Intrusion detection system for LEACH

Software: NetSim Standard v14.4, Visual Studio 2022

Project Download Link:

https://github.com/NetSim-TETCOS/IDS_in_LEACH-v14.4/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

1 Introduction

An Intrusion Detection System (IDS) is a security mechanism designed to monitor network traffic and detect unauthorized access or malicious activities. The Low-Energy Adaptive Clustering Hierarchy (LEACH) is a clustering-based protocol that divides the sensor nodes into clusters, with one node acting as the cluster head, responsible for collecting data from the cluster members this protocol is used in wireless sensor networks to reduce energy consumption and prolong network lifetime.

In the context of LEACH, an IDS can help to detect and prevent attacks that target the network's nodes, such as denial-of-service (DoS) attacks, sinkhole attacks, and compromised node attacks. IDS for the LEACH is a security solution designed specifically for WSNs that use the LEACH protocol. It works by monitoring the network traffic, analyzing it for anomalies or malicious activities, and taking appropriate actions to prevent them.

The goal of an IDS for LEACH in Netsim is to provide an additional layer of security to protect the network and the data transmitted through it. By detecting and responding to potential threats in a timely manner, the system can help prevent attacks and ensure the integrity and availability of the network.

2 Example

1. The **IDS_in_LEACH** workspace 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-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.

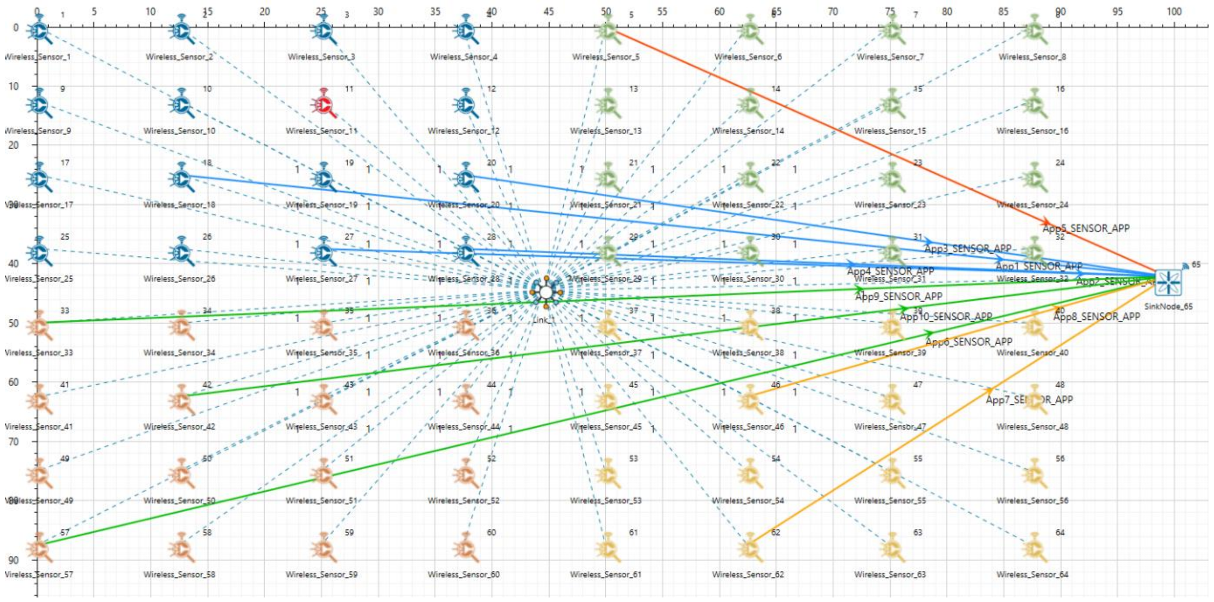
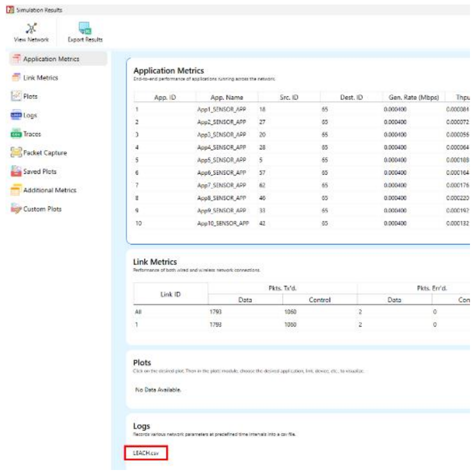


Figure 1: 64 sensors uniformly placed, with the SINKNODE

3. Wireless Link Properties
    - Channel Characteristics - Pathloss only
    - Path loss model - LOG_DISTANCE
    - Path loss exponent – 2.5
4. Run the simulation for 100 seconds.

