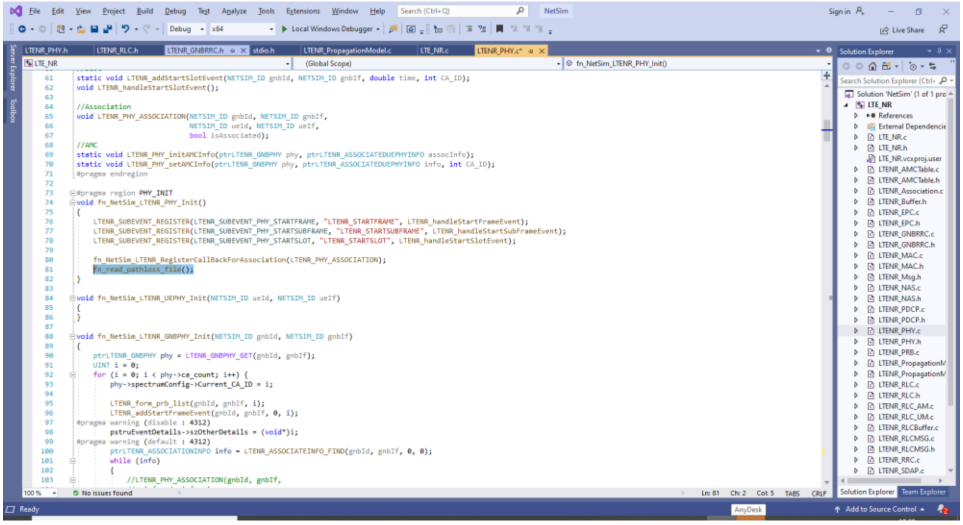
Steps to be done in NetSim to configure different path loss exponents:

To read the file content, we have added the following lines of code in LTENR_PHY.c file present inside LTE_NR project as shown below:



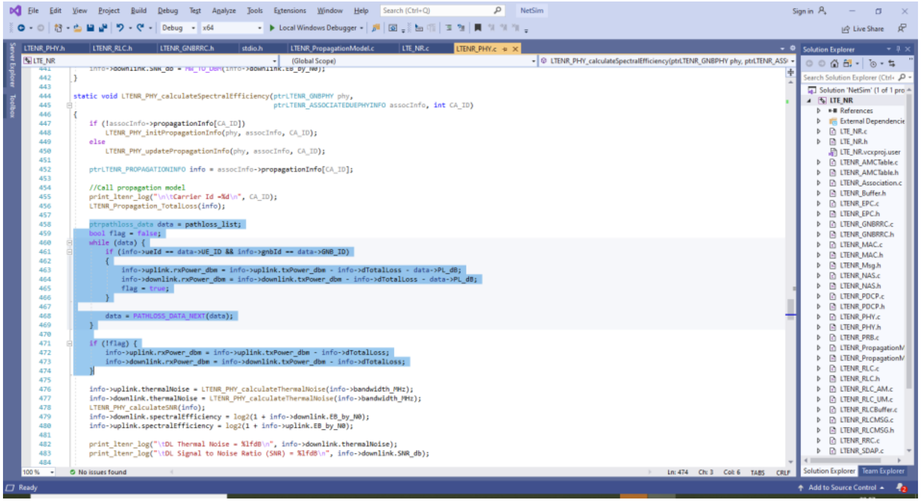
```
            NETSIM_ID i = 0;
            for (i = 0; i < phy->ca_count; i++) {
                LTENR_PHY_calculateSpectralEfficiency(phy, info, i);
            }
        }
        info = LTENR_ASSOCIATEDUEPHYINFO_NEXT(info);
    }
}

#pragma endregion
void fn_read_pathloss_file()
{
    char str[BUFSIZ];
    if (fileOpen == 0)
    {
        sprintf(str, "%s/%s", pszAppPath, "obstacle_loss.txt");
        fp = fopen(str, "r");
        if (!fp)
        {
            fnNetSimError("Unable to open obstacle loss file %s\n", str);
            perror(str);
            return;
        }
        char data[BUFSIZ];
        while (fgets(data, BUFSIZ, fp))
        {
            lskip(data);
            if (*data == '#' ||
                *data == 0)
                continue;
            fileOpen++;
         ptrpathloss_data list = PATHLOSS_DATA_ALLOC();
         sscanf(data, "$ENB_ID = %d UE_ID = %d PL_dB = %lf'\n'", &gnb_id, &ue_id, &PL_dB);
         list->GNB_ID = gnb_id;
         list->UE_ID = ue_id;
         list->PL_dB = PL_dB;
         PATHLOSS_DATA_ADD(pathloss_list, list);

        }

    }
}
```

We have added the following lines of code in fn_NetSim_LTENR_PHY_Init () present in LTE_PHY.c file.



