Abstract—This paper examines the problem of determining node location in ad-hoc sensor networks. Spatial localization is an important building block for wireless sensor networks. Beacons that know their position and serve as reference are a vital aspect of nearly every localization system. The intention of this paper is to examine the work done to locate nodes and the use of beacon nodes in order to decrease the error of nodes localization.

I. INTRODUCTION

In the near future, advances in processor, memory and radio technology will enable small and cheap nodes capable of wireless communication and significant computation capability. The availability of micro-sensors and low power wireless communications will enable the deployment of very dense, fully distributed sensor/actuator networks for a wide range of environmental monitoring applications (in-home, fire detection in forests, outdoor, atmospheric applications). Moreover these systems will eventually incorporate actuation, as well as sensing allowing these systems to influence and interact with their environment. These ad-hoc sensor networks will consist of nodes located arbitrarily and will be largely unattended. These unattended networks must self-configure and reconfigure to adapt to their environment and the availability of other nodes within the system.

One of the major challenges for researchers is to localize the sensor nodes with relatively high accuracy. For military, police radio networks, knowing the precise location of each person with a radio can be critical. In offices and in warehouses, object location and tracking applications are possible with large-scale ad-hoc networks of wireless tags. Thus, Global Positioning System (GPS) has been suggested as a means to obtain location information in ad-hoc networks [1]. But the straight forward of adding GPS to all nodes is not a viable solution due to the issues enumerated below.

- GPS cannot work indoors or in the presence of dense vegetation, foliage or other obstacles that block the line-of-sight from the GPS satellites.
- The size of GPS and its antenna increases the sensor node form factor. Sensor nodes are required to be small and unobtrusive.
- Finally, due to practical considerations as low-power consumption and cost preclude the use of GPS on all nodes. Instead of using GPS technology, distributed algorithms are proposed to locate small devices in wireless ad-hoc sensor network. These algorithms use either the cooperative ranging method based on Time-Of-Arrival (TOA), Time-Difference-Of-Arrival (TDOA), Angle-Of-Arrival (AOA), Received-Signal-Strength (RSSI) techniques, or hop count method. The basic idea behind these algorithms is the same in GPS technology, performing a triangulation between the nodes does not know its position (unknown node) and neighbour nodes knowing their positions (beacon nodes) to solve the set of geometric constraints. Beside, some studies [2] [3] show beacon placement and density in the transmission range of a given unknown node, affect the quality of spatial localization. Fixed beacon placement approaches such as uniform and very dense placement are not always viable and will be inadequate in very noisy environments in which sensor networks may be expected to operate.

This paper focus on the problem of nodes localization in wireless sensor ad-hoc networks. It is organized as follows: In the next section, the nodes localization problem is explained in term of error due to localization techniques (TOA, AOA, RSSI, and TDOA). In section 3, beacon nodes solution is studied to decrease the error. Two approaches are used in order to analyze the problem : arithmetic and via the direct method algorithm. Section 4, simulation results are explained. Section 5 is the conclusion.

II. NODES LOCALIZATION

In many proposed applications for wireless peer-to-peer and ad-hoc networks, knowing the location of the devices in the network is a key issue. One of the way to localize tiny sensor nodes is by local positioning. Devices communicates among them and perform a triangulation to determine their locations based on received signal strength (RSSI), time difference of arrival (TDOA), or time of arrival (TOA) technologies. The range estimate will be degrade due to multipath and noise in the channel and the inaccuracies of devices reference clocks. Error due to multipath can be reduced using very wide bandwidth like in the UWB technology (communication systems) or radar-like technology. The error due to inaccuracies of devices clock can be brought down using very accurate clocks with low parts-per-million (PPM) and low phase noise oscillators. In the RSSI technique, the pathloss model is frequently used for the fading channel, $p_{i,j}$ is the power transmitted by node $i$ and received by node $j$.

$$ p_{i,j} = p_0 - 10n\log_{10}\left(\frac{d_{i,j}}{d_0}\right) $$

(1)

$p_0$ is the received power in dB at a reference distance $d_0$ and $n$ is the pathloss exponent. The measured power, in error due to fading, is $\hat{p}_{i,j} = p_{i,j} + X_\sigma$. The random variable $X_\sigma$, represents the medium-scale fading in the channel and is typically reported to be zero-mean and Normal (in dB) with variance $\sigma_{dB}^2$ invariant with range. Then, the range estimate, $\hat{d}$, is ( [5])

$$ \hat{d}_{i,j} = d_0 \cdot 10^{\frac{p_{i,j} - X_\sigma}{10n}} = d_0 \cdot 10^{\frac{\hat{p}_{i,j}}{10n}} $$

(2)

From the imperfection of the radio channel, the error in time($\varepsilon_{t_{i,j}}$) is introduced.

$$ \varepsilon_{t_{i,j}} = t_{ij} - \frac{d_{i,j}}{c} $$

(3)

$\varepsilon_{t_{i,j}}$ is the error in time due to the channel and clock accuracy during the cooperative ranging between node $i$ and $j$, $t_{ij}$ is the TOA between node $i$ and $j$. 

