A Distributed Method for Localization in Large-Scale Sensor Networks based on Jarvis' March

Yassine Sabri
STIC Laboratory
Institute of Business and Engineering (ISGA)
El Jadida MOROCCO

Najib El Kamoun
STIC Laboratory
Chouaib Doukkali University, B.P: 20
El Jadida MOROCCO

ABSTRACT

This paper addresses target localization problem in a cooperative 3-D wireless sensor network (WSN). We employ a hybrid system that fuses distance and angle measurements, extracted from the received signal strength (RSS) and angle-of-arrival (AoA) information, respectively. Based on range measurement model and simple geometry, we derive a novel convex estimator based on Jarv's scan . The network is said to be uniquely localizable if there is a unique set of locations consistent with the given data.This paper presents an improved localization algorithm with high accuracy in large-scale Sensor networks with a large number of sensor nodes based on the Jarvis' March ,called SLSNJ. the Jarvis' March adapted here for our approximation technique to determining the convex hull of a set of sensors used instead of the Grid-Scan method,to take into account the requirements in memory, to make it scalable and rapidly convergent with small location estimation error.We verify our algorithm in various scenarios and compare it with AT-Dist method. Our simulation results show that the new estimators have excellent performance in terms of the estimation accuracy and convergence, and they confirm the effectiveness of combining two radio measurements in large-scale.

General Terms

Wireless Sensor Network (WSN), Routing

Keywords

Wireless Sensor Network (WSN), Routing, Multiple Sink, Localization, Geographic Routing

1. INTRODUCTION

In recent years wireless sensor networks (WSNs) have been used in various areas, like event detection (fires,floods) , monitoring (health care, industrial, agricultural, environmental) , energy-efficient routing , exploration (underground, deep water, outer space) , and surveillance , to name a few. A key element in many practical applications is to accurately determine the locations of sensors , namely in search and rescue missions or to enhance the network coverage. Although global positioning system (GPS) receivers can be used to locate the sensors, GPS is ineffective in indoor, dense urban and forest environments or canyons . Besides, installing a GPS receiver in each sensor would be extremely expensive in large-scale WSNs, which would restrict its applicability . Hence, development of localization strategies from different terrestrial radio frequency (RF) sources is of great practical interest. A wireless sensor network have been discussed for more than 30 years [1], but the vision of wireless sensor networks (WSNs) has been brought into reality only by the recent advances in wireless communications and electronics, which have enabled the development of low-cost, low-power and multi-functional sensors that are small in size and communicate over short distances. Existing researches for sensor localization mainly fall into two categories: range-based approaches and range-free approaches in **[10]**,**[13]**. Range-free approaches locate nodes using network connectivity information instead of accurate distance measurements between nodes.
This paper presents range-based method called Slsnj an improved of AT-Dist **[17]**. This method proposes a set of three rules and an approximation technique in order to assign either an exact position or an estimated position for each sensor node. The rules and the approximation technique use the data correlation between anchor positions and distances from nodes to anchors. As soon as a sensor node can apply one of rules, it obtains an exact position. Otherwise, by the approximation technique, it obtains an estimated position. With this approximation technique, using Jarvis' March **[11]** each sensor node defines a convex hull containing itself, according to the anchor positions and distances from it to anchors. To be located, this node computes an estimated position being the center of gravity of this convex hull.

Three important properties : first, a node can detect when its estimated position is relatively close to its real position. In this case this node becomes an estimated anchor and will be used by others nodes to obtain their positions. Second, some wrong informations (e.g. due to measure errors) can be eliminated related to defined sensor convex hull. These properties allow to obtain very good simulation results related to the methods described in **[15]**,**[18]** , **[20]** ,**[17]**, even if measure errors are introduced.third,Jarvis' March **[11]** fallowed us to reduce the consumption of CPU time (and therefore energy),But also allowed us to optimize including consumption of the memory,focusing not on the overall interpretation of network such as a type algorithm March-line but only on points of convex hull. consequently, We get to keep the functional properties of our localization technique despite change in network size with a minimum convergence time.

