Getting Started With TTM4133 Labs

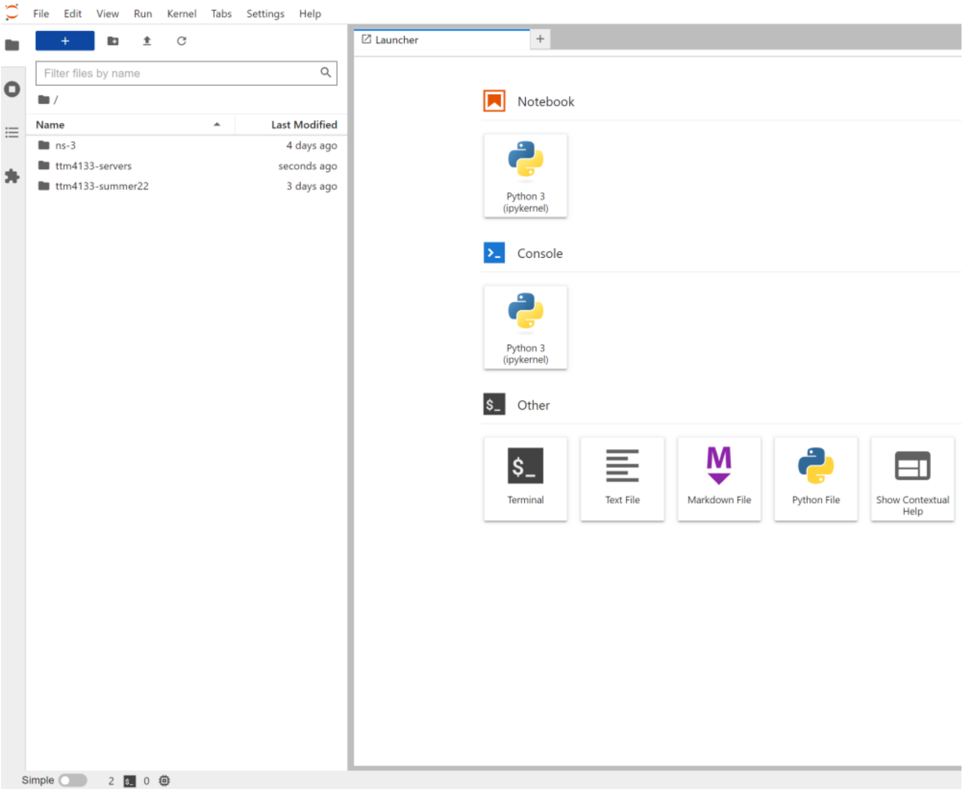
The following steps must be taken to create the environment for the testing of labs:

1) *Accessing the VM*

VMs for each group are created using NTNU's Openstack Infrastructure. They have been given public IPs, but the access is restricted for these IPs only to the NTNU's private network (hence NTNU's VPN is needed). The VM needs to be accessed using port numbers **9000** and **3000** for accessing the jupyter and react app interface. Use the following link template for doing this:

- Jupyter Interface

http://vm-public-ip:9000/

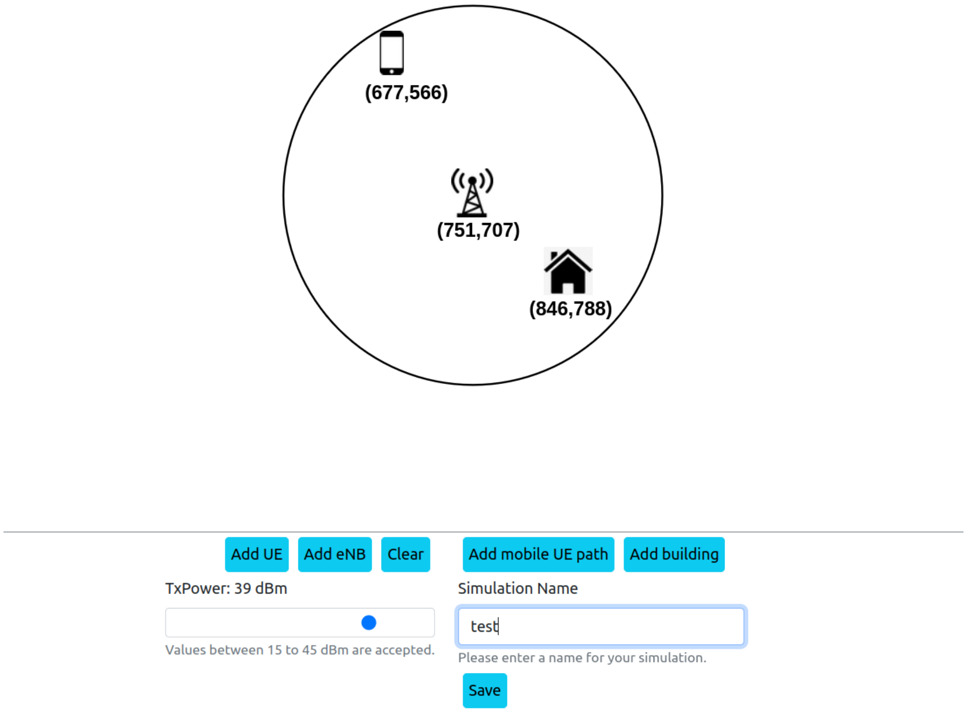


Name	Last Modified
ns-3	4 days ago
ttm4133-servers	seconds ago
ttm4133-summer22	3 days ago

Note:

- React Interface

http://vm-public-ip:3000/ [Required only for topology design-based tasks]



TxPower: 39 dBm

Values between 15 to 45 dBm are accepted.

