Channel notching in Cognitive Radio Networks

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

Project Download Link:
https://github.com/NetSim-TETCOS/CC_AODV_Project_v13.0/archive/refs/heads/main.zip

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.

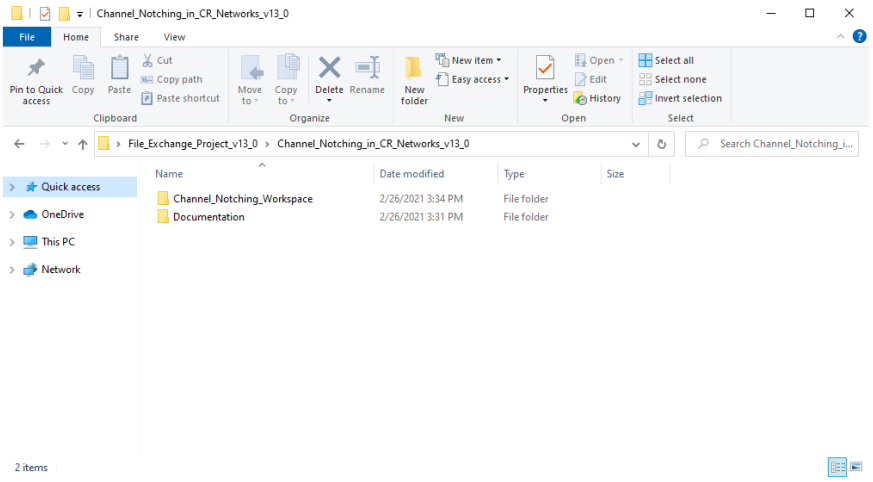
Channel Notching basically allows the primary and secondary users to co-exist in the same channel. This is achieved by allowing the secondary users to use the free sub channels which are not occupied by the primary users, as compared to the standard case (without channel notching), where the entire channel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

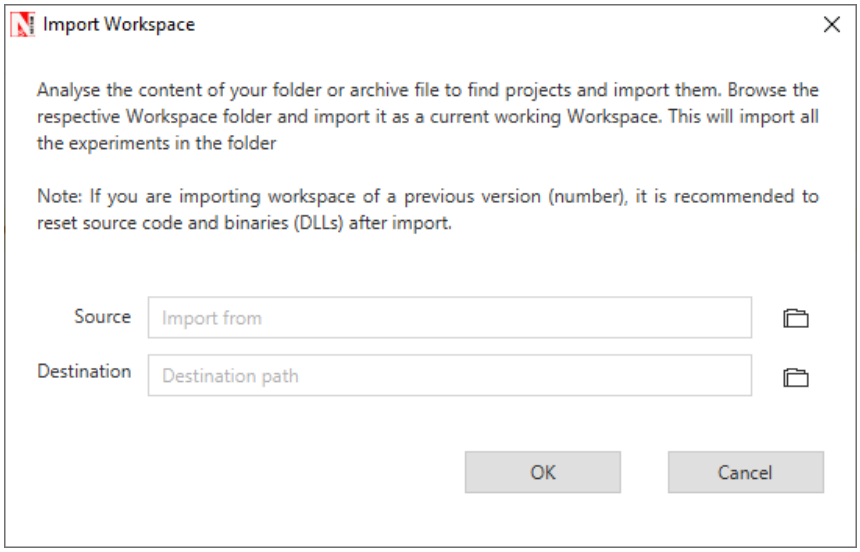
Note: Channel notching code will work only for a single Channel, single CR-CPE and for at-most one Incumbent.

Steps:

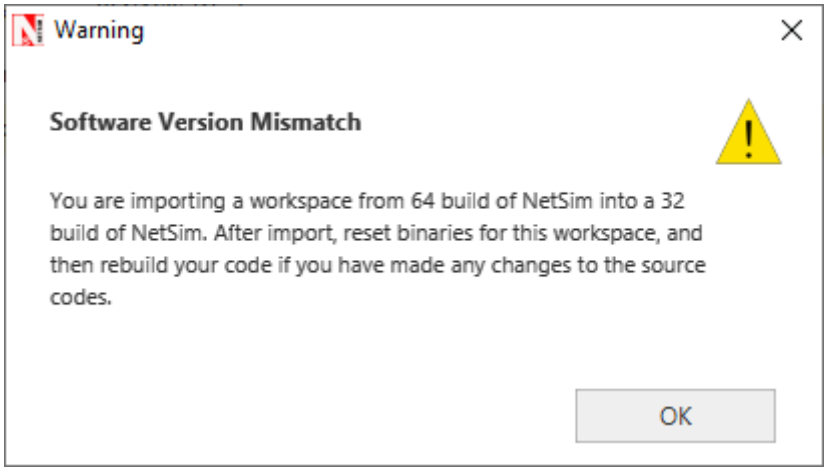
1. The downloaded project folder contains the folders Documentation and Channel Notching Workspace directory as shown below:



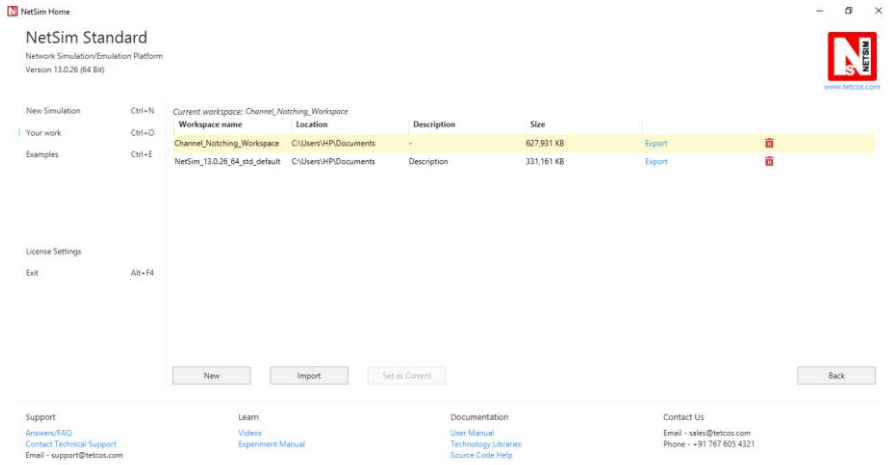
2. Import Channel_Notching_Workspace by going to Your work->Workspace Options->More Options in NetSim Home window. Click on Import. For source path select the Channel_Notching_Workspace.netsim_wsp file from the extracted project folder. Also set the destination path where you want the workspace to be setup.



3. While importing the workspace, if the following warning message indicating Software Version Mismatch is displayed, you can ignore it and proceed.