The rest of the paper is organized as follows: In Section 2, introduces basic notions for this problem.In Section 3, we summarize related work on localization algorithms. In Section 4, we present our new localization algorithm Slsnj. In Section 5, we evaluate the proposed scheme through comprehensive simulation studies and compare it with other localization techniques. We conclude the paper in Section 6.

2. MODEL

Extensive approaches have been proposed to locate sensor nodes in WSNs.In this paper ,we focuse on static networks. Moreover, it assumes that all sensors have identical reachability radius r. However, it is easy to adapt our method to sensors having different reachability radius. A wireless sensor networks is represented as a bidirectional graph $G(V, E)$ where V is the set of n nodes representing sensors and E is the set of m edges representing communication links. If two nodes $u, v \in V$ are neighbors, then they are linked that means distance between u and v is smaller than r. The set of neighbors for a node $u \in V$ is noted $N(u)$.
Anchor nodes have knowledge of their location through some other means, such as GPS or simply explicit programming. The set of anchors is noted Λ. The set of neighbor anchors for a node u is noted $N_\Lambda(u)(N_\Lambda(u) = N(u) \cap \Lambda)$ and the set of non-neighbor anchors is noted $\overline{N_\Lambda}(u)(\overline{N_\Lambda}(u) = \Lambda/N_\Lambda(u))$. Note that all identical nodes (anchors or others nodes) have the same capabilities (energy, processing, communication, ...). The coordinate of a position of node u is noted (x_u, y_u).$\mathcal{P}$ is the set of all possible positions in a network. Our method construct the convex hull of a point cloud $\mathcal{S}_u$ for each node u,this convex hull is noted $conv(\mathcal{S}_u)$. The localization modules (eg, GPS or Galileo) are expensive and consumers of energy,for this our method seeks to use the least possible anchors with the Nodes can use technology measures distances as ToA, RSSI, AoA. So, when it receives a signal from a transmitter, a node deduces that it is located on the circle centered on the transmitter. The exact distance between two nodes u and v is noted d_{uv} . Two neighbor nodes u, v know d_{uv} (via ToA, ...). The estimated distance is noted $\hat{d}_{uv}$. The following section explains how to obtain these estimated distance. the set of circles built from the knowledge of anchor neighbors is noted $\mathcal{C}_{N_\Lambda}$,the set of circles built from the knowledge of non-anchor neighbors is noted $\mathcal{C}_{\overline{N_\Lambda}}$.

ϵ is the distance between the estimated position $(x_{u_{estm_i}}, y_{u_{estm_i}})$ of the sensor u and the summit furthest from convex hull$Conv(\mathcal{S})$. Let d_{err} being the distance between the estimated position of a node and its real position, representing the position error. The node knows that d_{err} . By using a predefined threshold, if $d_{err} \leq threshold$ then the node has an estimation close to its real position. In this case the node becomes an estimated anchor and broadcasts its position .

