

Article

Localization Algorithm with On-line Path Loss Estimation and Node Selection

Albert Bel ^{*}, José López Vicario and Gonzalo Seco-Granados

Signal Processing for Communications and Navigation Group, Telecommunications and Systems Engineering Department, Universitat Autònoma de Barcelona, Edifici Q Campus de la UAB, Bellaterra (Cerdanyola del Vallès), Barcelona 08193, Spain

* Author to whom correspondence should be addressed; E-Mail: albert.bel@uab.cat; Tel.: +34-935-813-843; Fax: +34-935-814-031.

Received: 6 May 2011; in revised form: 17 June 2011 / Accepted: 21 June 2011 /

Published: 1 July 2011

Abstract: RSS-based localization is considered a low-complexity algorithm with respect to other range techniques such as TOA or AOA. The accuracy of RSS methods depends on the suitability of the propagation models used for the actual propagation conditions. In indoor environments, in particular, it is very difficult to obtain a good propagation model. For that reason, we present a cooperative localization algorithm that dynamically estimates the path loss exponent by using RSS measurements. Since the energy consumption is a key point in sensor networks, we propose a node selection mechanism to limit the number of neighbours of a given node that are used for positioning purposes. Moreover, the selection mechanism is also useful to discard bad links that could negatively affect the performance accuracy. As a result, we derive a practical solution tailored to the strict requirements of sensor networks in terms of complexity, size and cost. We present results based on both computer simulations and real experiments with the Crossbow MICA2 motes showing that the proposed scheme offers a good trade-off in terms of position accuracy and energy efficiency.

Keywords: wireless sensor networks; localization; cooperative; distributed; Received Signal Strength (RSS)

1. Introduction

In recent years, location estimation in wireless sensor networks (WSN) has raised a lot of interest from researchers [1-3]. In order to give sense to the measured data by a WSN, it is necessary in the majority of the environments to give the location of the nodes. The WSN localization techniques are used to obtain estimates of nodes position with initially unknown positions. In order to do that, those non-located nodes take the advantage of the knowledge of the positions of some sensors with known location (anchor nodes) and inter-node measurements. Those anchor nodes can obtain their locations by means of using a global positioning system (GPS), or by setting those nodes at known positions. Localization methods are normally divided in two phases [4]. The first phase consists in the estimation of distances between nodes. At the second phase, the localization algorithm computes the position of each node through different methods. Next, we present these two phases in detail:

1.1. Measurement Phase

In this subsection we present three different signal metrics used to obtain distances estimates.

- **Time Measurements** Distance estimates obtained through time measurements are usually estimated using different methods. It is possible to obtain the distance by means of using the time of arrival (TOA) of a signal [5]. When a node receives a message it extracts the distance through the measurement of the transmission time. Although a high accuracy can be achieved with this method, it is necessary to have a synchronized network with the same reference clock in all the nodes.

Another method based on time measurements is the Time Difference of Arrival (TDOA). In [3], two different methods are presented in accordance with the nature of the time estimates. First TDOA method [6] is based in the measurement of the difference between the arrival time of the same signal at two receivers. This method assumes that the receiver locations are known and the two receivers are perfectly synchronized. The second TDOA method [7] eliminates the necessity of having a synchronized network. It uses a combination of two kind of signals, e.g., RF and ultrasonic signals. The time difference between the first and the second signal is used as an estimate of the one-way acoustic propagation time. In this case nodes require extra hardware in order to be able to transmit different signals.

Finally, in order to avoid the necessity of a synchronized network or the use of different transmit signals, one can find in literature the Roundtrip Propagation time measurements [8]. In this method, one sender sends one signal to a node and this node retransmits the same signal to the sender node. Then this method measures the difference between the time when a signal is sent and the time when this signal comes back to the sender node. This time measurement is done with the same local clock, so it avoided the need of a synchronized network. Also, it is not necessary to send different signals. The major downside of this method is the necessity of a double transmission in order to obtain a time measurement and of knowing the time delay of transmission at the receiver node.

- **Angle Measurements** Methods using angle measurements are known as Angle or Direction of Arrival, AOA and DOA [9]. The angle is estimated with the use of directive antennas or array of antennas. The necessity of extra hardware is the major disadvantage of this method that could probably increase the size and cost of the nodes.
- **Received Signal Strength Measurements** In this case, the distance between two nodes is obtained by using the power of the received signal [10]. RSS-based distance estimations, in particular, are based on the well-known radio propagation path loss model. Compared to AOA or TOA-based techniques, this technique has become the most inexpensive because RSS signals can be obtained during normal transmissions. However, distance estimates obtained through RSS measurements present less accuracy than the other two methods.

1.2. Location-Update Phase

The second phase uses the previous distance estimates to start the location procedure. Existing algorithms could be classified as centralized versus distributed, and non-cooperative versus cooperative. In centralized algorithms, a central processor receives all the information and calculates the position of all the network nodes. The central processor probably will not have any processing limitation, but on the other hand, this solution limits the scalability of the system. Hence, centralized solutions are not very attractive for large scale networks, where distributed algorithms are preferred. The second classification differentiates between the cooperative and non-cooperative techniques [11]. We assume that a network is formed by nodes with known position, called reference or anchor nodes, and nodes without knowing their position, called non-located nodes. In non-cooperative techniques, non-located nodes are only able to communicate with reference nodes. On the other hand, cooperative techniques allow a non-located node to communicate with any node either anchor or non-located. In general, cooperative techniques can increase localization performance in terms of both accuracy and coverage.

1.3. Contribution

In this paper, we focus on a cooperative distributed localization method based on RSS measurements. RSS measurements become the simplest choice in order to reduce the complexity and the cost of the nodes. Although distance estimates obtained through RSS measurements have a lower accuracy, we try to reduce the adverse effects introduced by modelling problems. The choice of a distributed strategy is motivated by the desire to reduce the necessity of transmitting all the network information to a central node. With this adoption each node modifies in the second phase its own state through those estimated metrics and the nodes state information. We also adopt a cooperative technique. Although cooperative techniques could increase localization accuracy, cooperation with distant nodes could introduce a higher degradation in the estimate depending on the estimation method used [12]. This is because the error introduced in the measurements can be multiplicative to the distance when RSS measurements are considered [13]. In addition, allowing the cooperation with more nodes increase the consumption of energy. The introduction of node selection strategies allow to the localization algorithms to minimize the energy consumption and the cooperation with further nodes while maintaining location accuracy.

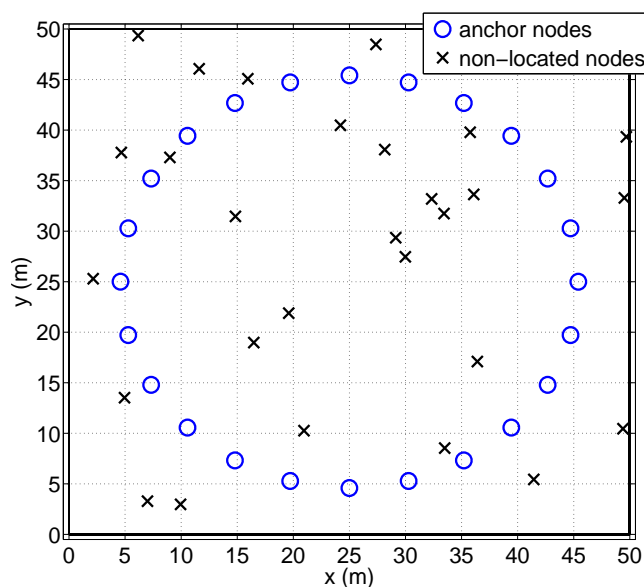
Other sources of error that affect RSS-based distance estimations are shadowing and multipath signals, which complicate the modelling of the channel that nodes need to know *a priori*. A previous measurement campaign is usually carried out in order to obtain a proper model. The introduction of an on-line estimation of the propagation model helps to the RSS-based method to adapt to the scenario without the necessity of an off-line calibration. We propose a cooperative distributed positioning algorithm that dynamically estimates the propagation model that best fits the propagation environment by means of RSS measurements only.

Furthermore, two node selection criteria are proposed in order to reduce the number of cooperative nodes. By doing so, the scheme only selects those nodes providing accurate distance measurements. Another benefit is that the energy consumption is reduced. As a result, accuracy *vs.* energy consumption trade-off is significantly improved. As shown in the paper, experimental results carried out with Crossbow MICA2 motes validates the proposed scheme.

