# Localization using RSS-based Trilateration in Wireless Sensor Networks

March 2023

**Applicable Release**: NetSim v13.3 or higher, www.tetcos.com
**Applicable Version(s)**: Standard and Pro
**Project download link**: See Appendix-1. The URL has the configuration files (scenario, settings, and other related files) of the examples discussed in this analysis for users to import and run in NetSim

## Introduction

In an ad hoc wireless sensor network the nodes make measurements of their environment, and then these measurements are used to carry out some global computation. Often, in this process, it becomes necessary to determine from which location a measurement came. Sensor network nodes may be too small (in terms of size and available energy) to carry a GPS (global positioning system) receiver. Some applications may require the nodes to be placed indoors, where GPS signals may not penetrate. Hence GPS-free techniques for location determination become important. Localization is the process of finding the absolute or relative location of a sensor node. In this project, the positions of some nodes are assumed to be known and the positions of other nodes are calculated using the power of signals received from the nodes whose position is known.

## Anchor Nodes

Anchor nodes are sensor nodes that have known position information. These nodes are typically equipped with GPS receivers or manually placed

at known coordinates. The anchor nodes provide reference points for determining the position of the unknown nodes. The anchor nodes also help in the calibration of the localization algorithm, as their positions are accurately known.

## Unknown Nodes

Unknown nodes are sensor nodes that have unknown position information. These nodes are also known as non-anchor nodes. Their position is estimated using localization techniques. In this method, the position of the unknown node is determined by measuring the power of radio signals received from the anchor nodes. By triangulating the signal strength from multiple anchor nodes, the position of the unknown node can be estimated.

## Actual location and Computed location

The actual distance between the anchor nodes and the unknown node can be precisely calculated using the function

```
fn_calculate_localisation_distance (Dev1, Dev2)
```

where `Dev1` is the unknown node and `Dev2` is the anchor node. This function can be used in this project because the sensor's location is *within* the NetSim simulation environment, making it possible to obtain its exact position using this API. Our objective is to determine the location of a node through ranging, which involves computing the location using the received signal strength. We then compare the estimated position obtained from trilateration algorithms with the actual position of the sensor.

## RSS based distance estimation

We estimate the distance between the nodes using the Received Signal Strength (RSS) and assume the simple log-distance path-loss model, which relates the RSS to the transmit power and the path-loss exponent. The accuracy of the location estimation is highly dependent on the knowledge of the path loss exponent. In this project, we assume perfect knowledge of the path-loss exponent to ensure accurate location estimates.

# 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 wish to determine is at some (x, y).

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

$$(x - x_1)^2 + (y - y_1)^2 = r_1^2 \tag{1}$$

$$(x - x_2)^2 + (y - y_2)^2 = r_2^2 \tag{2}$$

$$(x - x_3)^2 + (y - y_3)^2 = r_3^2 \tag{3}$$

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 the radial distance between transmitter and receiver devices. If we assume that no shadowing, no fading and no interference, then we have a deterministic function between RSS and distance. For example, this could be based on the log distance path loss model as explained in the next section.

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

$$x^2 - 2xx_1 + x_1^2 + y_1^2 - 2yy_1 + y_1^2 = r_1^2 \tag{4}$$

$$x^2 - 2xx_2 + x_2^2 + y_2^2 - 2yy_2 + y_2^2 = r_2^2 \tag{5}$$

$$x^2 - 2xx_3 + x_3^2 + y_3^2 - 2yy_3 + y_3^2 = r_3^2 \tag{6}$$

Subtracting the equation (5) from the equation (4)

$$2x(x_2 - x_1) + 2y(y_2 - y_1) = r_1^2 - r_2^2 - x_1^2 - y_1^2 + x_2^2 + y_2^2 \tag{7}$$

and equation (6) from the equation (5), we get

$$2x(x_3 - x_2) + 2y(y_3 - y_2) = r_2^2 - r_3^2 - x_2^2 - y_2^2 + x_3^2 + y_3^2 \tag{8}$$

