Channel notching in Cognitive Radio Networks

Software Recommended: NetSim Standard v12.0 (32/64 bit), Visual Studio 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/Channel_Notching_in_CR_Networks_v12.0.git

Introduction:

In Cognitive Radio networks, the secondary user (CR CPE) actively senses for the presence of the primary user (Incumbent). If the CR_CPE detects the primary user, then UCS Notifications will be sent by the secondary user to the base station. UCS notifications are generated at the end of the quiet period. Upon receiving the UCS notification BS checks for possible interference between Primary and secondary users. If interference is detected, secondary users vacate the channel and will be moved to a different vacant channel if available.

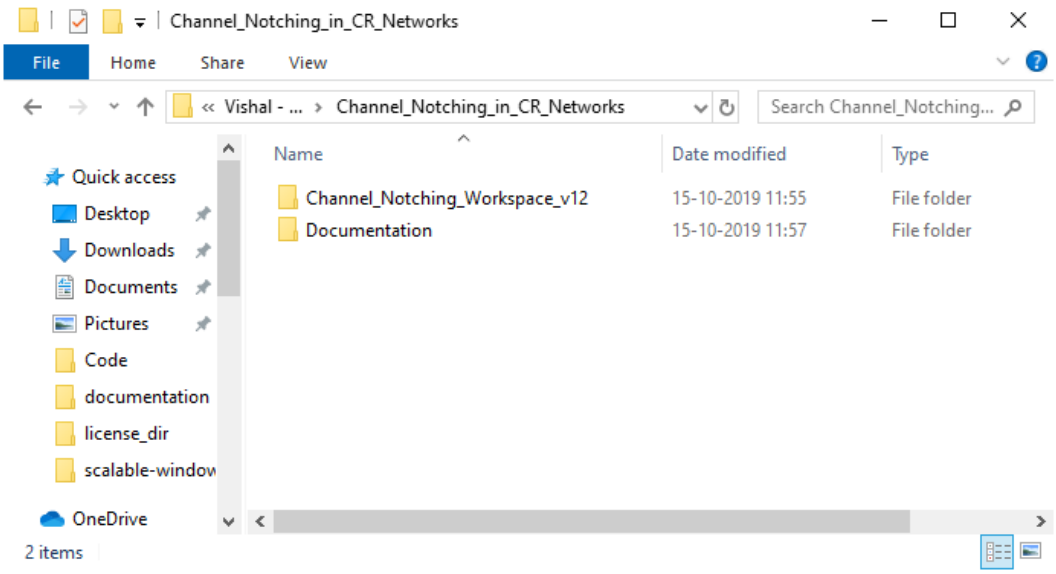
Channel Notching basically allows the primary and secondary users to co-exist in the same channel. This is achieved by allowing the secondary users to use the free sub channels which are not occupied by the primary users, as compared to the standard case (without channel notching), where the entire channel is blocked even if some of the sub channels of that channel are being used by the Incumbent.

Hence, in cases where there are limited available channels, using Channel Notching will help in achieving higher throughput and channel utilization.