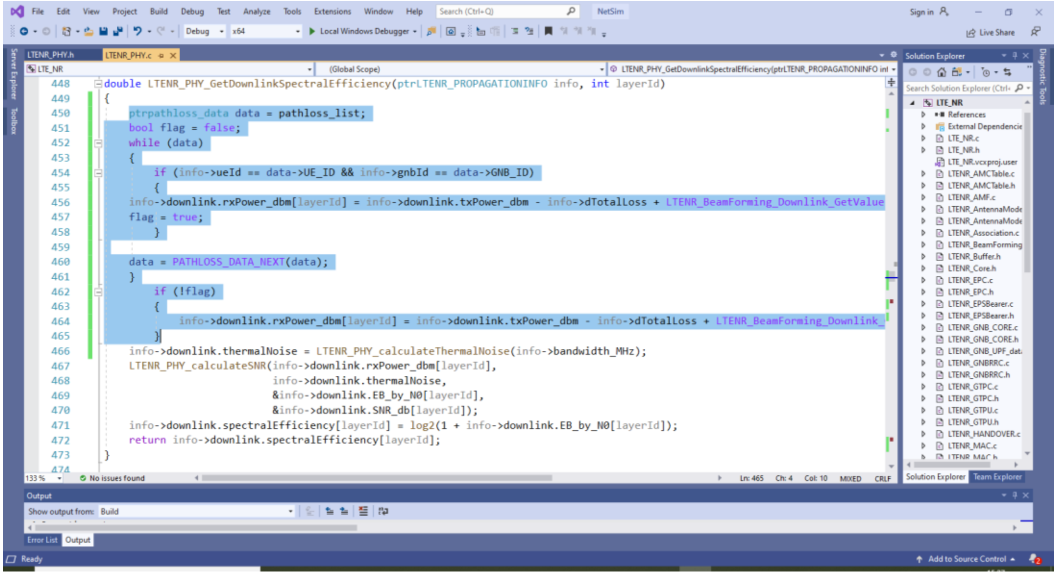
```c
static void LTENR_addStartSlotEvent(NETSIM_ID gnbId, NETSIM_ID gnbIf, double time, int CA_ID);
void LTENR_handleStartSlotEvent();

//Association
void LTENR_PHY_ASSOCIATION(NETSIM_ID gnbId, NETSIM_ID gnbIf,
                           NETSIM_ID ueId, NETSIM_ID ueIf,
                           bool isAssociated);
//AMC
static void LTENR_PHY_initAMCInfo(ptrLTENR_GNBPHY phy, ptrLTENR_ASSOCIATEDUEPHYINFO assocInfo);
static void LTENR_PHY_setAMCInfo(ptrLTENR_GNBPHY phy, ptrLTENR_ASSOCIATEDUEPHYINFO info, int CA_ID);
#pragma endregion

#pragma region PHY_INIT
void fn_NetSim_LTENR_PHY_Init()
{
    LTENR_SUBEVENT_REGISTER(LTENR_SUBEVENT_PHY_STARTFRAME, "LTENR_STARTFRAME", LTENR_handleStartFrameEvent);
    LTENR_SUBEVENT_REGISTER(LTENR_SUBEVENT_PHY_STARTSUBFRAME, "LTENR_STARTSUBFRAME", LTENR_handleStartSubFrameEvent);
    LTENR_SUBEVENT_REGISTER(LTENR_SUBEVENT_PHY_STARTSLOT, "LTENR_STARTSLOT", LTENR_handleStartSlotEvent);

    fn_NetSim_LTENR_RegisterCallBackForAssociation(LTENR_PHY_ASSOCIATION);
    fn_read_pathloss_file();
}

void fn_NetSim_LTENR_UEPHY_Init(NETSIM_ID ueId, NETSIM_ID ueIf)
{
}

void fn_NetSim_LTENR_GNBPHY_Init(NETSIM_ID gnbId, NETSIM_ID gnbIf)
{
    ptrLTENR_GNBPHY phy = LTENR_GNBPHY_GET(gnbId, gnbIf);
    UINT i = 0;
    for (i = 0; i < phy->ca_count; i++) {
        phy->spectrumConfig->Current_CA_ID = i;

        LTENR_form_prb_list(gnbId, gnbIf, i);
        LTENR_addStartFrameEvent(gnbId, gnbIf, 0, i);
#pragma warning (disable : 4312)
        pstruEventDetails->szOtherDetails = (void*)i;
#pragma warning (default : 4312)
        ptrLTENR_ASSOCIATIONINFO info = LTENR_ASSOCIATEINFO_FIND(gnbId, gnbIf, 0, 0);
        while (info)
        {
            //LTENR_PHY_ASSOCIATION(gnbId, gnbIf,
```