3 Results and discussion

- You will note that the sensors directly start transmitting packets without route establishment since the routes are statically defined in LEACH.
- In LEACH.csv file in the result dashboard, you will also note that the cluster heads keep changing dynamically in Clusters 2, 3 and 4.



Application Metrics
End-to-end performance of applications running across the network.

App. ID	App. Name	Src. ID	Dest. ID	Gen. Rate (Mbps)	Thpu
1	App1_SENSOR_APP	18	65	0.000400	0.000084
2	App2_SENSOR_APP	27	65	0.000400	0.000072
3	App3_SENSOR_APP	20	65	0.000400	0.000056
4	App4_SENSOR_APP	28	65	0.000400	0.000064
5	App5_SENSOR_APP	5	65	0.000400	0.000188
6	App6_SENSOR_APP	57	65	0.000400	0.000164
7	App7_SENSOR_APP	62	65	0.000400	0.000176
8	App8_SENSOR_APP	46	65	0.000400	0.000220
9	App9_SENSOR_APP	33	65	0.000400	0.000192
10	App10_SENSOR_APP	42	65	0.000400	0.000132

Link Metrics
Performance of both wired and wireless network connections.

Link ID	Pkts. Tx'd.		Pkts. Err'd.	
	Data	Control	Data	Con
All	1793	1060	2	0
1	1793	1060	2	0

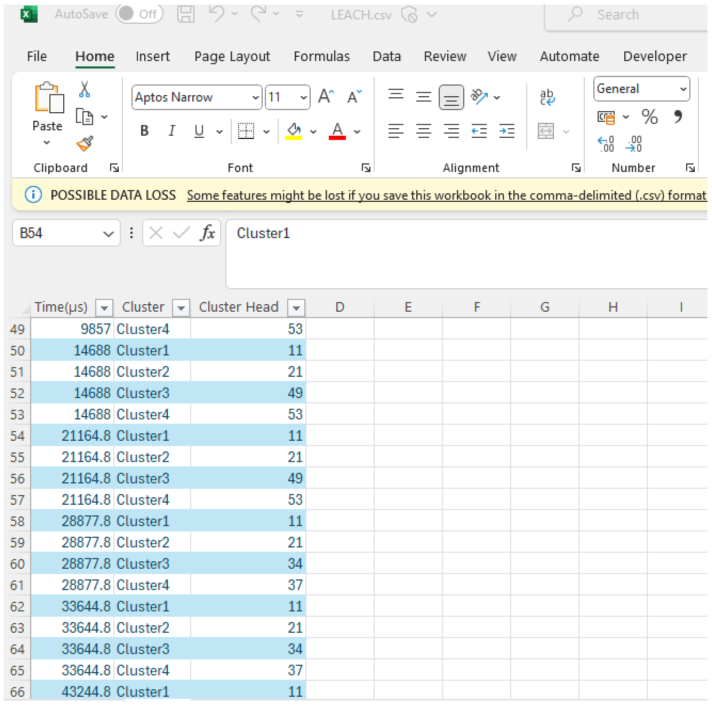
Plots
Click on the desired plot. Then in the plots module, choose the desired application, link, device, etc., to visualize.

No Data Available.

Logs
Records various network parameters at predefined time intervals into a csv file.

LEACH.csv

Figure 2: LEACH log file in result dashboard.

