Sink Hole Attack in AODV

Software Recommended: NetSim Standard v13.0, Visual Studio 2017/2019

Secure URL for the GitHub repository:

https://github.com/NetSim-TETCOS/Secure_AODV_Project_v13_0/archive/refs/heads/main.zip

Sinkhole attack is one of the severe attacks in wireless Ad hoc network. In sinkhole Attack, a compromised node or malicious node advertises wrong routing information to produce itself as a specific node and receives whole network traffic. After receiving whole network traffic, it can either modify the packet information or drop them to make the network complicated. Sinkhole attacks affect the performance of Ad hoc networks protocols such as DSR, AODV protocol.

Implementation in AODV:

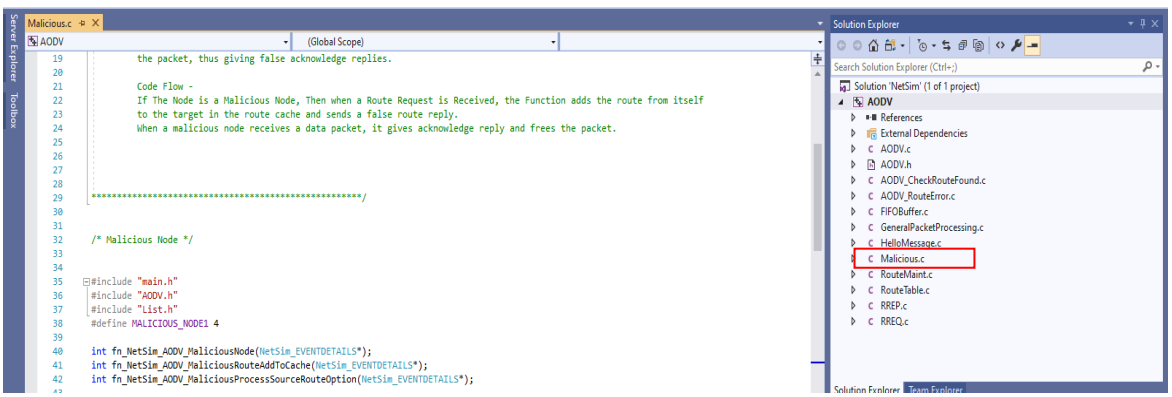
- In AODV the source broadcasts RREQ packet during Route Discovery.
- The destination on receiving the RREQ packet replies with a RREP packet containing the route to reach the destination.
- But Intermediate nodes can also send RREP packet to the source if they have a route to the destination in their route cache.
- Using this as an advantage the malicious node adds a fake route entry into its route cache with the destination node as its next hop.
- On receiving the RREQ packet from the source the malicious node sends a fake RREP packet with the fake route.
- The source node on receiving this packet observes this as a better route to the destination.
- All the Network Traffic is attracted towards the Sinkhole (Malicious Node) and it can either modify the packet Information or simply drop the packet.

A file **malicious.c** is added to the AODV project which contains the following functions:

- fn_NetSim_AODV_MaliciousNode ( )
  This function is used to identify whether a current device is malicious or not in-order to establish malicious behavior.

- fn_NetSim_AODV_MaliciousRouteAddToCache ()
  This function is used to add a fake route entry into the route cache of the malicious device with its next hop as the destination.

- fn_NetSim_AODV_MaliciousProcessSourceRouteOption ()
  This function is used to drop the received packets if the device is malicious, instead of forwarding the packet to the next hop.