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

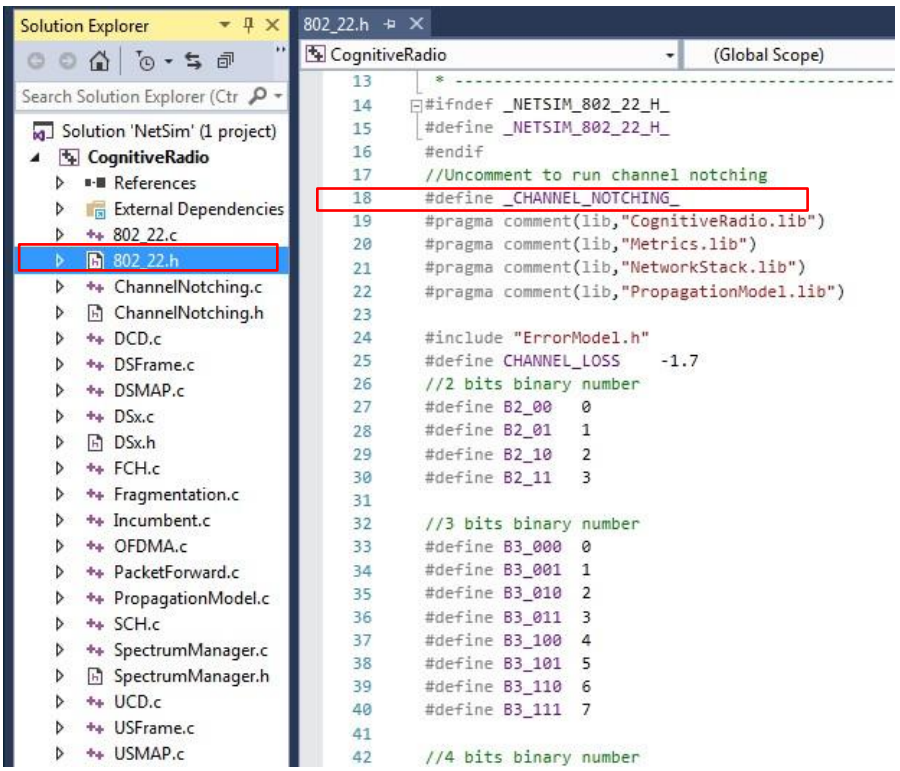


5. Open the Source codes in Visual Studio by going to Your work-> Workspace Options and Clicking on Open code button.

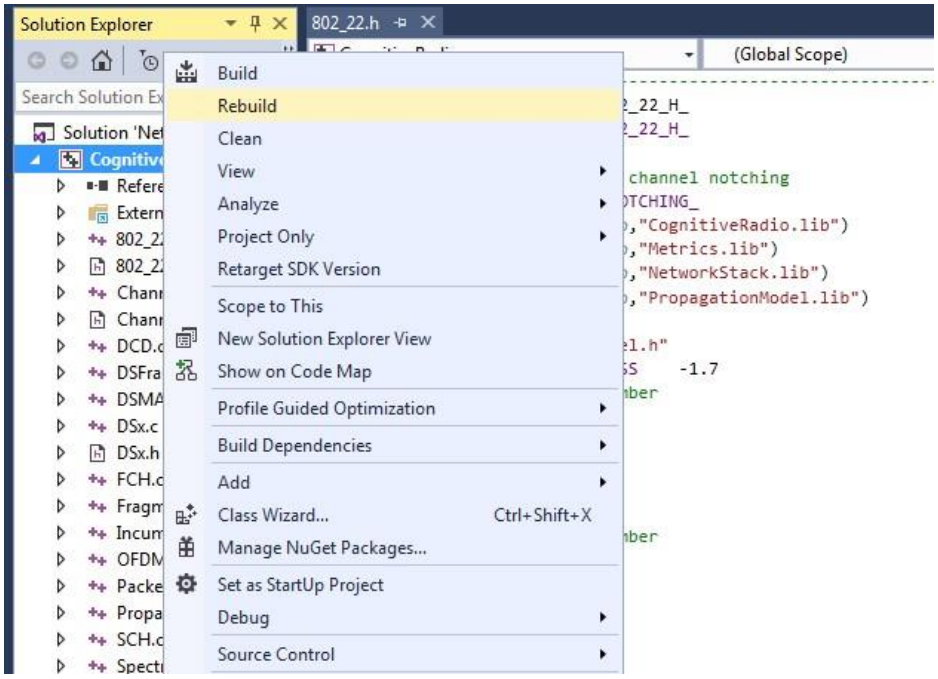
6. In the Solution Explorer, go to **Cognitive Radio > 802_22.h** and open it.

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

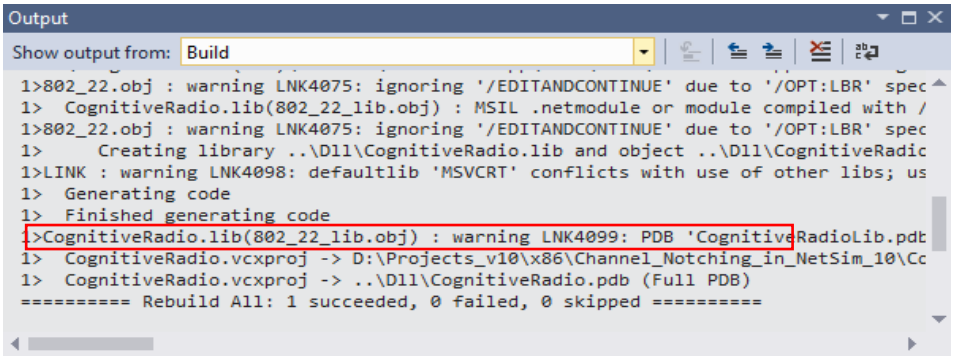
#define _CHANNEL_NOTCHING_



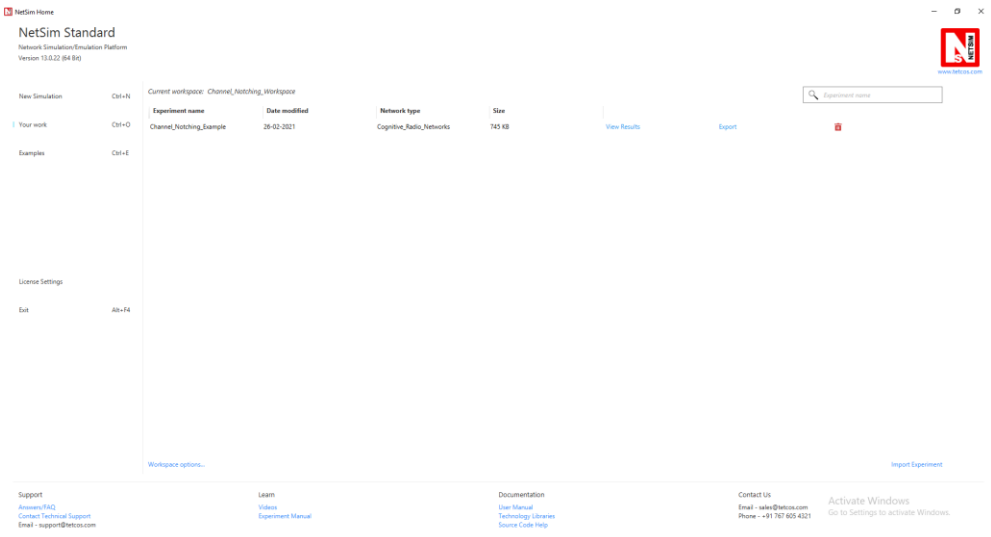
8. Right click on **Solution Explorer-> Rebuild project**



