Modelling Obstacles between UEs and eNB in NetSim LTE

Software Recommended: NetSim Standard v12. (32-bit/64-bit), Visual Studio 2017/2019

Follow the instructions specified in the following link to clone/download the project folder from GitHub using Visual Studio:
https://tetcos.freshdesk.com/support/solutions/articles/14000099351-how-to-clone-netsim-file-exchange-project-repositories-from-github-

Other tools such as GitHub Desktop, SVN Client, Sourcetree, Git from the command line, or any client you like to clone the Git repository.

Note: It is recommended not to download the project as an archive (compressed zip) to avoid incompatibility while importing workspaces into NetSim.

Secure URL for the GitHub repository:

https://github.com/NetSim-TETCOS/MODELING_OBSTACLES_in_LTE_v12.0.git

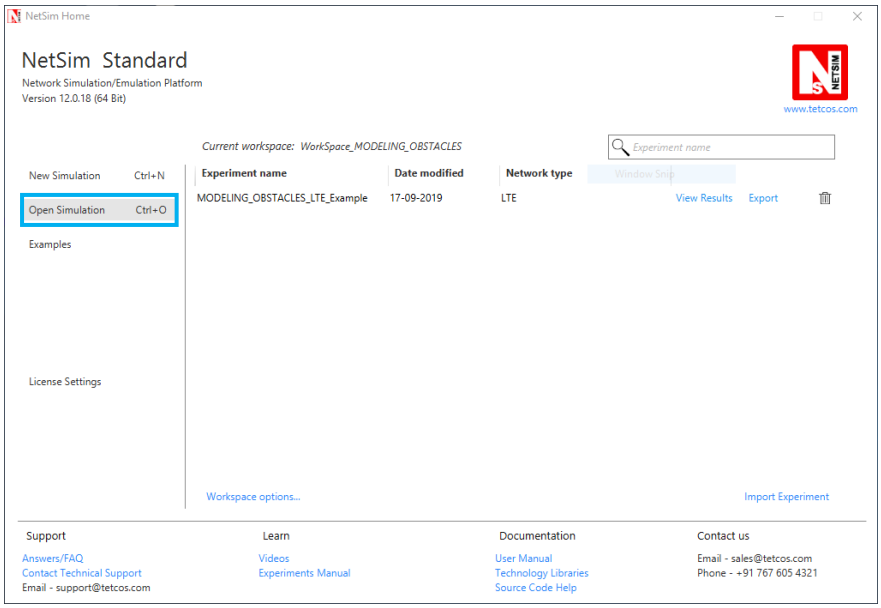
Users can model obstacles and varied channel conditions between the eNB and the connected UEs, by modifying the underlying LTE code.

This is required because, as of **NetSim v11.0**, in the GUI, the wireless link (between one eNB and the connected UEs) properties are same i.e. if we change in one link it reflects in all the other links of UEs connected to same eNB.

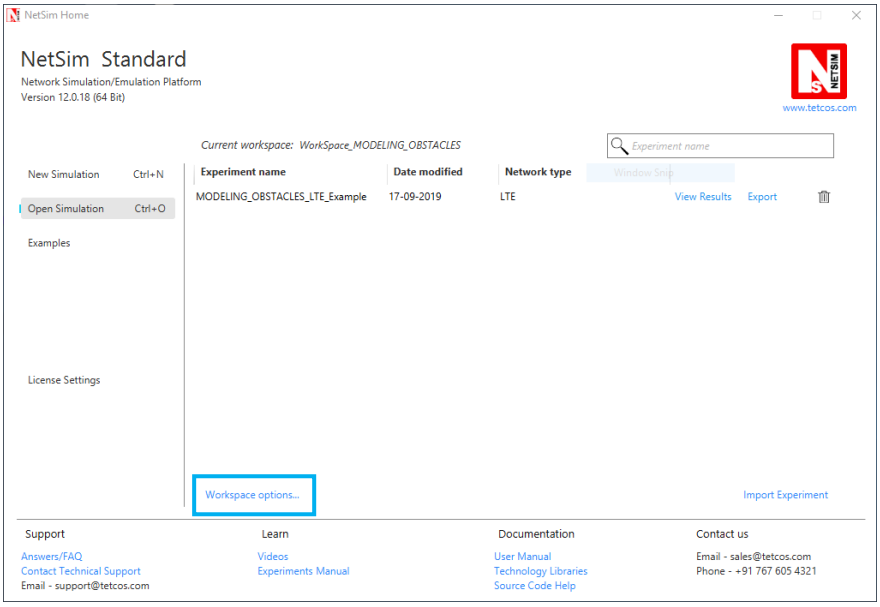
Obstacles are modelled by adding an attenuation (in dB) value. Varying channel conditions are modelled by changing the pathloss exponent between the eNB and connected UEs.

