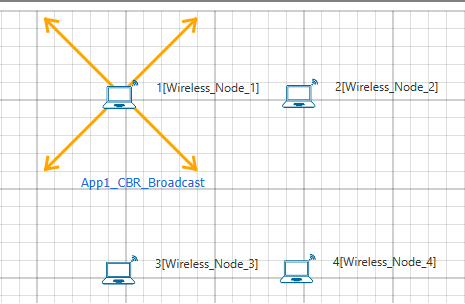
**Rebroadcasting packet in NetSim MANET\VANETs**

**Software Used:** NetSim Standard v11.1 (32bit/ 64bit), Microsoft Visual Studio 2015/2017

**Broadcasting:**

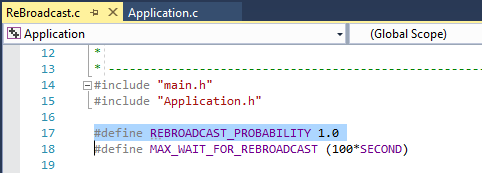
Broadcasting is the process of sending a message from one node to all other nodes in an ad-hoc network. It is a fundamental operation for communication in ad-hoc networks as it allows for the update of network information and route discovery at every node.

**Rebroadcasting:**



Wireless Node 1 initiates a broadcast message and the message is received by nodes 2, 3 and 4. 2, 3 and 4 rebroadcast the message if they have not broadcasted that before. Furthermore, this implementation involves a Rebroadcast\_Probability based on which the nodes resend the packets.

**Probability-based rebroadcasting** - The decision of rebroadcasting is based upon a random probability. This probability may be as simple as flipping a coin or it may be very complex involving probabilities which include parameters such as node density, duplicate packets received, battery power or a particular nodes participation within the network etc. Users can change the Rebroadcast\_Probability macros present in Rebroadcast.c file as shown below:



**Rebroadcasting in NetSim:**

To implement this project in NetSim, we have created an additional Rebroadcast.c file inside Application project. The file contains the following functions:

* void rebroadcast\_packet();

This function is used to rebroadcast the packet.

* static bool isRebroadcastAllowed();

This function is used to check whether rebroadcasting is allowed or not.

* void rebroadcast\_add\_packet\_to\_info();

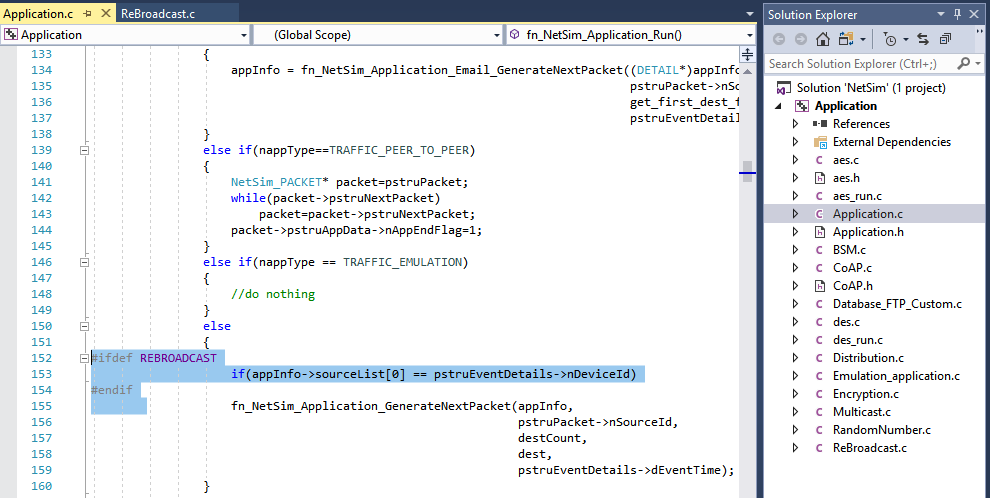
This function is used to add the packet to rebroadcast list.