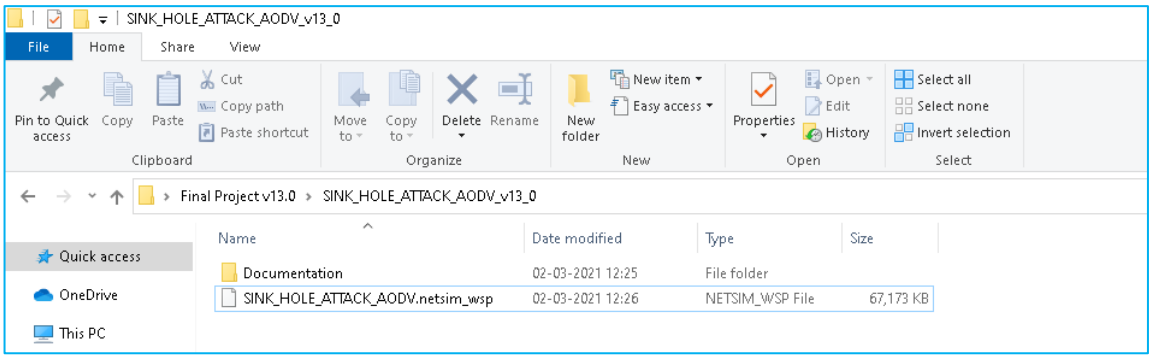


You can set any device as malicious and you can have more than one malicious node in a scenario.

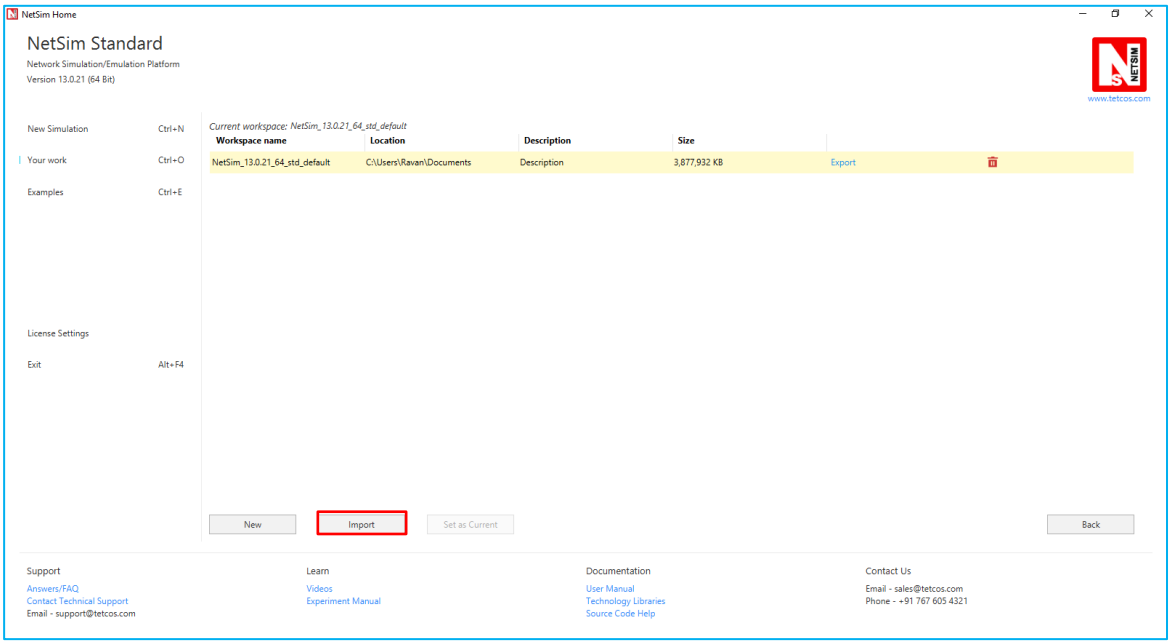
Device id's of malicious nodes can be set inside the fn_NetSim_AODV_MaliciousNode () function.

Steps:

1. The downloaded project folder contains the folders Documentation, and SINK_HOLE_ATTACK_AODV.netsim_wsp directory as shown below:

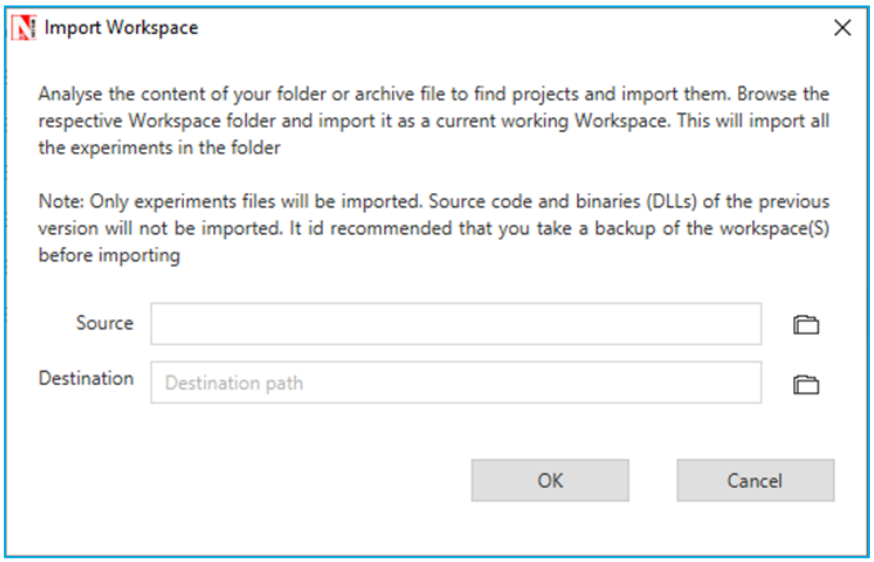


2. Import SINK_HOLE_ATTACK_AODV.netsim_wsp by going to Your work->Workspace Options- >More Options in NetSim Home window. Then select Import as shown below:

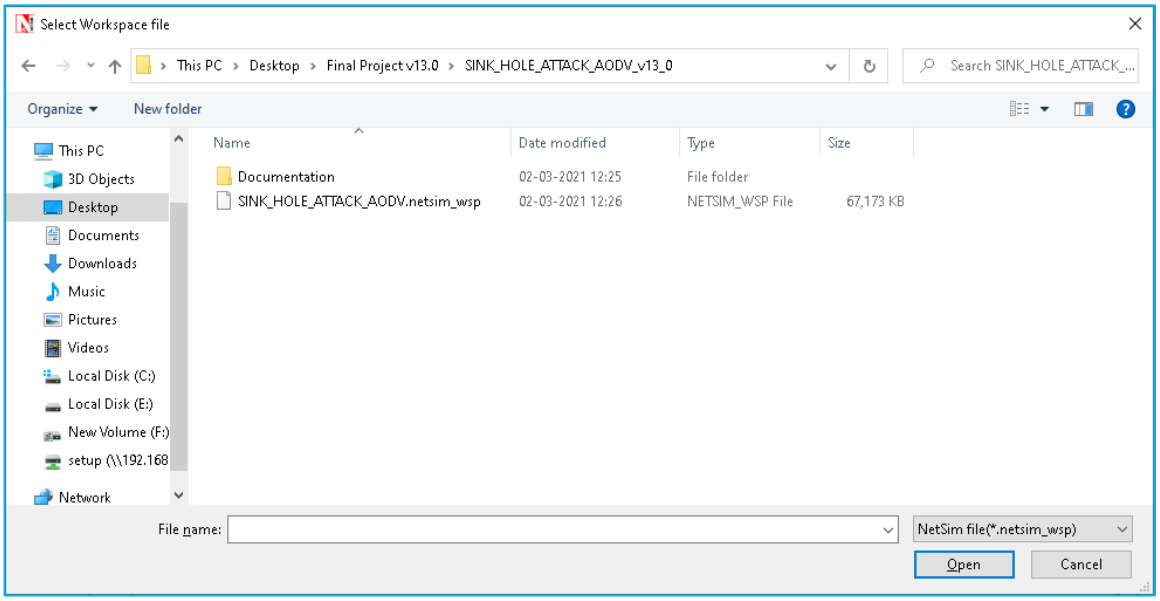


3. This will display a window were users need to give the source file (exported workspace file) and the Destination, the path where the workspace is to be imported to and then click on ok.