And then the following lines in LTENR_PHY_calculateSpectralEfficiency() present in LTENR_PHY.c file.



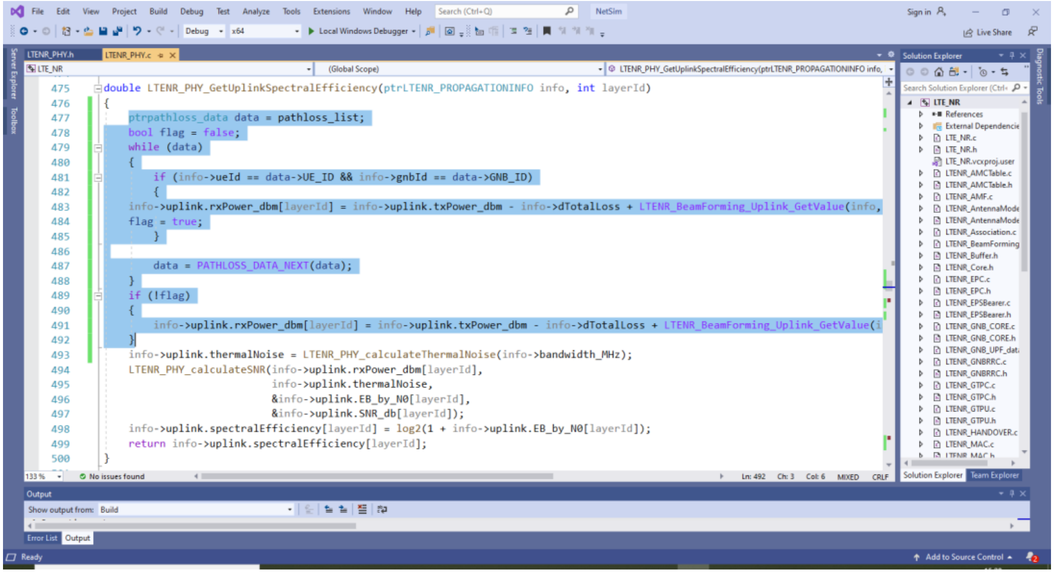
```c
    info->downlink.SNR_db = MW_TO_DBM(info->downlink.EB_by_N0);
}

static void LTENR_PHY_calculateSpectralEfficiency(ptrLTENR_GNBPHY phy,
                                                  ptrLTENR_ASSOCIATEDUEPHYINFO assocInfo, int CA_ID)
{
    if (!assocInfo->propagationInfo[CA_ID])
        LTENR_PHY_initPropagationInfo(phy, assocInfo, CA_ID);
    else
        LTENR_PHY_updatePropagationInfo(phy, assocInfo, CA_ID);

    ptrLTENR_PROPAGATIONINFO info = assocInfo->propagationInfo[CA_ID];

    //Call propagation model
    print_ltenr_log("\n\tCarrier Id =%d\n", CA_ID);
    LTENR_Propagation_TotalLoss(info);

    ptrpathloss_data data = pathloss_list;
    bool flag = false;
    while (data) {
        if (info->ueId == data->UE_ID && info->gnbId == data->GNB_ID)
        {
            info->uplink.rxPower_dbm = info->uplink.txPower_dbm - info->dTotalLoss - data->PL_dB;
            info->downlink.rxPower_dbm = info->downlink.txPower_dbm - info->dTotalLoss - data->PL_dB;
            flag = true;
        }

        data = PATHLOSS_DATA_NEXT(data);
    }

    if (!flag) {
        info->uplink.rxPower_dbm = info->uplink.txPower_dbm - info->dTotalLoss;
        info->downlink.rxPower_dbm = info->downlink.txPower_dbm - info->dTotalLoss;
    }

    info->uplink.thermalNoise = LTENR_PHY_calculateThermalNoise(info->bandwidth_MHz);
    info->downlink.thermalNoise = LTENR_PHY_calculateThermalNoise(info->bandwidth_MHz);
    LTENR_PHY_calculateSNR(info);
    info->downlink.spectralEfficiency = log2(1 + info->downlink.EB_by_N0);
    info->uplink.spectralEfficiency = log2(1 + info->uplink.EB_by_N0);

    print_ltenr_log("\tDL Thermal Noise = %lfdB\n", info->downlink.thermalNoise);
    print_ltenr_log("\tDL Signal to Noise Ratio (SNR) = %lfdB\n", info->downlink.SNR_db);
```