3. RELATED WORKS

Many methods assume that some sensors in networks know their exact positions (by human intervention, GPS, ...). These sensors are called anchors. There are two categories among these methods : first, the range-free localization schemes which deduce estimated positions for all nodes in the network with only coordinates of anchors. Techniques described in **[10]**, **[14]**,**[9]** are examples of these methods. Second, the range-based localization which use techniques allowing to calculate distances between two neighbor sensors. The most popular methods in order to compute the range with two neighbor nodes are RSSI, ToA and TDoA **[19]** and AoA **[7]** : RSSI (Received Signal Strength Indicator) measures the power of the signal at the receiver. With the power transmission information, the effective propagation loss can be calculated and either theorical or empirical models are used to translate this loss into distance. ToA / TDoA (Time of arrival / Time difference of arrival) translates directly the prapagation time into distance if the signal propagation speed is known. For example, the most basic localization system using ToA techniques is GPS in [21]. AoA (Angle of arrival) estimates the angle at which signals are received and uses simple geometric relationships to calculate node positions.
Among localization methods in wireless sensor networks, the most popular are the methods of Niculescu and Nath APS in **[15]** , Savvides,al **[20]** and Savarese,al **[18]**. These methods use the same execution scheme. This plan contains three steps : first, anchors broadcast their position. Second, each node estimates distances with anchors. Each node derives an estimation of its position from its anchor distances. Finally, a refinement process is performed in order to improve accuracy of estimations. In **[12]** , Langendoen and Reijers provide a detailed comparative survey for each step of these methods. After the distance estimation step, there are two techniques in order to calculate node position : either multilateration, described above, used by **[18]** et **[15]** , or Min-Max technique, used by **[20]** : the main idea is to construct, for each node, a bounding box related to anchor positions and estimated distances, and then to determine the intersection of these boxes. . in AT-Dist **[17]**,This method based on the method for estimating distance Sum-Dist used by **[20]** and on a method based on the intersection of the disc centered by anchors nodes for each sensor that seeks its position,the intersection of these disc provided an area and the center of gravity of this area considered as the estimated position.AT-Dist method exploits the location error when the error is below certain threshold the sensor is also starting to broadcast its position estimated accompanied by localization error as the anchor.The implementation uses by this method to represent the network and areas constructed is Grid-Scan described in section *3.0.1*.

3.0.1 AT-Dist. In [17], authors present an interesting localization method. In a first time, nodes determine their positions with a position error bound using anchors positions, and when this position error bound goes below a given threshold on a node, this node is considerated as an estimated anchor and other nodes uses this information to improve the knowledge of their positions. Resulting localization information are provided with a position error bound, which is interesting as it can be used for geographical routing for example [16]. Simulation results show that AT-dist method performs accurate localization of the nodes when distance measurement errors are small results are clearly less accurate in large scal . Each node represents the network by a grid. The length of a grid side is set of $0.1r$ in order to guarantee that estimation accuracy is not noticeably compromised. When a node receives an anchor position, it increments the cases in the grid that may be its position :

—if the node and the anchor are not neighbors : all cases between the two circles : one with radius equals to r and the other with radius equals to estimated distance returned by Sum-Dist.

—if the node and the anchor are neighbors : all cases on the circle having as center the anchor of radius equals to the range.

Figure 1 represents an example of Grid-Scan : when node X receives the position of B (resp. C, D), it increments all cases being between the two circles centered in B (resp. C, D). The zone containing X is defined by the area composed by the cases with the maximum score. In figure 6 this zone is defined by cases equal to 3. X calculates the center of gravity of this zone and obtains an estimated position.

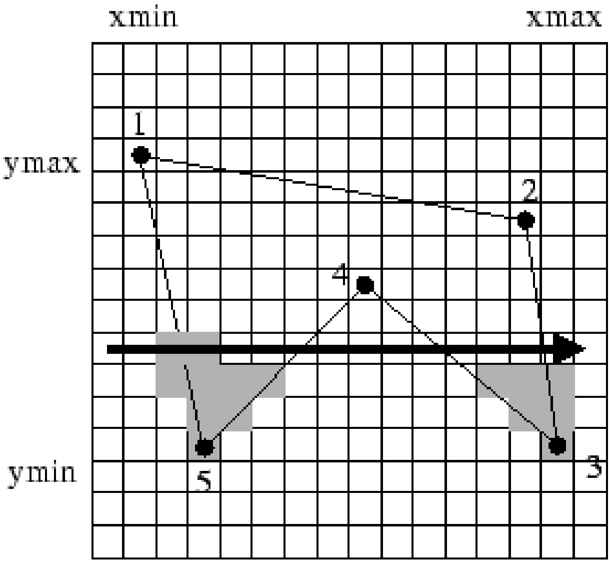


Fig. 1. *Grid-Scan*

3.0.1.1 Advantages and Drawbacks . Moreover, in an initial phase,each sensor must keep in its memory the field of interest of a sampled mannerusing grid Marchmethod,and anchors needs to flood the whole network, and then additional communications are added to improve sensors localization. This leads to an important exchange of messages.

