

Intrusion detection system for MANET

Software Recommended: NetSim Standard v12.1/v12.2 32-bit/ 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-fileexchange-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:

v12.1: https://github.com/NetSim-TETCOS/Intrusion_Detection_System_for_MANETS_v12.1.git

v12.2: https://github.com/NetSim-TETCOS/Intrusion_Detection_Systems_for_MANETs_v12.2.git

Note: The cloned project directory will contain the documentation specific to the NetSim version (v12.1/v12.2).

The following steps show how a user can run the IDS in NetSim to detect a malicious node, and then setup a new route to the destination avoiding the malicious node

- Creating Malicious nodes for a particular network scenario is explained in Malicious.c file
- To detect the intruder and to send data via a new route, the following files are added in DSR and IEEE802_11:

➤ **Pathrater.c** :

This file contains code for avoiding the malicious node and finding a new route (once the IDS detects the malicious node) in networks running DSR in Layer 3. Note that this system would work only for UDP and not for TCP, since TCP involves receiving ack's from the destination

If `_NETSIM_PATHRATER_` is defined, the code is used to validate routes. When the Node is a Malicious Node and a Route Reply is processed, the Function verifies the route reply in the route cache and checks for the black listed node i.e.,malicious node. When a malicious node is found, that route entry is deleted from the cache.

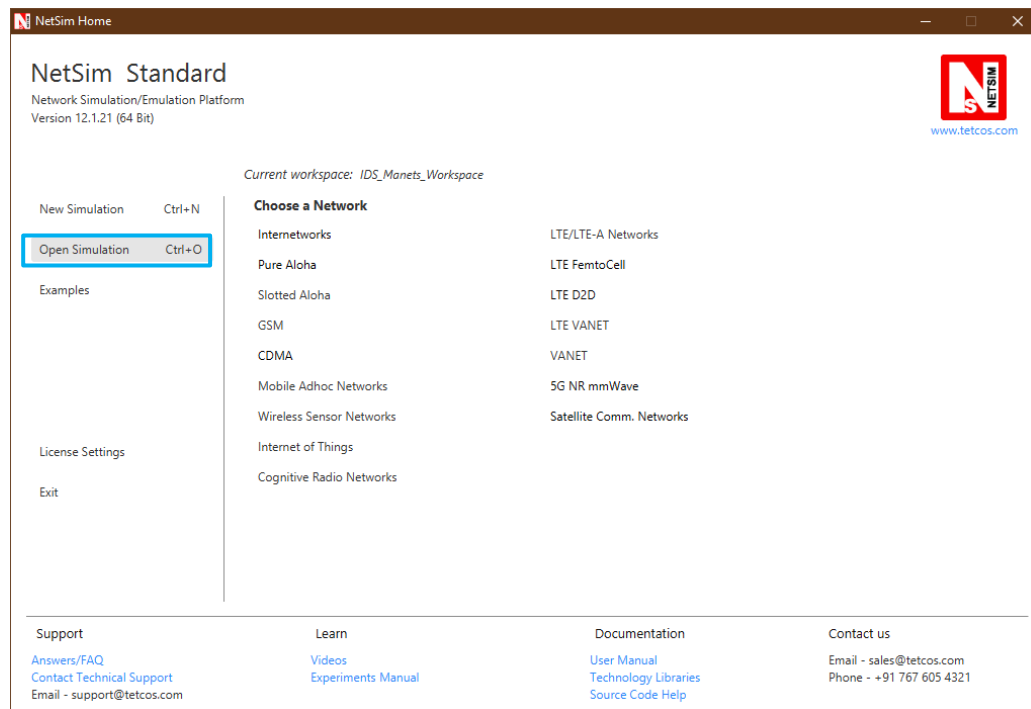
➤ **Watchdog.c**

This file contains code for the IDS and is added in IEEE802_11 operating in Layer 2.

