Modelling Obstacles between UEs and eNB in NetSim LTE

Software: 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 to simulate

- Open the Source codes in Visual Studio by going to Your work-> Workspace Options and Clicking on Open code button in NetSim Home Screen window
- Right click on Solution in Solution Explorer and select rebuild solution.

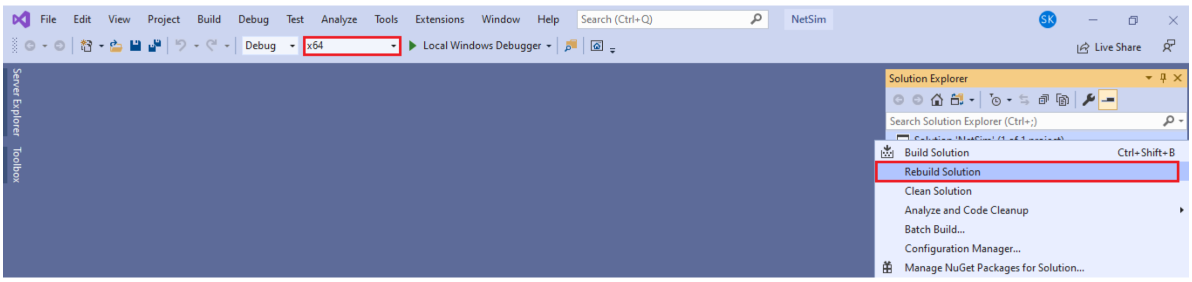


Figure 1: Screen shot of NetSim project source code in Visual Studio

- Upon rebuilding, libLTE.dll will get created in the bin_x86/ bin_x64 folder.

Example

- The **WorkSpace_MODELLING_OBSTACLES** comes with a sample network configuration that are already saved. To open this example, go to Your work in the Home screen of NetSim and click on the **MODELING_OBSTACLES_LTE_Example** from the list of experiments.
- Click and drop 1EPC, 1 wired node, 1eNB and 7UEs as per the below.

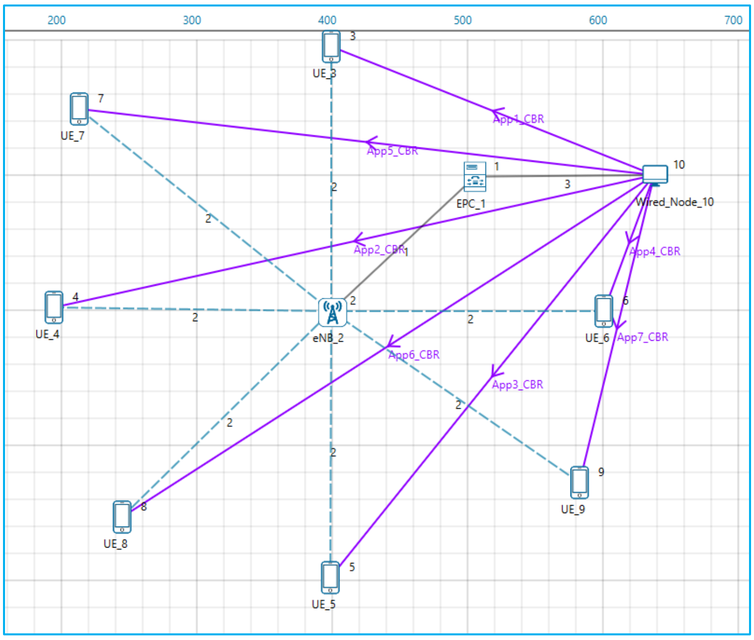


Figure 2: Modeling obstacles network scenario

- 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.
- After simulation, note down the throughputs available in the metrics window.