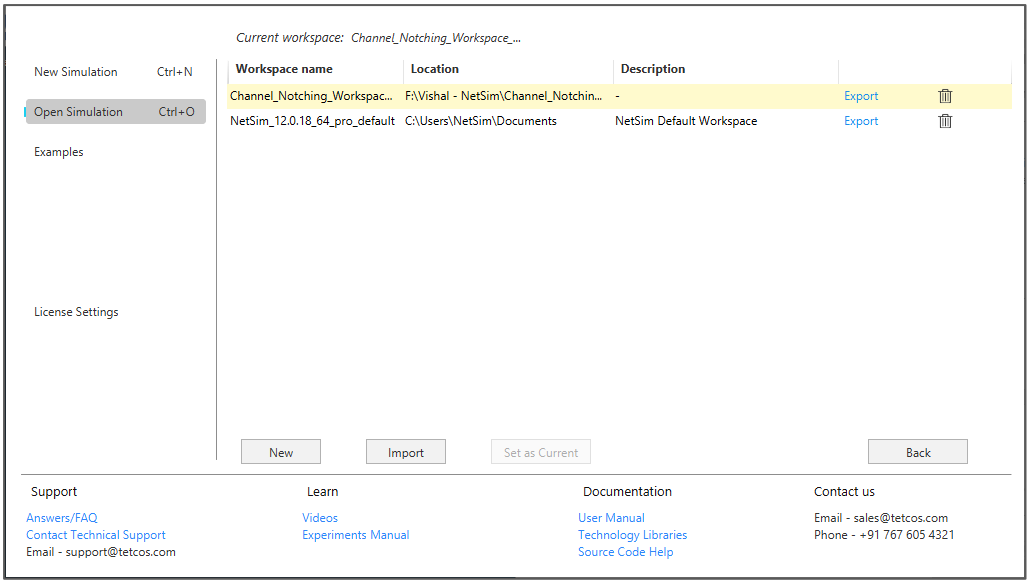
Note: Channel notching code will work only for a single Channel, single CR-CPE and for at-most one Incumbent.

Steps:

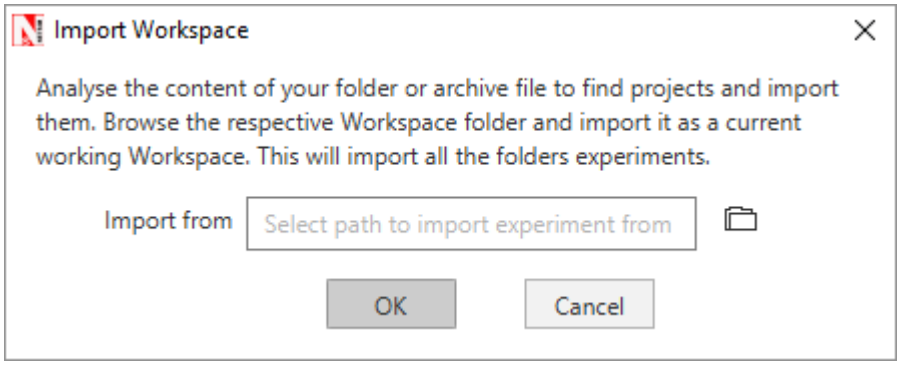
1. The downloaded project folder contains the folders Documentation and Channel Notching Workspace directory as shown below:



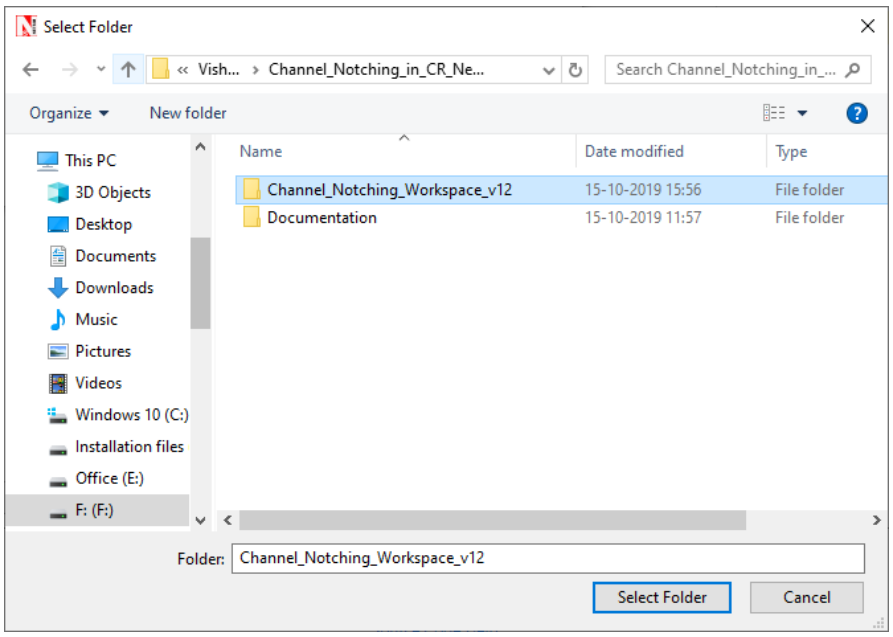
2. Import Channel_Notching_Workspace_v12 by going to Open Simulation->Workspace Options->More Options in NetSim Home window. Then select Import as shown below:



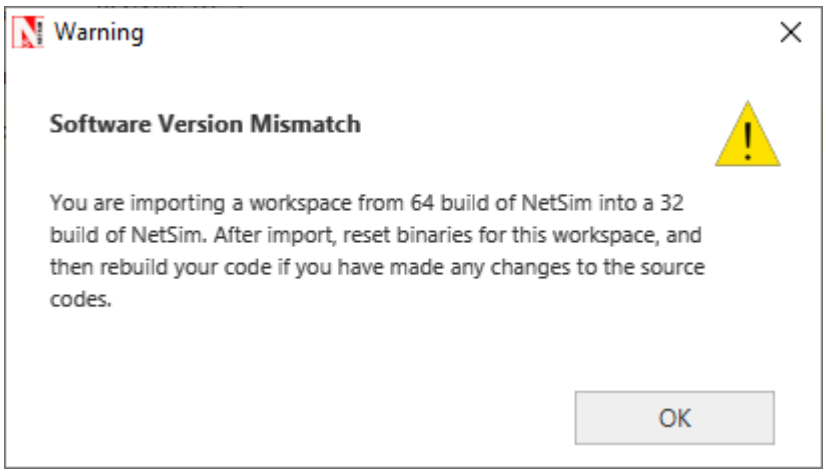
3. It displays a window where users need to give the path of the workspace folder and click on OK as shown below:



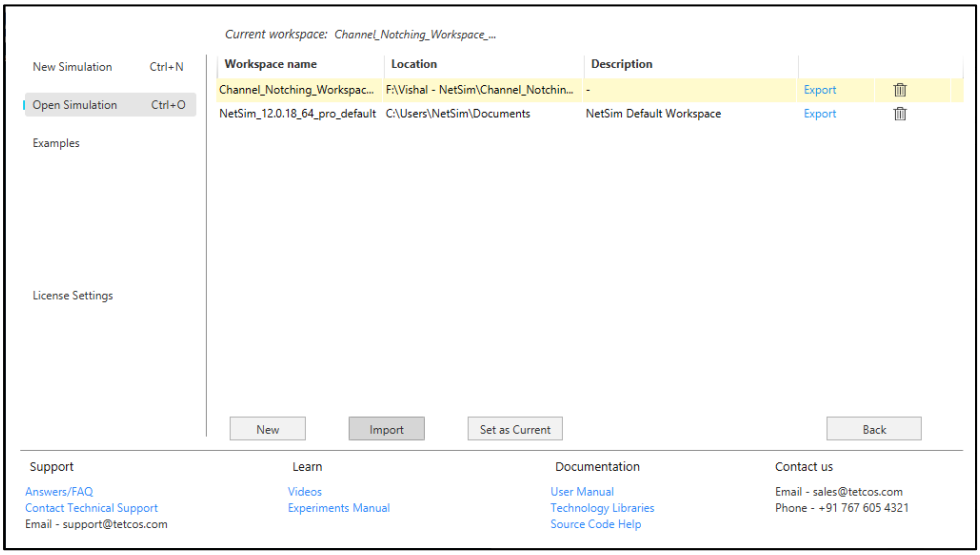
4. Browse to the Channel_Notching_Workspace_v12 folder and click on select folder as shown below:



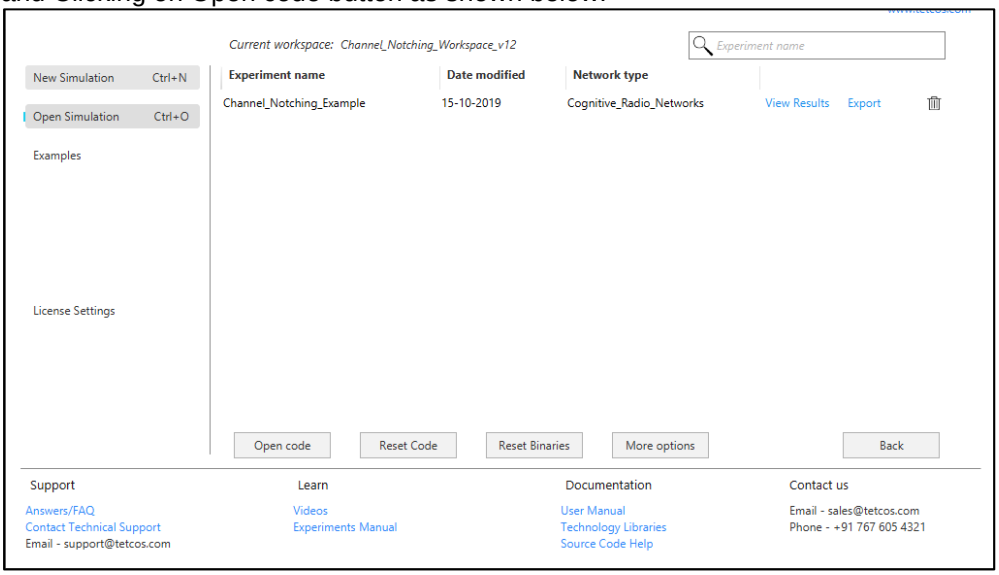
5. After this click on OK button in the Import Workspace window.
6. While importing the workspace, if the following warning message indicating Software Version Mismatch is displayed, you can ignore it and proceed.



7. The Imported workspace will be set as the current workspace automatically. To see the imported workspace, click on Open Simulation->Workspace Options->More Options as shown below:



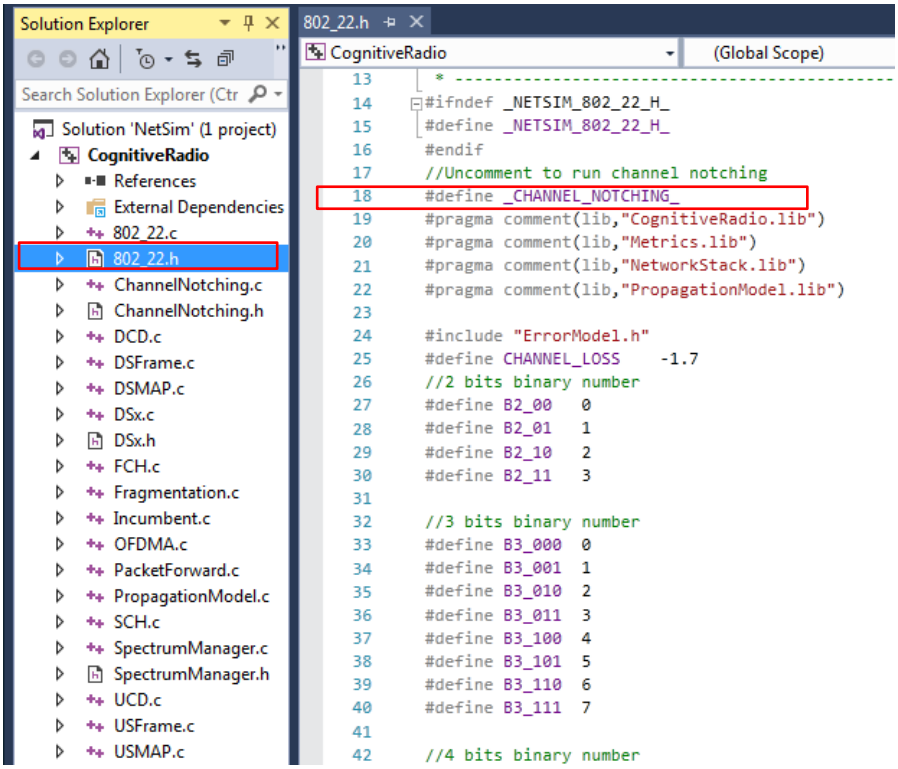
8. Open the Source codes in Visual Studio by going to Open Simulation-> Workspace Options and Clicking on Open code button as shown below:



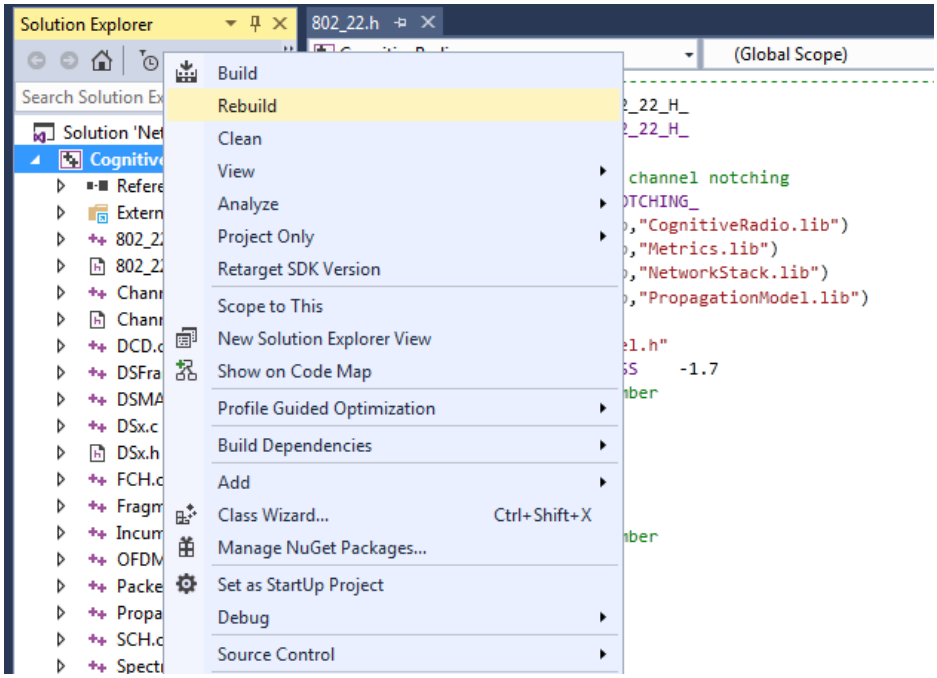
9. In the Solution Explorer, go to **Cognitive Radio → 802_22.h** and open it.