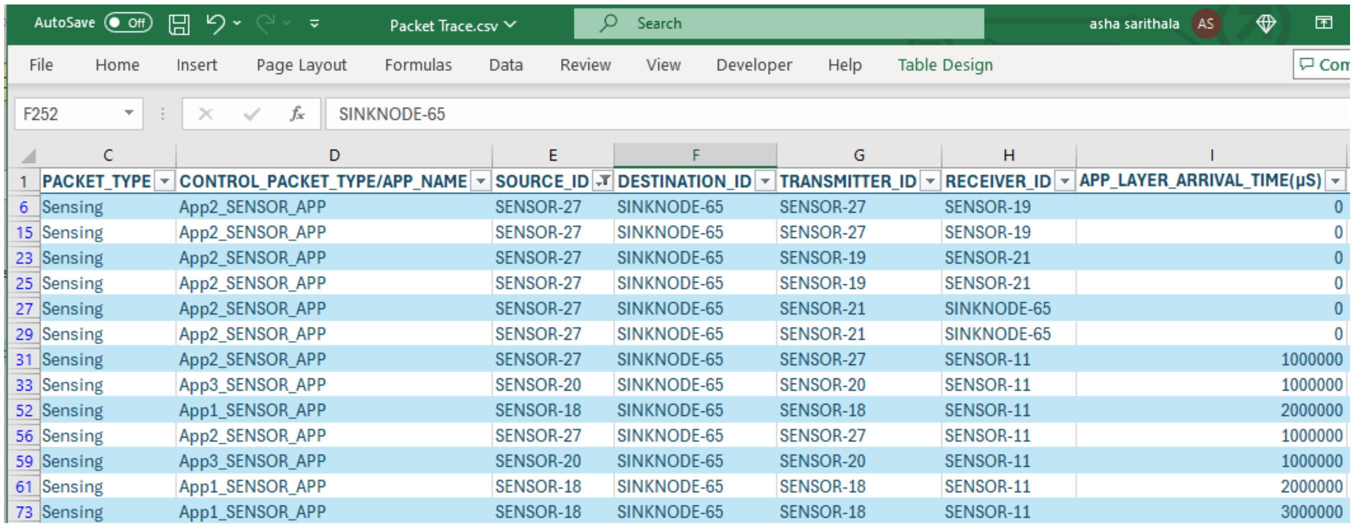


POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format

	Time(µs)	Cluster	Cluster Head
49	9857	Cluster4	53
50	14688	Cluster1	11
51	14688	Cluster2	21
52	14688	Cluster3	49
53	14688	Cluster4	53
54	21164.8	Cluster1	11
55	21164.8	Cluster2	21
56	21164.8	Cluster3	49
57	21164.8	Cluster4	53
58	28877.8	Cluster1	11
59	28877.8	Cluster2	21
60	28877.8	Cluster3	34
61	28877.8	Cluster4	37
62	33644.8	Cluster1	11
63	33644.8	Cluster2	21
64	33644.8	Cluster3	34
65	33644.8	Cluster4	37
66	43244.8	Cluster1	11

Figure 3: LEACH log file screenshot showing the cluster heads at different time

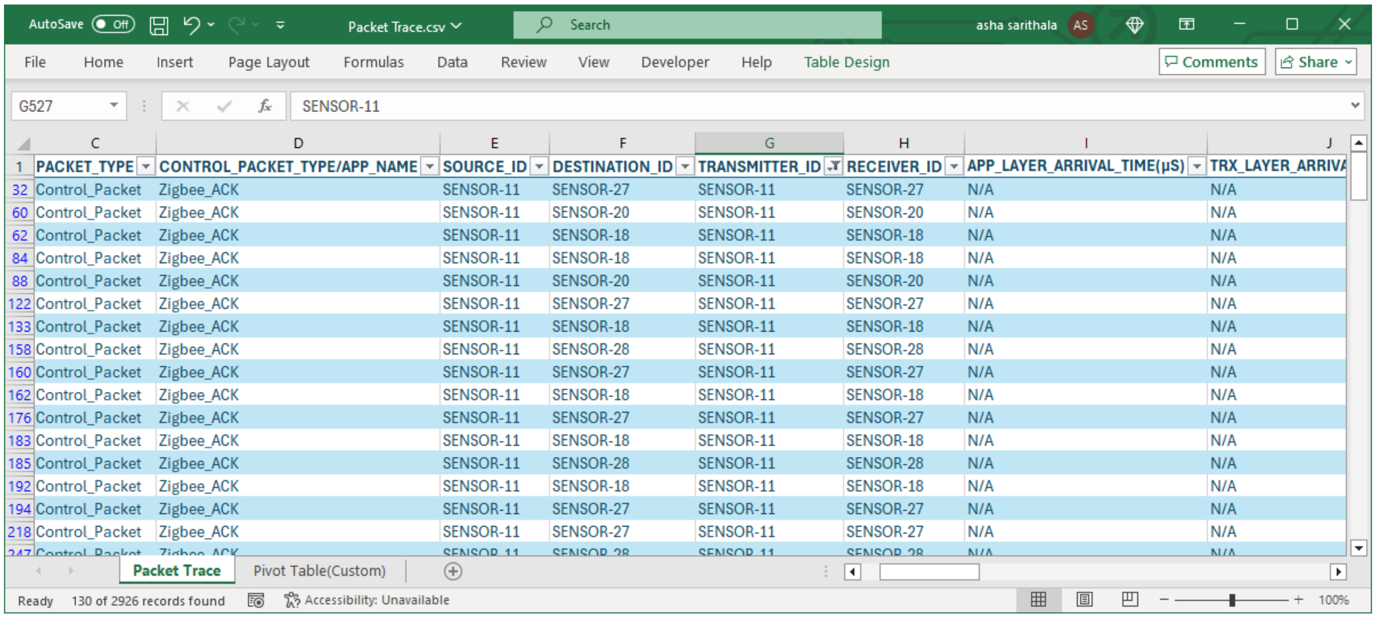
- In cluster1, initially the cluster members transmit packets to malicious node (device id 11) since it advertises false battery information to become a cluster head. Per the original code setting the Watchdog timer is set to 2 seconds and the failure threshold is set to 20 packets. You would notice that around 59 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 as we are checking on cluster 1 where malicious node is present and application is only configured from this particular node. You will be able to see that the receiver id is sensor-11 from 1s till 59s of simulation time and then it is changed when it gets blacklisted.