* static void cleanup\_broadcast\_info();

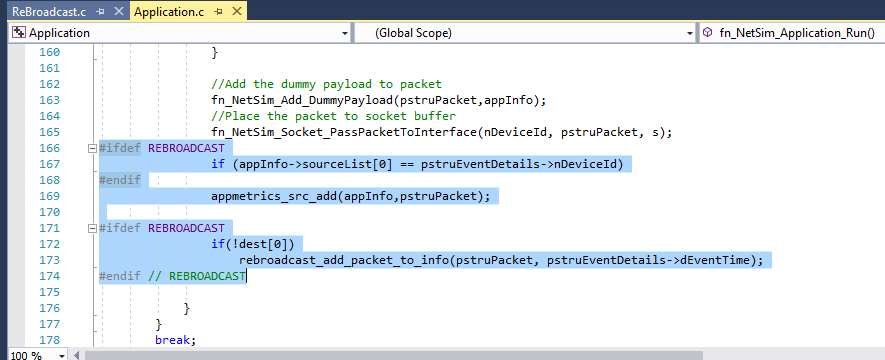
This function is used to clean the broadcast information.

**Code modifications done in NetSim:**

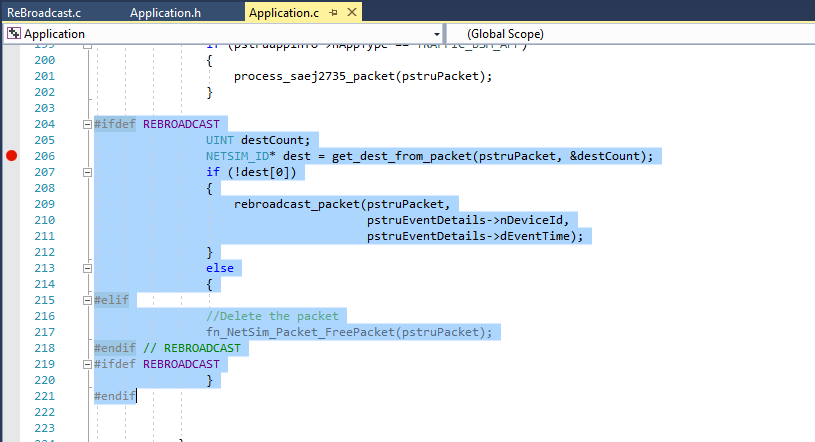
1. We have added the following lines of code in fn\_NetSim\_Application\_Run() function in the APPLICATION\_OUT\_EVENT present in Application.c file inside Application project. This is used to generate next broadcast packet if the current device is present in the source list.

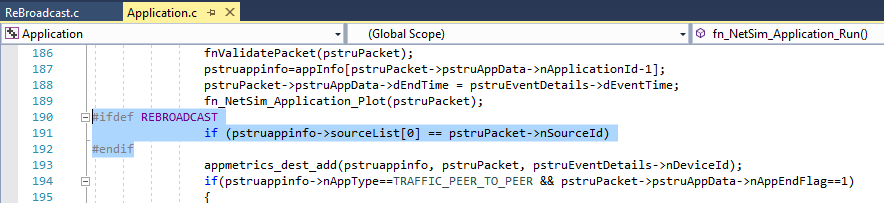


1. The following lines of code are added in the same fn\_NetSim\_Application\_Run() function in the APPLICATION\_OUT\_EVENT present in Application.c file inside Application project. The code checks if the destination is ‘0’ i.e., Broadcast packet, then it adds the packet to rebroadcast list.

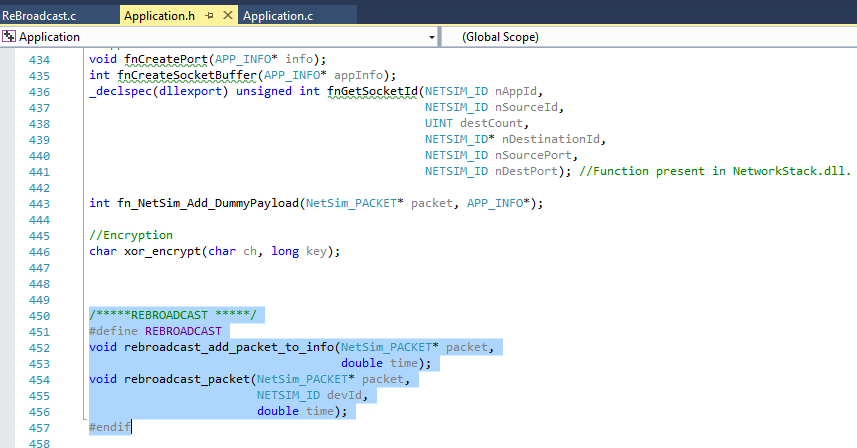


1. Now add the following code in fn\_NetSim\_Application\_Run() function in APPLICATION\_IN\_EVENT present in Application.c file inside Application project. It checks whether the destination is ‘0’ or not. If it is ‘0’, then it rebroadcasts the packet or else deletes the packet.