If `_NETSIM_WATCHDOG_` is defined, a watchdog timer starts the moment a packet is sent. Once a packet is forwarded to next hop node, the current node checks for watchdog timer duration if the packet is getting forwarded further on to destination node or not.

The malicious node doesn't forward packets that it receives. The watchdog timer in the node (which forwarded the packet to the malicious node) expires. A counter is present which measures the number of times the watchdog timer expires (in other words the number of packets sent out but not forwarded by the next hop node). Once this counter's value reaches the failure threshold the next hop is marked by the current node as a malicious node.

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



- Click on **Workspace options**

NetSim Home

NetSim Standard

Network Simulation/Emulation Platform
Version 12.1.21 (64 Bit)

Current workspace: *IDS_Manets_Workspace*

Experiment name | Date modified | Network type

Experiment name	Date modified	Network type
IDS_Experiment	13-03-2020	MANET

View Results | Export |

Workspace options...

Import Experiment

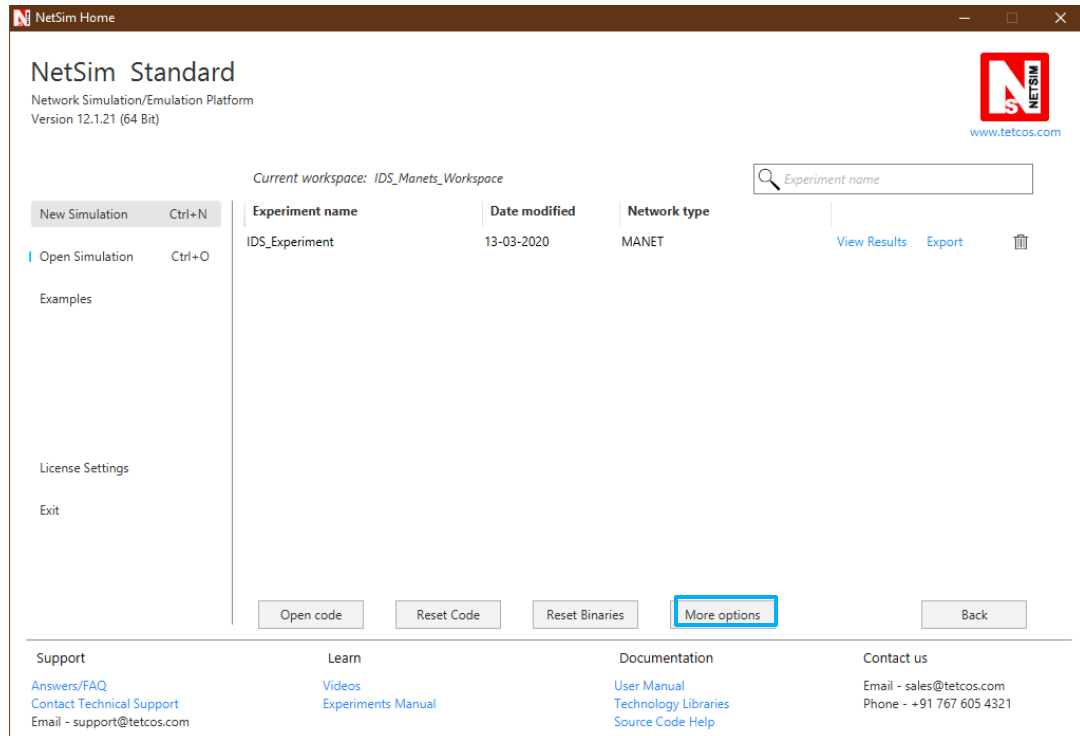
Support: [Answers/FAQ](#), [Contact Technical Support](#), [Email - support@tetcos.com](mailto:support@tetcos.com)

Learn: [Videos](#), [Experiments Manual](#)