Steps:

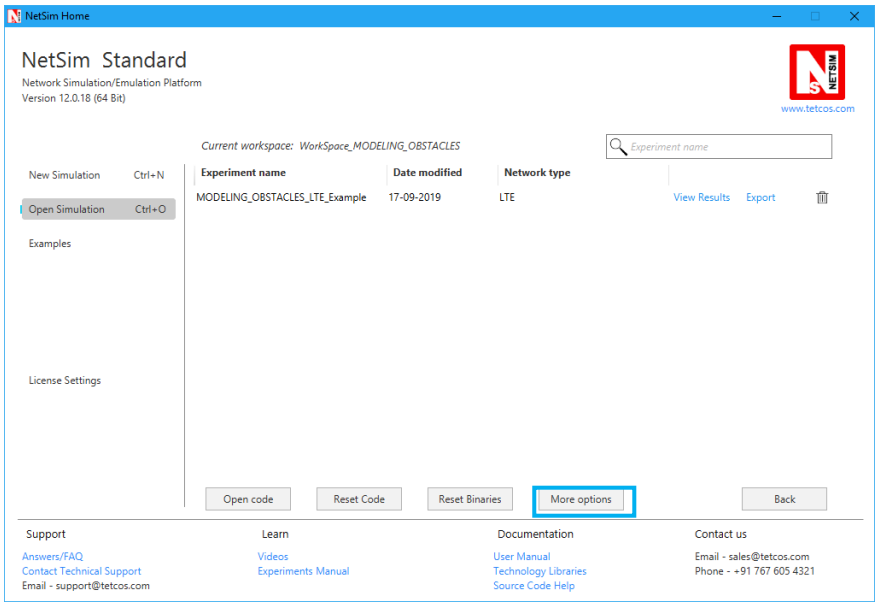
- After you unzip the downloaded project folder, Open NetSim Home Page click on **Open Simulation** option,