2. RSS-Based Distance Estimation

Let us consider a wireless sensor network with N nodes. There are N_1 nodes, whose exact locations are known (anchor nodes). The rest of the nodes $N_2 = N - N_1$ do not know their position (non-located nodes). The main goal is to estimate the location of the non-located nodes with the help of anchor nodes and the rest of nodes in the network by means of a cooperative strategy. Concerning the anchor nodes placement, we follow the approach presented in [14], where it is shown that the best anchor placement is a centered circumference with radius equal to the root-mean-square (rms) of the non-located nodes distances to the center. One can see in Figure 1 an example of the scenario deployment.

Figure 1. Example of an scenario deployment.



As previously commented, we consider an RSS-based distributed cooperative algorithm for location estimation. The first phase of the algorithm consists in obtaining internode distances through RSS measurements. Power received is modelled through the well-known radio-propagation path loss and

shadowing model [12]. Received Signal Strength (RSS) can be expressed as the power received in node j from a signal transmitted by node i , P_{ij} , as:

$$RSS_{ij} = P_{ij} = P_0 - 10\alpha_{ij} \log_{10} d_{ij} - v_{ij} \text{ (dBm)} \quad (1)$$

where P_0 is the power received in dBm at 1 m distance, d_{ij} is the distance between nodes i and j , parameter α_{ij} is the path loss exponent and $v_{ij} \sim \mathcal{N}(0, \sigma_v^2)$ represents log-normal shadow-fading effects, where the value of the standard deviation σ_v depends on the characteristics of the environment. Given the power received RSS_{ij} in Equation (1), an ML estimate of the actual distance can be derived as:

$$\delta_{ij} = 10^{\frac{P_0 - RSS_{ij}}{10\alpha_{ij}}} \quad (2)$$

3. On-Line Localization Algorithm

Once the relative distances between nodes are obtained, the position estimates for each non-located node are estimated by means of the least squares criterion. Position estimates are calculated by obtaining the set of non-located node positions and path loss exponents that minimize the difference between estimated distances at the first phase and the distances computed using such position estimates. In particular, the problem consists in minimizing the following cost function:

$$C_{LS}(\mathbf{x}, \boldsymbol{\alpha}) = \sum_{i=1}^{N_2} \sum_{j \in S_i} (\delta_{ij}(\alpha_{ij}) - d_{ij}(\mathbf{x}_i, \mathbf{x}_j))^2 \quad (3)$$

where $d_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|$ is the distance between nodes i and j , calculated with the estimated position (or real coordinates if node j is an anchor) of nodes i and j , S_i is the group of nodes (anchor and non-located) that cooperates in the position estimation of non-located node i , \mathbf{x} are the coordinates of nodes, and $\boldsymbol{\alpha}$ the set of all path loss exponents of all links. The minimization of the cost function of Equation (3) is carried out in a distributed fashion. We adopt a distributed strategy due to its scalability and robustness. We obtain the minimization of Equation (3) by computing its derivative with respect to x_i :

$$\frac{\partial C_{LS}}{\partial \mathbf{x}_i} = \sum_{j \in S_i} \frac{\partial (\delta_{ij} - d_{ij})^2}{\partial \mathbf{x}_i} + \sum_{k \in S_i} \frac{\partial (\delta_{ki} - d_{ki})^2}{\partial \mathbf{x}_i} \quad (4)$$

Notice that the second part of the sum can be omitted. This is because reciprocal channels are not assumed and, then, the value of δ_{ki} cannot be estimated at node i . As a result, the cost function adopted by each node can be rewritten as:

$$C_{DLS}(\mathbf{x}_i, \alpha_{i1}, \alpha_{i2}, \dots, \alpha_{iN_{S_i}}) = \sum_{j \in S_i} (\delta_{ij}(\alpha_{ij}) - d_{ij}(\mathbf{x}_i, \mathbf{x}_j))^2 \quad (5)$$

Here, the objective is to minimize the difference between both distances by optimizing the node coordinates and the set of path loss exponents. The node coordinates (\mathbf{x}_i) and the set of all path loss exponents ($\alpha_{ij} \forall j \in S_i$) affect the computation of both distances, d_{ij} and δ_{ij} , respectively.

In order to solve the cost function of Equation (5) we adopt the Gauss-Seidel algorithm [15]. This non-linear algorithm is based on a circular iterative optimization with respect to one set of variables while maintaining the rest of the variables fixed. Hence, the minimizations are carried out successively

for each component. If we consider a generic cost function F that depends on a set of variables β , the desired minimization of F is formally defined as [15]:

$$\beta_i(t+1) = \arg \min_{\beta_i} F(\beta_1(t+1), \dots, \beta_{i-1}(t+1), \beta_i, \beta_{i+1}(t), \dots, \beta_m(t)) \quad (6)$$

At time instant $t+1$, the value of the component β_i is optimized. Components from β_1 to β_{i-1} have been already minimized whereas components from β_{i+1} to β_m (being m the total number of components) have not been optimized. In our problem, cost function Equation (5) depends on the components $(\mathbf{x}_i, \alpha_{S_i})$. By using the Gauss-Seidel approach, we can divide the optimization in two steps: firstly a minimization of the cost function by means of optimizing the node coordinates (fixing the path loss exponents) is carried out; secondly, another minimization is done by means of the optimization of the path loss exponent (fixing the nodes coordinates). As the convergence of the non-linear Gauss-Seidel algorithm can be established using a descent approach (see [15]), both minimizations are carried out through a gradient descent mechanism. Furthermore, the gradient descent method is a low computational method and it is distributable and scalable. It is a method that does not necessarily deliver the global optimal solution in non-convex problems, as it is the case of the problem presented here, unless a good starting point is available. For that reason, a study of the initial point's impact on the behaviour of the algorithm is presented later on. Our results show that in practice, with an appropriate initialization, the gradient descent technique converges to a good solution.

3.1. Optimization of the Node Coordinates

We initially obtain a minimization of the cost function by optimizing the nodes coordinates. We first maintain all the α_{ij} fixed $\forall j \in S_i$. As $d_{ij}(\mathbf{x}_i, \mathbf{x}_j) = f(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|$ depends on the coordinates \mathbf{x}_i the gradient of the cost function is:

$$\begin{aligned} \nabla_{\mathbf{x}_i} C_{DLS}(\mathbf{x}_i, \alpha_{S_i}) &= \nabla_{\mathbf{x}_i} \left(\sum_{j \in S_i} (\delta_{ij}(\alpha_{ij}) - \|\mathbf{x}_i - \mathbf{x}_j\|)^2 \right) \\ &= \sum_{j \in S_i} (\delta_{ij}(\alpha_{ij}) - d_{ij}(\mathbf{x}_i, \mathbf{x}_j)) \mathbf{e}_{ij} \end{aligned} \quad (7)$$

where $\mathbf{e}_{ij} = \frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|}$ is the unit vector that takes the orientation between the node i and node j . So, the estimate of \mathbf{x}_i , can be iteratively computed by using the gradient descent algorithm as follows:

$$\hat{\mathbf{x}}_i(t+1) = \hat{\mathbf{x}}_i(t) + \gamma \sum_{j \in S_i} (\delta(\alpha_{ij}) - d(\mathbf{x}_i, \mathbf{x}_j)) \mathbf{e}_{ij} \quad (8)$$

The use of a gradient descent method requires initial values for the position estimates ($\hat{\mathbf{x}}(t=0)$). Several methods can be found in the literature to obtain appropriate initial values. In that direction, MultiDimensional Scaling (MDS) is proposed in [12] to obtain such values. It is a good option but it increases the computational cost. Another option can be based on a random initialization. It is a simple method but a higher number of iterations is required for algorithm convergence. We propose to initialize each $\hat{\mathbf{x}}_i(0)$ as the weighted mean of the coordinates of the n_{anch} nearest anchor nodes. When the node

i sets its own group S_i (see Section 4), it has to find the nearest anchor nodes and computes a weighted mean as:

$$\hat{\mathbf{x}}_i(t=0) = \sum_{a=1}^{n_{anch}} \mathbf{x}_a \left| \frac{RSS_{ia}}{\sum_{a=1}^{n_{anch}} RSS_{ia}} \right| \quad (9)$$

where \mathbf{x}_a are the coordinates of the anchor $a \in S_i$.