10. If you want to enable **Channel Notching**, uncomment (if commented)

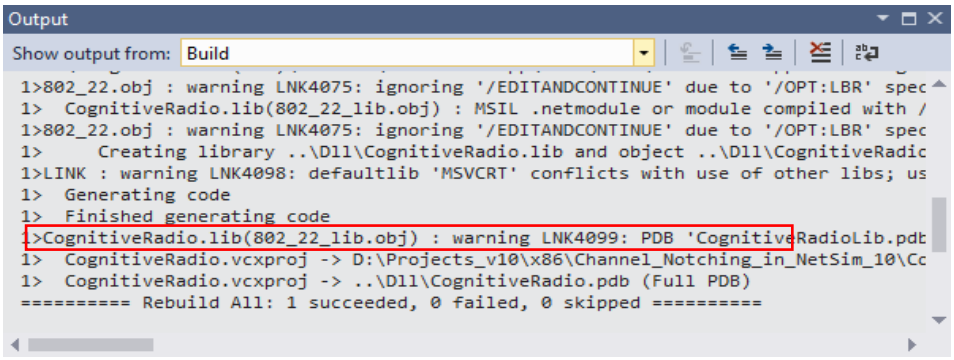
#define _CHANNEL_NOTCHING_



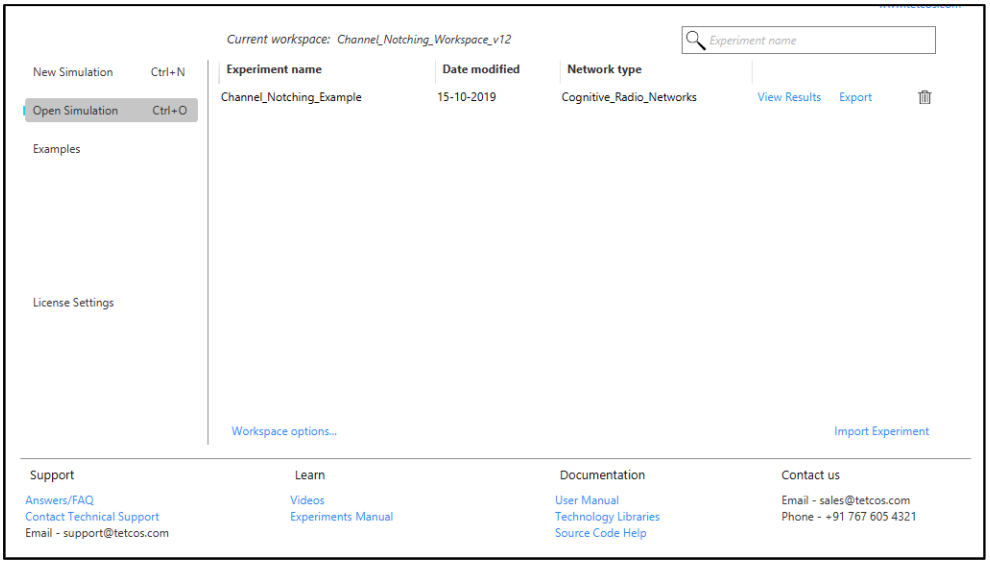
11. Right click on **Cognitive Radio p**roject→ **Rebuild**



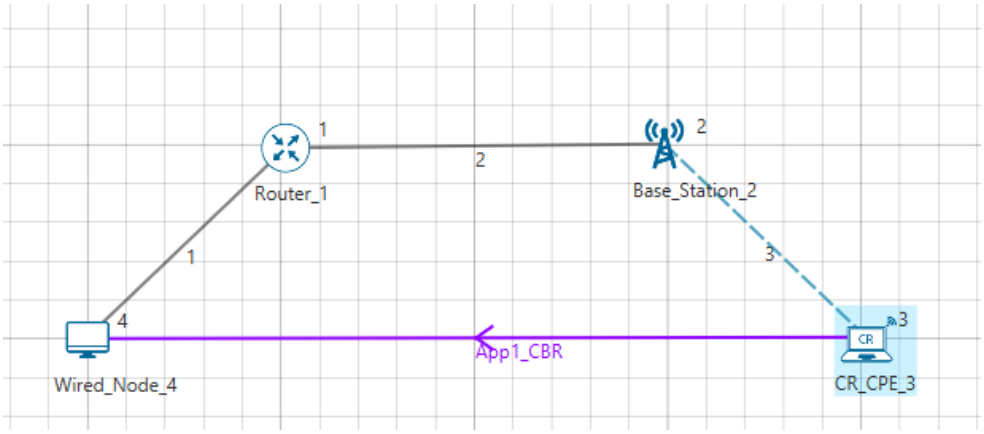
12. You should see a message in the **Output** window as shown in the following figure.