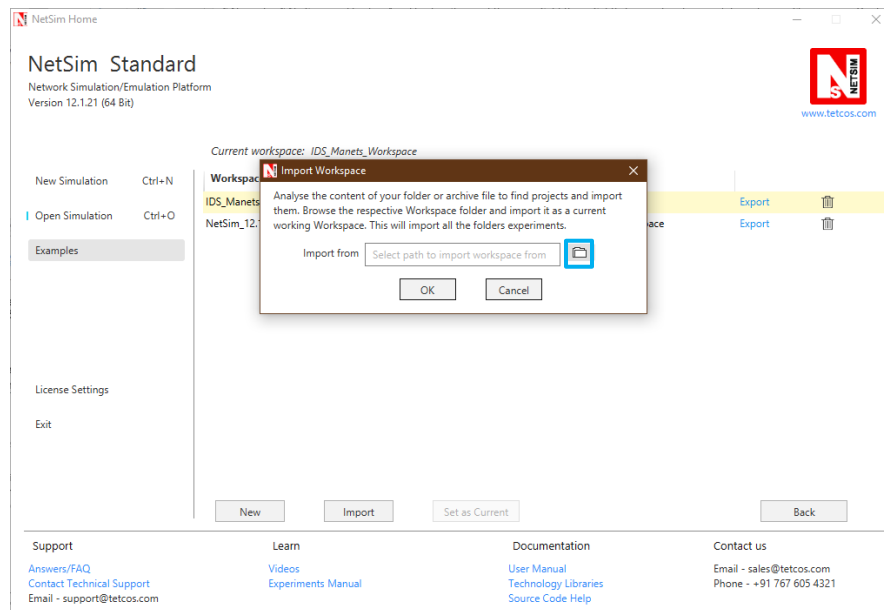
Documentation: [User Manual](#), [Technology Libraries](#), [Source Code Help](#)

Contact us: [Email - sales@tetcos.com](mailto:sales@tetcos.com), [Phone - +91 767 605 4321](tel:+917676054321)