With this estimation we could achieve an initial value closer to the final solution without an important increase of the computational cost. As we are assuming an equal path loss, the selection of the nearest anchor could not reflect the reality (see Figure 2). For that reason, we present an study of the influence of the number of anchor nodes used in the initial estimation of the non-located nodes inside the network. Figure 3 shows the results obtained. One could observe that better results are obtained when a number of anchor nodes equal to 1 is used to calculate the weighted mean. We use a fixed number of $n_{anch} = 1$ in the sequel. At first sight, this result could seem strange because, normally, it is better to use as many nodes as possible. But at the initial time instant of our algorithm we have not still estimated the path loss exponent values. Hence, the higher RSS_{ij} , the lower distance estimate δ_{ij} . If the number of anchors nodes n_{anch} has a greater value, we can increase the probability of having a further anchor node. Hence, we can estimate an initial position far away from the real position. For that reason it is better to select only the anchor node with the highest RSS.

Figure 2. Example of an initial position with different values of n_{anch} .

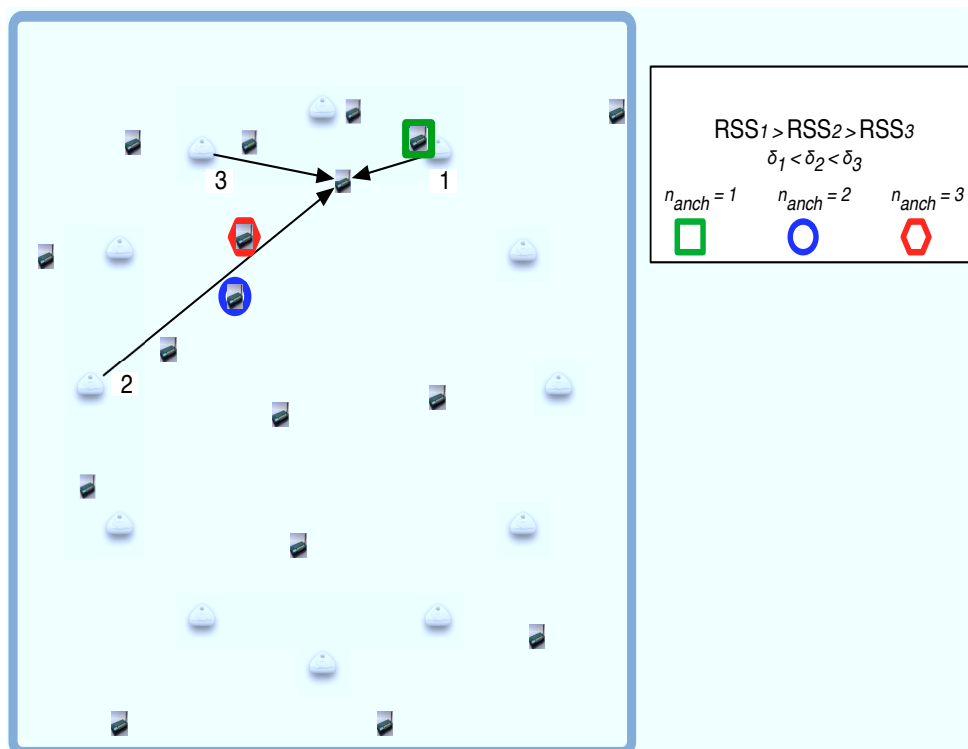
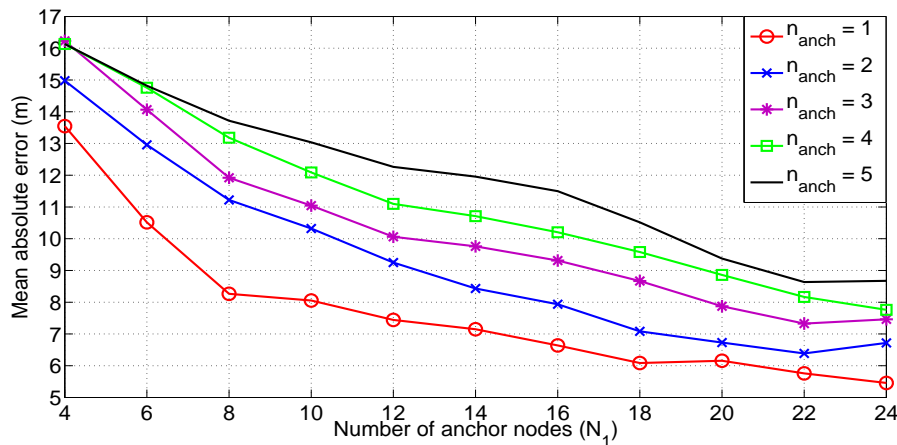


Figure 3. Mean absolute error vs. number of anchor nodes (N_1).



3.2. Optimization of the Path Loss Exponents

As previously commented, the objective of this paper is the improvement of RSS-based localization algorithm. However, the use of RSS measurements requires an accurate scenario modelling. In order to obtain a good model, a thorough previous measurement campaign should be carried out. A change in the environment becomes in the necessity of repeating the measurement campaign. By adopting an on-line estimation of the path loss exponent, the algorithm can be adapted to the scenario without the need of performing another measurement campaign. With on-line estimation we can also avoid the equal path loss exponent assumption for all links.

Following the Gauss-Seidel approach, we now minimize the cost function of Equation (5) fixing the nodes coordinates \mathbf{x} . As the $\delta_{ij} = f(\alpha_{ij}) = 10^{\frac{P_0 - RSS_{ij}}{10\alpha_{ij}}}$ depends on the path loss exponent α_{ij} , the cost function becomes:

$$C_{DLS}(\mathbf{x}_i, \boldsymbol{\alpha}_{S_i}) = \sum_{j \in N_{S_i}} (10^{\frac{P_0 - RSS_{ij}}{10\alpha_{ij}}} - d_{ij}(\mathbf{x}_i, \mathbf{x}_j))^2 \tag{10}$$

As each δ_{ij} is a function of α_{ij} , but not on α_{ik} with $k \neq j \in S_i$ (independent links between nodes), we can minimize cost function for each individual link. In that case, the fixed variables are the coordinate estimates and the rest of the path loss exponents ($\alpha_{ik} \forall k \neq j$). The gradient of cost function of Equation (10) is:

$$\begin{aligned} \nabla_{\alpha_{ij}} C_{DLS}(\mathbf{x}_i, \boldsymbol{\alpha}_{S_i}) &= \nabla_{\alpha_{ij}} ((10^{\frac{P_0 - RSS_{ij}}{10\alpha_{ij}}} - d_{ij}(\mathbf{x}_i, \mathbf{x}_j))^2) \\ &= -\log(10) \frac{P_0 - RSS_{ij}}{5} \frac{1}{\alpha_{ij}^2} \delta_{ij} (\delta_{ij} - d_{ij}(\mathbf{x}_i, \mathbf{x}_j)) \end{aligned} \tag{11}$$

Each node estimates their own path loss exponents for all the links. This is the major difference between our proposal and the proposed method in [16], which is centralized. Our proposal is a distributed method that minimizes the cost function through an iterative gradient descent strategy. Finally, it is worth noting that restrictions are applied to the set of solutions for α_{ij} . More specifically, the following feasible set of solutions is considered in order to avoid undesirable results:

$$\alpha_{min} \leq \alpha_{ij} \leq \alpha_{max} \tag{12}$$

where we have considered that α_{min} and α_{max} are equal to 2 and 5, respectively. This selection is based on the knowledge of the kind of deployment scenario.

3.3. Localization Algorithm

With the obtained path loss exponent estimations, we can update distance estimates δ_{ij} and recalculate coordinates estimates $\hat{\mathbf{x}}_i$. Following the Gauss-Seidel approach, both previous procedures are repeated in a circular fashion until convergence. A scheme of the global procedure of the location algorithm can be found in the Algorithm 1, with t_{iter1} and t_{iter2} being the required number of iterations to converge to the final solution.