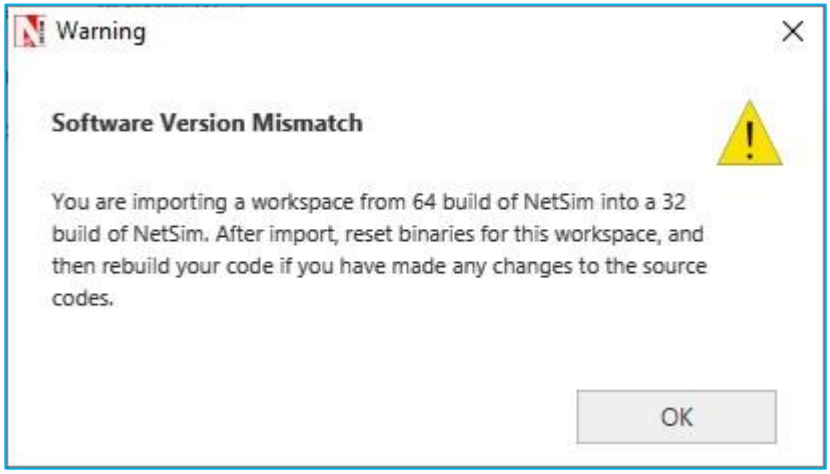
   Note: Only exported workspaces with ".netsim_wsp" extension can be imported.



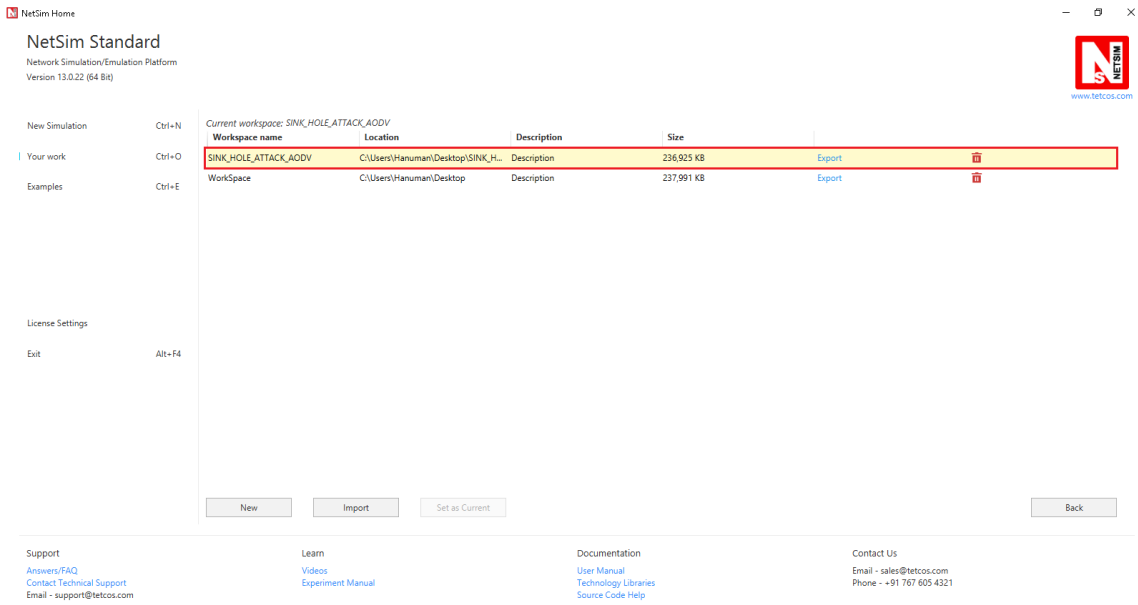
4. Browse to the SINK_HOLE_ATTACK_AODV folder and click on select folder as shown below.



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.