AutoSave Off
Packet Trace.csv
Search
asha sarithala AS
File Home Insert Page Layout Formulas Data Review View Developer Help Table Design
Com
F252
SINKNODE-65

	C	D	E	F	G	H	I
1	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID	APP_LAYER_ARRIVAL_TIME(µS)
6	Sensing	App2_SENSOR_APP	SENSOR-27	SINKNODE-65	SENSOR-27	SENSOR-19	0
15	Sensing	App2_SENSOR_APP	SENSOR-27	SINKNODE-65	SENSOR-27	SENSOR-19	0
23	Sensing	App2_SENSOR_APP	SENSOR-27	SINKNODE-65	SENSOR-19	SENSOR-21	0
25	Sensing	App2_SENSOR_APP	SENSOR-27	SINKNODE-65	SENSOR-19	SENSOR-21	0
27	Sensing	App2_SENSOR_APP	SENSOR-27	SINKNODE-65	SENSOR-21	SINKNODE-65	0
29	Sensing	App2_SENSOR_APP	SENSOR-27	SINKNODE-65	SENSOR-21	SINKNODE-65	0
31	Sensing	App2_SENSOR_APP	SENSOR-27	SINKNODE-65	SENSOR-27	SENSOR-11	1000000
33	Sensing	App3_SENSOR_APP	SENSOR-20	SINKNODE-65	SENSOR-20	SENSOR-11	1000000
52	Sensing	App1_SENSOR_APP	SENSOR-18	SINKNODE-65	SENSOR-18	SENSOR-11	2000000
56	Sensing	App2_SENSOR_APP	SENSOR-27	SINKNODE-65	SENSOR-27	SENSOR-11	1000000
59	Sensing	App3_SENSOR_APP	SENSOR-20	SINKNODE-65	SENSOR-20	SENSOR-11	1000000
61	Sensing	App1_SENSOR_APP	SENSOR-18	SINKNODE-65	SENSOR-18	SENSOR-11	2000000
73	Sensing	App1_SENSOR_APP	SENSOR-18	SINKNODE-65	SENSOR-18	SENSOR-11	3000000

Figure 4: NetSim Packet trace after filtering Sensor 18, 20, 27 and 28 as source ID

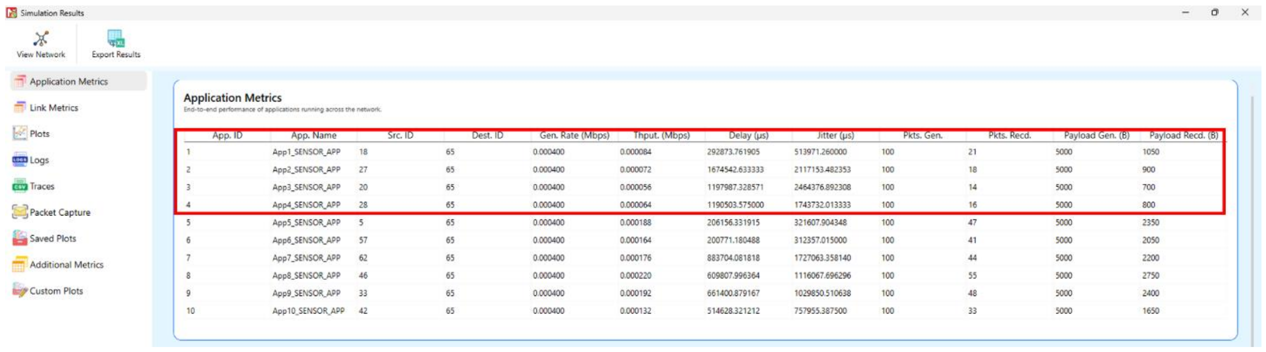
- Now undo filter in Source_Id column and apply filter to transmitter_Id column by selecting only Sensor-11. You will be able to see that no data packets are forwarded by the malicious node.