Simulation Name

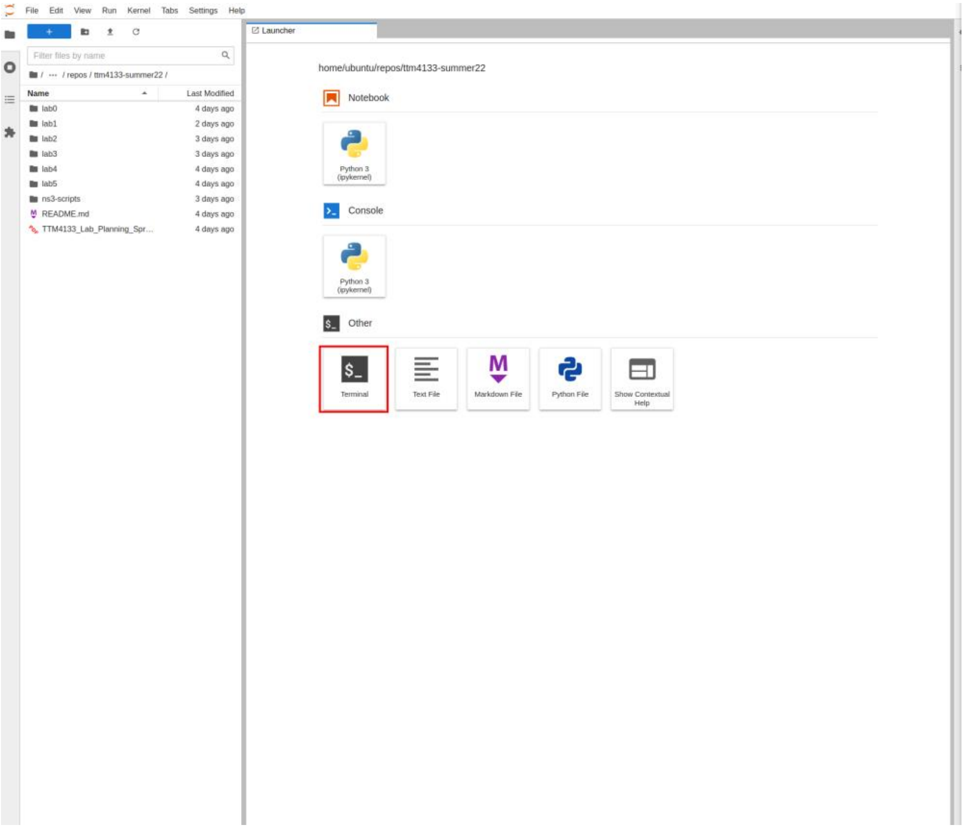
test

Please enter a name for your simulation.

2) *Cloning the files*

The following steps allow the synchronization of latest files from the lab repository:

Step 1: Getting the terminal access: Before we clone the files, we need to open the terminal tab in jupyter interface. In the jupyter interface, Go to *File -> New Launcher -> Other (Terminal)*



Name	Last Modified
lab0	4 days ago
lab1	2 days ago
lab2	3 days ago
lab3	3 days ago
lab4	4 days ago
lab5	4 days ago
ns3-scripts	3 days ago
README.md	4 days ago
TTM4133_Lab_Planning_Spr…	4 days ago

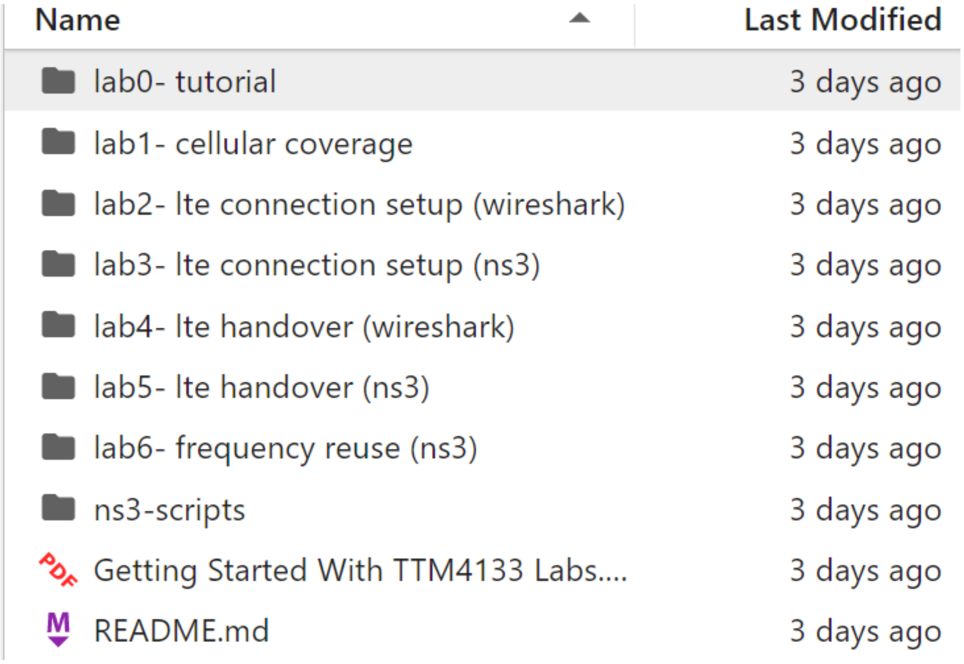
<u>Step 2:</u> <u>Changing the path:</u> The path for cloning must be ***/home/repos/***

- **cd /home/ubuntu/repos/**

<u>Step 3:</u> <u>Getting files from github:</u> The required files for different files are available at the following repository: https://github.com/kashifme224/ttm4133-spring23.