The following lines in LTENR_PHY_GetDownlinkSpectralEfficiency() present in LTENR_PHY.c file.



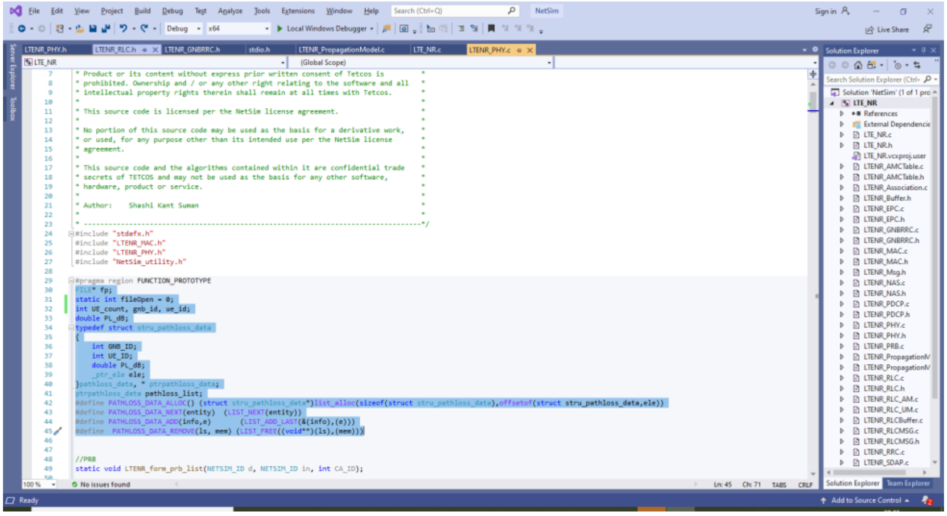
```
double LTENR_PHY_GetDownlinkSpectralEfficiency(ptrLTENR_PROPAGATIONINFO info, int layerId)
{
    ptrpathloss_data data = pathloss_list;
    bool flag = false;
    while (data)
    {
        if (info->ueId == data->UE_ID && info->gnbId == data->GNB_ID)
        {
    info->downlink.rxPower_dbm[layerId] = info->downlink.txPower_dbm - info->dTotalLoss + LTENR_BeamForming_Downlink_GetValue
    flag = true;
        }

    data = PATHLOSS_DATA_NEXT(data);
    }
        if (!flag)
        {
            info->downlink.rxPower_dbm[layerId] = info->downlink.txPower_dbm - info->dTotalLoss + LTENR_BeamForming_Downlink_
        }
    info->downlink.thermalNoise = LTENR_PHY_calculateThermalNoise(info->bandwidth_MHz);
    LTENR_PHY_calculateSNR(info->downlink.rxPower_dbm[layerId],
                           info->downlink.thermalNoise,
                           &info->downlink.EB_by_N0[layerId],
                           &info->downlink.SNR_db[layerId]);
    info->downlink.spectralEfficiency[layerId] = log2(1 + info->downlink.EB_by_N0[layerId]);
    return info->downlink.spectralEfficiency[layerId];
}
```