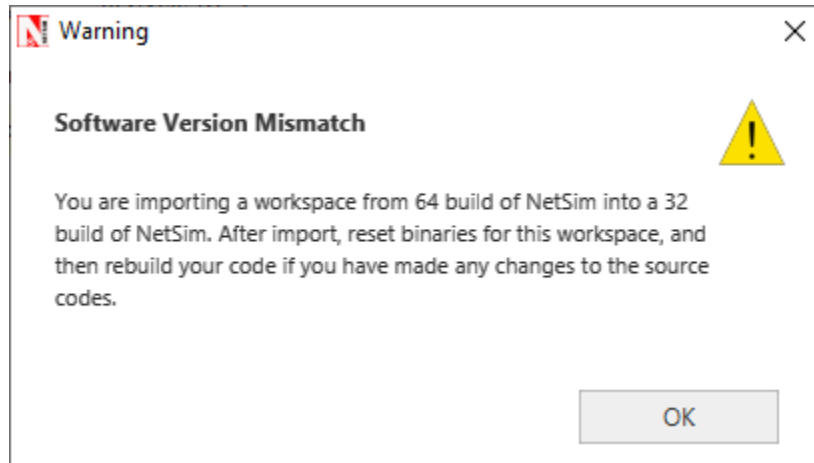
- Click on **More Options**,



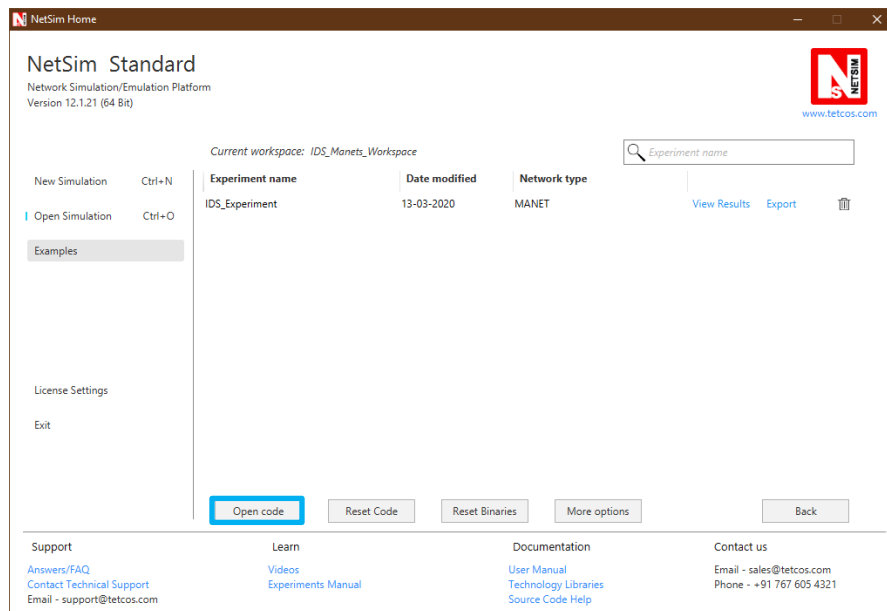
- Click on **Import**, browse the extracted folder path and go into *Intrusion_Detection_System* workspace directory. Click on Select folder button and then on **OK**.



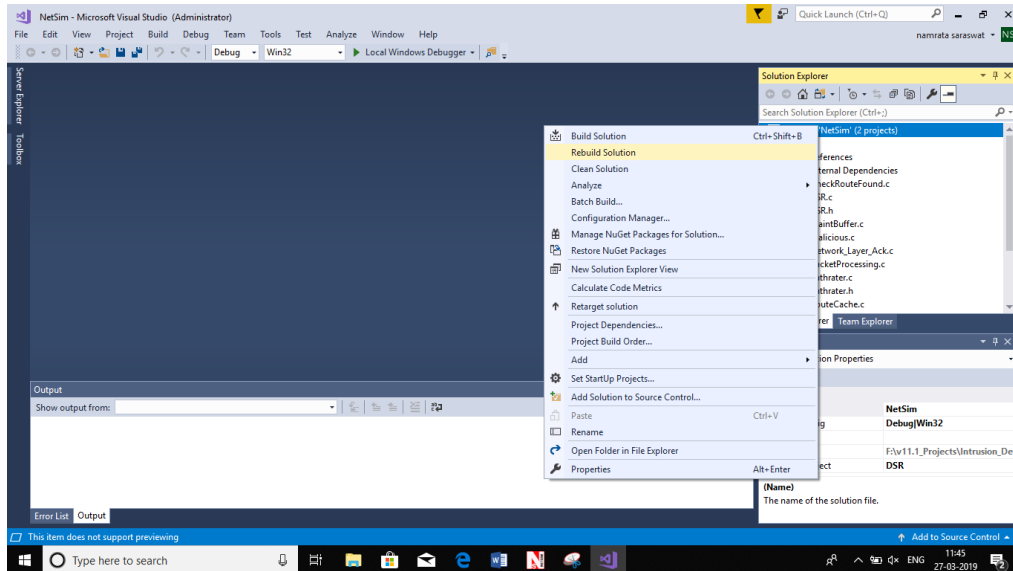
- While importing the workspace, if the following warning message indicating Software Version Mismatch is displayed, you can ignore it and proceed.