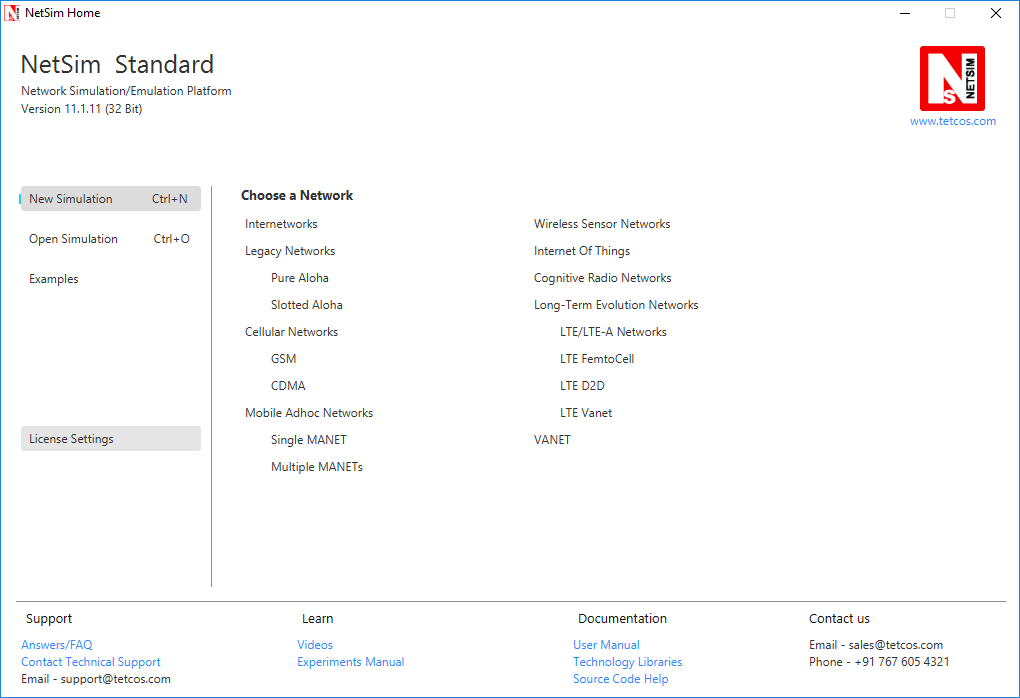


1. We have added the following function declarations in Application.h file.

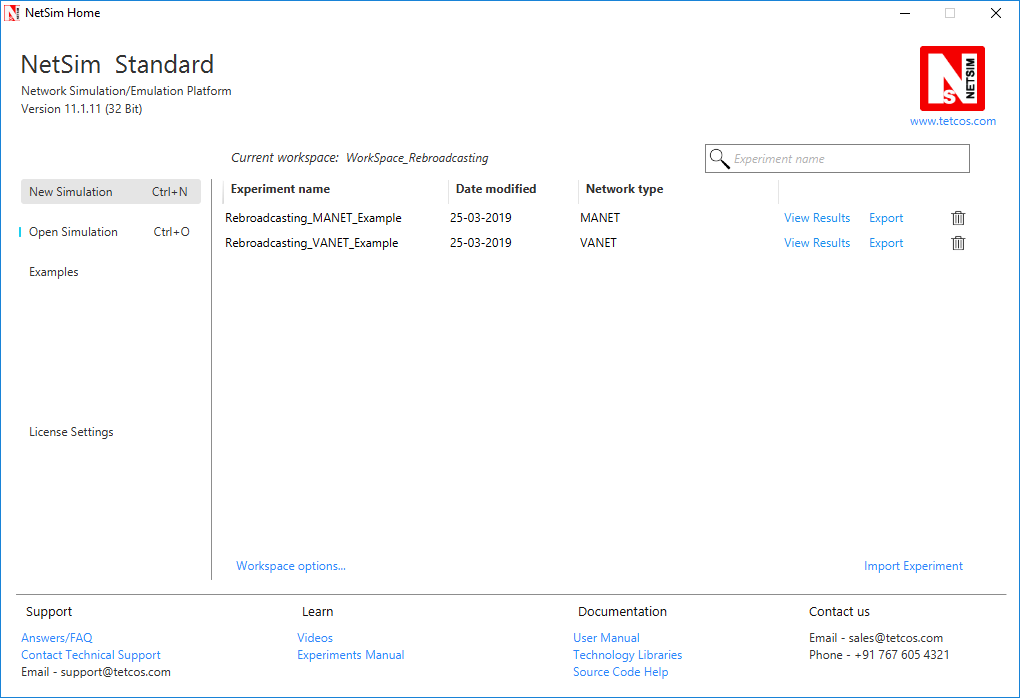