4. SCALABLE LOCALIZATION SENSOR NETWORK BASED ON JARVIS MARCH

4.1 Description

Initially, each anchor broadcasts its position. A node can therefore be deduced the distance between each of the anchors We use the technique SumDist (Savvides et al., 2002) for estimating distances adding the distances between separated sensor nodes of an anchor.Upon receiving the position of a anchor, a node considers the following cases:

—if it receives directly the position of the anchor, he deduces they are neighbors and therefore it located on the circle centered at the anchor or radius of a circle is r (this Circle belongs to).

—if it receives the position by an intermediate node, it concluded that it is not neighbor of the anchor and therefore it is not inside the circle of radius r centered in anchor (this Circle belongs to).

So,when a node u receives a position of an anchor A, it estimates the distance to this anchor with Sum-Dist and draws one or two circles. In fact, if $(A \in N_\Lambda(u))$, u knows d_{Au} and deduces that it is on the circle $\mathcal{C}_{Au}$ of radius equals to d_{Au} and centered in A. If $(A \notin N_\Lambda(u))$ then u knows that it is not inside the circle of center A and radius r otherwise A and u would be neighbors. Moreover, u knows the estimated distance to A, $\hat{d}_{Au}$deduced by Sum-Dist. By triangular inequality, $\hat{d}_{Au} \leq d_{Au}$. u applies this technique to each received anchor position. So, u is inside the circle $\mathcal{C}_{\overline{Au}}$ of center A and radius $\hat{d}_{Au}$. Thus, the intersection of circles defines a cloud of points $\mathcal{S}_u$. the center of gravity of the convex hull of this cloud $conv(\mathcal{S}_u)$ represents the estimated position of u.

To summarize, for each node $u \in V/\Lambda$, the envelope obtained as follow : Initialization of the algorithm:

$$\mathcal{S}_0 = \mathcal{P} \tag{1}$$

$$\mathcal{C}_{N_\Lambda(u)_0} = \mathcal{C}_{\overline{N_\Lambda}(u)_0} = \{\emptyset\} \tag{2}$$

When a node u receives a message controle P from anchor node a_i neighbor:

If $a_i \in N_\Lambda(u)$:

The circle centered at a_i and of radius $d^2_{ua_i}$:

$$\mathcal{C}_{ua_i} = \{(x_i, y_i) \in \mathcal{P} | (x_i - x_a)^2 + (y_i - y_a)^2 = d_{ua_i}\} \tag{3}$$

Construction of intersection points of a circle $\mathcal{C}_{ua_i}$ with the old circles $\mathcal{C}_{(u)_{i-1}}$,keeping only the points inside a circle centered at u and of radius $\epsilon_{u_{i-1}}$:

$$\mathcal{W}_{u_i} = \{(x_i, y_i) \in (\mathcal{C}_{(u)_{i-1}} \cap \mathcal{C}_{ua_i}) | (x_i - x_{u_{estm_i}})^2 + (y_i - y_{u_{estm_i}})^2 \leq \epsilon^2_{u_{i-1}} \tag{4}$$

Cleaning the old cloud of points $\mathcal{S}_{i-1}$,keeping only the points inside a circle centered at u and of radius d_{ua_i}:

$$\mathcal{Z}_{u_i} = \{(x_i, y_i) \in \mathcal{S}_{i-1} | (x_i - x_a)^2 + (y_i - y_a)^2 \leq d^2_{ua_i}\} \tag{5}$$