Results and discussion

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

- **Without obstacles**

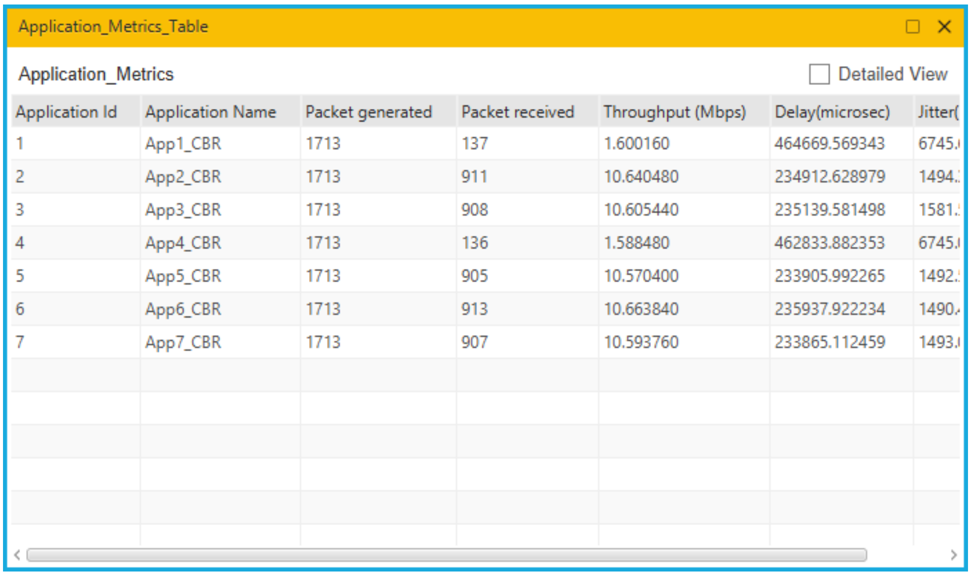
Application_Metrics_Table

Application_Metrics Detailed View

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

Figure 3: Throughputs for Without obstacles

- **With obstacles**



Application_Metrics_Table

Application_Metrics ☐ Detailed View

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.

Figure 4: Throughputs for with obstacles

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

Appendix: NetSim source code modifications

Changes to fn_read_pathloss_file(), in LTENR_PHY.c, within LTE_NR project

---

/* This code is used to generate obstacle_loss.txt */

```
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);

}

}
}
```

Changes to fn_NetSim_LTENR_PHY_Init() in LTENR_PHY.c, within LTE_NR project

---

/* This code is used to read pathloss file */

```
#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();
}
```

Changes to LTENR_PHY_calculateSpectralEfficiency() in LTENR_PHY.c, within LTE_NR project

---

```
/* Call propagation model */

    //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;
    for (UINT i = 0; i < info->downlink.layerCount; i++)
    {
    while (data)
    {
    if (info->ueId == data->UE_ID && info->gnbId == data->GNB_ID)
    {
    info->downlink.rxPower_dbm[i] = info->downlink.txPower_dbm - info->dTotalLoss +
    LTENR_BeamForming_Downlink_GetValue(info,i) - data->PL_dB;
    flag = true;
    }

    data = PATHLOSS_DATA_NEXT(data);
    }
    if (!flag)
    {
    info->downlink.rxPower_dbm[i] = info->downlink.txPower_dbm - info->dTotalLoss +
    LTENR_BeamForming_Downlink_GetValue(info, i);
    }
```

Changes to LTENR_PHY_GetDownlinkSpectralEfficiency() in LTENR_PHY.c, within LTE_NR project

---

```
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(info,layerId) - data->PL_dB;
flag = true;
}
```

```
data = PATHLOSS_DATA_NEXT(data);
}
if (!flag)
{
info->downlink.rxPower_dbm[layerId] = info->downlink.txPower_dbm - info->dTotalLoss +
LTENR_BeamForming_Downlink_GetValue(info, layerId);
}
```

Changes to LTENR_PHY_GetUplinkSpectralEfficienc y()in LTENR_PHY.c, within LTE_NR project

---

```
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,layerId)- data->PL_dB;
flag = true;
}

data = PATHLOSS_DATA_NEXT(data);
}
if (!flag)
{
info->uplink.rxPower_dbm[layerId] = info->uplink.txPower_dbm - info->dTotalLoss +
LTENR_BeamForming_Uplink_GetValue(info, layerId);
}
```

---

the following lines added starting of LTENR_PHY.c file.

```
#pragma region HEADER_FILES
#include "stdafx.h"
#include "LTENR_MAC.h"
#include "LTENR_PHY.h"
#pragma endregion

#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)))
```

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.