AutoSave Off
Packet Trace.csv
Search
asha sarithala AS
File Home Insert Page Layout Formulas Data Review View Developer Help Table Design
Comments Share
G527
SENSOR-11
Packet Trace Pivot Table(Custom)
Ready 130 of 2926 records found
Accessibility: Unavailable
100%

	C	D	E	F	G	H	I	J
1	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID	APP_LAYER_ARRIVAL_TIME(µS)	TRX_LAYER_ARRIVA
32	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-27	SENSOR-11	SENSOR-27	N/A	N/A
60	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-20	SENSOR-11	SENSOR-20	N/A	N/A
62	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-18	SENSOR-11	SENSOR-18	N/A	N/A
84	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-18	SENSOR-11	SENSOR-18	N/A	N/A
88	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-20	SENSOR-11	SENSOR-20	N/A	N/A
122	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-27	SENSOR-11	SENSOR-27	N/A	N/A
133	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-18	SENSOR-11	SENSOR-18	N/A	N/A
158	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-28	SENSOR-11	SENSOR-28	N/A	N/A
160	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-27	SENSOR-11	SENSOR-27	N/A	N/A
162	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-18	SENSOR-11	SENSOR-18	N/A	N/A
176	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-27	SENSOR-11	SENSOR-27	N/A	N/A
183	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-18	SENSOR-11	SENSOR-18	N/A	N/A
185	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-28	SENSOR-11	SENSOR-28	N/A	N/A
192	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-18	SENSOR-11	SENSOR-18	N/A	N/A
194	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-27	SENSOR-11	SENSOR-27	N/A	N/A
218	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-27	SENSOR-11	SENSOR-27	N/A	N/A
247	Control_Packet	Zigbee_ACK	SENSOR-11	SENSOR-28	SENSOR-11	SENSOR-28	N/A	N/A

Figure 5: 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).