New cloud of points $\mathcal{S}_i$:

$$\mathcal{S}_i = \mathcal{Z}_{u_i} \cup \mathcal{W}_{u_i}, i \geq 3 \tag{6}$$

The circle $\mathcal{C}_{ua_i}$ joins the old circles $\mathcal{C}_{N_\Lambda(u)_{i-1}}$:

$$\mathcal{C}_{N_\Lambda(u)_i} = \mathcal{C}_{ua_i} \cup \mathcal{C}_{N_\Lambda(u)_{i-1}} \tag{7}$$

Same effect occurs when a node u receives a message controle P from anchor node a_i not neighbor :

if $a_i \notin \overline{N_\Lambda}(u)$:

$$\mathcal{C}_{\overline{ua_i}} = \{(x_i, y_i) \in \mathcal{P} | (x_i - x_a)^2 + (y_i - y_a)^2 = \hat{d}^2_{ua_i}\} \tag{8}$$

$$\mathcal{W}_{u_i} = \{(x_i, y_i) \in (\mathcal{C}_{(u)_{i-1}} \cap \mathcal{C}_{\overline{ua_i}}) | (x_i - x_{u_{estm_i}})^2 + (y_i - y_{u_{estm_i}})^2 \leq \epsilon^2_{u_{i-1}} \tag{9}$$

$$\mathcal{Z}_{u_i} = \{(x_i, y_i) \in \mathcal{S}_{i-1} | r^2 \leq (x_i - x_a)^2 + (y_i - y_a)^2 \leq \hat{d}^2_{ua_i}\} \tag{10}$$

$$\mathcal{S}_i = \mathcal{Z}_{u_i} \cup \mathcal{W}_{u_i}, i \geq 3 \tag{11}$$

$$\mathcal{C}_{\overline{N_\Lambda}(u)_i} = \mathcal{C}_{\overline{ua_i}} \cup \mathcal{C}_{\overline{N_\Lambda}(u)_{i-1}} \tag{12}$$

$$\mathcal{C}_{(u)_i} = \mathcal{C}_{N_\Lambda(u)_i} \cup \mathcal{C}_{\overline{N_\Lambda}(u)_i} \tag{13}$$

The end for each node we will have a set of points $\mathcal{S}_u$ of the cloud:

$$\mathcal{S}_u = \{p_1, p_2, p_3, \cdots, p_n\} \tag{14}$$

Calculate the convex hull $\mathcal{S}_u$ based on Jarvis' March:

$$conv(\mathcal{S}_u) = \{\sum_{n=0}^{n} \alpha_i p_i | \alpha_i \geq 0, \sum_i \alpha_i = 1\} \tag{15}$$

The new estimation error ϵ_{u_i}:

$$\epsilon_{u_i} = \max_{p \in conv(\mathcal{S})} d(p, u_{estm_i}) \tag{16}$$

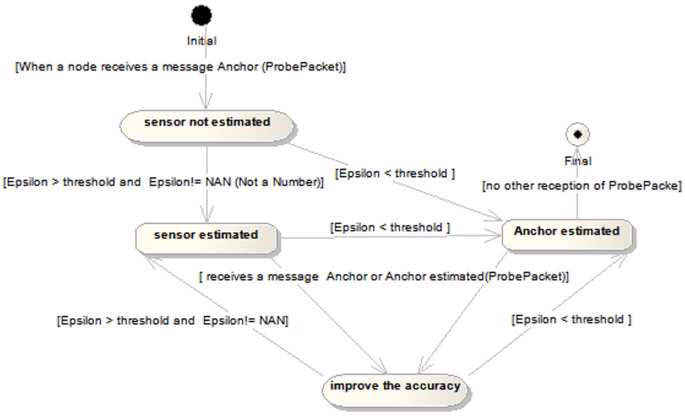

Initial
[When a node receives a message Anchor (ProbePacket)]
sensor not estimated
[Epsilon > threshold and Epsilon!= NAN (Not a Number)]
[Epsilon < threshold]
Final
[no other reception of ProbePacke]
sensor estimated
[Epsilon < threshold]
Anchor estimated
[receives a message Anchor or Anchor estimated(ProbePacket)]
[Epsilon > threshold and Epsilon!= NAN]
[Epsilon < threshold]
improve the accuracy

