Intrusion detection system for LEACH

Software: NetSim Standard v13.1 64-bit, Visual Studio 2019

Project Download Link:

https://github.com/NetSim-TETCOS/IDS_for_LEACH_v13.0/archive/refs/heads/main.zip

Follow the instructions specified in the following link to download and setup the Project in NetSim:

https://support.tetcos.com/en/support/solutions/articles/14000128666-downloading-and-setting-up-netsim-file-exchange-projects

Steps to simulate

1. Open the Source codes in Visual Studio by going to Your work-> Source Code and Clicking on Open code button in NetSim Home Screen window.
2. Right click on the solution in the solution explorer and select Rebuild.

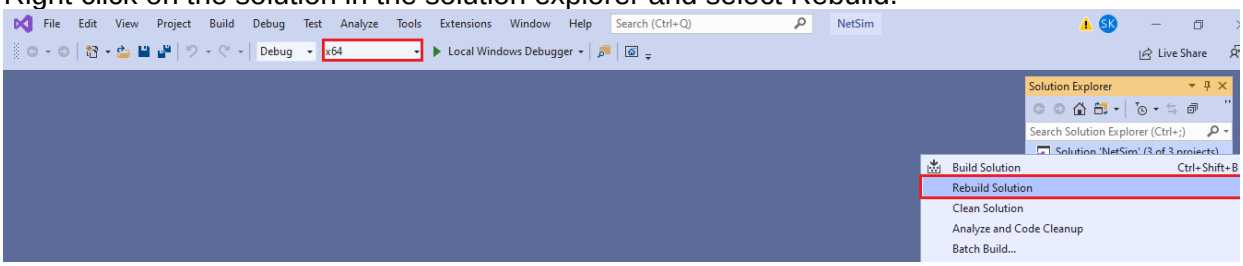


Figure 1: Screen shot of NetSim project source code in Visual Studio

3. Upon rebuilding, **libZigbee.dll** and **libDSR.dll** will automatically get updated in the respective binary folder of the current workspace.

Example

1. The **WorkSpace_IDS_in_LEACH** comes with a sample network configuration that are already saved. To open this example, go to Your work in the Home screen of NetSim and click on the **IDS_in_LEACH_Example** from the list of experiments.
2. This Network is created in WSN Network as per the Number of clusters and size of clusters that are set in the LEACH code. By default, the code runs for a scenario with 64 sensors uniformly placed, with the SINKNODE placed as per the screenshot shown below.