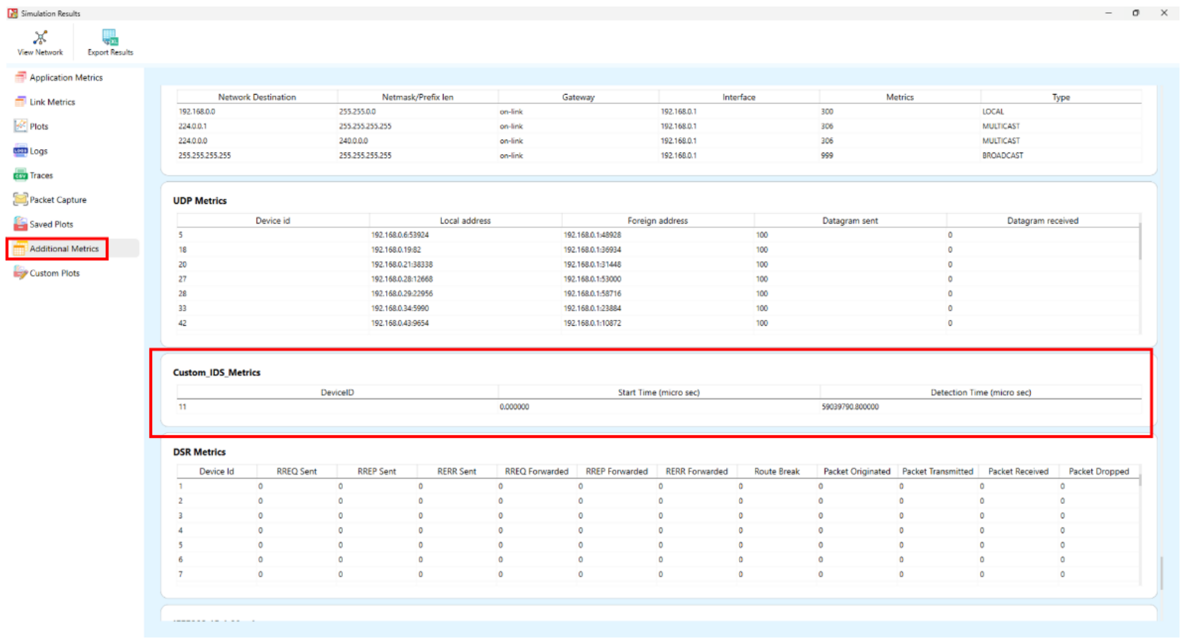
Application Metrics

End-to-end performance of applications running across the network.

App. ID	App. Name	Src. ID	Dest. ID	Gen. Rate (Mbps)	Thput. (Mbps)	Delay (µs)	Jitter (µs)	Pkts. Gen.	Pkts. Recd.	Payload Gen. (B)	Payload Recd. (B)
1	App1_SENSOR_APP	18	65	0.000400	0.000084	292873.761905	513971.260000	100	21	5000	1050
2	App2_SENSOR_APP	27	65	0.000400	0.000072	1674542.633333	2117153.482353	100	18	5000	900
3	App3_SENSOR_APP	20	65	0.000400	0.000056	1197987.328571	2464376.892308	100	14	5000	700
4	App4_SENSOR_APP	28	65	0.000400	0.000064	1190503.575000	1743732.013333	100	16	5000	800
5	App5_SENSOR_APP	5	65	0.000400	0.000188	206156.331915	321607.904348	100	47	5000	2350
6	App6_SENSOR_APP	57	65	0.000400	0.000164	200771.180488	312357.015000	100	41	5000	2050
7	App7_SENSOR_APP	62	65	0.000400	0.000176	883704.081818	1727063.358140	100	44	5000	2200
8	App8_SENSOR_APP	46	65	0.000400	0.000220	609807.996364	1116067.696296	100	55	5000	2750
9	App9_SENSOR_APP	33	65	0.000400	0.000192	661400.879167	1029850.510638	100	48	5000	2400
10	App10_SENSOR_APP	42	65	0.000400	0.000132	514628.321212	757955.387500	100	33	5000	1650

Figure 6: Application Metrics Window

- The time at which a malicious node is detected can be obtained from the CUSTOM METRICS (IDS METRICS). These metrics can be accessed by navigating to the additional metrics section, where you can find the start time (the time from which a node becomes malicious) and the detection time (the time at which the node was added to the blacklist).



Network Destination	Netmask/Prefix len	Gateway	Interface	Metrics	Type
192.168.0.0	255.255.0.0	on-link	192.168.0.1	300	LOCAL
224.0.0.1	255.255.255.255	on-link	192.168.0.1	306	MULTICAST
224.0.0.0	240.0.0.0	on-link	192.168.0.1	306	MULTICAST
255.255.255.255	255.255.255.255	on-link	192.168.0.1	999	BROADCAST

UDP Metrics

Device id	Local address	Foreign address	Datagram sent	Datagram received
5	192.168.0.6:53924	192.168.0.1:48928	100	0
18	192.168.0.19:82	192.168.0.1:36934	100	0
20	192.168.0.21:38338	192.168.0.1:31448	100	0
27	192.168.0.28:12668	192.168.0.1:53000	100	0
28	192.168.0.29:22956	192.168.0.1:58716	100	0
33	192.168.0.34:5990	192.168.0.1:23884	100	0
42	192.168.0.43:9654	192.168.0.1:10872	100	0

Custom_IDS_Metrics

DeviceID	Start Time (micro sec)	Detection Time (micro sec)
11	0.000000	59039790.800000

DSR Metrics

Device Id	RREQ Sent	RREP Sent	RERR Sent	RREQ Forwarded	RREP Forwarded	RERR Forwarded	Route Break	Packet Originated	Packet Transmitted	Packet Received	Packet Dropped
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0

Figure 7: 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 which is present in DSR Project.
- Clustering and cluster head election is explained in LEACH.c file which is present in DSR Project.