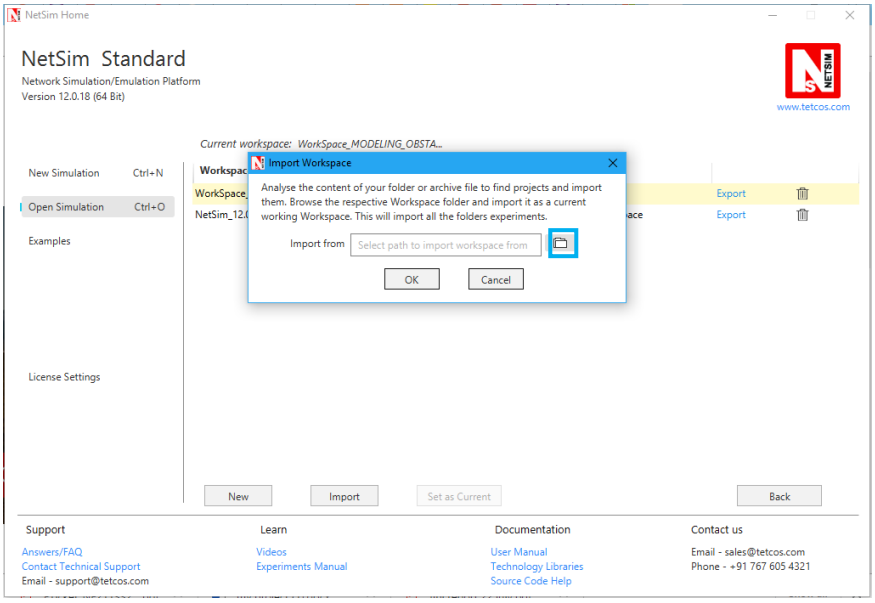
- Click on **Workspace options**



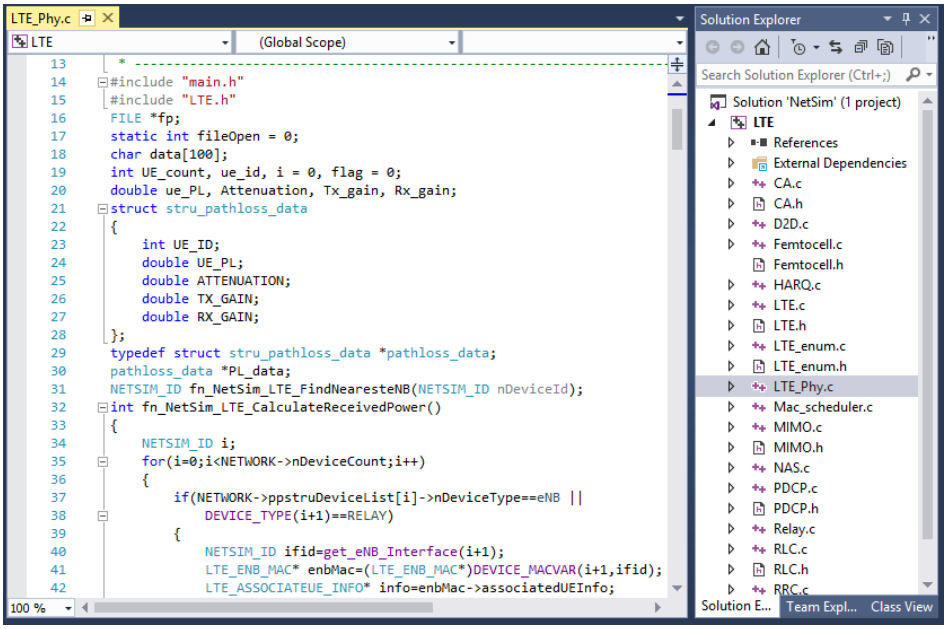
- Click on **More Options,**



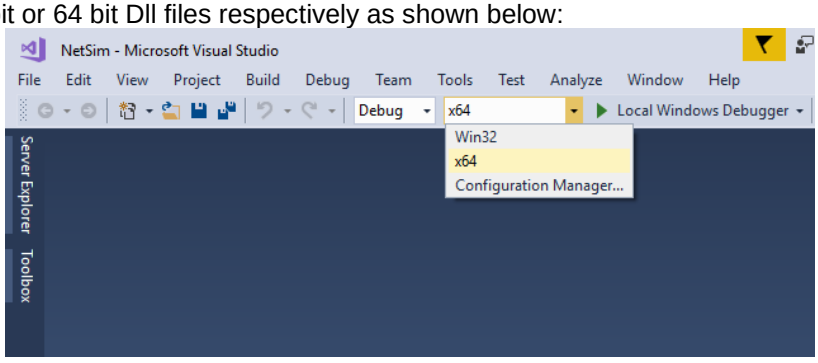
- Click on **Import**, browse the extracted folder path and go into WorkSpace_MODELING_OBSTACLES directory. Click on Select folder and then on **OK.**