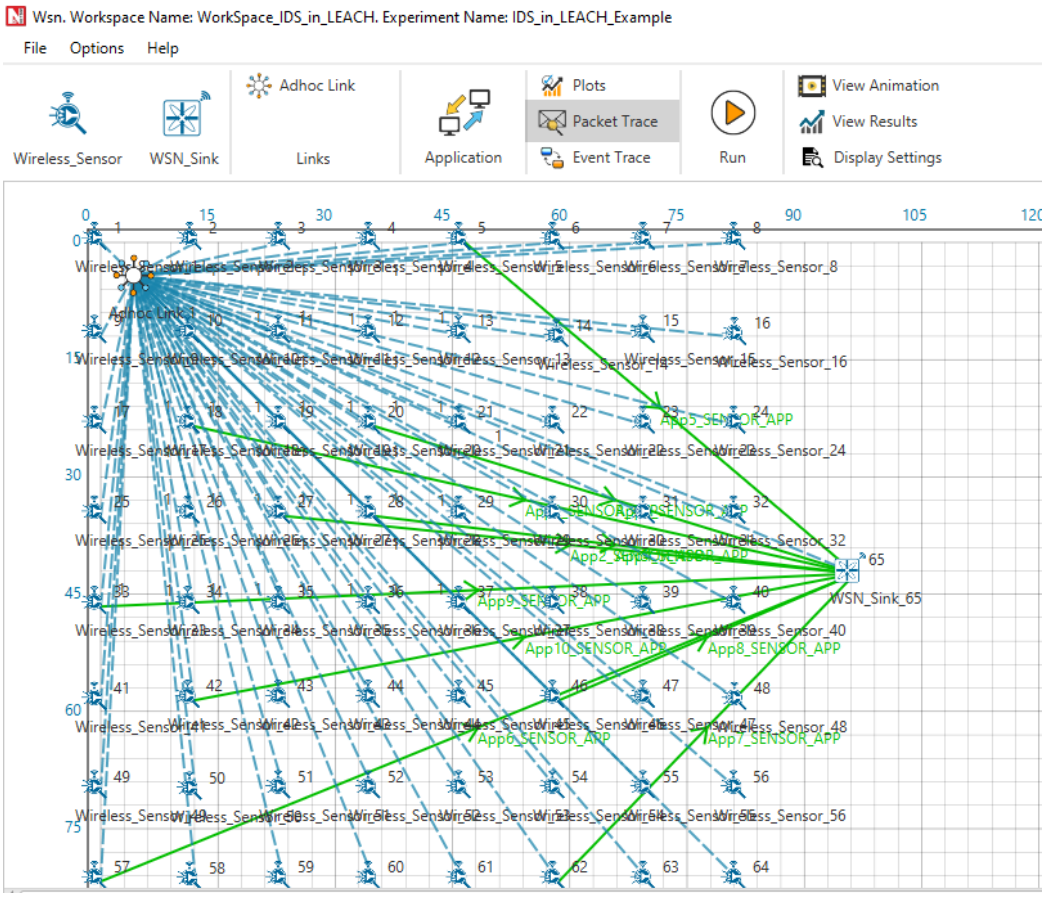
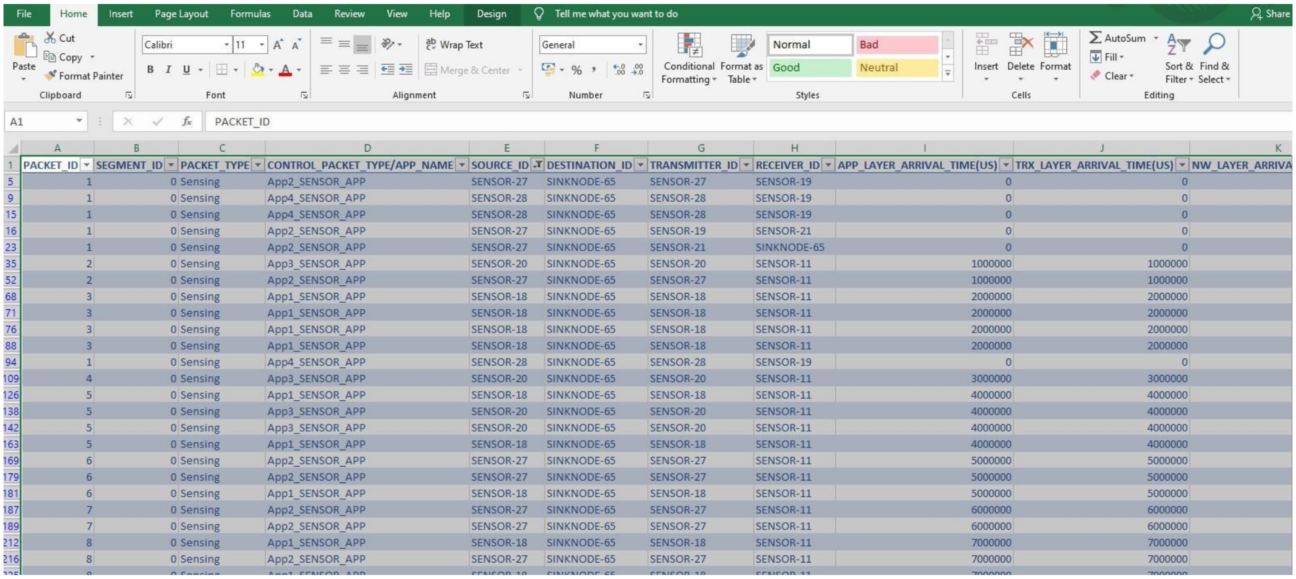


Figure 2: 64 sensors uniformly placed, with the SINKNODE

3. Wireless Link Properties
    - Channel Characteristics - Pathloss only
    - Path loss model - LOG_DISTANCE
    - Path loss exponent - 3
4. Run the simulation.