- We can use the following command from the terminal tab: **git clone https://github.com/kashifme224/ttm4133-spring23**

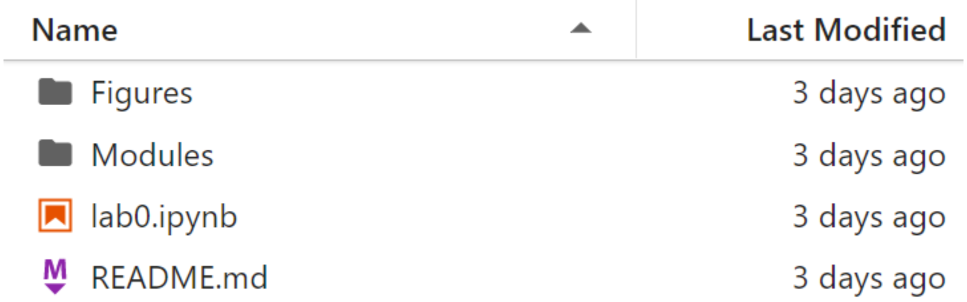
Expected cloned files in the main repository will be the <u>lab folders</u> and <u>ns3-scripts</u> folder:



Name	Last Modified
lab0- tutorial	3 days ago
lab1- cellular coverage	3 days ago
lab2- lte connection setup (wireshark)	3 days ago
lab3- lte connection setup (ns3)	3 days ago
lab4- lte handover (wireshark)	3 days ago
lab5- lte handover (ns3)	3 days ago
lab6- frequency reuse (ns3)	3 days ago
ns3-scripts	3 days ago
Getting Started With TTM4133 Labs....	3 days ago
README.md	3 days ago

- **labx-name** has the jupyter notebook file and the script files for data processing
- **Ns3-scripts** hold the ns3 script files for the labs.

Each lab folder consists of:



Name	Last Modified
Figures	3 days ago
Modules	3 days ago
lab0.ipynb	3 days ago
README.md	3 days ago

- **labx.ipynb** has the jupyter notebook file and the script files for data processing
- **Figures** have the associated graphics used in the notebooks
- **Modules** has the associated python code for performing different simulation and data processing tasks in the notebook.

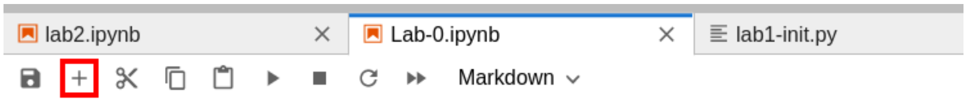
3) *Placement of files*

The cloned files have **ns3-scripts** that must be moved inside the ns3 distribution already provided in the VMs. Copy the script for the respective lab titled 'labx.cc' (where, x is the lab number) to ***/repos/ns-3/scratch.*** The rest of the files can be under the same location as they are cloned into. Execute the following command to do this from the cloned repository:

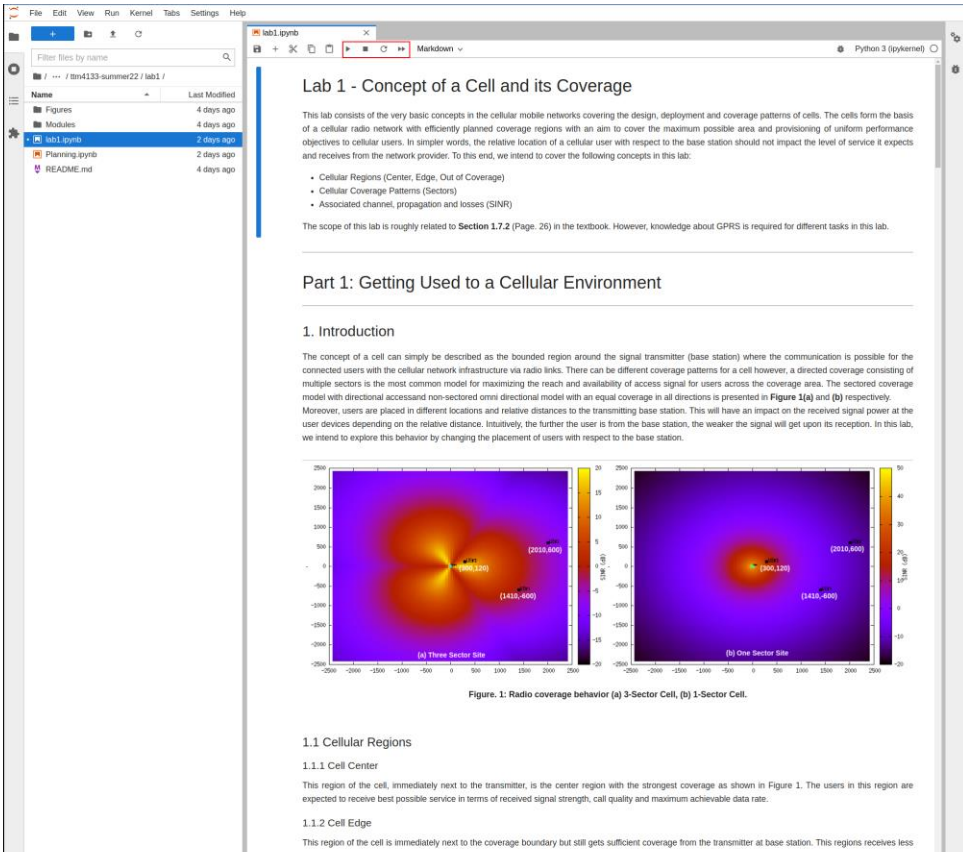
- *cp /home/ubuntu/repos/ttm4133-spring23/ns3-scripts/labx.cc /home/ubuntu/repos/ns-3/scratch*

4) *Testing the Labs*

There are two types of cells in jupyter. One for executable commands and code and the other for simple text (based on markdown language). To create a markdown cell, one must create a new cell using the plus (+) button from the top bar of the jupyter notebook.



To run the jupyter notebook for a lab, just access the jupyter interface via a browser and start running the cells using the play button as highlighted below:



Lab 1 - Concept of a Cell and its Coverage

This lab consists of the very basic concepts in the cellular mobile networks covering the design, deployment and coverage patterns of cells. The cells form the basis of a cellular radio network with efficiently planned coverage regions with an aim to cover the maximum possible area and provisioning of uniform performance objectives to cellular users. In simpler words, the relative location of a cellular user with respect to the base station should not impact the level of service it expects and receives from the network provider. To this end, we intend to cover the following concepts in this lab:

- Cellular Regions (Center, Edge, Out of Coverage)
- Cellular Coverage Patterns (Sectors)
- Associated channel, propagation and losses (SINR)

The scope of this lab is roughly related to **Section 1.7.2** (Page. 26) in the textbook. However, knowledge about GPRS is required for different tasks in this lab.

Part 1: Getting Used to a Cellular Environment

1. Introduction

The concept of a cell can simply be described as the bounded region around the signal transmitter (base station) where the communication is possible for the connected users with the cellular network infrastructure via radio links. There can be different coverage patterns for a cell however, a directed coverage consisting of multiple sectors is the most common model for maximizing the reach and availability of access signal for users across the coverage area. The sectored coverage model with directional accessand non-sectored omni directional model with an equal coverage in all directions is presented in **Figure 1(a)** and **(b)** respectively. Moreover, users are placed in different locations and relative distances to the transmitting base station. This will have an impact on the received signal power at the user devices depending on the relative distance. Intuitively, the further the user is from the base station, the weaker the signal will get upon its reception. In this lab, we intend to explore this behavior by changing the placement of users with respect to the base station.

Figure. 1: Radio coverage behavior (a) 3-Sector Cell, (b) 1-Sector Cell.