- Go to home page, Click on **Open Simulation → Workspace options → Open code**



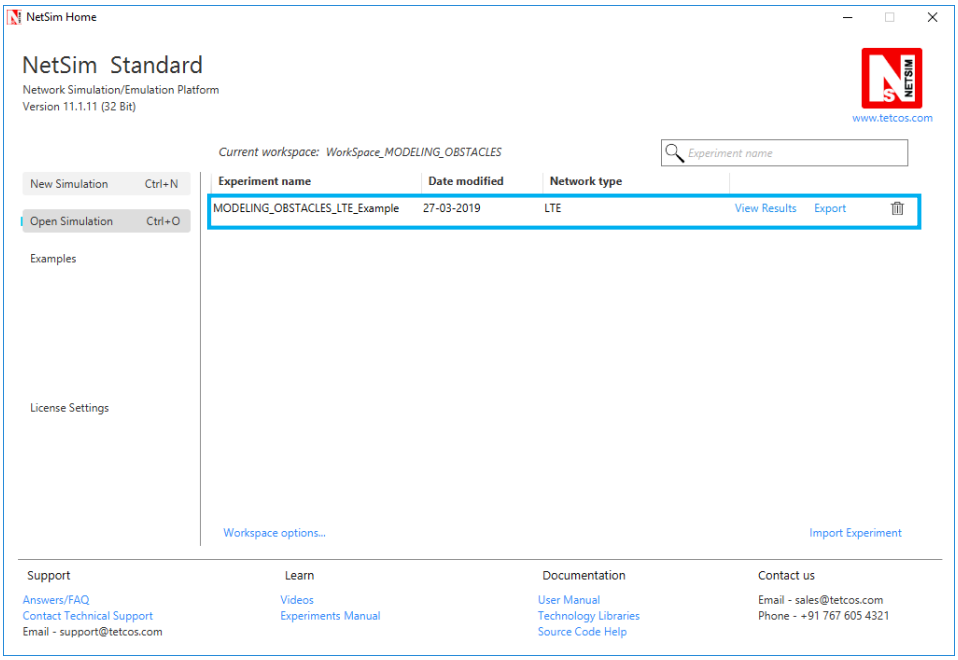
- Based on whether you are using NetSim 32 bit or 64 bit setup you can configure Visual studio to build 32 bit or 64 bit Dll files respectively as shown below:



1. Right click on Solution in Solution Explorer and select rebuild solution

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

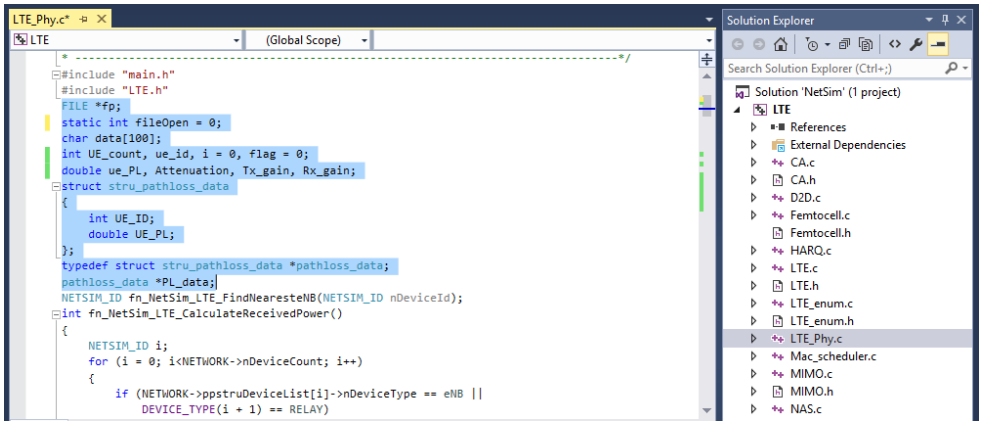
3. Go to NetSim home page, click on **Open Simulation**, Click on **MODELING_OBSTACLES_LTE_Experiment**.