Algorithm 1 LS Localization Algorithm with On-Line Path Loss Estimation

for $t = 1$ to t_{iter1} **do**

Coordinate Estimation:

for $t = 1$ to t_{iter2} **do**

for $i = 1$ to N_2 **do**

$$\hat{\mathbf{x}}_i(t) = \hat{\mathbf{x}}_i(t-1) + \gamma_x \sum_{j \in S_i} (\delta_{ij} - \hat{d}_{ij}) \mathbf{e}_{ij}$$

end for

end for

Path Loss Estimation:

for $t = 1$ to t_{iter2} **do**

for $i = 1$ to N_2 **do**

for $j = 1$ to N_{S_i} **do**

$$\hat{\alpha}_{ij}(t) = \hat{\alpha}_{ij}(t-1) - \gamma_\alpha \log(10) \frac{P_0 - RSS_{ij}}{5} \frac{1}{\alpha_{ij}^2} \delta_{ij} (\delta_{ij} - \hat{d}_{ij})$$

$$\delta_{ij} = 10^{\frac{P_0 - P_{ij}}{10 \hat{\alpha}_{ij}}}$$

end for

end for

end for

end for

It is important to note that our proposal requires a higher computational complexity than that required by a pure RSS-based algorithm (without node selection and path loss estimation). In order to reflect this point, we have evaluated the required complexity in terms of number of operations for both cases. In the conventional RSS-based algorithm the total number of operations is equal to:

$$t_{iter1} \times t_{iter2} \times N_2 \times (2N_{S_i} + 2) \quad (13)$$

In our proposal, the number of operation is:

$$t_{iter1} \times t_{iter2} \times N_2 \times ((2N_{S_i} + 2) + 7N_{S_i}) \quad (14)$$

where the additional term with respect to the former expression is due to the estimation of the path loss. Although it may seem that our proposal results in a larger computational complexity, this is not

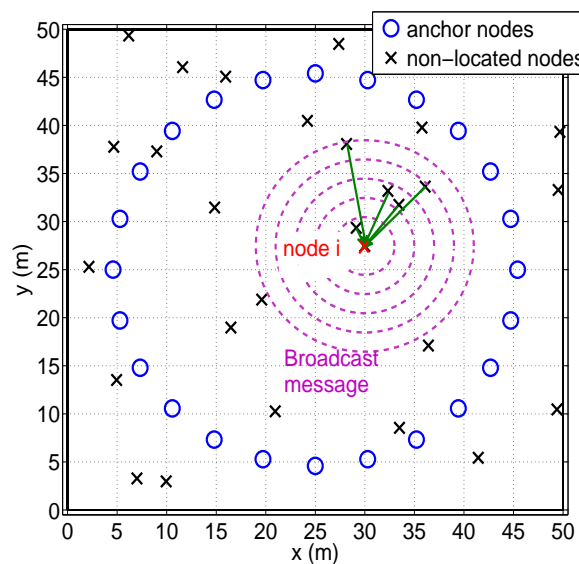
necessarily the case because our selection mechanism allows us to reduce the number of nodes S_i in each group, possibly leading to a complexity equal to or smaller than that of the conventional RSS-based algorithm. Indeed, for the scenarios addressed here, simulation results reflect that the average number of operations is 18% lower in our proposal than that carried out by the conventional RSS-based algorithm.

4. Energy Consumption

Wireless sensor networks nodes rely on low data rates, very long battery life (several months or even years) and very low computational complexity associated with the processing and communication of the collected information across the WSN. In order to maintain the battery life, the reduction of the energy consumption is an important point in WSN.

Taking into account the localization algorithm presented in the previous section, we can present an energy consumption model based on the required number of transmissions (see Figure 4).

Figure 4. Example of the creation of a group S_i .



At a first moment each node i has to create its own group of cooperating nodes S_i . At the beginning, each node i sends a broadcast message with its coordinates x_i . Only those nodes that receive this message answer with their node ID and their location coordinates. With the received messages, each non-located node can create its own S_i group. Once these groups are created, the exchange of messages are only done between cooperating nodes. At this time, the total amount of energy consumed by the network follows the model presented in [17]:

$$\epsilon = (\mu_{R_x} + \mu_{T_x}) \left(\sum_{i=1}^{N_2} N_{S_i} - N_2 \right) \kappa \quad (15)$$

where κ is the number of iterations of the algorithm, N_{S_i} the number of nodes inside S_i , and μ_{T_x} and μ_{R_x} are the energy consumption dedicated for peer to peer transmission and reception procedures, respectively. We have to mention that our model presents some differences compared with the model presented in [17]. We suppose that the energy per transmission is always the same instead of having an

energy consumption depending on time. Furthermore, we only take into account the energy consumption at the transmission and reception time. Our node selection mechanism produces a reduction in the number of transmitted messages. For that reason we do not take care about the energy consumption at sensing time, as in the cited reference. Hence, we only present an energy model based on the consumption at the transmission and reception time.

It is worth noting that energy consumption is an increasing function on the number of cooperating nodes (N_{S_i}). In the next section we propose different node selection methods that allows us to reduce the energy consumption (reducing the number of nodes inside S_i).

5. Node Selection Mechanisms

The authors presented in [18] a Node-Selection Least Squares (NS-LS) location algorithm. The idea is to obtain a good trade-off in terms of position accuracy versus energy consumption. As discussed in [18], the derivation of the optimal selection criterion is not possible. For that reason, the authors presented a sub-optimal scheme based on the received power threshold (RSS_{th}). In other words, only nodes with RSS higher than the RSS_{th} were allowed for cooperation. This criterion becomes in a simple scheme suitable for a hardware restricted WSN. In particular, the choice of the RSS_{th} value was designed to assure a minimum number of anchor nodes inside the cooperating nodes group (S_i). In accordance to this value, N_m , different trade-off points in the energy consumption versus accuracy can be achieved. Results showed that $N_m = 3$ allows the algorithm to achieve an excellent trade-off. Concerning the relation between RSS_{th} and N_m , we derived in [18] an analytical procedure to obtain the required RSS_{th} that assures the desired value of N_m . To do so, we first considered a uniform distribution for the positions of the non-located nodes and obtained the mean number of anchor nodes inside a circumference of radius r_{th} as:

$$N_m \approx \sum_{j=1}^{N_1} \pi r_{th}^2 \frac{1}{A} = \frac{N_1 \pi r_{th}^2}{A} \quad (16)$$

where A is the total area of the considered scenario and N_1 is the total number of anchor nodes in the scenario. Then, by taking into account the existing relation between received power and coverage range radius based on the path loss propagation model, the idea was to select the appropriate received power threshold RSS_{th} assuring that N_m anchors nodes is inside the coverage range radius. By considering r_{th} in the expression above as the coverage range radius, we finally established the following relation between RSS_{th} and N_m (for more in depth explanation see [18]):

$$RSS_{th} = \left(\frac{N_1 \pi P_0^{2/\alpha}}{N_m A} \right)^{\alpha/2} \quad (17)$$

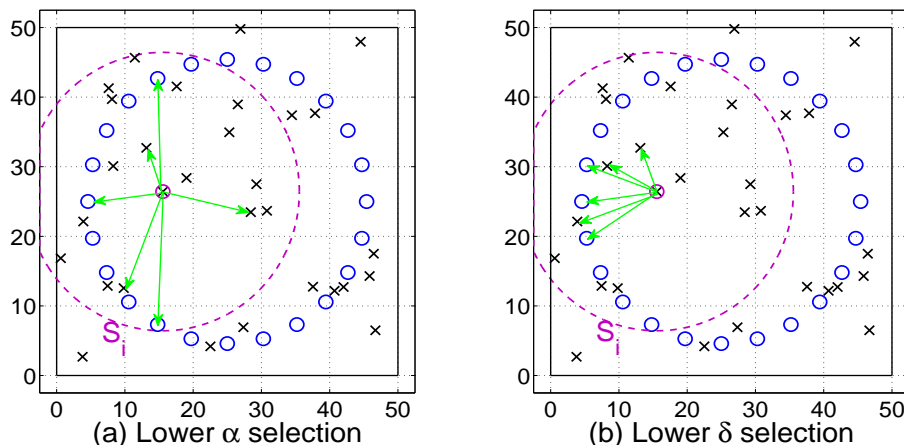
5.1. Selection Mechanisms

In this subsection, we derive two node selection algorithm for the proposed cooperative localization scheme. Since in this work we consider the adoption of on-line path loss estimation, we need to adapt the scheme presented in [18] to the new context. The use of RSS thresholds, in particular, is not the

optimal choice in an environment where path loss exponents of the different links can be quite different: the node with the highest RSS is not necessarily the best one.

In particular we present two node selection mechanisms that depend on the path loss estimates obtained through the Algorithm 1. With those selection mechanisms we try to reduce the number of nodes that cooperates in the location algorithm presented in Section 3 (see Figure 5). The different criteria are: low path loss selection and low distance selection. Hence, both node selection schemes depend on path loss estimates. First, the low path loss selection tries to find the nodes with best channel conditions. Second, the low distance criterion selects those nodes that have a low distance estimate δ_{ij} . It is possible that these selection criterion could not select a node nearby, because distance estimates could not reflect the reality (δ values depend on the quality of $\hat{\alpha}$ estimates and the shadowing effects).