1.1 Cellular Regions

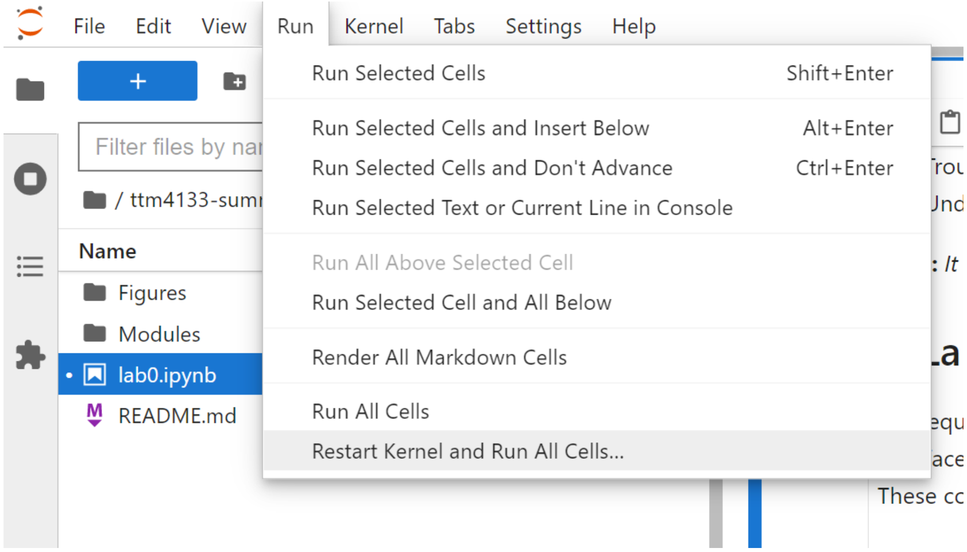
1.1.1 Cell Center

This region of the cell, immediately next to the transmitter, is the center region with the strongest coverage as shown in Figure 1. The users in this region are expected to receive best possible service in terms of received signal strength, call quality and maximum achievable data rate.

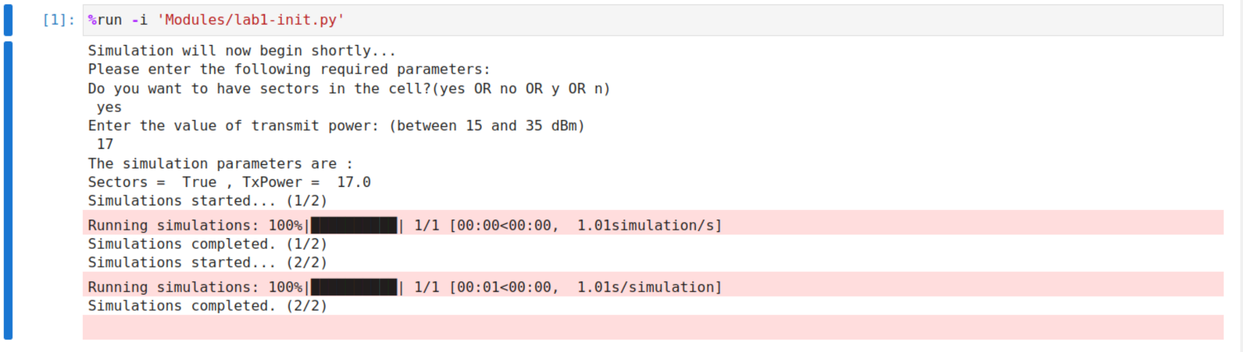
1.1.2 Cell Edge

This region of the cell is immediately next to the coverage boundary but still gets sufficient coverage from the transmitter at base station. This regions receives less

- The **play button** ▶ executes the selected cell and after completing the current cell selects the next cell. The cells can also be executed by selecting the cell first and using (Shift + Enter) key combination. Another option is the utilization of the taskbar option ’Run’ and the selecting the required run cell option.