**Steps:**

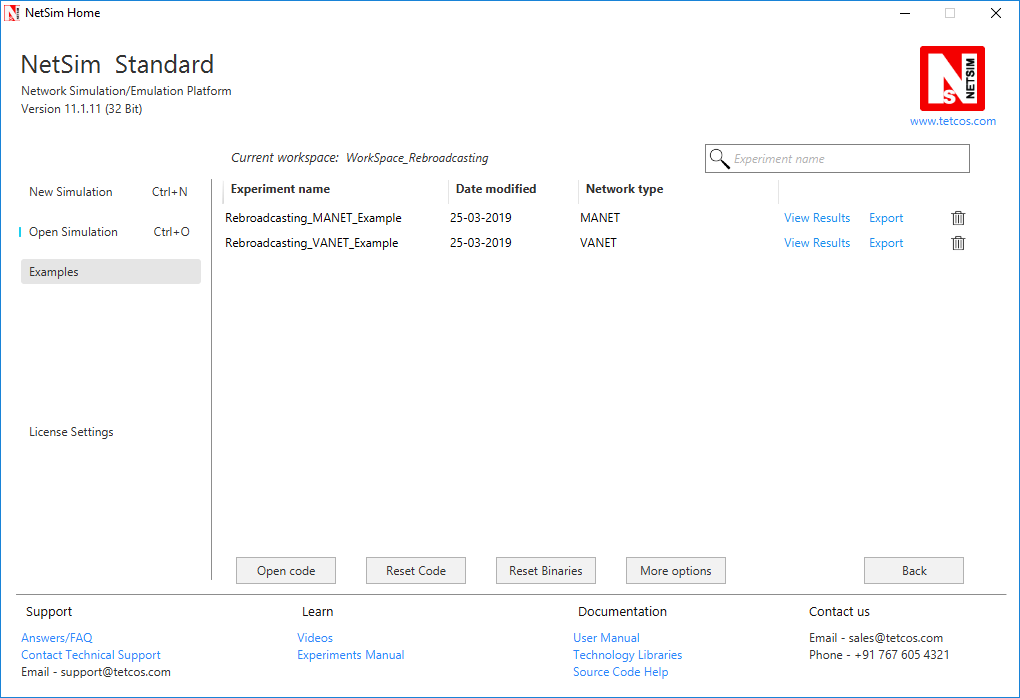
* After you unzip the downloaded project folder, Open NetSim Home Page click on **Open Simulation** option,



