

Localization in WSN

Software: NetSim Standard v13.3, Microsoft Visual Studio 2022

Project Download Link:

<https://github.com/NetSim-TETCOS/Localization-in-WSN-v13.3/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>

Introduction

Localization is the process of finding the physical or relative location of a sensor node as data and information are useless if the nodes have no idea of their geographical positions. GPS (Global positioning system) is the simplest method for localization of nodes, but it becomes expensive if large number of nodes exists in each network.

Anchor Nodes

Sensor nodes with known location information are called “Anchor nodes”. Typically, anchor nodes obtain their location information by using a global positioning system (GPS), or by manually being placed at defined coordinates.

Unknown Nodes (Non-Anchor nodes)

Sensor nodes with unknown location information are called “Non-Anchor nodes” or “Unknown nodes”. Localization is estimated through communication between localized node and unknown node for determining their geometrical placement or position. Location is determined by means of distance and angle between nodes.

Trilateration

Trilateration is the estimation of the position of a point unambiguously based on the measurements of distances from three or more known reference locations.

In NetSim we assume the position of three sensors $S_1(x_1, y_1)$, $S_2(x_2, y_2)$ and $S_3(x_3, y_3)$ are known. Let us assume the sensor U whose position we seek to determine is at some (x, y) .

We also know that the distance $S_1 \rightarrow U = r_1$, $S_2 \rightarrow U = r_2$ and $S_3 \rightarrow U = r_3$. Then we get the set of equations

$$(x - x_1)^2 + (y - y_1)^2 = r_1^2$$

$$(x - x_2)^2 + (y - y_2)^2 = r_2^2$$

$$(x - x_3)^2 + (y - y_3)^2 = r_3^2$$

How are r_1 , r_2 and r_3 known (or obtained)? Generally, this is based on received signal strength (RSS), which is a function of radial distance between transmitter and receiver devices. If we make the assumption that of (i) no shadow fading and (ii) no interference, then we have a deterministic function between RSS and distance. For example, this could be based on the log distance pathloss model.

Expanding out the squares in the equations 1, 2 and 3 above

$$x^2 - 2xx_1 + x_1^2 + y^2 - 2yy_1 + y_1^2 = r_1^2$$

$$x^2 - 2xx_2 + x_2^2 + y^2 - 2yy_2 + y_2^2 = r_2^2$$

$$x^2 - 2xx_3 + x_3^2 + y^2 - 2yy_3 + y_3^2 = r_3^2$$

Expand out the squares and subtract the second equation from the first

$$2x(x_2 - x_1) + 2y(y_2 - y_1) = r_1^2 - r_2^2 - x_1^2 - y_1^2 + x_2^2 + y_2^2$$

and third equation from second, we get.

$$2x(x_3 - x_2) + 2y(y_3 - y_2) = r_2^2 - r_3^2 - x_2^2 - y_2^2 + x_3^2 + y_3^2$$

Use the distance equation. If your unknown point is (x, y) and known points are (xi, yi) which are at distances ri from unknown point, then you get three equations:

This is a system of two equations with two unknowns:

$$Ax + By = C$$

$$Dx + Ey = F$$

The values of x and y is obtained from the below equations:

$$x = \frac{CE - FB}{EA - BD}$$

$$y = \frac{CD - AF}{BD - AE}$$

Computing r_1, r_2 and r_3

The distance between the anchor nodes and the sensor whose position is known, is computed using the NetSim API

DEVICE_DISTANCE (Dev1, Dev2)

where **Dev1** is the anchor node and **Dev2** is the sensor, whose position is known.

Usage of this API is possible since the sensor (whose position is unknown) is also “within” the NetSim simulation environment. Hence, the simplification is carried out for this representative file exchange project.

A better method would be to compute the signal strength at the sensor (whose position is unknown) from all three anchor nodes namely S_1, S_2 and S_3 . Then the distances r_1, r_2 and r_3 can be obtained by applying the appropriate pathloss equation.

Time of arrival

An alternate approach to computing distance is to use Time-of-arrival. NetSim currently

assumes NIL propagation delay between any two nodes in a Wireless sensor network. This is because the time taken for an electromagnetic signal to propagate would be of the order $\frac{10m}{3 \times 10^8 m/s} = 3.33 \times 10^{-8} s$ which is around one hundredth ($\frac{1}{100} = 0.01$) of a microsecond.

On the other hand, a packet transmission time is around $\frac{50(B) \times 8}{250 \times 10^3 bps} = 1.6 \times 10^{-3} s$. This is of the order of thousands (1000s) of microseconds.

A small deviation would lead to extremely large errors and hence this example does not explore the use time of arrival for purposes of distance estimation.

Example

1. The WorkSpace_Localization_WSN comes with a sample network configuration that is already saved. To open this example, go to Your work in the Home screen of NetSim and click on the Localization_in_WSN_Example from the list of experiments.