Figure 5. Example of both node selection methods.



- **Low Path Loss Selection** Given all the estimates of $\hat{\alpha}_{ij}$, the first selection mechanism selects those nodes that have the lowest values for the path loss exponent. In other words, by sorting the path loss exponents of the nodes inside the coverage of node i :

$$\hat{\alpha}_{i1} \leq \hat{\alpha}_{i2} \leq \dots \leq \hat{\alpha}_{in} \quad 1, \dots, n \in S_i$$

where $\hat{\alpha}_{i1}$ and $\hat{\alpha}_{in}$ are the lowest and highest exponent, respectively. We select the nodes with the lowest values:

$$S_i^{NS} = \{i1, i2, \dots, in_\alpha\}$$

with n_α standing for the number of selected nodes. With this selection mechanism we are interested in the selection of nodes that have better propagation conditions.

- **Low Distance Selection** Now, the idea is to select the closest nodes to node i . We want to use nearer nodes in order to reduce distance error estimates, because, as shown in [13], the shadowing effect introduces errors multiplicative to the distance. By sorting now the distance estimate of the nodes inside the coverage of node i :

$$\delta_{i1} \leq \delta_{i2} \leq \dots \leq \delta_{in} \quad 1, \dots, n \in S_i$$

where in this case δ_{i1} and δ_{in} are the lowest and highest distance estimate, respectively. The new group of cooperating nodes becomes:

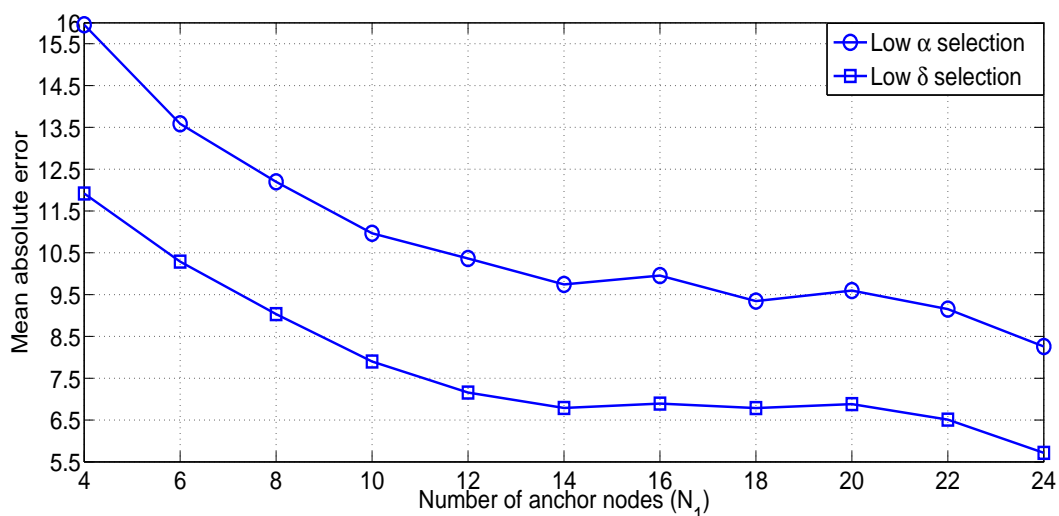
$$S_i^{NS} = \{i1, i2, \dots, in_\delta\}$$

with n_δ standing for the number of selected nodes.

5.2. Selection Mechanisms Performance

Once the node selection mechanisms are presented, we show the performance of both mechanisms in order to choose the appropriate one. First of all it is necessary to compare both methods. It is shown in Figure 6 that the low distance selection outperforms the low path loss selection in all the scenarios. As we said previously, the selection of those links with a lower value of α_{ij} could not correspond to the closest nodes. As we are using an RSS-based algorithm, the error introduced at the measurements done at the first phase is multiplicative to the estimated distance. The probability of selecting distant nodes is higher. As a consequence, the performance is affected.

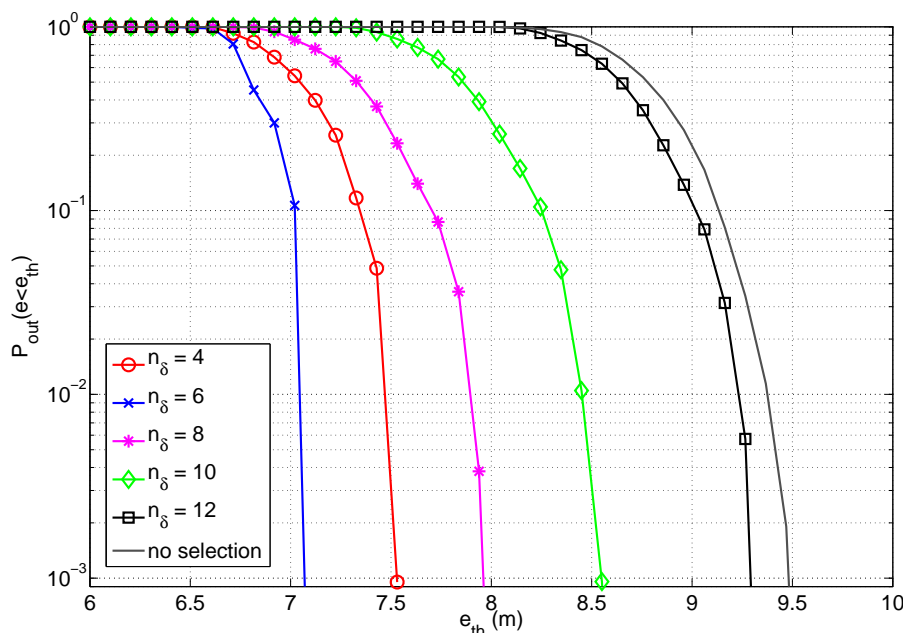
Figure 6. Mean absolute error versus number of anchor nodes (N_1) ($n_\delta = n_\alpha = 6$).



On the other hand, we also have to know which is the optimum number of cooperating nodes. Analytically, it is not straightforward to obtain the value of n_δ that optimize the system behaviour. Then, numerical evaluation is needed to obtain this value. In principle, the optimum number of n_δ is scenario dependent. However, our results have shown that a value equal to 6 is generally the best choice. This means that actually the optimum n_δ does not depend on the fine-grained distribution of the nodes, but rather on general parameters of the scenario (e.g., positioning in 2D or 3D, overall distribution of the nodes in the area under study, etc.). It is shown in Figure 7 the probability of outage for different values of n_δ . This probability of outage is the probability of having an error higher than an error threshold. Analysing the results presented, one can see that having a low value of cooperating nodes reduces the possibility of having anchor nodes (anchors are nodes with true information of their exact location) inside the cooperating group. On the other hand, having more cooperating nodes implies the existence of further nodes. This fact implies a higher error of the distance estimates. It is important to achieve a

good trade-off between having a group big enough to lodge an anchor but low enough to not use further nodes with a high error at distance estimates. A number of n_δ equal to 6 offers the best results in terms of position accuracy. For that reason we adopt this value in the sequel.

Figure 7. Outage probability ($N_1 = 18$).



6. Numerical Results

This section presents the performance of the proposed location algorithm with on-line path loss estimation and node selection. We present both simulations and experimental results obtained in an indoor scenario. We consider that the path loss exponents take values between a maximum value of 5 and a minimum value of 2 (The uniform distribution of the path loss exponents between 2 and 5 are based on experimental results obtained in [16]). Hence, we simulate path loss values with a uniform distribution ($\alpha \in \mathcal{U}(2, 5)$). We assume in our algorithm an initial value of the path loss equal to 3.5, which is the middle value of the random values used in the uniform distribution. The experimental parameters are shown in Table 1.

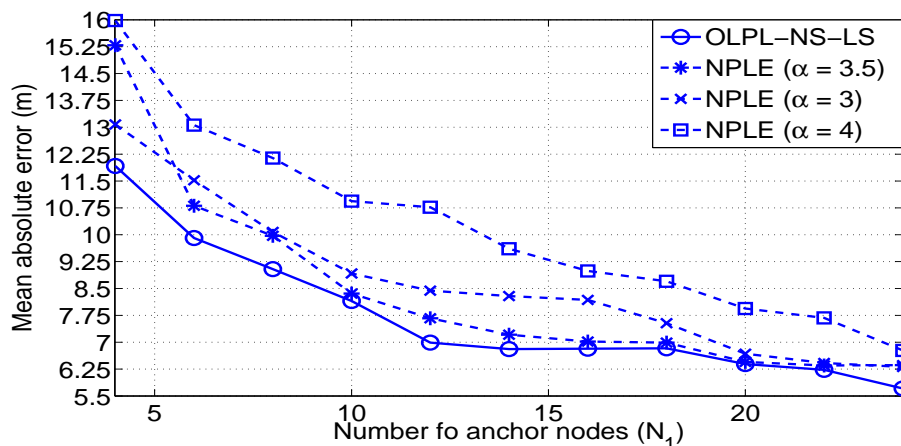
Table 1. Simulation Parameters.