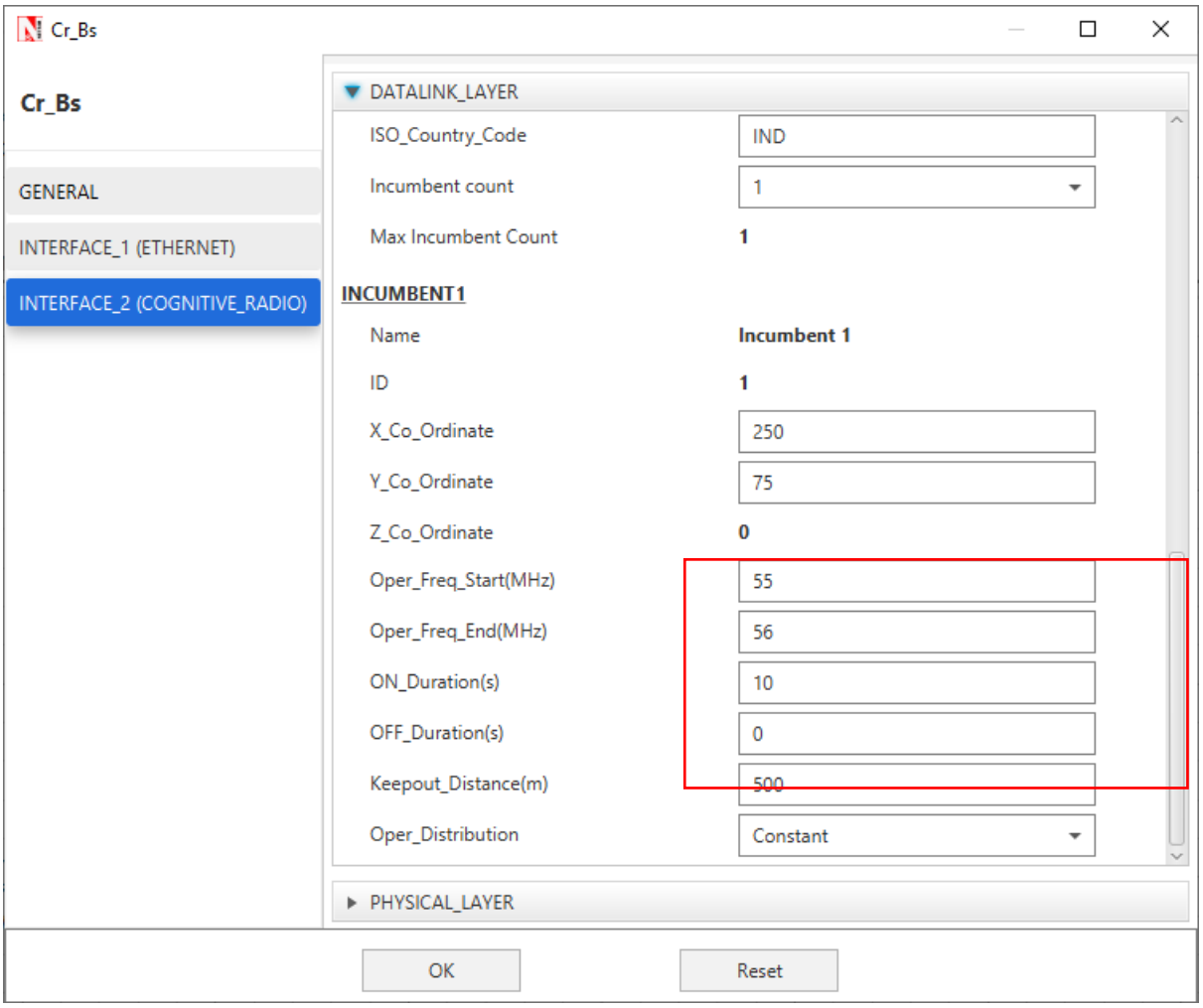
13. Then Channel_Notching_Workspace_v12 comes with a sample configuration that is already saved. To open this example, go to Open Simulation and click on the that is present under the list of experiments as shown below:



14. The scenario looks like



15. And set BS properties as



16. Run the scenario for both the cases: with channel notching and without it. The throughputs obtained will be **0.002949** and **0.000000** respectively