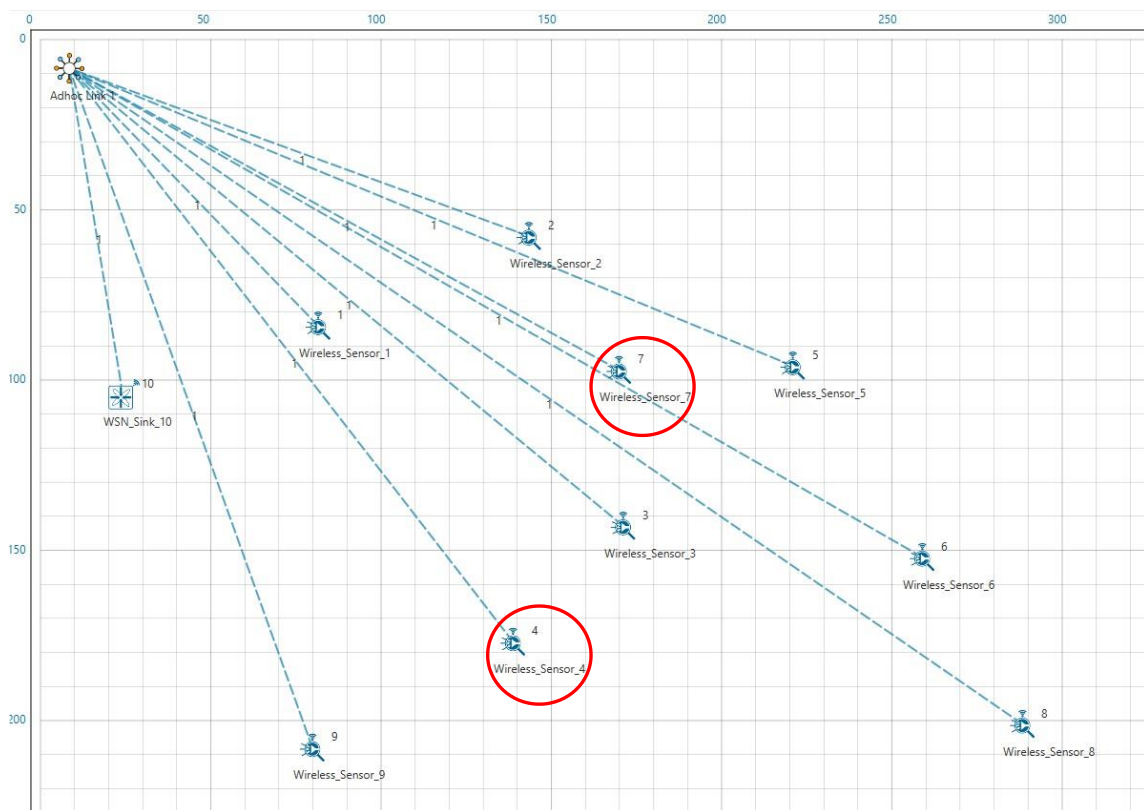


Figure 1: Network scenario with representation of unknown nodes

2. Run simulation after the network scenario gets loaded.

Results and discussion

After simulation, **localisation.txt** file can be viewed from log files section of results dashboard window in NetSim. The localisation.txt file logs the unknown node IDs, received powers from all anchor nodes to unknown nodes, anchor node IDs based on highest received powers and the position or coordinates of the unknown nodes with variation in time as shown below.

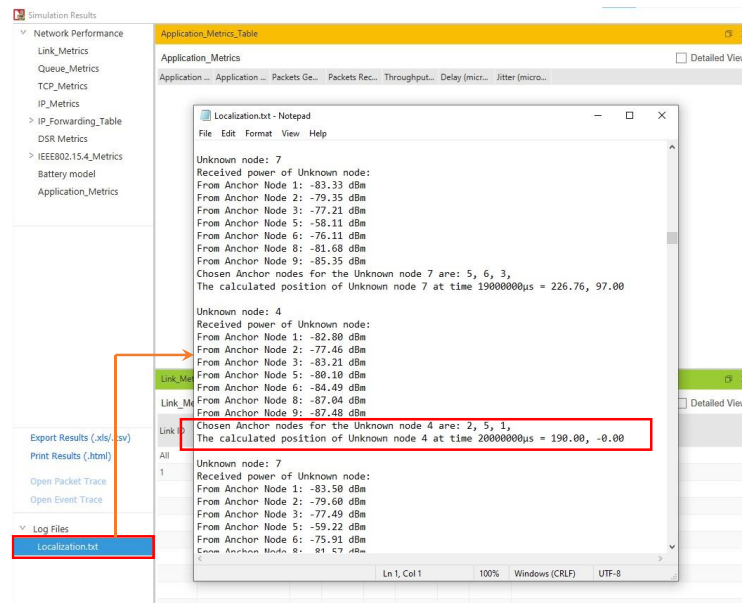


Figure 2: localization.txt log file present in results dashboard

Appendix: NetSim source code modifications

Localization in NetSim

- To implement Localization, we have added **Localisation.c** file in Zigbee project. The file contains the following functions:
 - `int fn_NetSim_localisation();` //This function is used to find the anchor nodes based on the highest received powers received at unknown sensors from anchor nodes.
 - `int fn_NetSim_trilateration_method();` //This function is used to implement the trilateration method to calculate the position / location of the unknown sensor.
 - `bool IsUnknownNode();` //This function is used to check whether the given node is unknown node or not.
 - `bool determine_anchor_node();` //This function is used to check whether the given node is anchor node or not.

Set unknown node number in Localisation.c

/* Users can give their own unknown node IDs and unknown node count in **Localisation.c** file. NetSim knows all the positions of sensor nodes. Localization is used to find the position of unknown nodes and then comparing this position with NetSim sensor positions */

```
// Set up unknown node
int unknown_node_count =2;
int unknown_node_IDs[10] = {4, 7};
```

Added fn_NetSim_Mobility_Run() function in mobility.c file

/* Since the unknown nodes are mobile, this function is user calculate the new positions of the unknown node whenever a node moves */

```
fn_NetSim_Localisation();
memcpy(pos,cor,sizeof* pos);
mobility_pass_position_to_animation(pstruEventDetails->nDeviceId,
```

```
                                pstruEventDetails->dEventTime,pos);  
//Add event for next point  
pstruEventDetails->dEventTime+=pstruMobilityVar->dCalculationInterval;  
fnpAddEvent(pstruEventDetails);  
pstruEventDetails->dEventTime-=pstruMobilityVar->dCalculationInterval;  
}
```