Fig. 2. State machine diagram for Sensor node not estimated

The main design of the Slsnj, which is a simple finite state machine. As shown in figure 2, a node running Slsnj is in one of four states at any time: (i) Sensor not estimated, (ii) Sensor estimated, (iii) estimated Anchor,and (iv) improve the accuracy. Transitions between the states are triggered by events. After the Slsnj protocol is initiated, the node enters the Sensor not estimated state,Whenever the node receives a broadcasting ProbePacket packet, the node enters the Sensor not estimated state and uses this packet to estimate its postion,after this stage of estimation the node switches to another state is depending on the value of the estimation error found,if espilon¡threshold the node enters in estimated Anchor state else it enters in Sensor estimated state .In the latter two states a node is still waiting of probpacket packet from anchor or estimated Anchor nodes to enter in improve the accuracy state and improve its accuracy.when there will be no more ProbePacket, the node switches to the state final and considered as estimated with an error of precison.

An example is illustrated in figure 1. X receives positions of anchors A, B and C. It estimates distances $\hat{d}_{AX}$, $\hat{d}_{BX}$, $\hat{d}_{CX}$ with Sum-Dist. Since all an- chors are not neighbors of X then X is not inside circles centered respectively in A, B, C with a radius equals to r but it is inside circles with radius equal to $\hat{d}_{AX}$, $\hat{d}_{BX}$, $\hat{d}_{CX}$. The intersection of these circles defines the cloud points $\mathcal{S}_X$ for a node X .X computes the center of gravity of the convex hull $conv(\mathcal{S}_X)$ of this cloud and estimates its position in $G2$.

4.2 Slsnj properties

Our localization technique meets three very important properties who have a significant impact on its performance:

—First, a node knows if its estimated position is close to its real position. Let ϵ be the distance between the center of gravity and the point, in the zone, furthest away from the center of gravity. Let d_{err} being the distance between the estimated position of a node and its real position, representing the position error. The node knows that $d_{err} \leq \epsilon$. By using a predefined $threshold$, if $\epsilon \leq threshold$ then the node has an estimation close to its real position. In this case the node becomes an estimated anchor and broadcasts its position and its ϵ . When a node applies the approximation technique with an estimated anchor radius, it takes into account ϵ.

Consider a sensor X calculating its position with an estimated anchor A. If they are neighbors, X trace two circles (belongs to $\mathcal{C}_{N_\Lambda}$) centered in A of radius $d_{AX} \pm \epsilon$ and deduce that it is between these two circles. If they are not neighbors, X deduces that it is not inside the circles centered at A of radius $r - \epsilon$and belongs to a circle of radius $d_{AX} + \epsilon$,the definitions (4),(6),(9) and (11) become :
si $a_i \in N_\Lambda(u)$:

$$\mathcal{C}_{ua_i} = \{(x_i, y_i) \in \mathcal{P} | (x_i - x_a)^2 + (y_i - y_a)^2 = (d_{ua_i} \pm \epsilon_{u_i})^2\} \tag{17}$$

$$\mathcal{Z}_{u_i} = \{(x_i, y_i) \in \mathcal{S}_{i-1} | (x_i - x_a)^2 + (y_i - y_a)^2 \leq (d_{ua_i} \pm \epsilon_{u_i})^2\} \tag{18}$$