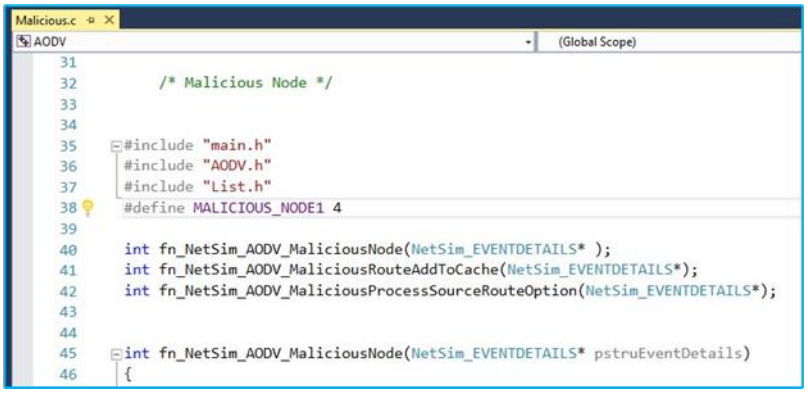


7. The Imported workspace will be set as the current workspace automatically. To see the imported workspace, click on Your work->Workspace Options->More Options as shown below:

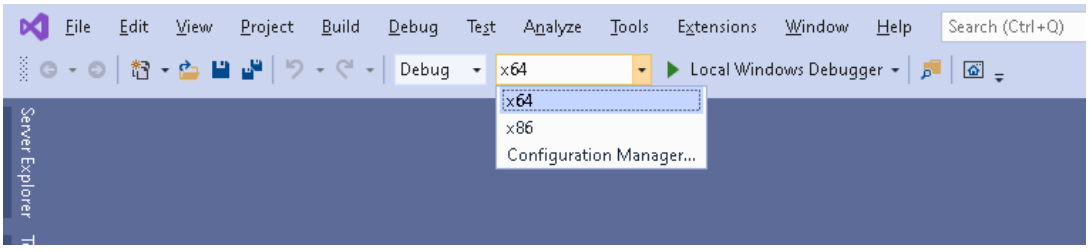


8. Open the Source codes in Visual Studio by going to Your work-> Workspace Options and Clicking on Open code button as shown below:

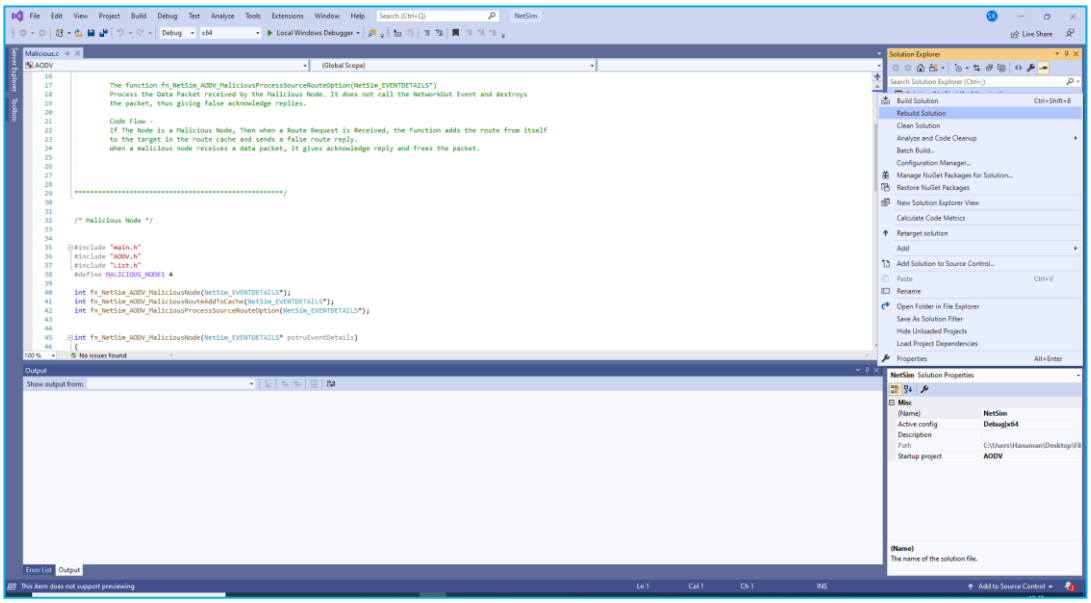
- Expand AODV project and open Malicious.c file.
- Set malicious node id.