Beside, Previously in the literature, some authors suggest that the error in space due to the redundancy of the algorithms(iterative). It is
not easy to model, because each algorithms have their own manner of implementing the error on the coordinates. Let take a gradient error. This idea is close to the one exposed by R. Nagpal et al ([8]). $E_j$ is the total error in space on $X$ and $Y$ coordinates if we restrain our study in 2D. In other words, $E_j$ is the residual error after one iteration. Thus, the coordinates updates are:

$$\Delta x_j = \frac{\delta E_j}{\delta x_j} \text{and} \Delta y_j = \frac{\delta E_j}{\delta y_j}$$

(4)

where $0 < \alpha << 1$. Thus, the error is decreasing after each iteration. Now, we define the error in space ($\varepsilon_{S_j}$).

$$\varepsilon_{S_j} = \nabla \mathbf{E}_j$$

(5)

Thus, we have two kinds of error: in time due to radio channel and hardware imperfection summarizes in $\varepsilon_{tij}$; and the error in space on the coordinates with ($\varepsilon_{S_j}$).

### III. BEACON NODES

Beacon nodes which know their position and serve as a reference are a vital aspect of the nearly every localization system. Authors previously said ([3]) that beacon placement strongly affects the quality of localization system. The beacon density and their spatial placement control the granularity of localization. Most of articles dealt with the nodes location issue without taking into account the "beacon" parameter (except [3] and [2]). Intuitively, the idea of beacon nodes improving the quality of nodes pops up in our mind. But very dense placement is not a good solution due to two major issues:

- The cost of the beacon may preclude very dense beacon placement. Power considerations may require that only a restricted smaller subset of beacon nodes be active at any given time so as to prolong system lifetime.[3]
- Self-interference: At very high densities, the probability of collisions among signals transmitted by the beacons increases. Therefore even if we had unlimited numbers of beacons, we would like to limit their use.

After giving a definition of a beacon node and an intuitive idea about how it can influence the node location problem, two approaches are used to have a deep view of the phenomenon. The first view will show the beacon improvement through an arithmetic example. The second part is through the direct method to find the coordinate of the tag.

### A. Arithmetic method

We place four nodes (Beacon/unknown) at the edge of a square figure. But if it is a beacon, then the position is exactly at the edge of a square. If it is an unknown node, which the position was already estimated, the position is around the edge, in a circle which radii is equal to $\gamma$, the error on the coordinates $(X, Y)$.

$$\gamma = \sqrt{(x + \alpha)^2 + (y + \alpha)^2}$$

(6)

The goal is to calculate the error on the coordinates of the node put a the center of the figure in 2-D. A special triangulation (an average of the coordinates) is performed as describe in the paper [6]). The first case is when four anchor nodes are at the edge of the square and the coordinates of the node at center are calculated. A residual error is introduced on the coordinates of the first node at the center independently of the results given by our simple triangulation. This error models the imperfection of the channel (TOA error in the cooperative ranging). If this hypothesis is not made, no error is introduced and the coordinates of the nodes at the center of the square are always exact. After the first unknown node gets its coordinates, we define an other set of nodes with 3 beacon nodes and the unknown node previously at the center of the square and an other unknown node is put at the center of the new square, as shown in Figure 1. The X coordinate in the scenario with 4 anchor nodes is

$$X_{u1} = \sum_{i=1}^{4} \frac{X_i}{4} + \alpha = X_{a1} + \alpha$$

(7)

where, $X_{a1}$ is the coordinate of the unknown node 1 and $X_i$ is the coordinate of the beacon node $i$. $X_{a1}$ is defined as $X_{a1} = \sum_{i=1}^{4} \frac{X_i}{4}$. Now, if at the edge one of the anchor nodes is replaced by the unknown node $U_{a1}$.