The following lines in LTENR_PHY_GetUplinkSpectralEfficiency() present in LTENR_PHY.c file.



```
double LTENR_PHY_GetUplinkSpectralEfficiency(ptrLTENR_PROPAGATIONINFO info, int layerId)
{
    ptrpathloss_data data = pathloss_list;
    bool flag = false;
    while (data)
    {
        if (info->ueId == data->UE_ID && info->gnbId == data->GNB_ID)
        {
    info->uplink.rxPower_dbm[layerId] = info->uplink.txPower_dbm - info->dTotalLoss + LTENR_BeamForming_Uplink_GetValue(info,
    flag = true;
        }

        data = PATHLOSS_DATA_NEXT(data);
    }
    if (!flag)
    {
        info->uplink.rxPower_dbm[layerId] = info->uplink.txPower_dbm - info->dTotalLoss + LTENR_BeamForming_Uplink_GetValue(i
    }
    info->uplink.thermalNoise = LTENR_PHY_calculateThermalNoise(info->bandwidth_MHz);
    LTENR_PHY_calculateSNR(info->uplink.rxPower_dbm[layerId],
                           info->uplink.thermalNoise,
                           &info->uplink.EB_by_N0[layerId],
                           &info->uplink.SNR_db[layerId]);
    info->uplink.spectralEfficiency[layerId] = log2(1 + info->uplink.EB_by_N0[layerId]);
    return info->uplink.spectralEfficiency[layerId];
}
```

the following lines added starting of LTENR_PHY.c file.



```
* Product or its content without express prior written consent of Tetcos is       *
* prohibited. Ownership and / or any other right relating to the software and all  *
* intellectual property rights therein shall remain at all times with Tetcos.      *
*                                                                                  *
* This source code is licensed per the NetSim license agreement.                   *
*                                                                                  *
* No portion of this source code may be used as the basis for a derivative work,   *
* or used, for any purpose other than its intended use per the NetSim license      *
* agreement.                                                                       *
*                                                                                  *
* This source code and the algorithms contained within it are confidential trade   *
* secrets of TETCOS and may not be used as the basis for any other software,       *
* hardware, product or service.                                                    *
*                                                                                  *
* Author:    Shashi Kant Suman                                                     *
*                                                                                  *
* ---------------------------------------------------------------------------------*/
#include "stdafx.h"
#include "LTENR_MAC.h"
#include "LTENR_PHY.h"
#include "NetSim_utility.h"

#pragma region FUNCTION_PROTOTYPE
FILE* fp;
static int fileOpen = 0;
int UE_count, gnb_id, ue_id;
double PL_dB;
typedef struct stru_pathloss_data
{
    int GNB_ID;
    int UE_ID;
    double PL_dB;
    _ptr_ele ele;
}pathloss_data, * ptrpathloss_data;
ptrpathloss_data pathloss_list;
#define PATHLOSS_DATA_ALLOC() (struct stru_pathloss_data*)list_alloc(sizeof(struct stru_pathloss_data),offsetof(struct stru_pathloss_data,ele))
#define PATHLOSS_DATA_NEXT(entity)  (LIST_NEXT(entity))
#define PATHLOSS_DATA_ADD(info,e)       (LIST_ADD_LAST(&(info),(e)))
#define  PATHLOSS_DATA_REMOVE(ls, mem) (LIST_FREE((void**)(ls),(mem)))


//PRB
static void LTENR_form_prb_list(NETSIM_ID d, NETSIM_ID in, int CA_ID);
```

Create a obstacle_loss.txt file and paste it in the install directory of NetSim would look something like **“<MODELLING_OBSTACLES_IN_LTE_v13.0 path>\bin\bin_x64”** and the file format should be

#Obstacle pathloss file. Naming: obstacle_loss.txt

#Place this file in "workspace/bin/bin_x64" folder of NetSim

#The format of this file is

#1st parameter - ENB ID

#2nd parameter - UE ID

#3rd parameter – Obstacle pathloss in dB (A positive loss value which implies a negative gain)