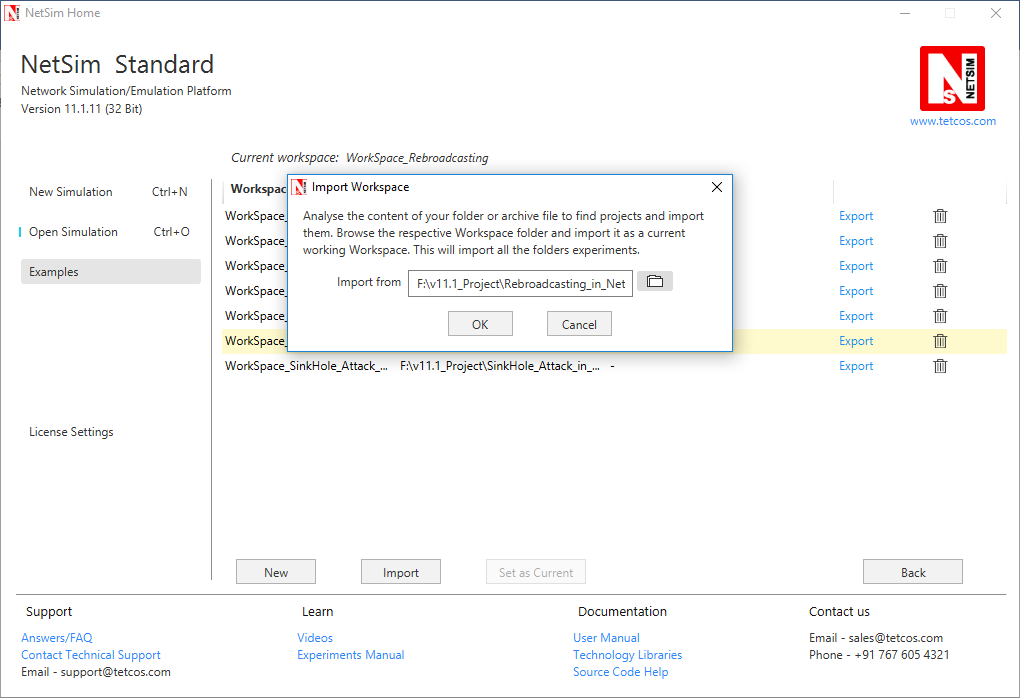
* Click on **Workspace options**



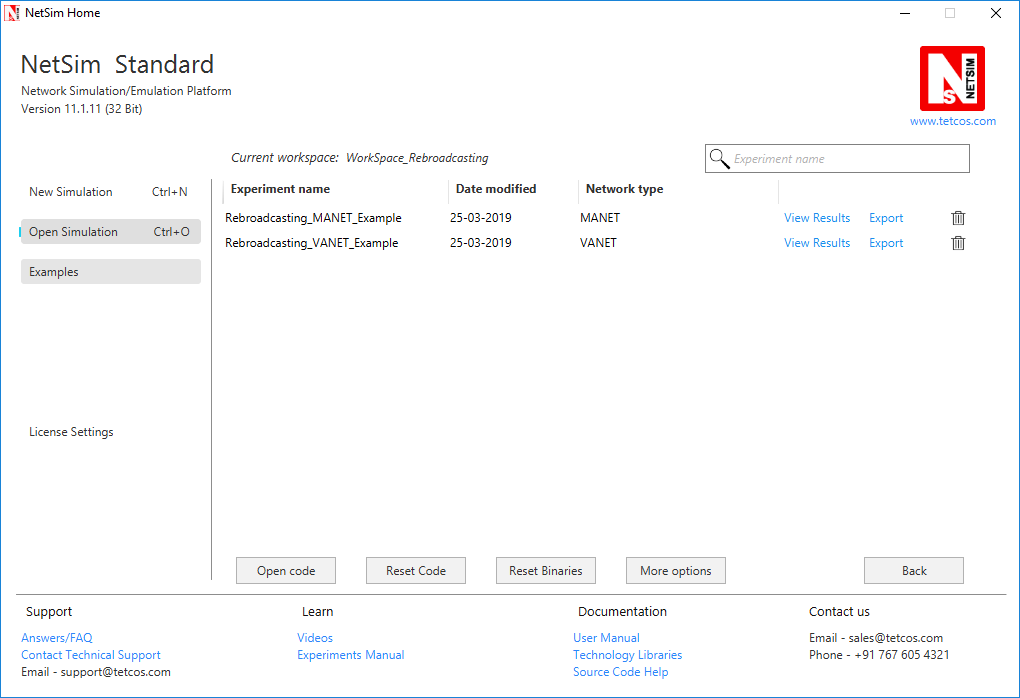
* Click on **More Options,**



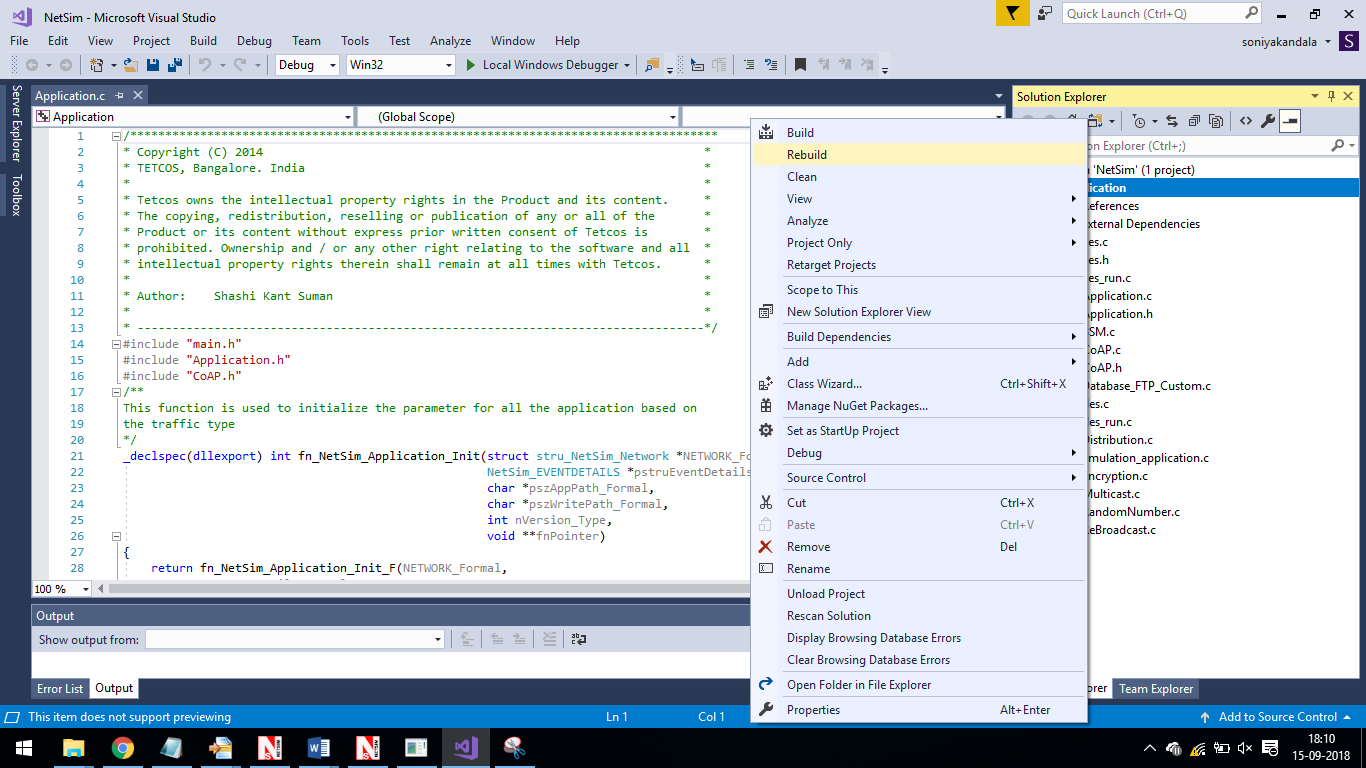
* Click on **Import**, browse the extracted folder path and go into the WorkSpace\_Rebroadcasting directory. Click on the select folder button and then on **OK.**