$$X_{u2} = \sum_{i=1}^{3} \frac{X_i}{4} + \frac{X_{u1}}{4} + \alpha = X_{a2} + \frac{\alpha}{4} + \alpha$$

(8)

But, if instead of replacing one anchor node, we replace more than one by an unknown node with error on the coordinate the same type than $X_{a1}$, we have for 2 beacon nodes:

$$X_{u2} = \sum_{i=1}^{2} \frac{X_i}{4} + 2 \cdot \frac{X_{u1}}{4} + \alpha = X_{u2} + \frac{2 \cdot \alpha}{4} + \alpha$$

(9)

for 1 beacon node:

$$X_{u2} = \sum_{i=1}^{1} \frac{X_i}{4} + 3 \cdot \frac{X_{u1}}{4} + \alpha = X_{u2} + \frac{3 \cdot \alpha}{4} + \alpha$$

(10)

for only unknown nodes type $X_{u1}$:

$$X_{u2} = 2 \cdot \frac{X_{u1}}{4} + \alpha = X_{u2} + \frac{4 \cdot \alpha}{4} + \alpha$$

(11)

In this simple example, we can see the error on the X coordinate take value $\{\alpha + \frac{\alpha}{4}, \frac{2 \cdot \alpha}{4} + \alpha, \frac{3 \cdot \alpha}{4} + \alpha, 2 \cdot \alpha\}$. For the second unknown node, we obtain a singleton of values on the error coordinate. If we calculate the error on the error coordinate for a third node we will obtain : $4 \cdot 7 = 28$ type of errors. For example, case where 2 beacon nodes, node U1 and U2 (with first value of the error).

$$X_{u3} = \sum_{i=1}^{2} \frac{X_i}{4} + \frac{X_{u1}}{4} + \frac{X_{u2}}{4} + \alpha = X_{u3} + \frac{2 \cdot \alpha}{4} + \frac{\alpha}{4} + \alpha$$

(12)

We will stop here our investigation in this arithmetic problem. Through this simple example, we show that the overall error in space on the coordinates is reduced. In the next section, we will see two more realistic algorithm to locate nodes.

### B. Direct method [7]

Thanks to the direct method, we will simulate different scenario with different density of unknown nodes and anchor nodes in the neighborhood of the node which wants to know its coordinates (called sometimes 'tag'). First we will introduce a bit of theory on the direct method and how the error is taking into account.
In the cartesian system, the range(distance) between sensor i and the node which wants to know its coordinate \((x, y, z)\) is given by
\[
\sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2} = c(t_i - t_o)
\] (13)

\((x_i, y_i, z_i)\) is the coordinates of the sensor respectively, \(c\) is the speed of light, \(t_i\) is the signal TOA at sensor i, and \(t_o\) is the unknown transmit time at the tag sensor.

First, the neighbors of the tag have to be differentiated in two categories: one hand the unknown nodes which previously set up their coordinates with any algorithms. On the other hand, there are the beacon nodes. If N neighbors are in the neighborhood of the tag \(i = 1, 2, 3, 4, \ldots, N\). Then, \(i = 1, 2, 3, 4, \ldots, k\) are unknown nodes and \(i = k + 1, \ldots, N\) are the anchor nodes.

In order to see the impact of the neighborhood on the tag’s coordinates, instead of introducing a gaussian error on the TOA measurements as its commonly done here, we will introduce an error on the coordinates\((\alpha)\) depending on if its an anchor \((\alpha = 0)\) or its an unknown \((\alpha \neq 0)\). Then, the equation below can be rewritten taken into account our hypothesis.

\[
\sqrt{(x-(x_i+\alpha))^2 + (y-(y_i+\beta))^2 + (z-(z_i+\gamma))^2} = c(t_i-t_o)
\] (14)

Where
\[
(\text{Anchor}) \quad \alpha = 0 \quad \text{Unknown} \quad \alpha \approx N(0, \sigma)
\]

\[
(\text{Anchor}) \quad \beta = 0 \quad \text{Unknown} \quad \beta \approx N(0, \sigma)
\]

\[
(\text{Anchor}) \quad \gamma = 0 \quad \text{Unknown} \quad \gamma \approx N(0, \sigma)
\]

In the case of the unknown node, the error on the coordinates is modelled as zero mean gaussian distributed. Before showing the results, for reminder we will give the equations explained comprehensively in the paper [7]. In order to simplify the calculus, we hide the error \(x_i + \alpha, y_i + \beta, z_i + \gamma\) in \(x_i, y_i, z_i\). Using the equation (10), and squaring both sides gives
\[
(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2 = c^2(t_i-t_o)^2 \quad i = 1, 2, 3, 4
\] (15)