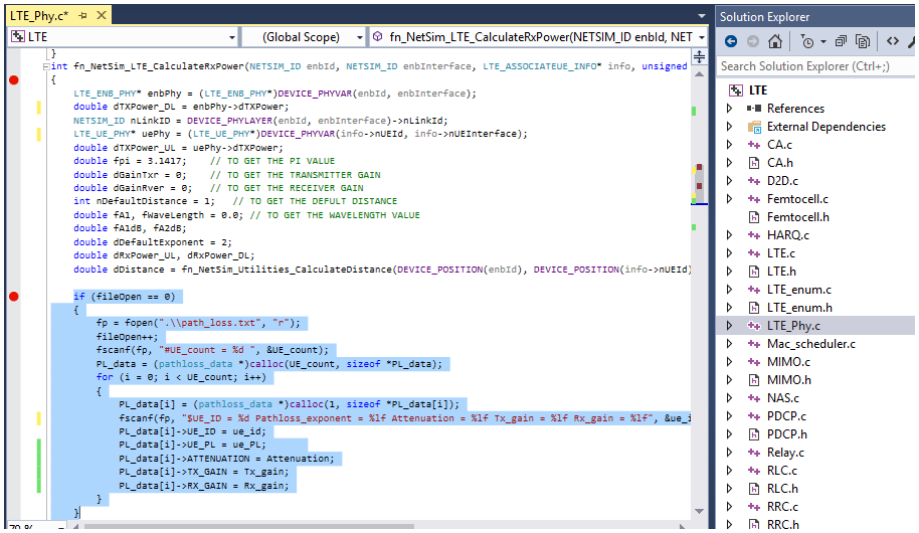
4. After simulation, note down the throughputs available in the metrics window.

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

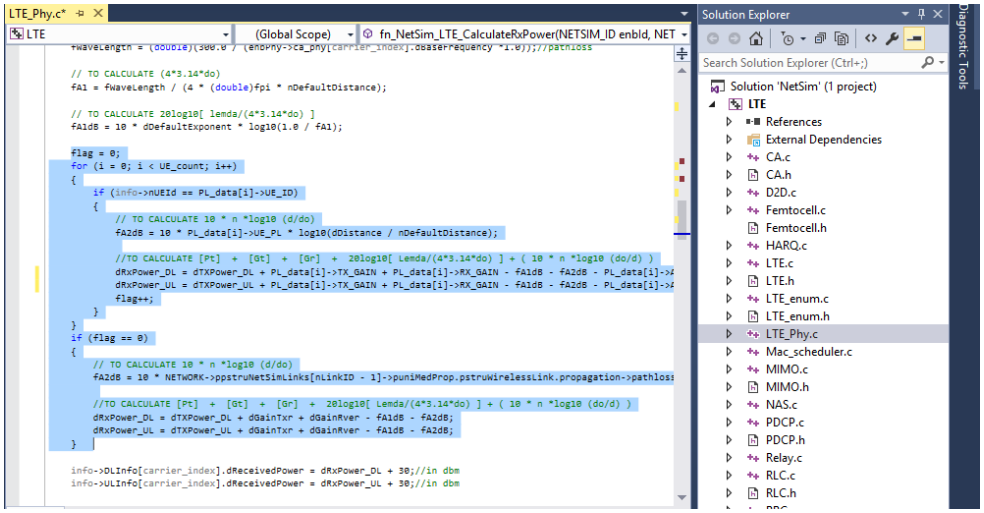
We have added the following lines of code in LTE_PHY.c file present inside LTE project as shown below:



To read the file content, we have added the following lines of code in fn_NetSim_LTE_CalculateRxPower() present in LTE_PHY.c file.



And then the following lines in fn_NetSim_LTE_CalculateRxPower() present in LTE_PHY.c file.