Simulation Parameters	Parameter Value
Size of Sensor Field	50 × 50 m
Number of Non-Located Nodes (N_2)	30
Path Loss Exponent α_{ij}	2–5
Standard Deviation σ_v	1 dB
First-Meter RSS P_0	−50 dBm
Anchor Radius	20.4 m
Energy Consumption to Transmit or Receive μ_{T_x} or μ_{R_x}	400 nJ

6.1. Computer Simulation

The assumption of different path loss exponent for each link allows us to simulate a more realistic scenario to that presented in [18]. With this assumption the importance of doing an on-line estimation of the path loss exponent (*i.e.*, a good propagation model) is reflected in Figure 8. With the proposed solution we achieve a gain in terms of position accuracy that oscillates between 2 and 0.5 meters, compared with that achieved with the Non Path Loss Estimation (NPLE) algorithm. Different values of path loss exponents have been simulated and the best result is always achieved with our on-line path loss estimation and node selection least squares algorithm (OLPL-NS-LS). Having an on-line estimation of the path loss not only has good consequences in the localization performance accuracy but also makes it possible for the algorithm to adapt to possible changes in the scenario. Furthermore, the lowest gain is obtained when the number of anchor nodes is higher than 16 and this is not a usual value in a realistic scenario.

Figure 8. Mean absolute error versus number of anchor nodes (N_1) (solid line: lower distance selection, dashed line: non-path loss estimation).



6.2. Comparison with Existing Methods

In this subsection we will compare our proposed algorithm OLPL-NS-LS with two different existing solutions: a distributed method based on a Maximum Likelihood algorithm (ML) and centralised algorithm based on Multidimensional Scaling (MDS). We apply the on-line path loss estimation to all the methods in order to achieve a fair comparison between them. Only our OLPL-NS-LS method present the node selection method proposed. We will compare the performance of our method in terms of both energy consumption and positioning accuracy.

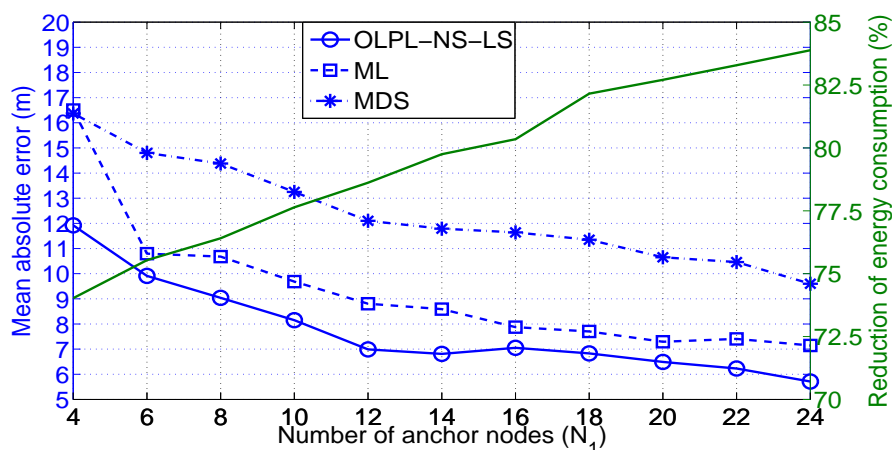
The ML localization algorithm [19] used is based on the minimization of the following cost function carried out with a distributed iterative method:

$$C_{ML}(\mathbf{x}) = \sum_{i=1}^{N_2} \sum_{j \in S_i} (\log_{10}(\delta_{ij}(\alpha_{ij})) - \log_{10}(d_{ij}(\mathbf{x}_i, \mathbf{x}_j)))^2 \quad (18)$$

The MDS algorithm is a simple centralized approach that builds a global map using classical MDS [20]. MDS works well on networks with relatively uniform node density but less well on more irregular networks.

As observed in Figure 9 our OLPL-NS-LS algorithm outperforms the ML localization algorithm. It is important to remark that a gain between 0.5 and 1.5 meters is obtained with our algorithm. This gain is achieved thanks to the selection algorithm. The gain achieved with respect to the MDS localization algorithm is more remarkable. In that case we are comparing a distributed method (our OLPL-NS-LS) with a centralized method (MDS). On the one hand, a centralized method includes more distant nodes. Then, nodes with a high error on their distance estimates are used. On the other hand, all possible nodes inside each group S_i are also used in the path loss estimation process. Probably, these nodes that are not near to a node i would not have a similar propagation conditions compared to those nodes that are closer. For that reason a node selection scheme allows to reduce the mean absolute error results in an RSS-based localization algorithm. It is also important to remark on the reduction of the energy consumption. According to our energy consumption model, the use of a reduced cooperating group S_i produces a reduction in the energy consumed by the network. With the use of a number of cooperating nodes $n_\delta = 6$, our OLPL-NS-LS algorithm achieves a percentage of reduction between 74% and 83% compared to the energy consumed by a method without node selection mechanism (see Figure 9).

Figure 9. Mean absolute error versus number of anchor nodes (N_1) (solid line: On-Line Path Loss and Node Selection Least Squares, dashed line: Maximum Likelihood, dash-dotted line: MultiDimensional Scaling).

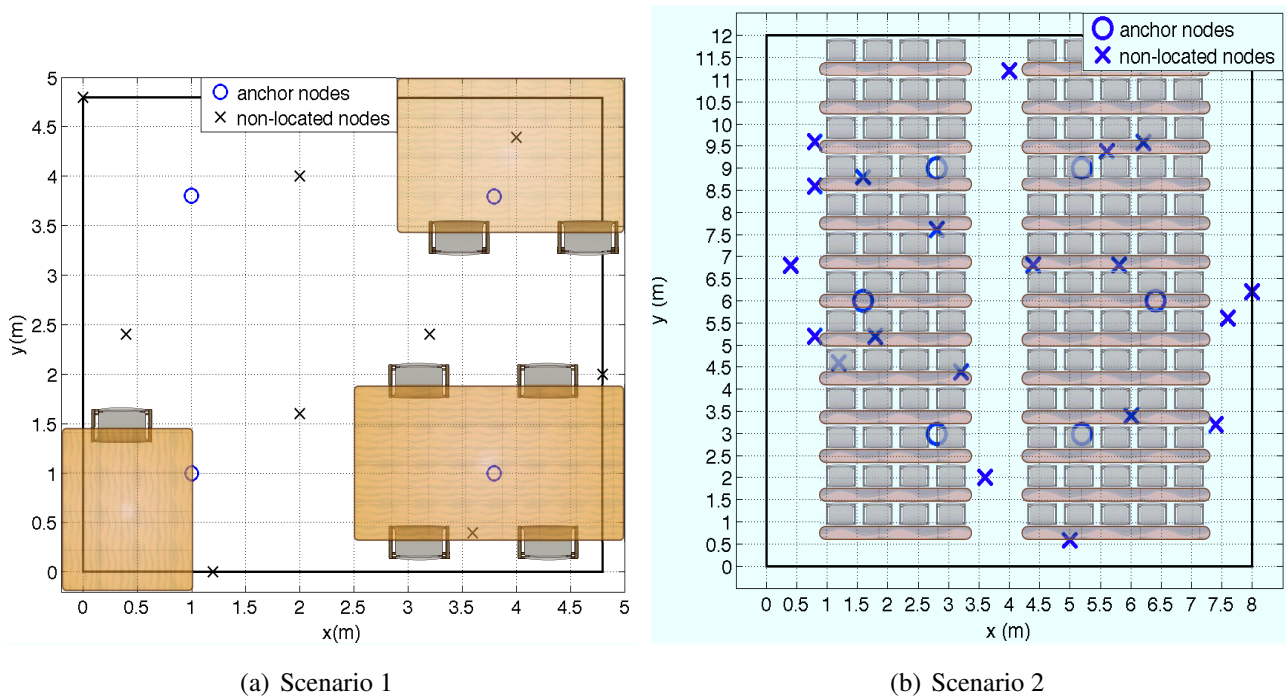


6.3. Experimental Results

In order to check the performance in a real scenario, we have carried out measurements in different indoor scenarios with the Mica2 motes @915MHz of Crossbow [21]. We present two different indoor scenarios shown in Figure 10.