- 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:

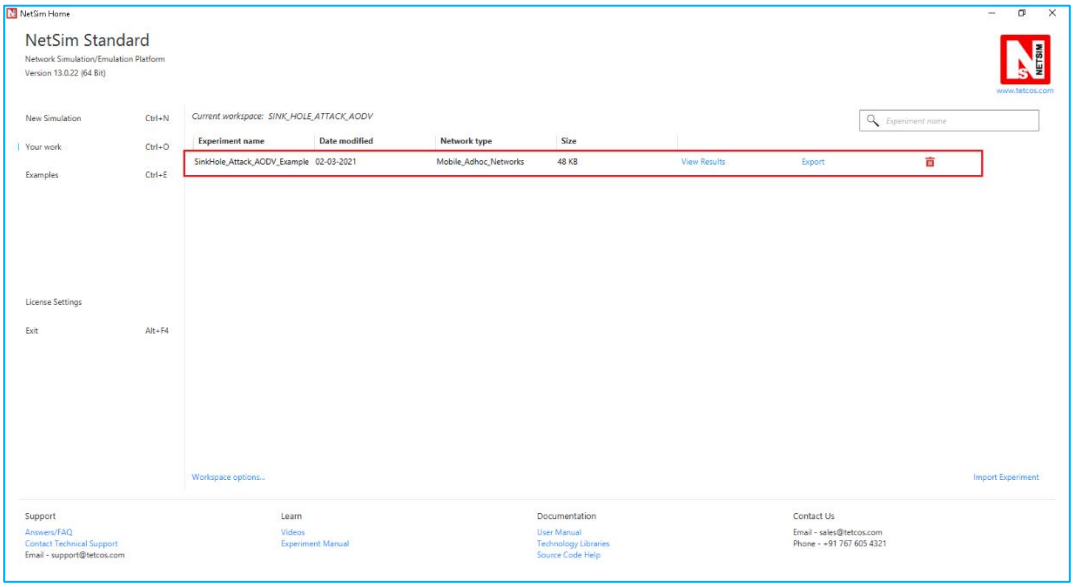


- Now right click on solution explorer and select Rebuild.



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

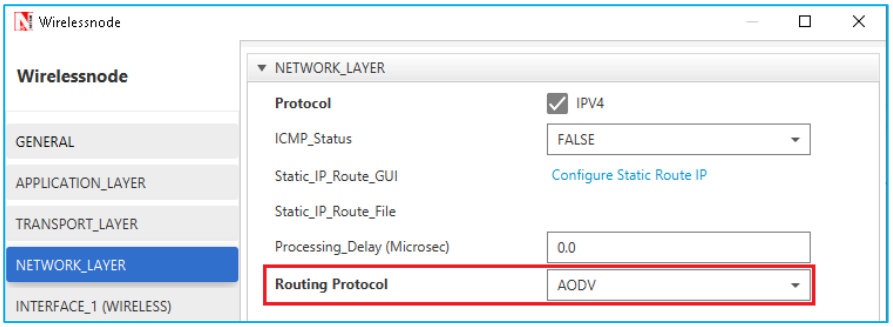
- Then Sink_Hole_Attack_AODV comes with a sample configuration that is already saved. To open this example, go to Your work and click on Experiment that is present under the list of experiments as shown below:



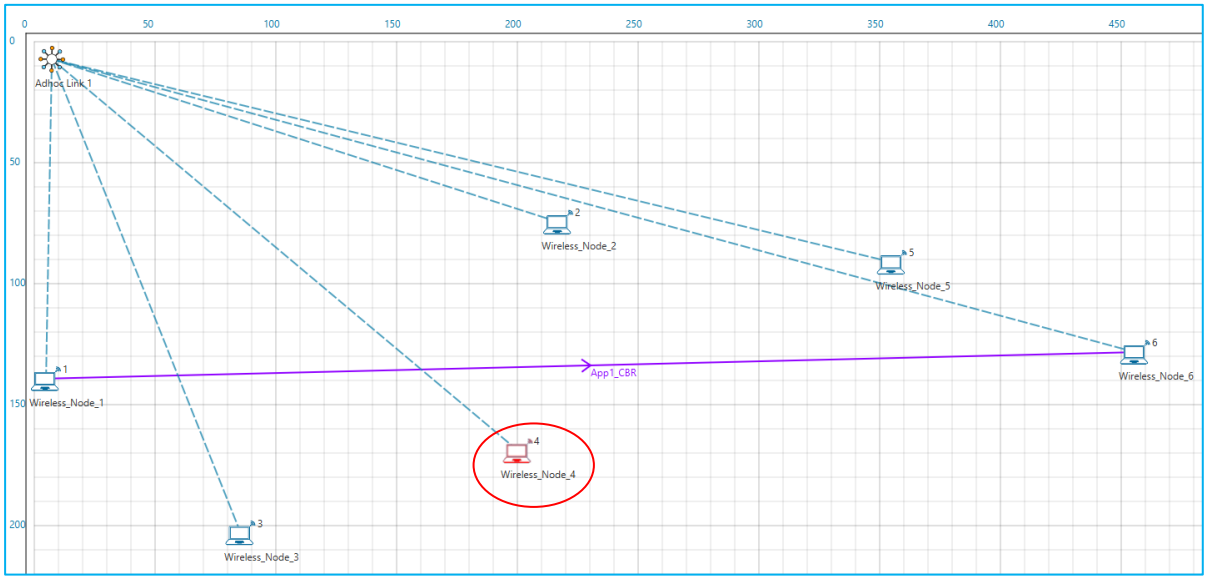
- The following steps illustrates the creation of the above sample configuration file:

- Create a network scenario in MANET with UDP running in the Transport Layer.

- Change the Network Layer Routing protocol to **AODV**.



- For example, you can create a scenario as shown in the following screenshot:



Scenario Steps: (Wireless Node 6)

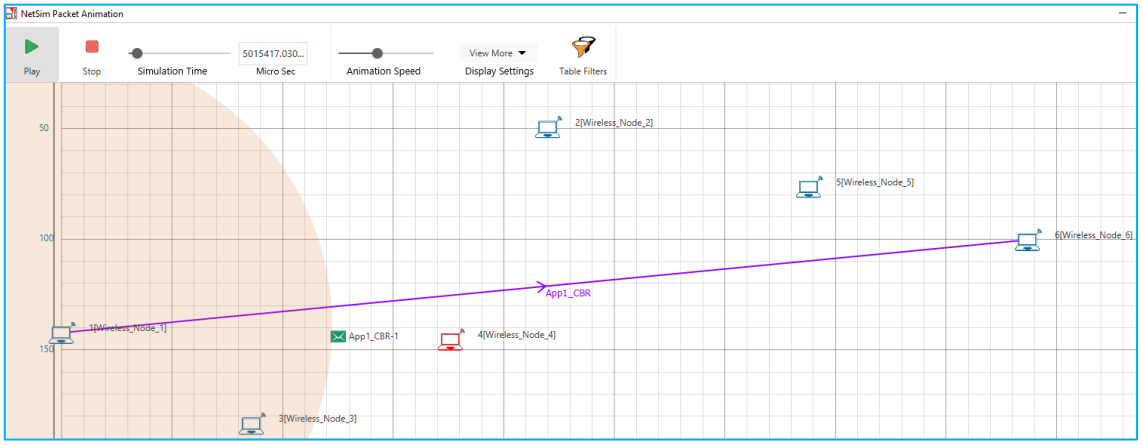
- Source – Device id 1 Destination
- Device id 6
- Sinkhole (Malicious node) Device id 4

Link properties (Adhoc Link1)