In this, the unknowns are $x$, and $y$ (x, y), while the knowns are ($x_i$, $y_i$) which are at distances $r_i$ from an unknown point. This is basically a system of two equations with two unknowns:

$$Ax + By = C \tag{9}$$

$$Dx + Ey = F \tag{10}$$

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

$$x = \frac{(CE - FB)}{(EA - BD)} \tag{11}$$

$$y = \frac{(CD - AF)}{(BD - AE)} \tag{12}$$

# Computing unknown node position in NetSim

The received powers from all anchor nodes to the unknown node are computed using API

`GET_RX_POWER_dbm (Dev1, Dev2, Time)`

where `Dev1` is the anchor node, `Dev2` is the unknown node and `Time` is event time. Next, in the log distance path-loss model, the path-loss varies with distance per the equation

$$pathloss = 20 \times \log(\frac{\lambda}{4 \times \pi \times d_0}) + 10 \times \eta \times \log(\frac{d_0}{d}) \tag{13}$$

To get the distance between the unknown node and the anchor node, we rearrange the terms in the above equation to get

$$distance(d) = \frac{d_0}{10^{(\frac{-pathloss-(\frac{\lambda}{4\times\pi\times d_0})}{10\times\eta})}} \tag{14}$$

where, $\eta$ is the path-loss exponent. NetSim allows the user to set $2 \leq \eta \leq 5$, and this value of $\eta$ is assumed to be known. The reference distance ($d_0$) is set to 1m, and the path-loss model is applicable only for distances greater than $d_0$. The wavelength ($\lambda$) is calculated using the speed of light ($c$) and the transmit frequency ($f$), i.e., $\lambda = \frac{c}{f}$.

Once the unknown node receives signals from at least three anchor nodes, it can use trilateration to accurately compute its own position.

An alternate approach to computing distance is to use Time-of-arrival. NetSim currently assumes zero propagation delay between any two nodes in a Wireless sensor network. This is because the time taken for an electromagnetic signal to propagate 10m would be of the order of $10/(3 \times 10^8) = 3.33 \times 10^-8s$ which is of the order of (1/100)th of a microsecond. On the other hand, a packet transmission time is around $(50(B) \times 8)/(250 \times 10^3 bps) = 1.6 \times 10^-3s$. This is of the order of thousands of microseconds. A small deviation would lead to extremely large errors and hence this article does not explore the use of time of arrival for purposes of distance estimation.

# Simulating localization in NetSim

In the following section, we describe how localization is simulated in NetSim. The Workspace_Localization_WSN contains this scenario. To open this scenario, go to Your work on the home screen of NetSim and click on Localization_in_WSN_Example from the list of experiments. The scenario shown in Figure 1

Next, run the simulation.

# Simulation Results

After simulation, **Localization_Log.csv** file can be viewed from log files section of the results dashboard window. This file logs the Time, Unknown IDs, Anchor IDs, received powers from all anchor nodes to unknown nodes, actual position or coordinates of unknown nodes, calculated position or coordinates of unknown nodes, calculated distance and actual distance of unknown nodes.

Pathloss model is the reduction in attenuation of a signal as it propagates through a network. For a fixed distance between source and destination