At the first scenario the total number of nodes with unknown position, N_2 , is 9 (see Figure 10(a)), the number of anchor nodes, N_1 , is 4. These nodes are located in a 4.8 m \times 4.8 m scenario. The second scenario (see Figure 10(b)) is composed by $N_2 = 20$, $N_1 = 6$ and $N_1 = 4$ in a scenario of 8 m \times 12 m.

Figure 10. Experimental scenarios.

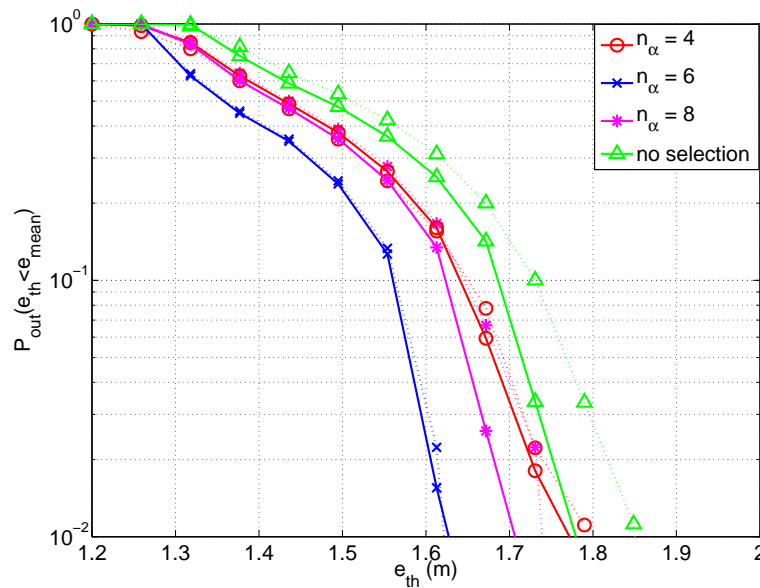


Scenario 1 It is shown in Figure 11(a) the probability of having an error above an error threshold. We present this graphic in order to verify the better results achieved in Figure 7 with only 6 nodes cooperating. We also present simulation results with the same conditions as the real scenario. On the one hand, the best result achieved is with a value of n_δ equal to 6 as achieved in the large scale case. On the other hand, simulation results are very similar to that achieved with the experimental scenario. Furthermore, Figure 11(b) shows the mean absolute error and the percentage of reduction compared to the consumption of the location algorithm without node selection. The best result is again achieved with n_δ equal to 6. With this value, the percentage of reduction in terms of energy consumption is about 50%. As observed, experimental results are quite similar to simulation results in this case. It is also presented the comparison of behaviour between our OLPL-NS-LS and the NPLE with different values of the path loss exponent (α). Our OLPL-NS-LS achieves always the best result compare to that achieve with all the NPLE presented (see Figure 11(b)). Best behaviour is achieved with a fixed value of $\alpha = 3.5$. The differences in terms of mean absolute error oscillates between 10 cm when $\alpha = 3.5$ and 0.6 cm when $\alpha = 2$. Results of this experimental scenario show the gains obtained by considering the proposed real time path loss estimation with respect to the case adopting an equal path loss exponent for all the links.

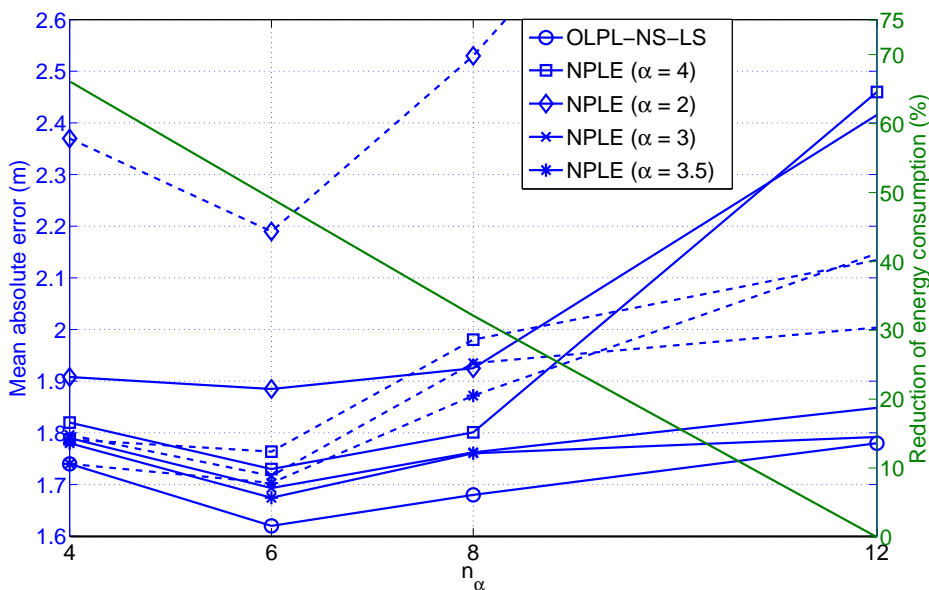
Scenario 2 As commented before, we present two different results: one for a number of 4 anchor nodes and the other for 6 anchor nodes. Figure 12 shows both results. As before experimental and simulation results are presented, showing a similar behaviour between experimental an simulation performance.

It is also remarkable that, as in the previous Figures, the best performance is achieved when the number of cooperating nodes is equal to 6.

Figure 11. Results at Scenario 1.



(a) Outage probability (solid line: experimental results, dashed line: simulation results.)



(b) Mean absolute error versus number of cooperating nodes (n_δ) (solid line: experimental results, dashed line: simulation results)

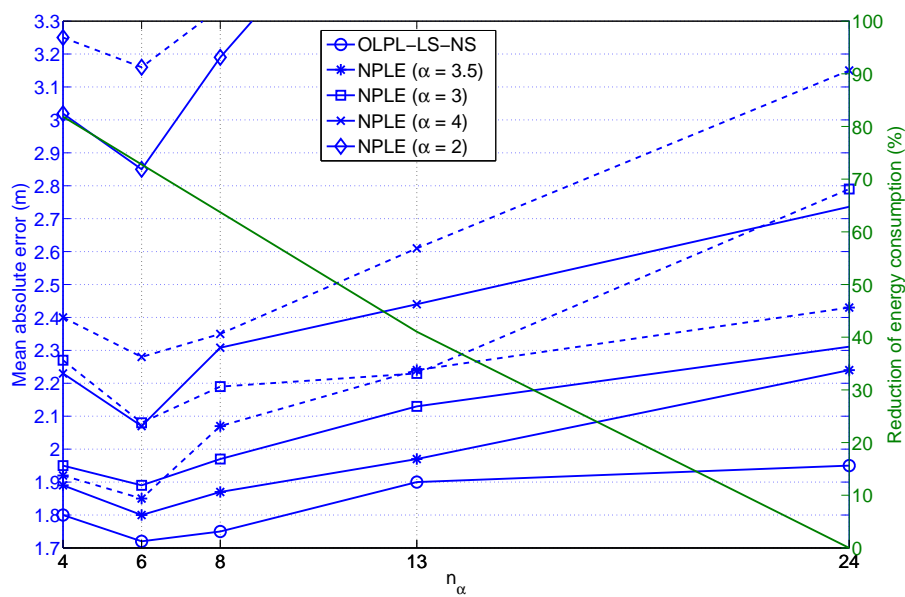
We could observe in Figure 12(a) that although we have increased the number of nodes and the scenario dimensions, the performance of the algorithm is similar to that achieved in the scenario shown in Figure 10(a). We have a higher scenario but also more nodes that can be closer. The problem is that the accuracy achieved is equal to 1.7 m in average.