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



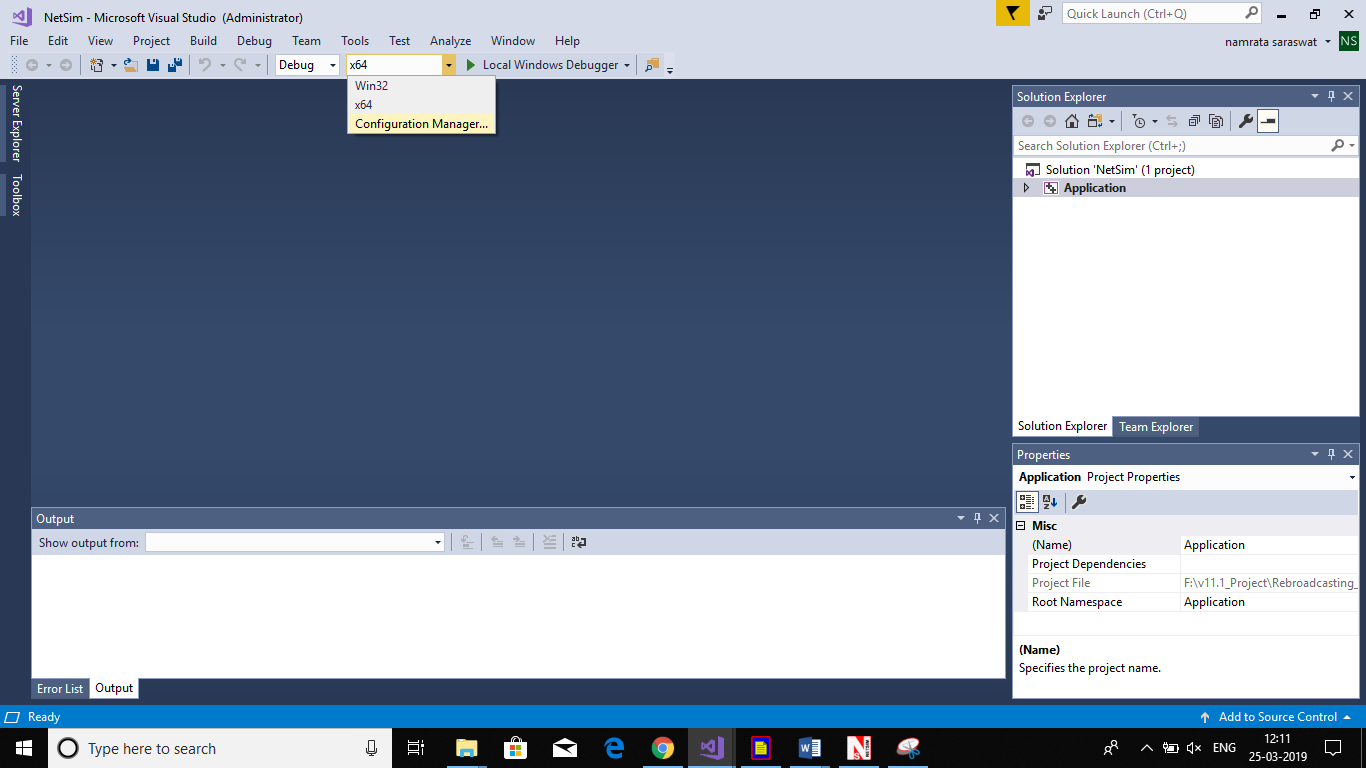
* Right click on Solution in Solution Explorer and select ‘Rebuild solution’.



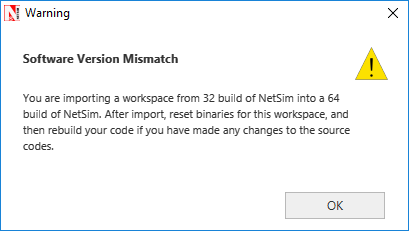
* Upon rebuilding, **libApplication.dll** will automatically get updated in the respective bin folder of the current workspace.