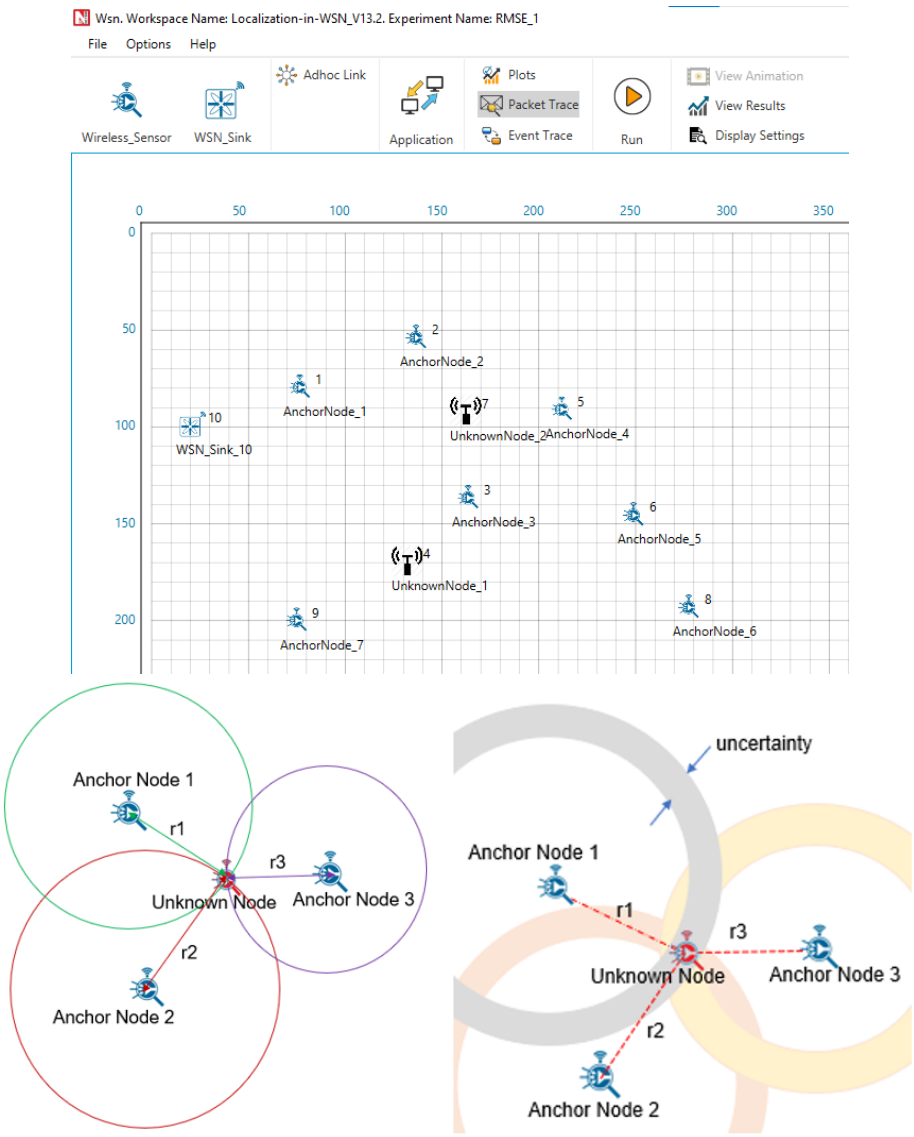


Figure 1: Top: Network scenario of unknown nodes and anchor nodes in NetSim, Bottom: Illustration of Trilateration explained in equations (1) through (12)

pathloss is same. So, if you set channel characteristic to pathloss only model, you can see in the Figure 3 that calculated distance and actual distance are same.

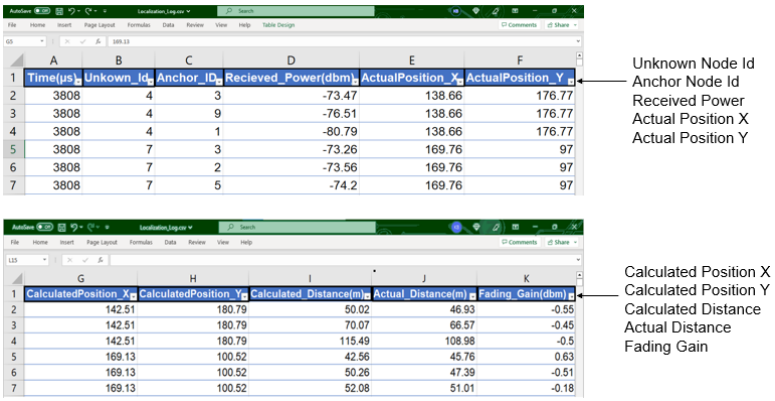


Figure 2: NetSim writes the Localization_Log.csv log file which can be viewed from the results dashboard

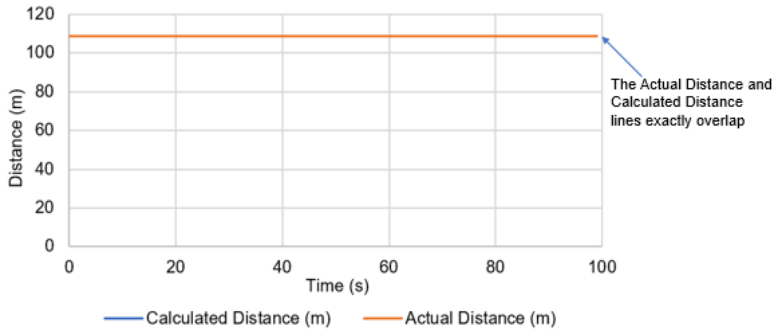


Figure 3: Distance vs Time Plot for unknown node 4 with respect to anchor node 1 in case of pathloss only model

When fading model is enabled, users can observe differences between the calculated distance and the actual distance in the Figure 4. This is because of the variation in attenuation of a signal with time. Notice that mean of the calculated distance is approximately equal to the actual distance.