If we observe the results achieved in Figure 12(b), the accuracy obtained is now 1.2 m. Increasing the number of anchor nodes in 2 contributes in a benefit of 0.5 m in the accuracy. Another important point is the benefit of 0.5 m, in terms of accuracy, when we are using our proposal with on-line path loss

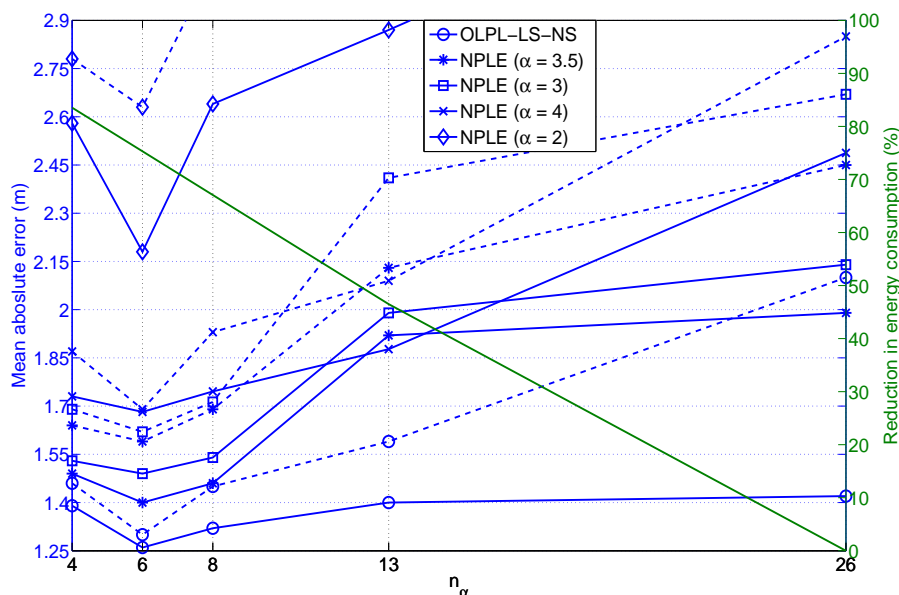
estimation compared to an algorithm without path loss estimation results. Finally, the reduction in terms of energy consumption is of, approximately, 75% in both scenarios.

In both Figures, one can observe the benefits of the proposed OLPL-NL-LS approach when compared with the case of assuming a constant path loss exponent. We achieve always a better result in terms of position accuracy with our proposal. For example, in the scenario 1 we achieved a minimum difference of 10 cm when $\alpha = 3.5$ and a maximum difference of 1 m when $\alpha = 2$. In the scenario 2 the results show that, when $\alpha = 3.5$ the difference is 15 cm and when the $\alpha = 2$ the difference is 0.9 m. We always achieve a benefit when the path loss is estimated on-line.

Figure 12. Mean absolute error versus number of cooperating nodes (n_δ) (solid line: experimental results, dashed line: simulation results.)



(a) 4 anchor nodes



(b) 6 anchor nodes

7. Conclusions

In this paper we have proposed a distributed cooperative RSS-based location algorithm with on-line path loss exponent estimation and node selection. Although RSS-based algorithm constitutes the simplest method, the necessity of having an accurate propagation model is revealed by the results shown. The introduction of a distributed on-line path loss estimation allows the algorithm to infer a good propagation model in a simple way. The need of an off-line calibration is avoided. Furthermore, we have presented an adaptive solution to track possible changes in the environment. Since wireless sensor networks are energy-constrained networks, we have also presented a node selection criterion that reduces the number of cooperating nodes. It has been shown that having a reduced number of cooperating nodes allows us to reduce the energy consumption without affecting the accuracy. Besides, we have presented experimental results that validate the proposed method, and we have compared it with ML and MDS, showing the efficiency in terms of the trade-off between energy consumption and accuracy.

Acknowledgements

The work has been supported by the Spanish Government under projects TEC2008-06305, the Catalan Government under grant 2009 SGR 298, and the Chair of Knowledge and Technology Transfer Parc de Recerca UAB—Santander.

References

1. Bachrach, J.; Taylor, C., Localization in sensor networks. In *Handbook of Sensor Networks: Algorithms and Architectures*; Stojmenovic, I., Ed.; Wiley and Sons, Inc.: New York, NY, USA, 2005.
2. Boukerche, A.; Oliveira, H.; Nakamura, E.; Loureiro, A. Localization systems for wireless sensor networks. *Wirel. Commun.* **2007**, *14*, 6-12.
3. Mao, G.; Fidan, B. *Localization Algorithms and Strategies for Wireless Sensor Networks*; Information Science Reference: Muenchen, Germany, 2009.
4. Gustafsson, F.; Gunnarsson, F. Mobile positioning using wireless networks: Possibilities and fundamental limitations based on available wireless network measurements. *IEEE Signal Process. Mag.* **2005**, *22*, 41-53.
5. Gezici, S.; Tian, Z.; Giannakis, G.; Kobayashi, H.; Molisch, A.; Poor, H.; Sahinoglu, Z. Localization via ultra-wideband radios: A look at positioning aspects for future sensor networks. *IEEE Signal Process. Mag.* **2005**, *22*, 70-84.
6. Mao, G.; Fidan, B.; Anderson, B.D. Wireless sensor network localization techniques. *Comput. Netw.* **2007**, *51*, 2529-2553.
7. Priyantha, N.B.; Chakraborty, A.; Balakrishnan, H. The Cricket Location-Support System. In *Proceedings of the 6th ACM International Conference on Mobile Computing and Networking, MobiCom 2000*, Boston, MA, USA, August 2000; pp. 32-43.
8. McCrady, D.; Doyle, L.; Forstrom, H.; Dempsey, T.; Martorana, M. Mobile ranging using low-accuracy clocks. *IEEE Trans. Microwave Theory Tech.* **2000**, *48*, 951-958.

9. Niculescu, D.; Nath, B. *Ad Hoc Positioning System (APS) Using AOA*. In *Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications, INFOCOM 2003*, San Francisco, CA, USA, 30 March–3 April 2003; Volume 3, pp. 1734–1743.
10. Yang, J.; Chen, Y. *Indoor Localization Using Improved RSS-Based Lateration Methods*. In *Proceedings of the 28th IEEE Conference on Global Telecommunications, GLOBECOM'09*, Piscataway, NJ, USA, November 30–December 4 2009.
11. Wymeersch, H.; Lien, J.; Win, M. *Cooperative localization in wireless networks*. *Proc. IEEE* **2009**, *97*, 427–450.
12. Li, X. *Collaborative localization with received-signal strength in wireless sensor networks*. *IEEE Trans. Veh. Technol.* **2007**, *56*, 3807–3817.
13. Patwari, N.; Ash, J.; Kyperountas, S.; Hero, A.O., I.; Moses, R.; Correal, N. *Locating the nodes: Cooperative localization in wireless sensor networks*. *IEEE Signal Process. Mag.* **2005**, *22*, 54–69.
14. Ash, J.; Moses, R. *On Optimal Anchor Node Placement in Sensor Localization by Optimization of Subspace Principal Angles*. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP-2008*, Las Vegas, NV, USA, 31 March–4 April 2008; pp. 2289–2292.
15. Bertsekas, D.P.; Tsitsiklis, J.N. *Parallel and Distributed Computation: Numerical Methods (Optimization and Neural Computation)*; Athena Scientific: Lexington, MA, USA, 1997.
16. Mazuelas, S.; Bahillo, A.; Lorenzo, R.; Fernandez, P.; Lago, F.; Garcia, E.; Blas, J.; Abril, E. *Robust indoor positioning provided by real-time rssi values in unmodified WLAN networks*. *IEEE J. Sel. Top. Signal Process.* **2009**, *3*, 821–831.
17. Zou, Y.; Chakrabarty, K. *Energy-Aware Target Localization in Wireless Sensor Networks*. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, Fort Worth, TX, USA, 23–26 March 2003; pp. 60–67.
18. Bel, A.; Vicario, J.L.; Seco-Granados, G. *Node Selection for Sooperative Localization: Efficient Energy vs. Accuracy Trade-Off*. In *Proceedings of the 5th IEEE International Symposium on Wireless Pervasive Computing*, Modena, Italy, May 2010; pp. 307–312.
19. Denis, B.; Pierrot, J.B.; Abou-Rjeily, C. *Joint distributed synchronization and positioning in UWB ad hoc networks using TOA*. *IEEE Trans. Microwave Theory Tech.* **2006**, *54*, 1896–1911.
20. Shang, Y.; Ruml, W.; Zhang, Y.; Fromherz, M.P.J. *Localization from Mere Connectivity*. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing, MobiHoc '03*, Annapolis, MD, USA, 1–3 June 2003; pp. 201–212.
21. Crossbow Technology. Available online: <http://www.xbow.com/> (accessed on 29 June 2011).