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



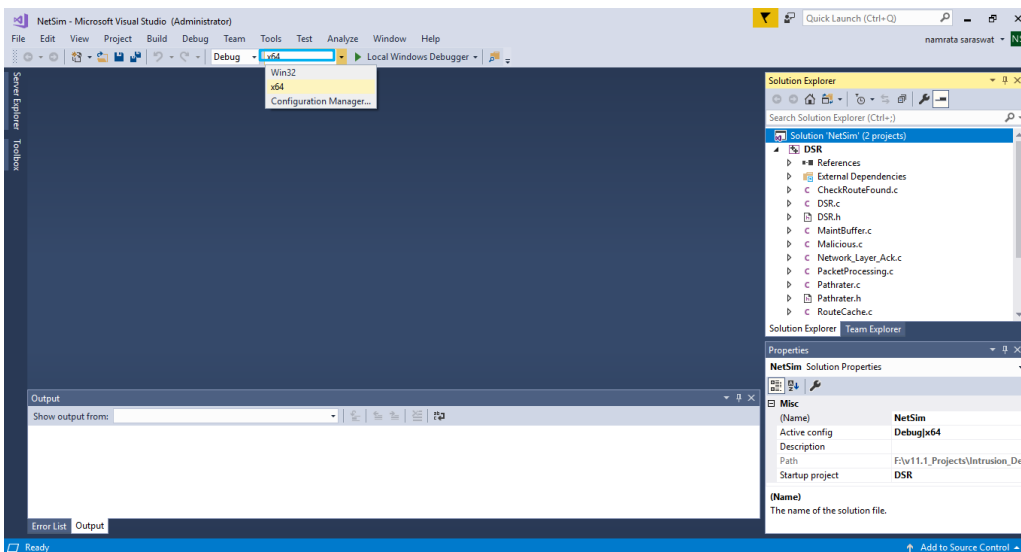
- Right click on the solution and select rebuild.



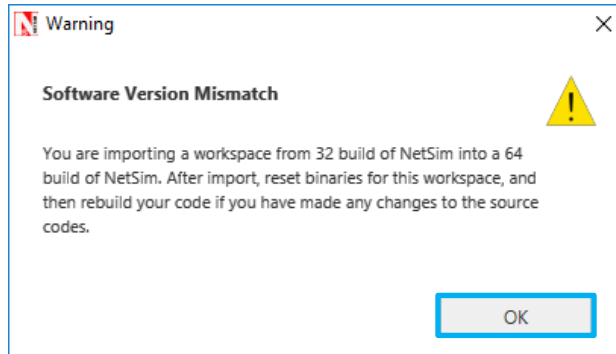
- Upon rebuilding, **libIEEE802_11.dll** and **libDSR.dll** will automatically get updated in the respective bin folder of the current workspace.

Note:

1. While 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:

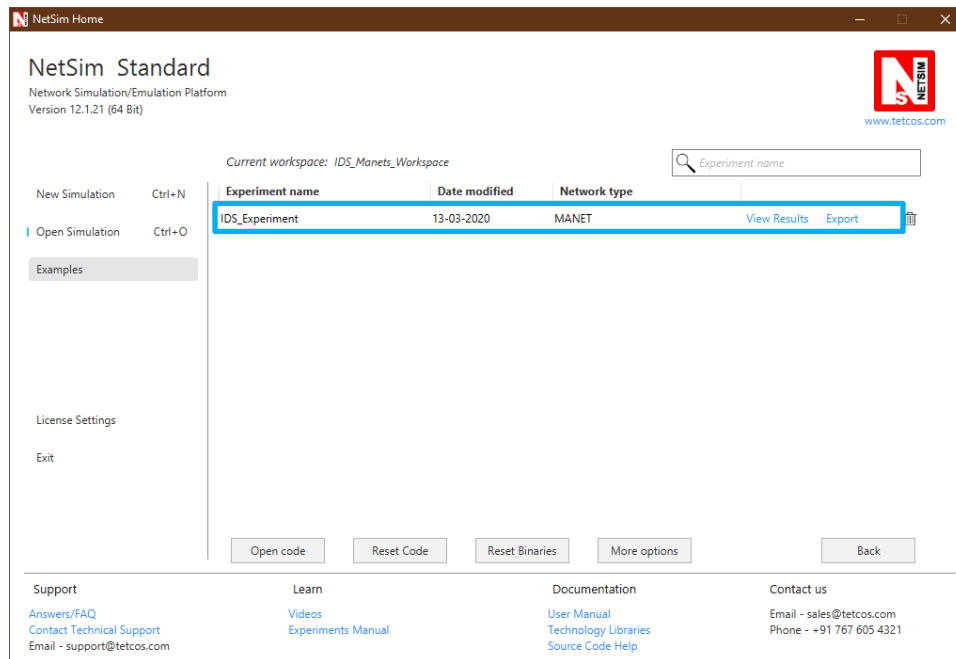


2. While importing the project in NetSim 64bit version, it will display popup as **“Software version mismatch”**, ignore the warning message and click in **ok**.

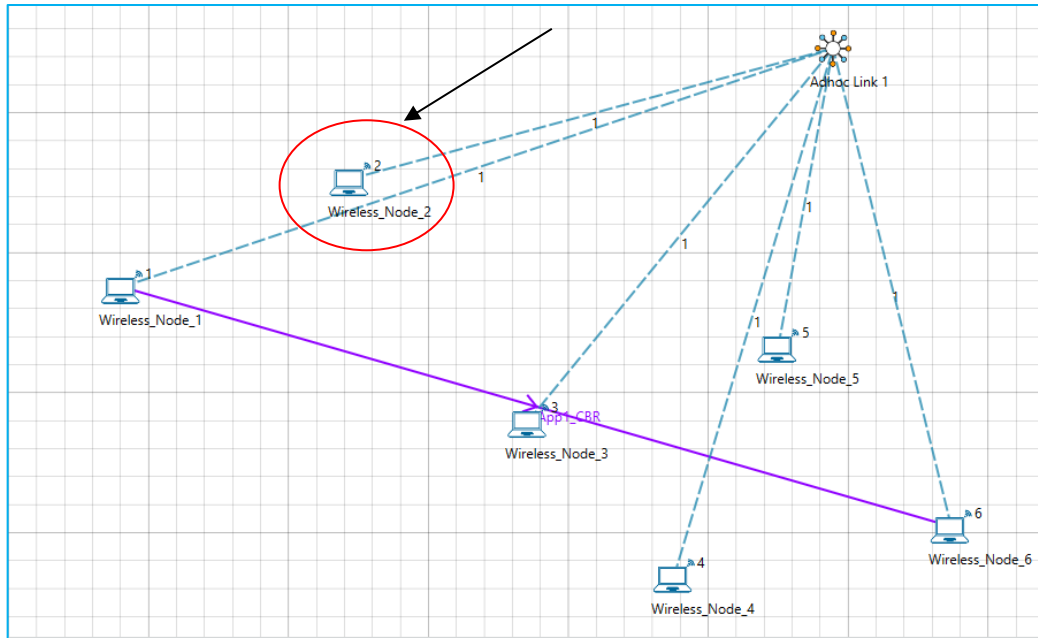


Next, to run the IDS code, follow these steps:

Step 1: Go to NetSim home page, click on **Open Simulation**, Click on **IDS_Experiment**.