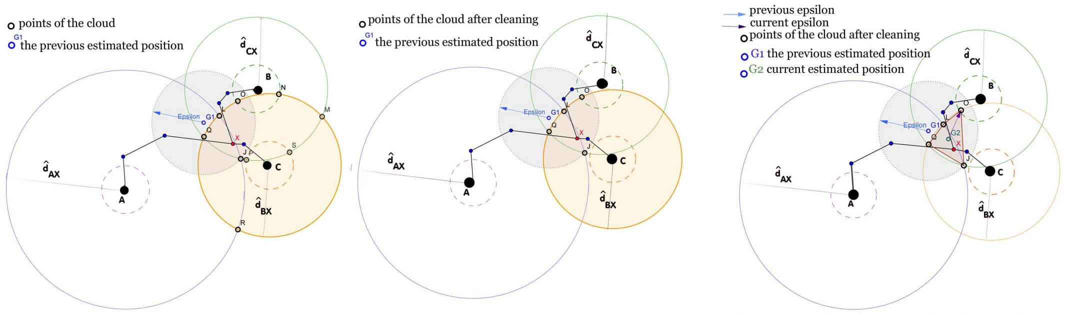
if $a_i \notin \overline{N_\Lambda}(u)$:

$$\mathcal{C}_{\overline{ua_i}} = \{(x_i, y_i) \in \mathcal{P} | (x_i - x_a)^2 + (y_i - y_a)^2 = (\hat{d}_{ua_i} \pm \epsilon)^2\} \tag{19}$$

$$\mathcal{Z}_{u_i} = \{(x_i, y_i) \in \mathcal{S}_{i-1} | (r - \epsilon)^2 \leq (x_i - x_a)^2 + (y_i - y_a)^2 \leq (\hat{d}_{ua_i} + \epsilon)^2\} \tag{20}$$

—Second, a node can detect if some informations are wrong. This case is illustrated in esxpresion $\mathcal{W}_{u_i}$ .With its bound error ϵ , nodes reject the cloud points that are outside of circle centered at its estimated position and of radius ϵ.for example, when a node u detects a point of its cloud $\mathcal{S}_u$ it outside in the circle centered at u of radius ϵ will not take it into account . This property is defined by the expression $\mathcal{W}_{u_i}$.

—Third, Jarvis' March **[11]** allowed us to calculate the convex hull $conv(\mathcal{S})$ a cloud of points with a very optimum complexity , of order $O(n \log(n))$ with n the number of points of the cloud, which allowed us to reduce consumption of CPU time (and therefore energy), but also allowed us to optimize particularly the consumption of memory storage ,focusing not on global interpretation of the network as in an algorithm of type Grid-Scan presented in *3.0.1*, but only on points of the cloud. The improvement made allowed us to retain the properties functional Our localization technique despite the change in network size, and efficiently localize the nodes (continuously) and with a certain level of quality in different scales.


points of the cloud
G1 the previous estimated position
points of the cloud after cleaning
previous epsilon
current epsilon
G1 the previous estimated position
G2 current estimated position
Epsilon
G1
G2
A
B
C
X
J
N
M
S
R
$\hat{d}_{AX}$
$\hat{d}_{BX}$
$\hat{d}_{CX}$

Step 1: Construction of the cloud of points when x receives the information of localization from the anchor C

Step 2: cleaning the point cloud based on the previous epsilon

Step 2: construction of the convex hull and the calculation of the new position estimated G2

Fig. 3. Estimated position for X at the reception of localization information from C anchor

5. EXPRIMENTATIONS ET RSULTATS

5.1 Environnement de simulation

Experiments were built upon the J-Sim simulator [9] dedicated to WSN simulations. It is a compositional, component-based simulation environment. It is built upon the concept of autonomous component programming model. J-Sim is developed entirely in Java. The signal attenuation due to obstacles or other factors (e.g. use of unidirectional antennas) is simulated in J-Sim. Therefore, the vicinity of a node in terms of transmission range is not necessarily spherical.Note that there several simulators in the literature such as GlomoSim**[4]** , OMNET++**[5]** , OPNET**[3]** , NS-2**[2]** . The MAC layer is considered perfect and the transmission of messages are without loss in our simulation.