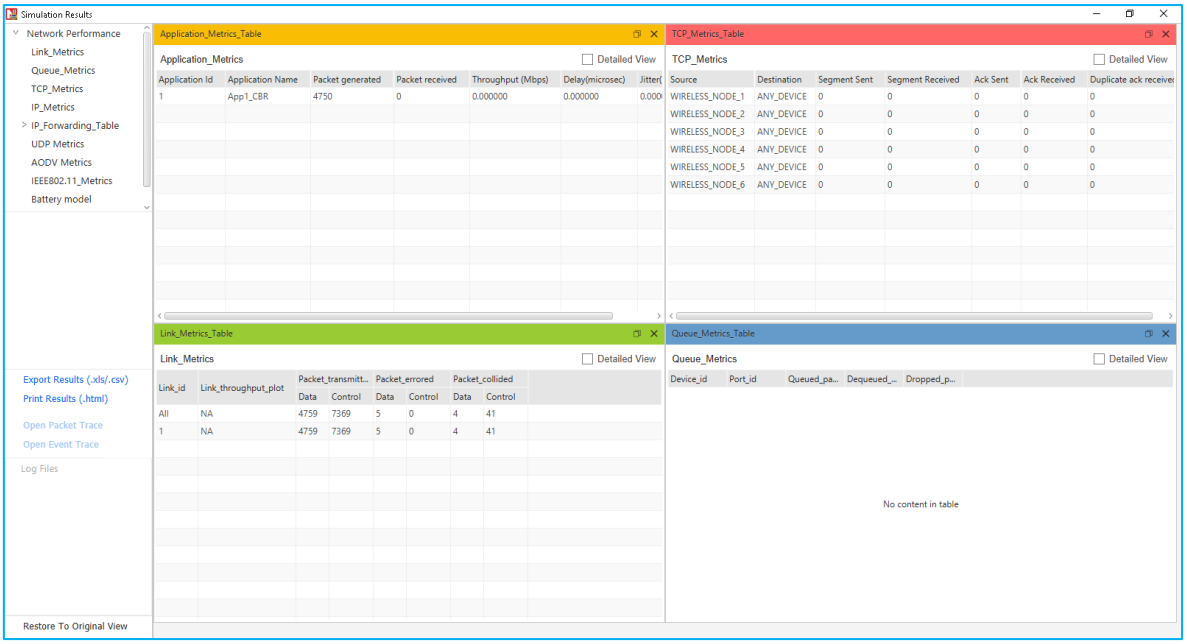
- Channel characteristics – Pathloss only
- Path Loss model – LOG DISTANCE
- Path Loss Exponent: 3

9. Run the Simulation for 100 seconds.

10. View the packet animation. You will find that the malicious node (Device id 4) gives Route Reply on receiving Route Request and attracts packets towards it. You will also find that the malicious node does not forward the packets that it receives.



11. This will have a direct impact on the Application Throughput which can be observed in the Application Metrics table present in NetSim Simulation Results window.


Simulation Results
Network Performance
Link_Metrics
Queue_Metrics
TCP_Metrics
IP_Metrics
IP_Forwarding_Table
UDP Metrics
AODV Metrics
IEEE802.11_Metrics
Battery model
Export Results (.xls/.csv)
Print Results (.html)
Open Packet Trace
Open Event Trace
Log Files
Restore To Original View
Application_Metrics_Table
Application_Metrics
Detailed View
Application Id
Application Name
Packet generated
Packet received
Throughput (Mbps)
Delay(microsec)
1
App1_CBR
4750
0
0.000000
0.000000
TCP_Metrics_Table
TCP_Metrics
Detailed View
Source
Destination
Segment Sent
Segment Received
Ack Sent
Ack Received
Duplicate ack receive
WIRELESS_NODE_1
WIRELESS_NODE_2
WIRELESS_NODE_3
WIRELESS_NODE_4
WIRELESS_NODE_5
WIRELESS_NODE_6
ANY_DEVICE
Link_Metrics_Table
Link_Metrics
Detailed View
Link_id
Link_throughput_plot
Packet_transmitt...
Packet_errored
Packet_collided
Data
Control
All
NA
Queue_Metrics_Table
Queue_Metrics
Detailed View
Device_id
Port_id
Queued_pa...
Dequeued_...
Dropped_p...
No content in table