#This obstacle pathloss will get added to the regular pathloss thereby

#reducing the signal power at receiver

#Ex: To set an obstacle pathloss of 50dB between 1 to 2 you have to set it

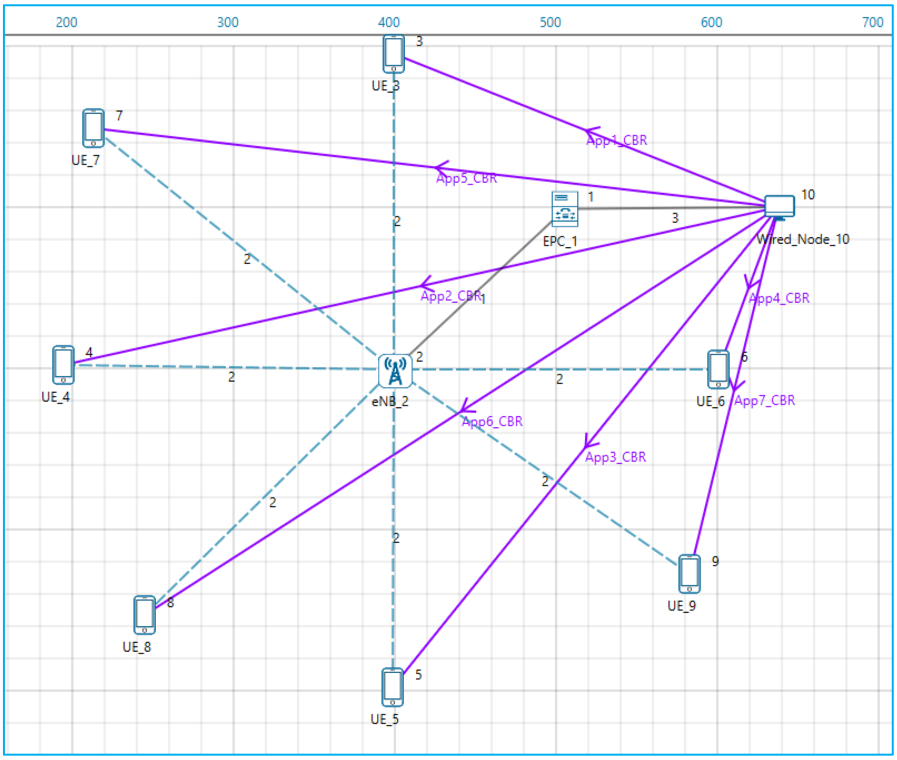
$ENB_ID = 2 UE_ID = 3 PL_dB = 50

$ENB_ID = 2 UE_ID = 6 PL_dB = 50

First line represents the number of UEs (whose path loss value needs to be changed). In the above sample, the numbers of UEs are 7, while the UEs which will be impacted by obstacle losses are 2. The second line represents UE id and the path loss exponent of the gNB-UE link and so on.

Settings to be done to create the network scenario:

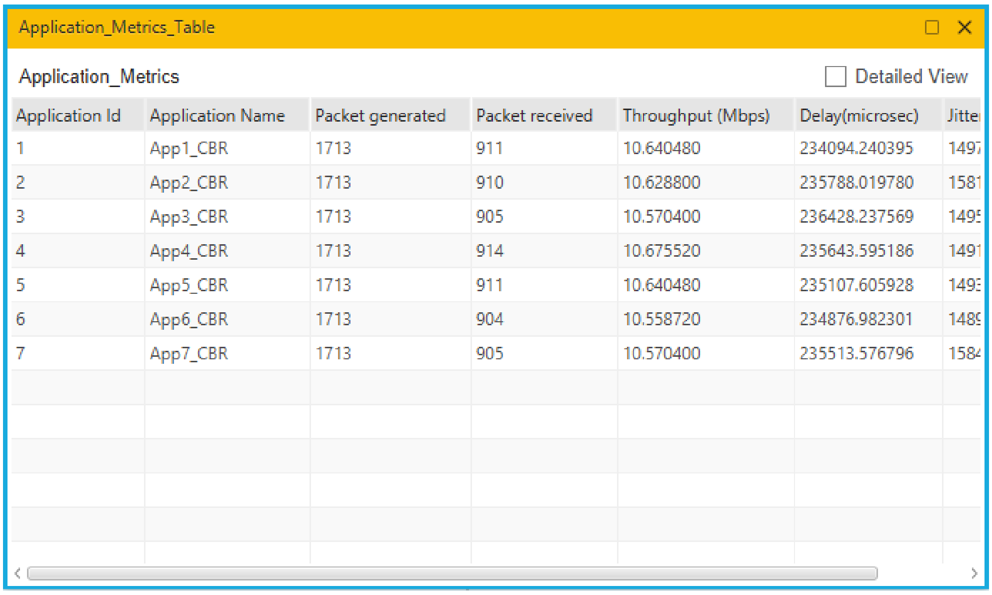
- Click and drop 1EPC, 1 wired node, 1eNB and 7UEs as per the below screenshot



- Create applications from wired node to all UEs with packet size 1460Bytes and Inter arrival Time 584µs.
- Set channel characteristics as Path loss only, LOS_Mode as USER_DEFINED, and LOS_Probability as 1.