Subtracting (2) for \(i=1\) from (2) for \(i = 2, 3, 4\) produces
\[
c t_o = \frac{1}{2}(t_1 + t_i) + \frac{1}{2c(t_1 - t_o)}(\beta_1 - 2x_i x - 2y_i y - 2z_i z) i = 2, 3, 4
\] (16)

where
\[
x_{i1} = x_i - x_1
\]
\[
y_{i1} = y_i - y_1
\]
\[
z_{i1} = z_i - z_1
\]
\[
\beta_{i1} = \beta_i = x_i^2 + y_i^2 + z_i^2 - (x_1^2 + y_1^2 + z_1^2)
\]

Define the TDOA between sensors i and j as \(\Delta_{ij} = t_i - t_j\). Eliminating \(t_o\) from (3) yields
\[
a_1 x + b_1 y + c_1 z = g_1
\] (17)

where
\[
a_1 = \Delta_{12} x_{31} - \Delta_{13} x_{21}
\]
\[
b_1 = \Delta_{12} y_{31} - \Delta_{13} y_{21}
\]
\[
c_1 = \Delta_{12} z_{31} - \Delta_{13} z_{21}
\]
\[
g_1 = \frac{1}{2}(c_1 \Delta_{12} \Delta_{13} \Delta_{23} + \Delta_{12} \beta_{31} - \Delta_{13} \beta_{21})
\]

and
\[
a_2 x + b_2 y + c_2 z = g_2
\] (18)

where
\[
a_2 = \Delta_{23} x_{41} - \Delta_{14} x_{21}
\]
\[
b_2 = \Delta_{23} y_{41} - \Delta_{14} y_{21}
\]
\[
c_2 = \Delta_{23} z_{41} - \Delta_{14} z_{21}
\]
\[
g_2 = \frac{1}{2}(c_2 \Delta_{12} \Delta_{14} \Delta_{24} + \Delta_{12} \beta_{41} - \Delta_{14} \beta_{21})
\]

Combining (4) and (5) yields
\[
x = Az + B
\] (19)

where
\[
A = \frac{b_1 c_2 - b_2 c_1}{a_1 b_2 - a_2 b_1}
\]
\[
B = \frac{c_1 b_2 - c_2 b_1}{a_1 b_2 - a_2 b_1}
\]

and
\[
y = Cz + D
\] (20)

where
\[
C = \frac{a_2 c_1 - a_1 c_2}{a_1 b_2 - a_2 b_1}
\]
\[
D = \frac{a_2 b_1 - a_1 b_2}{a_1 b_2 - a_2 b_1}
\]

Then, substitution of (6) and (7) back into (3) with \(i=2\) produces
\[
c(t_1 - t_o) = Ez + F
\] (21)

where
\[
E = \frac{1}{2A^2 + 2B^2 - C^2 - D^2 + I}
\]
\[
F = \frac{1}{2A^2 + 2B^2 - C^2 - D^2 + I}(2Ax_1 + By_1 - z_1 - EF)
\]
\[
I = (B - x_1)^2 + (D - y_1)^2 + z_1 - F^2
\]

The two solutions to (9) are
\[
z = - \frac{H}{2G} \pm \sqrt{\frac{H}{2G}^2 - \frac{I}{G}}
\] (23)

The two estimated \(z\) values(if both are reasonable) are then substituted back into (6)and(7) to produce the coordinates \(x\) and \(y\), respectively. However, there is only one desirable solution. We can either dismiss one of the solution if it has no physical meaning, or do the average of the two values if they are very close.

C. Nonlinear Optimization Method [9]

The last method - in order to determine coordinates of a tag- is the DFP quasi-Newton algorithm. The function is defined as
\[
f(p) = \sum_{i=1}^{N} \left[ (x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 - c(t_i - t_o) \right]^2
\] (24)

where
\[
p = [x, y, z, t_o]^T
\] (25)

is the vector of the unknown coordinates \((x, y, z)\) and the unknown transmit time\(t_o\). Also \(t_i\) is the estimated TOA at base station i. Clearly, the objective function is the summation of the squared range errors of all sensors. The optimization purpose is to minimize this objective function to produce the optimal position estimate. The solution of this equation is found iteratively by
\[
p_{k+1} = p_k - \alpha B_k g_k
\] (26)

where \(p_k\) is the vector of the estimated position coordinates and the
estimated transmit time at the kth iteration, and \( \alpha \) is the step size. Also \( g_k \) is the gradient of the objective function given by
\[
g_k = \nabla f(x, y, z, t_v)|_{p=p_k} = \begin{bmatrix} \frac{\delta f}{\delta x} |_{p=p_k} & \frac{\delta f}{\delta y} |_{p=p_k} & \frac{\delta f}{\delta z} |_{p=p_k} & \frac{\delta f}{\delta t_v} |_{p=p_k} \end{bmatrix}^T
\]
(27)