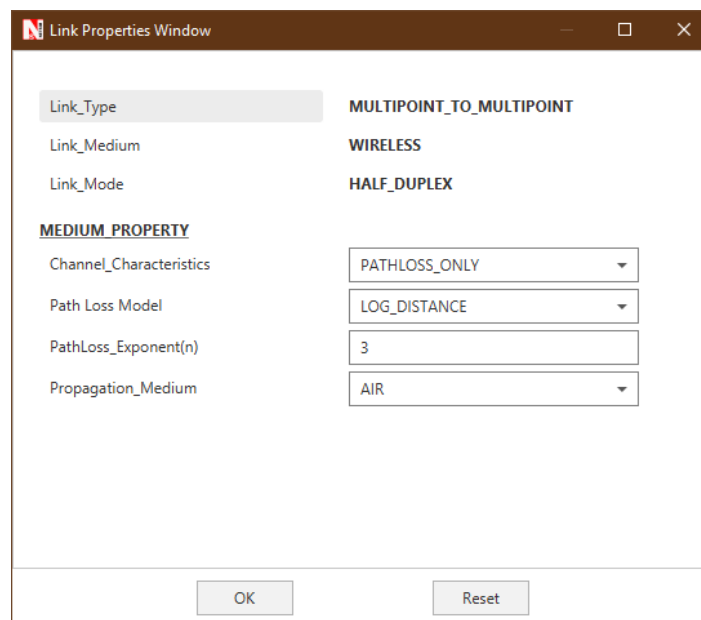


Malicious

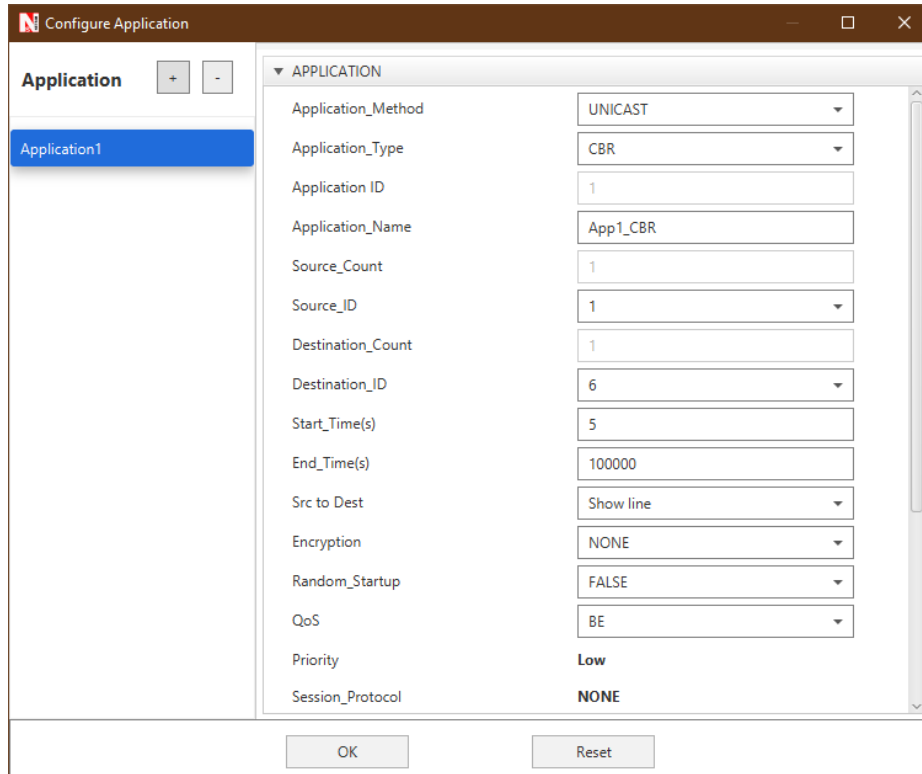


Step 2: Channel Characteristics is set to Pathloss only with LOG_DISTANCE as the path loss model. Path loss exponent is set to a high value 3

Example:

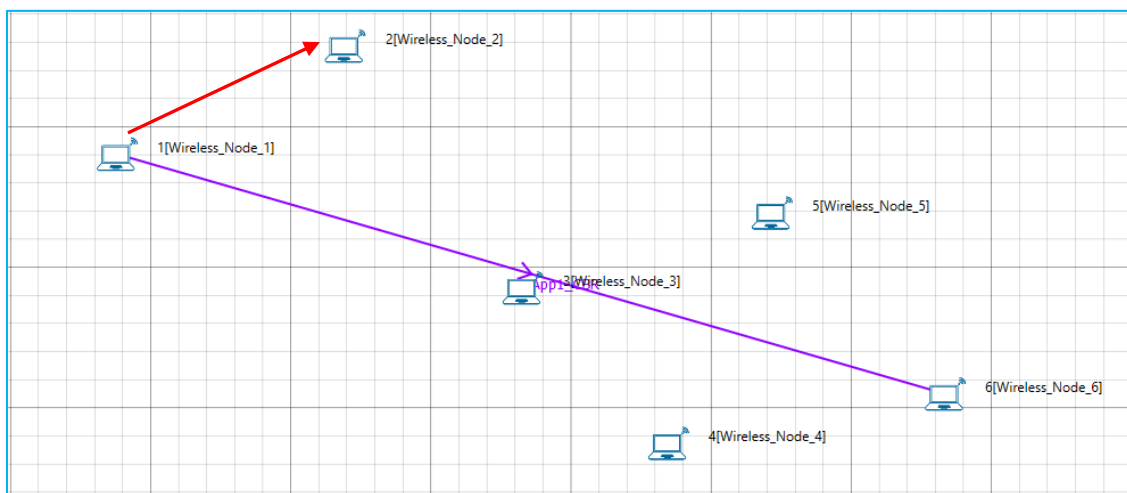


Step 3: An application is set between node 1 and node 6

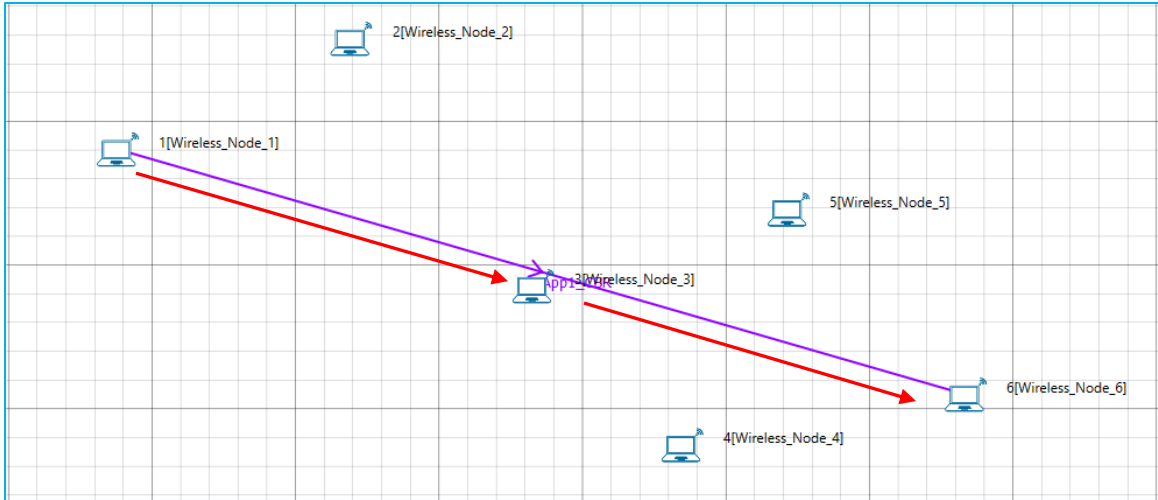


Step 4: Run the simulation

Step 5: View packet animation. Here you would notice initially all traffic would flow to the malicious nodes. Per the original code setting the Watchdog timer is set to 2 seconds and the failure threshold is set to 20 packets. So you would notice that around 25.76 seconds, the malicious node is detected and the route to destination would change in the subsequent route discovery process.



Initial flow of packets till node 2 detected as malicious



Flow of packets after node 2 is detected as malicious

The time at which a malicious node is detected can be obtained from the CUSTOM METRICS in the results window where the start time - time from which a node becomes malicious, detection time - time at which the node was added to blacklist can be obtained.

IDS_METRICS_Table		
IDS_Custom_Metrics <input type="checkbox"/> Detailed View		
DeviceID	Start Time (micro sec)	Detection Time (micro sec)
2	500000.000000	7386976.020000

Dedicated Metrics for IDS