1. For this implementation of LEACH, the number of Clusters is fixed as 4 and all the 4 clusters are equal. If the user wants to change it, then he/she must also change the static routing for the Cluster Heads and the ClusterElement array accordingly in **leach.c**

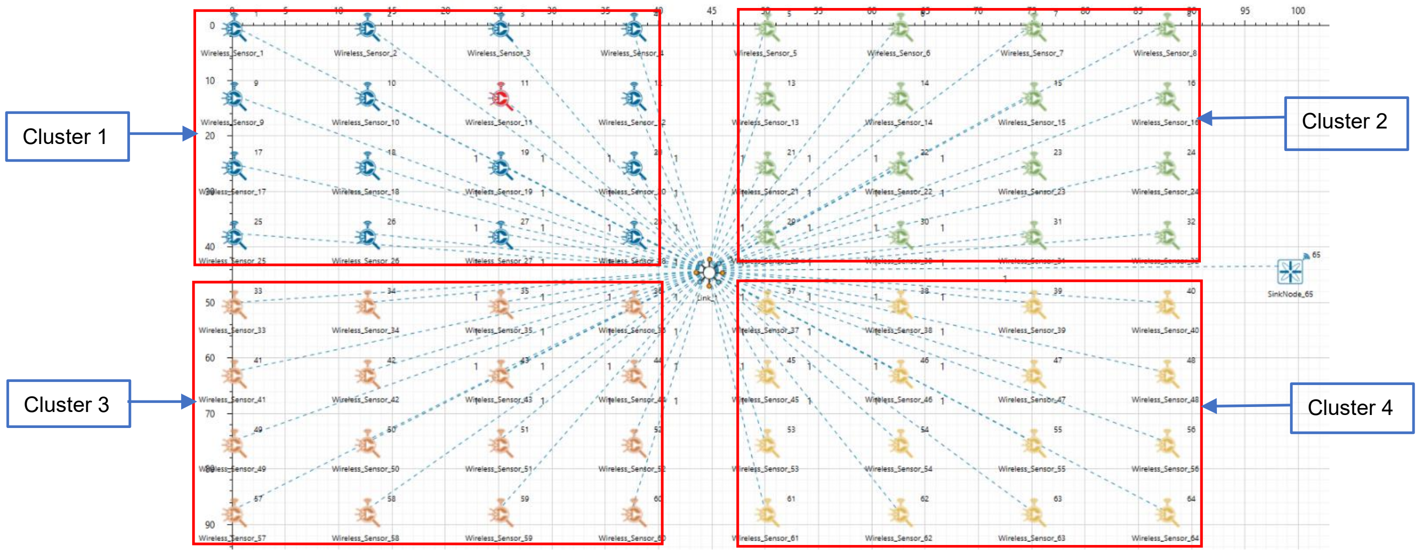
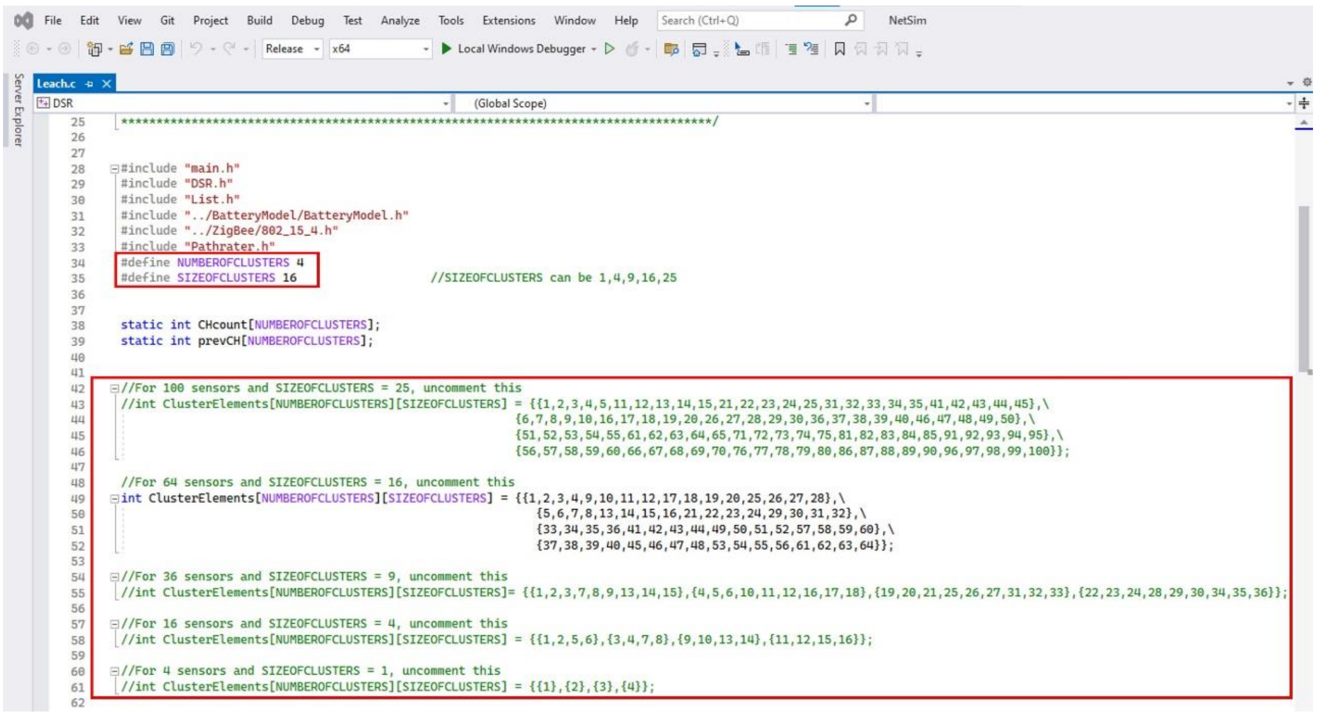