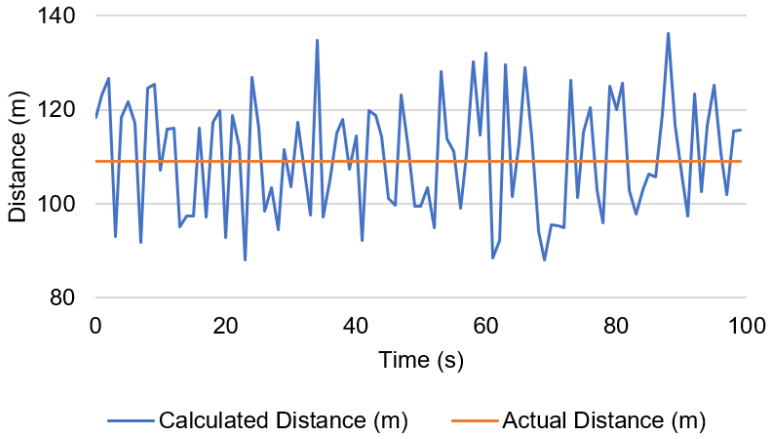


Figure 4: Distance vs Time Plot for unknown node 4 with respect to anchor node 1 in case of pathloss with fading loss model

# Results: With Fading

When using the log distance mean path-loss model, the distance between the anchor and the unknown node can be exactly obtained. This is because

$$Pr = Pt + 20\log(\frac{\lambda}{4 \times \pi \times d_0}) + 10 \times n \times \log(\frac{d_0}{d}) \tag{15}$$

where Pr is the received power, Pt is the transmit power, lambda is the wavelength, $d_0 = 1$m is the reference distance, and $d$ is the distance between the unknown node and the anchor node. Once Pr is measured, d can be exactly determined since all other variables are known. Next, when fading is present, the equation changes to

$$Pr = Pt + 20\log(\frac{\lambda}{4 \times \pi \times d_0}) + 10\eta log(\frac{d_0}{d}) + 10log(Y) \tag{16}$$

where the fading gain Y is modeled as a Rayleigh random variable with a shape parameter equal to 1. Therefore, at each measurement, the value of Pr is different since a different value would be drawn for Y. Thus the fading gain acts as *noise* in the path-loss measurements. Hence, for the inverse calculation of $d$, (see equation(14) ) at each RSS measurement, a slightly different d would be obtained. Note that the expectation of Y, $\mathbf{E}(Y) = 1$ and hence $\mathbf{E}(10 \times \log(Y)) = 0$. Therefore, the average (mean) value of d is very close to the actual distance.

The following plot show the variation of actual and measured distances for different scale parameters. The error - between the actual and calculated distance - decreases as we increase the scale parameter.

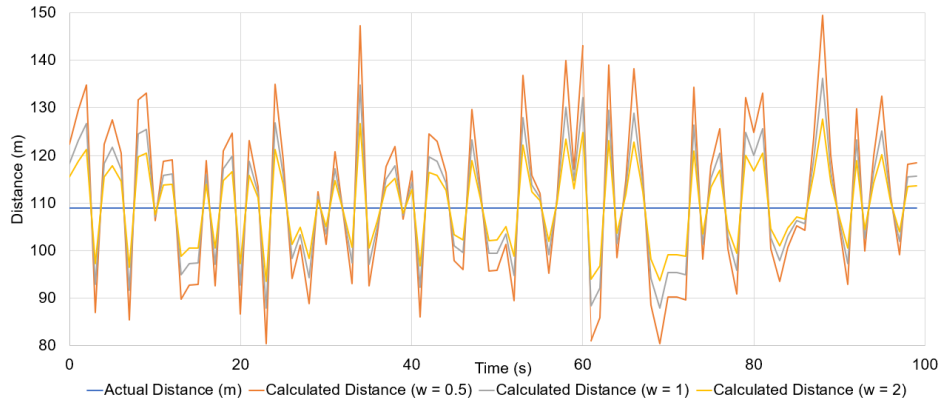


Figure 5: Distance vs Time Plot for unknown node 4 with respect to anchor node 1 for scale parameter 0.5, 1, and 2. This was obtained from data logged in Localization_Log.csv file