Results and discussion

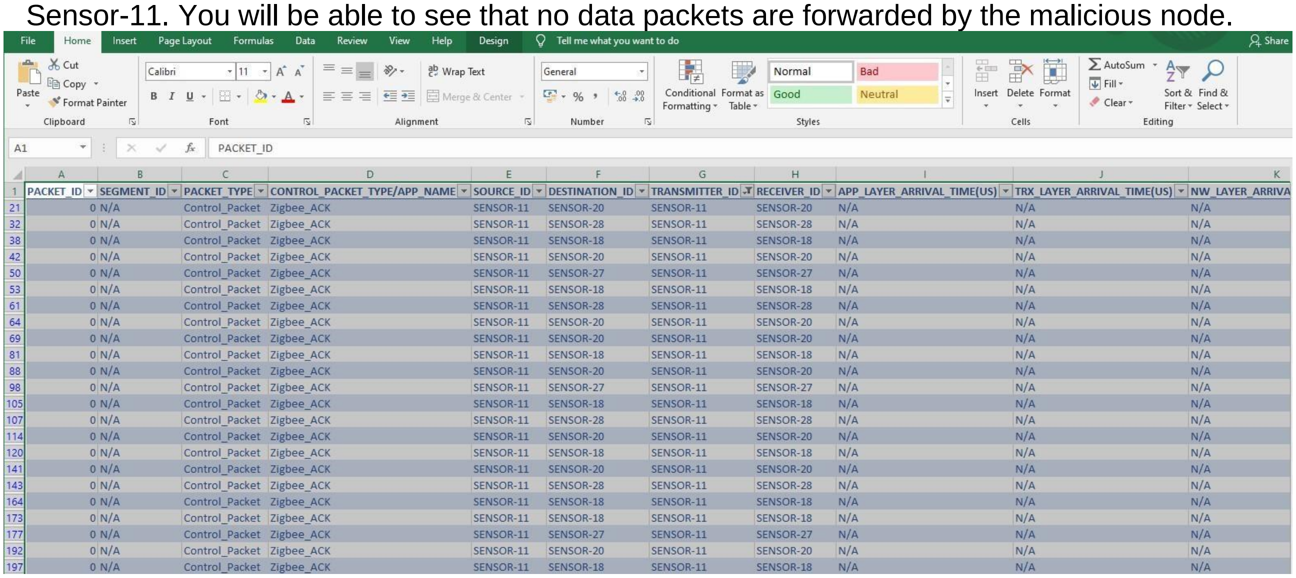
- View the packet animation. You will note that the sensors directly start transmitting packets without route establishment since the routes are statically defined in LEACH.
- You will also note that the cluster heads keep changing dynamically in Clusters 2, 3 and 4.
- In cluster1, initially the cluster members transmit packets to malicious node (device id 11) since it advertises false battery information to become a cluster head. Per the original code setting the Watchdog timer is set to 2 seconds and the failure threshold is set to 20 packets. You would notice that around 62 seconds, the malicious node is detected and then cluster head is elected dynamically based on the remaining energy of the sensor.
- This can be observed in Packet trace by applying filters to Source_ID column by selecting only Sensor-18, 20, 27 and 28. You will be able to see that the receiver id is sensor-11 from 1s till 62s of simulation time and then it is changed when it gets blacklisted.



	PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID	APP_LAYER_ARRIVAL_TIME(US)	TRX_LAYER_ARRIVAL_TIME(US)	NW_LAYER_ARRIVA
5	1	0	Sensing	App2_SENSOR_APP	SENSOR-27	SINKNODE-65	SENSOR-27	SENSOR-19	0	0	
9	1	0	Sensing	App4_SENSOR_APP	SENSOR-28	SINKNODE-65	SENSOR-28	SENSOR-19	0	0	
15	1	0	Sensing	App4_SENSOR_APP	SENSOR-28	SINKNODE-65	SENSOR-28	SENSOR-19	0	0	
16	1	0	Sensing	App2_SENSOR_APP	SENSOR-27	SINKNODE-65	SENSOR-19	SENSOR-21	0	0	
23	1	0	Sensing	App2_SENSOR_APP	SENSOR-27	SINKNODE-65	SENSOR-21	SINKNODE-65	0	0	
35	2	0	Sensing	App3_SENSOR_APP	SENSOR-20	SINKNODE-65	SENSOR-20	SENSOR-11	1000000	1000000	
52	2	0	Sensing	App2_SENSOR_APP	SENSOR-27	SINKNODE-65	SENSOR-27	SENSOR-11	1000000	1000000	
68	3	0	Sensing	App1_SENSOR_APP	SENSOR-18	SINKNODE-65	SENSOR-18	SENSOR-11	2000000	2000000	
71	3	0	Sensing	App1_SENSOR_APP	SENSOR-18	SINKNODE-65	SENSOR-18	SENSOR-11	2000000	2000000	
76	3	0	Sensing	App1_SENSOR_APP	SENSOR-18	SINKNODE-65	SENSOR-18	SENSOR-11	2000000	2000000	
88	3	0	Sensing	App1_SENSOR_APP	SENSOR-18	SINKNODE-65	SENSOR-18	SENSOR-11	2000000	2000000	
94	1	0	Sensing	App4_SENSOR_APP	SENSOR-28	SINKNODE-65	SENSOR-28	SENSOR-19	0	0	
109	4	0	Sensing	App3_SENSOR_APP	SENSOR-20	SINKNODE-65	SENSOR-20	SENSOR-11	3000000	3000000	
126	5	0	Sensing	App1_SENSOR_APP	SENSOR-18	SINKNODE-65	SENSOR-18	SENSOR-11	4000000	4000000	
138	5	0	Sensing	App3_SENSOR_APP	SENSOR-20	SINKNODE-65	SENSOR-20	SENSOR-11	4000000	4000000	
142	5	0	Sensing	App3_SENSOR_APP	SENSOR-20	SINKNODE-65	SENSOR-20	SENSOR-11	4000000	4000000	
163	5	0	Sensing	App1_SENSOR_APP	SENSOR-18	SINKNODE-65	SENSOR-18	SENSOR-11	4000000	4000000	
169	6	0	Sensing	App2_SENSOR_APP	SENSOR-27	SINKNODE-65	SENSOR-27	SENSOR-11	5000000	5000000	
179	6	0	Sensing	App2_SENSOR_APP	SENSOR-27	SINKNODE-65	SENSOR-27	SENSOR-11	5000000	5000000	
181	6	0	Sensing	App1_SENSOR_APP	SENSOR-18	SINKNODE-65	SENSOR-18	SENSOR-11	5000000	5000000	
187	7	0	Sensing	App2_SENSOR_APP	SENSOR-27	SINKNODE-65	SENSOR-27	SENSOR-11	6000000	6000000	
189	7	0	Sensing	App2_SENSOR_APP	SENSOR-27	SINKNODE-65	SENSOR-27	SENSOR-11	6000000	6000000	
212	8	0	Sensing	App1_SENSOR_APP	SENSOR-18	SINKNODE-65	SENSOR-18	SENSOR-11	7000000	7000000	
216	8	0	Sensing	App2_SENSOR_APP	SENSOR-27	SINKNODE-65	SENSOR-27	SENSOR-11	7000000	7000000	

Figure 3: NetSim Packet trace after filtering Sensor 18, 20, 27 and 28 as source ID

- Now undo filter in Source_Id column and apply filter to transmitter_Id column by selecting only Sensor-11. You will be able to see that no data packets are forwarded by the malicious node.



	PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID	APP_LAYER_ARRIVAL_TIME(US)	TRX_LAYER_ARRIVAL_TIME(US)	NW_LAYER_ARRIVA
21	0	N/A	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-20	SENSOR-11	SENSOR-20	N/A	N/A	N/A
32	0	N/A	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-28	SENSOR-11	SENSOR-28	N/A	N/A	N/A
38	0	N/A	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-18	SENSOR-11	SENSOR-18	N/A	N/A	N/A
42	0	N/A	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-20	SENSOR-11	SENSOR-20	N/A	N/A	N/A
50	0	N/A	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-27	SENSOR-11	SENSOR-27	N/A	N/A	N/A
53	0	N/A	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-18	SENSOR-11	SENSOR-18	N/A	N/A	N/A
61	0	N/A	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-28	SENSOR-11	SENSOR-28	N/A	N/A	N/A
64	0	N/A	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-20	SENSOR-11	SENSOR-20	N/A	N/A	N/A
69	0	N/A	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-20	SENSOR-11	SENSOR-20	N/A	N/A	N/A
81	0	N/A	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-18	SENSOR-11	SENSOR-18	N/A	N/A	N/A
88	0	N/A	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-20	SENSOR-11	SENSOR-20	N/A	N/A	N/A
98	0	N/A	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-27	SENSOR-11	SENSOR-27	N/A	N/A	N/A
105	0	N/A	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-18	SENSOR-11	SENSOR-18	N/A	N/A	N/A
107	0	N/A	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-28	SENSOR-11	SENSOR-28	N/A	N/A	N/A
114	0	N/A	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-20	SENSOR-11	SENSOR-20	N/A	N/A	N/A
120	0	N/A	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-18	SENSOR-11	SENSOR-18	N/A	N/A	N/A
141	0	N/A	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-20	SENSOR-11	SENSOR-20	N/A	N/A	N/A
143	0	N/A	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-28	SENSOR-11	SENSOR-28	N/A	N/A	N/A
164	0	N/A	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-18	SENSOR-11	SENSOR-18	N/A	N/A	N/A
173	0	N/A	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-18	SENSOR-11	SENSOR-18	N/A	N/A	N/A
177	0	N/A	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-27	SENSOR-11	SENSOR-27	N/A	N/A	N/A
192	0	N/A	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-20	SENSOR-11	SENSOR-20	N/A	N/A	N/A
197	0	N/A	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-18	SENSOR-11	SENSOR-18	N/A	N/A	N/A

Figure 4: Undo filter in Source_Id column and transmitter_Id column by selecting only Sensor-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. The throughput for applications 1, 2, 3 and 4 are less since the source ids belongs to cluster1 having malicious node (device id 11).
- 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.

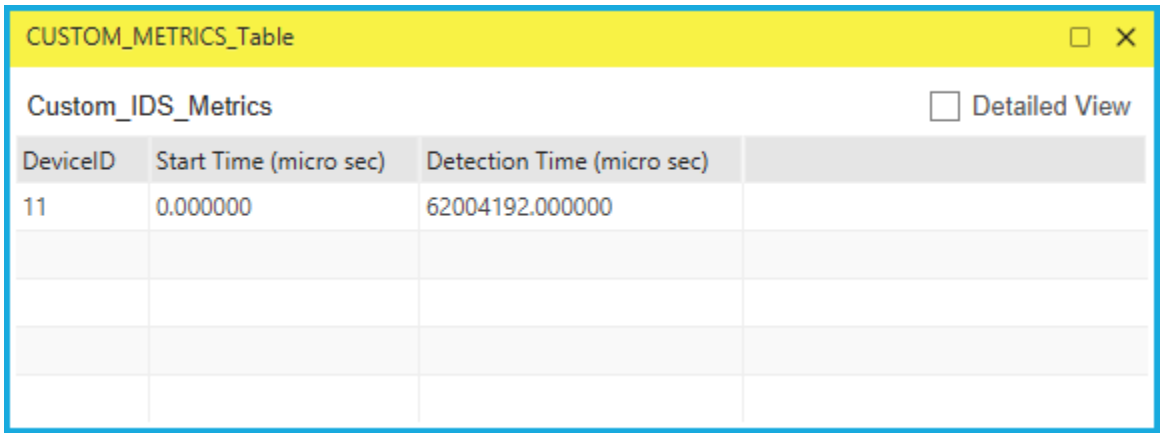


Figure 5: Dedicated Metrics for IDS

Files Used in this project

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.
- Clustering and cluster head election is explained in LEACH.c file.
- To detect the intruder and to send data via a new route, the following files are added in DSR and Zigbee:

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 blacklisted 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 Zigbee operating in Layer 3.
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 does not 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 hope is marked by the current node as a malicious node.