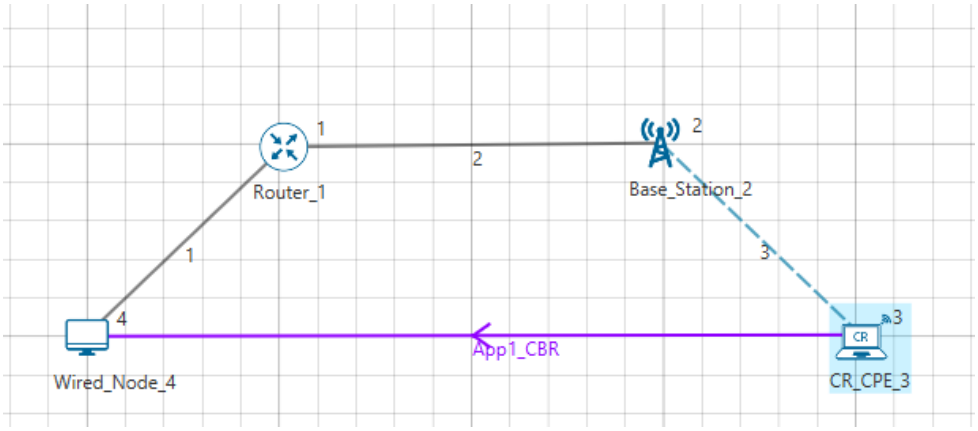
9. You should see a message in the **Output** window as shown in the following figure.



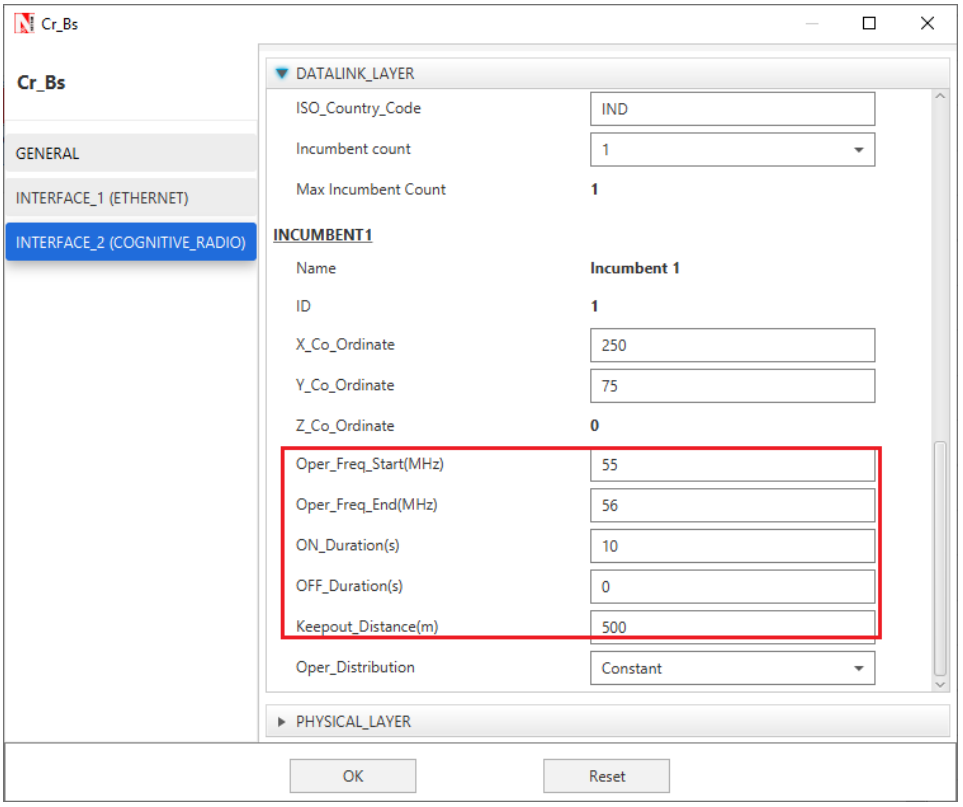
10. Then Channel_Notching_Workspace comes with a sample configuration that is already saved. To open this example, go to Your work and click on the that is present under the list of experiments as shown below:



11. The scenario looks like



12. And set BS properties as



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