Create a path_loss.txt file and paste it in the install directory of NetSim would look something like **"C:\Program Files\NetSim Standard\bin"** and the file format should be

#UE_count = 2

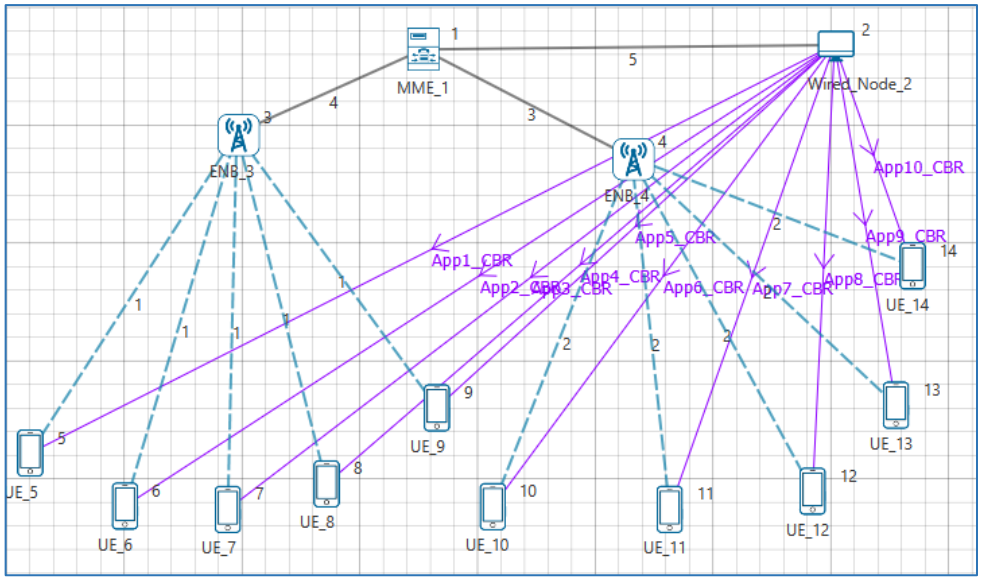
$UE_ID = 13 Pathloss = 4.7 Attenuation = 2 Tx_gain = 2 Rx_gain = 2

$UE_ID = 5 Pathloss = 4.7 Attenuation = 2 Tx_gain = 2 Rx_gain = 2

First line represents the number of UEs (whose path loss value needs to be changed). In the above sample, the numbers of UEs are 5. Second line represents UE id and the path loss exponent of particular UE link and so on.

Settings to be done to create the network scenario:

- Click and drop 1MME, 1 wired node, 2eNBs and 10UEs as per the below screenshot



- Create applications from wired node to all UEs with packet size 1460Bytes and Inter arrival Time 1168µs.
- Set channel characteristics as Path loss only, Path loss model as LOG DISTANCE and Path loss exponent to 3.5.

Results:

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_metrics Detailed View

Application Id	*Throughput Plot*	Application Name	Packet generated	Packet received	Throughput (Mbps)	Delay(microsec)	Jitter(microsec)
1	*Application_throughput_plot*	App1_CBR	42809	1103	0.257661	5456092.453309	13652.166969
2	*Application_throughput_plot*	App2_CBR	42809	216	0.050458	2862159.222222	29116.018605
3	*Application_throughput_plot*	App3_CBR	42809	2958	0.690989	11378458.258283	12181.381130
4	*Application_throughput_plot*	App4_CBR	42809	2041	0.476778	16345695.525723	15292.964706
5	*Application_throughput_plot*	App5_CBR	42809	4437	1.036483	4895779.391481	6880.842200
6	*Application_throughput_plot*	App6_CBR	42809	2177	0.508547	11480976.848875	20727.500000
7	*Application_throughput_plot*	App7_CBR	42809	3586	0.837690	427246.224205	3156.248815
8	*Application_throughput_plot*	App8_CBR	42809	1515	0.353904	32811514.508251	33101.812417
9	*Application_throughput_plot*	App9_CBR	42809	921	0.215146	3519287.800217	46575.260870
10	*Application_throughput_plot*	App10_CBR	42809	4006	0.935802	2926593.853220	5091.128090