File
Edit
View
Run
Kernel
Tabs
Settings
Help
Run Selected Cells Shift+Enter
Run Selected Cells and Insert Below Alt+Enter
Run Selected Cells and Don't Advance Ctrl+Enter
Run Selected Text or Current Line in Console
Run All Above Selected Cell
Run Selected Cell and All Below
Render All Markdown Cells
Run All Cells
Restart Kernel and Run All Cells...
Filter files by nar
/ ttm4133-sumr
Name
Figures
Modules
lab0.ipynb
README.md

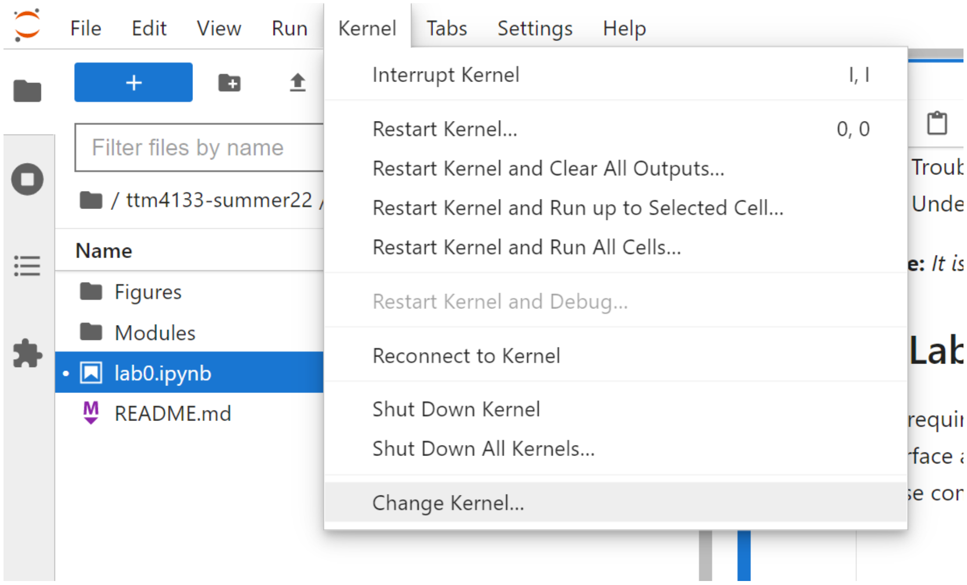
- The execution can continue cell wise throughout the notebook with any output displayed after the cell. A sample of the executed cell and output is shown as follows:



```
[1]: %run -i 'Modules/lab1-init.py'

     Simulation will now begin shortly...
     Please enter the following required parameters:
     Do you want to have sectors in the cell?(yes OR no OR y OR n)
      yes
     Enter the value of transmit power: (between 15 and 35 dBm)
      17
     The simulation parameters are :
     Sectors =  True , TxPower =  17.0
     Simulations started... (1/2)
     Running simulations: 100%|██████████| 1/1 [00:00<00:00,  1.01simulation/s]
     Simulations completed. (1/2)
     Simulations started... (2/2)
     Running simulations: 100%|██████████| 1/1 [00:01<00:00,  1.01s/simulation]
     Simulations completed. (2/2)
```

- The **stop button** ■ simply halts the execution.
- The **reset button** ↻ resets the distributed resources for the jupyter notebook and removes any variables. This is useful in case the notebook is freezing or not performing as expected. Another possible solution to refresh notebook kernel is to use the 'Kernel' tab from the taskbar and select the 'Restart Kernel' option as shown below:


File
Edit
View
Run
Kernel
Tabs
Settings
Help
+
Filter files by name
/ ttm4133-summer22 /
Name
Figures
Modules
lab0.ipynb
README.md
Interrupt Kernel I, I
Restart Kernel... 0, 0
Restart Kernel and Clear All Outputs...
Restart Kernel and Run up to Selected Cell...
Restart Kernel and Run All Cells...
Restart Kernel and Debug...
Reconnect to Kernel
Shut Down Kernel
Shut Down All Kernels...
Change Kernel...
Troub
Unde
e: It is
Lab
requi
face a
se cor