\( B_k \) is the inverse Hessian which is updated according to
\[
B_{k+1} = B_k + \frac{h_k h_k^T}{g_k} - \frac{B_k q_k q_k^T B_k}{q_k^T B_k q_k}
\]
(28)
where
\[
h_k = p_{k+1} - p_k
\]
\[
q_k = g_{k+1} - g_k
\]

To start the iteration, the initial position coordinates and the initial transmit time are required. The initial estimated values of the position coordinates may be chosen to be the mean position of all the active sensors or the area being monitored. The initial transmit time may be chosen to be some time point earlier than the earliest receive time, which will depend on the dimension of the monitored area. Both the step size \( \alpha \) and the first derivative \( g \) (the gradient) are updated during each iteration.

After describing two algorithms, it is high time to go to the simulation scenario and results part.

IV. SIMULATION RESULTS

As simulation model was developed to examine the influence of the number of anchor and unknown nodes in the neighborhood of a tag using either the direct method or the DFP algorithm to calculate the coordinates. The tool models a closed region with a number of sensors (anchor and unknown) and examines the accuracy with which a device in the area may be positioned in 3-D. The monitored area has been assumed to have a dimension of \( 70m \times 70m \times 50m \). The position of the sensors (delays estimation points) and the tag (device of interest) are randomly generated to obtain average performance.

As previously said, an error on the coordinates of the neighbors is introduced depending on if the neighbor is an unknown or anchor node. Then, if an unknown node, the error is zero mean gaussian distributed with \( \sigma \) variance. We have to underline, we chose an error not independent between coordinates. This choice can be justified with how the direct method works. It is less trivial for the DFP algorithm. At each value of \( \sigma \) examined, 70 simulations are run and averaged.

Figure 3 & 4 shows the results. The couples \((u,v)\)of numbers in the right corner of the figures mean \( u \) anchor nodes and \( v \) unknown nodes for the simulations. First, if we compare the graphs 2 by 2 \(([4, 8], [8, 4]), ([5, 9], [9, 5]), \cdots\), the curves with higher number of anchor nodes are not always under the curve with a smaller number of anchor nodes. Thus, we cannot conclude that beacon nodes decrease the error in space.

By the way, from the remark on what we called "key points", we can take a look on if the error is spatially distributed. In other words, in an area where it will be few or no beacon, the error on the \( X, Y, Z \) coordinates should be higher; or the further an unknown node is, the bigger the error might be. Figure 7&8 shows the results of the error spatially distributed. We put 4 anchors nodes at the center and randomly we distributed unknown nodes at a given radius from the center. Then, we run the two previous algorithms. Before commenting the results, the error has to be weighted by the number of unknown node in space.

The results from the figures confirm what we said previously. The further an unknown node is from the center, the bigger the error is.

V. CONCLUSION

In this article, we discuss about the error of node localization. In the first part, we show that there are two types of errors: an error in time \( t_{ij} \) due to channel and hardware imperfections and an error in space \( e_{ij} \) due to algorithm redundancy. Starting from these hypothesis, we try to see how beacon nodes can improve the error
in space. We used an arithmetic method and also two algorithms (Direct method and DFP algorithm). It was clear with the arithmetic approach beacon nodes in the neighborhood of an unknown node can decrease the error in space. In real case it is more complicated when we use the two algorithms. But, to localize all the nodes we use a cooperative ranging. Researcher in general used algorithms like [8],[10],[11]. Because of the accuracy. We did not chose this approach to stay at a simple level in order to be focus only on the beacon node solution. More investigations should be done with other algorithms. The question : does the beacon node help to decrease the error on the coordinate of the unknown node ? It helps, but considering many parameters as the density of beacon & unknown nodes in the neighborhood of an unknown nodes, the radius of the nodes,...An upper bound of the number of nodes in the neighborhood, will be the Signal-to-Noise-Ratio or the level of interferences in the neighborhood. This part has not be investigated in this present work.

REFERENCES