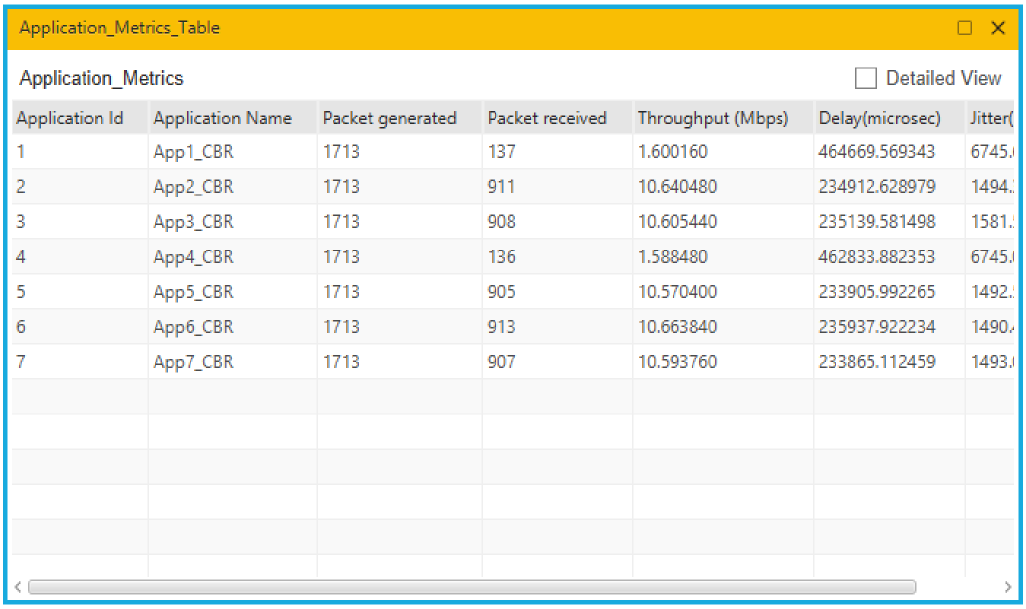
Results:

Without obstacles:



Application Id	Application Name	Packet generated	Packet received	Throughput (Mbps)	Delay(microsec)	Jitte
1	App1_CBR	1713	911	10.640480	234094.240395	1497
2	App2_CBR	1713	910	10.628800	235788.019780	1581
3	App3_CBR	1713	905	10.570400	236428.237569	1495
4	App4_CBR	1713	914	10.675520	235643.595186	1491
5	App5_CBR	1713	911	10.640480	235107.605928	1493
6	App6_CBR	1713	904	10.558720	234876.982301	1489
7	App7_CBR	1713	905	10.570400	235513.576796	1584

After simulation, note down the throughputs available in the simulation results window and compare with the previous results (Without Obstacles between UEs and eNB). Users can observe the change in throughputs



Application Id	Application Name	Packet generated	Packet received	Throughput (Mbps)	Delay(microsec)	Jitter(
1	App1_CBR	1713	137	1.600160	464669.569343	6745.
2	App2_CBR	1713	911	10.640480	234912.628979	1494.
3	App3_CBR	1713	908	10.605440	235139.581498	1581.
4	App4_CBR	1713	136	1.588480	462833.882353	6745.
5	App5_CBR	1713	905	10.570400	233905.992265	1492.
6	App6_CBR	1713	913	10.663840	235937.922234	1490.
7	App7_CBR	1713	907	10.593760	233865.112459	1493.

Comparison Table

Application_Id	Throughput (Mbps) Without_obstacle_loss	Throughput (Mbps) With_obstacle_loss
1	10.64	1.60
2	10.63	10.64
3	10.57	10.61

4	10.68	1.59
5	10.64	10.57
6	10.56	10.66
7	10.57	10.59

Table 1: Shows the variation in throughput with and without obstacle losses for UE2 and UE5, running App1 and App4