| Unknown Nodes | Anchor Nodes | Calculated Position X | Calculated Position Y | Actual Position X | Actual Position Y |
|---|---|---|---|---|---|
| 4 | 1 | 139.30 | 177.38 | 138.66 | 176.77 |
| | 3 | | | | |
| | 9 | | | | |
| 7 | 2 | 168.81 | 96.71 | 169.76 | 97.00 |
| | 3 | | | | |
| | 5 | | | | |

Table 1: The calculated and actual position coordinates of unknown nodes with fading enabled

# Root Mean Square Error (RMSE)

RMSE is a popular measure to estimate the accuracy of predicted values versus the actual or observed values.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(d_i^{calc} - d^{actual})^2}{n}} \tag{17}$$

Where 'n' is the count of distance measurements.
The RMSE calculations are shown in Table 2

| Scale Parameter | Unknown Nodes | Anchor Nodes | Mean Fading (dBm) | Std Dev Fading (dBm) | RMSE |
|---|---|---|---|---|---|
| 0.5 | 4 | 1 | -0.05 | 1.36 | 17.27 |
| | | 3 | -0.07 | 1.27 | 6.89 |
| | | 9 | -0.07 | 1.18 | 9.15 |
| | 7 | 2 | 0.26 | 1.31 | 7.05 |
| | | 3 | -0.04 | 1.26 | 6.73 |
| | | 5 | -0.14 | 1.37 | 8.3 |
| 1 | 4 | 1 | -0.03 | 0.96 | 12.12 |
| | | 3 | -0.05 | 0.90 | 4.85 |
| | | 9 | -0.05 | 0.83 | 6.42 |
| | 7 | 2 | 0.18 | 0.93 | 5.01 |
| | | 3 | -0.03 | 0.90 | 4.72 |
| | | 5 | -0.10 | 0.96 | 5.78 |
| 2 | 4 | 1 | -0.02 | 0.68 | 8.54 |
| | | 3 | -0.04 | 0.64 | 3.43 |
| | | 9 | -0.03 | 0.59 | 4.52 |
| | 7 | 2 | 0.13 | 0.66 | 3.57 |
| | | 3 | -0.02 | 0.63 | 3.22 |
| | | 5 | -0.07 | 0.68 | 4.05 |

Table 2: We see the calculated and actual position coordinates of unknown nodes, the fading mean and standard deviation, and the RMS error between predicted vs. actual position

# Additional Research Directions in Localization for NetSim Users

In this article, we demonstrated the feasibility and accuracy of using trilateration and the log-distance mean path-loss model for node localization in a wireless sensor network. Some new avenues for further research include:

1. What if the path-loss exponent is not known? In such cases, additional measurements and techniques are required for estimation. One approach is to use a large number of anchor nodes and RSS measurements to estimate both the path-loss exponent and the location of the unknown node. One could possibly solve this as a system of equations using a non-linear optimization technique.

2. Explore using machine learning algorithms to learn the path-loss exponent from a set of labeled data, and then use the learned model to estimate the location of the unknown node.

3. How does trilateration work when the unknown node is mobile? In such cases, the distance between the unknown node and the anchor nodes will change over time, and the trilateration algorithm needs to be updated frequently to reflect these changes. Additionally, the motion of the unknown node may introduce additional noise and errors in the distance measurements, which could reduce the accuracy.

4. Application of trilateration to underwater sensor networks where the path-loss equations are different given the underwater acoustic channel.

# Appendix-1: Project download link

`https://github.com/NetSim-TETCOS/Localization-in-WSN-v13.3/archive/refs/heads/main.zip`

# Appendix-2: Source code changes for implementing localization

We have made the following addition and modifications 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.
- **double fn_calculate_localisation_distance ();** //This function is used to find the distance between the unknown node and anchor node.
- **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.