**Note:**

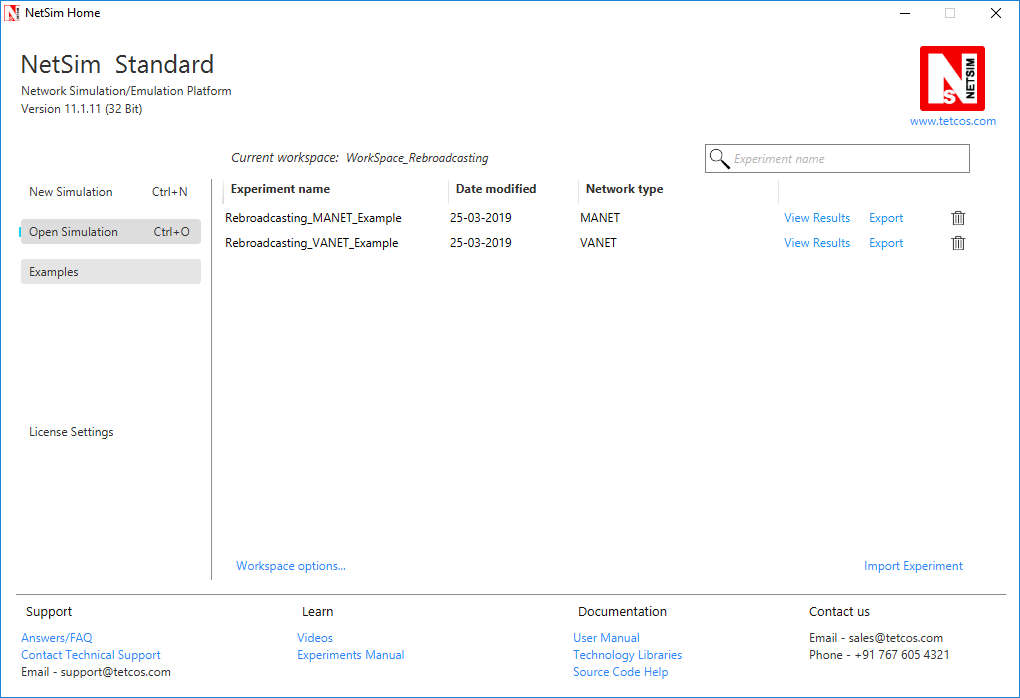
1. Based on whether you are using NetSim 32 bit or 64 bit setup you can configure Visual studio to build 32 bit or 64 bit Dll files respectively as shown below:



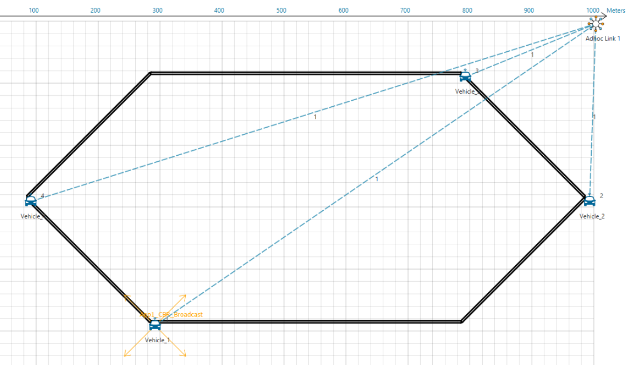
1. While importing the workspace, if the following warning message indicating Software Version Mismatch is displayed, you can ignore it and proceed.



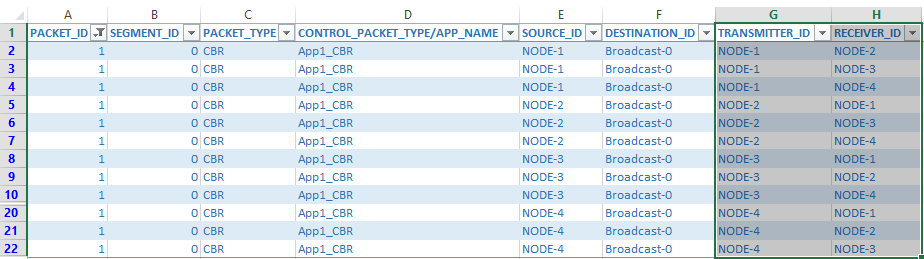
* Go to NetSim home page, click on **Open Simulation**, Click on **Rebroadcasting\_VANET\_Example/ Rebroadcasting\_MANET\_Example** and run the simulation for 100 seconds.



**VANET SCENARIO:**



* In the above scenario, Vehice-1 is broadcasting the packet and it is received by the Vehicles 2, 3 and 4. Then Vehices 2, 3, and 4 will rebroadcast the same packet based on the probability value in Rebroadcast.c file.
* After simulation, open Packet Trace and filter Packet\_Id to ‘1’ or any other id and observe that the nodes other than source are rebroadcasting the same packet.



* Note that Users SHOULD NOT use the performance metrics provided at the end of simulation but should rather calculate the network performance metrics from the packet trace.
* Users can also create their own network scenarios in **Single MANET/VANET** and run the simulation.