Figure 8: Sensor nodes present in each Clusters

To make 4 equal clusters the number of sensors must be 4,16,36,64,100. Depending on the number of sensors, the ClusterElements array must be defined. Here, it has been defined and commented for 4,16,36,64,100 sensors. Uncomment the one you want to use.



```
**********************************************************************************/


#include "main.h"
#include "DSR.h"
#include "List.h"
#include "../BatteryModel/BatteryModel.h"
#include "../ZigBee/802_15_4.h"
#include "Pathrater.h"
#define NUMBEROFCLUSTERS 4
#define SIZEOFCLUSTERS 16              //SIZEOFCLUSTERS can be 1,4,9,16,25


static int CHcount[NUMBEROFCLUSTERS];
static int prevCH[NUMBEROFCLUSTERS];


//For 100 sensors and SIZEOFCLUSTERS = 25, uncomment this
//int ClusterElements[NUMBEROFCLUSTERS][SIZEOFCLUSTERS] = {{1,2,3,4,5,11,12,13,14,15,21,22,23,24,25,31,32,33,34,35,41,42,43,44,45},\
                                                        {6,7,8,9,10,16,17,18,19,20,26,27,28,29,30,36,37,38,39,40,46,47,48,49,50},\
                                                        {51,52,53,54,55,61,62,63,64,65,71,72,73,74,75,81,82,83,84,85,91,92,93,94,95},\
                                                        {56,57,58,59,60,66,67,68,69,70,76,77,78,79,80,86,87,88,89,90,96,97,98,99,100}};

//For 64 sensors and SIZEOFCLUSTERS = 16, uncomment this
int ClusterElements[NUMBEROFCLUSTERS][SIZEOFCLUSTERS] = {{1,2,3,4,9,10,11,12,17,18,19,20,25,26,27,28},\
                                                           {5,6,7,8,13,14,15,16,21,22,23,24,29,30,31,32},\
                                                           {33,34,35,36,41,42,43,44,49,50,51,52,57,58,59,60},\
                                                           {37,38,39,40,45,46,47,48,53,54,55,56,61,62,63,64}};

//For 36 sensors and SIZEOFCLUSTERS = 9, uncomment this
//int ClusterElements[NUMBEROFCLUSTERS][SIZEOFCLUSTERS]= {{1,2,3,7,8,9,13,14,15},{4,5,6,10,11,12,16,17,18},{19,20,21,25,26,27,31,32,33},{22,23,24,28,29,30,34,35,36}};

//For 16 sensors and SIZEOFCLUSTERS = 4, uncomment this
//int ClusterElements[NUMBEROFCLUSTERS][SIZEOFCLUSTERS] = {{1,2,5,6},{3,4,7,8},{9,10,13,14},{11,12,15,16}};

//For 4 sensors and SIZEOFCLUSTERS = 1, uncomment this
//int ClusterElements[NUMBEROFCLUSTERS][SIZEOFCLUSTERS] = {{1},{2},{3},{4}};

```

Figure 9: Leach.c file present in DSR project

- To detect the intruder and to send data via a new route, the following files are added in DSR(Pathrater.c) and Zigbee(Watchdog.c)

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.