**Localisation.c source code**

/* Users can give their own unknown node IDs and unknown node count in Localisation.c file. */

//Set up unknown node

```
int unknown\_node\_count=2;
int unknown\_node\_IDs[10]={4, 7};
```

The function **fn_calculate_localisation_distance()** contains the following lines of code.

```
double pl_exp=info->propagation.pathlossVar.pathLossExponent;
IEEE802_15_4_PHY_VAR* phyVar=WSN_PHY(an);

fWaveLength=(double)(300.0/(phyVar->dFrequencyBand_MHz*1.0));

ref_dist_pathloss=20*log10(fWaveLength/(4*(double)fpi*phyVar->d0));
```

//Calculate Fading Loss

```
dFadingloss_db=_propagation_calculate_fadingloss
(find_propagation_info(an, 1,un, 1));
```

//Calculate Path Loss

```
dPathloss_db=MW_TO_DBM(phyVar->dTransmitterPower_mw)-
GET_RX_POWER_dbm(an, un, pstruEventDetails->dEventTime)-dFadingloss_db;
```

//Calculate Distance between Unknown and Anchor node

```
unknown_anchor_dist=
phyVar->d0/(pow(10,((-dPathloss_db-fA1dB)/(10*pl_exp))));
```

The function **fn_Netsim_trilateration_method()** contains the following lines of code.

```
double A[2][2]={ 0 };
double B[2][1]={ 0 };
double x=0, y=0;

A[0[0]=2(sen_det[unknown_id]>anchor_node[1]>x_possen_det[unknown_id]
->anchor_node[0]->x_pos);
A[0][1]=2*(sen_det[unknown_id]->anchor_node[1]->y_pos-sen_det[unknown_id]
->anchor_node[0]->y_pos);
A[1][0]=2*(sen_det[unknown_id]->anchor_node[2]->x_pos-sen_det[unknown_id]
->anchor_node[1]->x_pos);
A[1][1]=2*(sen_det[unknown_id]->anchor_node[2]->y_pos-sen_det[unknown_id]
->anchor_node[1]->y_pos);
```

```
B[0][0]=(sen_det[unknown_id]->anchor_node[1]->
x_pos*sen_det[unknown_id]->anchor_node[1]->x_pos)-\
(sen_det[unknown_id]->anchor_node[0]->x_pos*sen_det[unknown_id]
->anchor_node[0]->x_pos)+\
(sen_det[unknown_id]->anchor_node[1]->y_pos*sen_det[unknown_id]
->anchor_node[1]->y_pos)-\
(sen_det[unknown_id]->anchor_node[0]->y_pos*sen_det[unknown_id]
->anchor_node[0]->y_pos)+\
(sen_det[unknown_id]->anchor_node[0]->dist*sen_det[unknown_id]
->anchor_node[0]->dist)-\
(sen_det[unknown_id]->anchor_node[1]->dist*sen_det[unknown_id]
->anchor_node[1]->dist);

B[0][1]=(sen_det[unknown_id]->anchor_node[2]->x_pos*sen_det[unknown_id]
->anchor_node[2]->x_pos)-\
(sen_det[unknown_id]>anchor_node[1]>x_pos*sen_det[unknown_id]
->anchor_node[1]->x_pos)+\
(sen_det[unknown_id]->anchor_node[2]->y_pos*sen_det[unknown_id]
->anchor_node[2]->y_pos)-\
(sen_det[unknown_id]->anchor_node[1]->y_pos*sen_det[unknown_id]
->anchor_node[1]->y_pos)+\
(sen_det[unknown_id]->anchor_node[1]->dist*sen_det[unknown_id]
->anchor_node[1]->dist)-\
(sen_det[unknown_id]->anchor_node[2]->dist*sen_det[unknown_id]
->anchor_node[2]->dist);
```

//Unknown sensor coordinates

```
sen_det[unknown_id]->x_pos=(B[0][0]*A[1][1]-B[1][0]*A[0][1])/
(A[1][1]*A[0][0]-A[0][1]*A[1][0]);
sen_det[unknown_id]->y_pos=(B[0][0]*A[1][0]-A[0][0]*B[1][0])/
(A[0][1]*A[1][0]-A[0][0]*A[1][1]);
```

**fn_NetSim_localisation(pstruEventDetails->nDeviceId)** function is called when each time packet received in 802_15_4.c file.