In the field of localization in of wireless sensors networks and services,The scalability was analyzed as a problem of performance where enough variety of metrics were considered. These metrics are concentrated around the measurement of response time, , Consumption of resources and the number of messages exchanged between nodes.the factor scale most considered in the literature is the number of nodes.This section analyzes the performance of our method Slsnj following three metrics: :accuracy, storage space, complexity,In order for us show the good performance of our protocol in large networks. In order to allow easy comparison between different scenarios, range errors as well as errors on estimated positions are normalized to the radio range. For example, 50% of position error means a distance of half the range of the radio between the real and estimated positions. The percentage of range errors is noted δ.

5.2 Results

In figure 4 When the value of confidence is equal to 3, the obtained error mean is the best. In fact, when the value of confidence is higher than 3, the voting process is very strict and nodes cannot deduce their positions. Conversely, when the value of confidence is lower than 3, the voting process assigns in some times bad positions to sensors because it uses a few number of anchor positions and some wrong informations can be used. This comment is confirmed when increases. But, it is possible that this value increases when the percentage of range errors is higher than 15. In the next experiences the value of confidence is equal to 3.

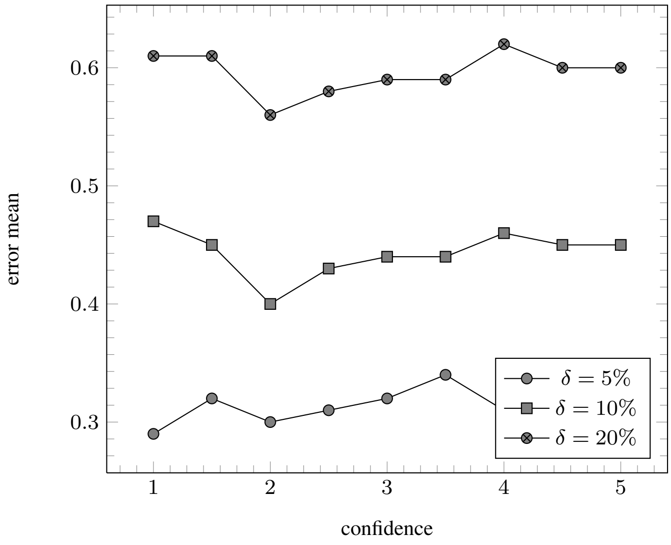


Fig. 4. *Slsnj : impact du seuil confidence*

5.2.1 The accuracy. We compared our algorithm Slsnj with the distributed method AT-Dist ,The positions to estimate are generated randomly on a surface $\mathcal{A} = L \times L$ with dimensions of experimentation varying between 100×100 to 800×800 and a density of anchor $d = 20$, each configuration obtained is repeated for each of the two methods.the range of the sensors was set at 14.

Globally, the positions determined by a localization algorithm represent a geometrical layout of the physical positions of the sensors. This layout must be compared to the ground truth, or known layout of the sensors. It is important therefore that not only the error between the estimated and real position of each node is minimised, but also that the geometric layout determined by the algorithm matches well the original geometric layout. In order to have a unified approach for evaluate the accuracy of our technique and a solid frame for analysis of the scalability,we propose to use two metrics.

—MAE(Mean Absolute Error): The simplest way to describe localization performance is to determine the residual error between the estimated and actual node positions for every node in the network, sum them and average the result. Broxton et al in [8] do this using the mean absolute error metric (MAE), which, for each of n nodes in the network, calculates the residual between the nodes estimated and actual coordinates.

$$MAE = \frac{\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 - (y_i - \hat{y}_i)^2}{n} \tag{21}$$

avec (x_i, y_i) Positions relles et $(\hat{x}_i, \hat{y}_i)$ les postillons estims.

—GDE (Global Distance Error): As discussed briefly at the start , it is important for the accuracy metric to reflect not only the positional error in terms of distance, but also in terms of the geometry of the network localization result. GDE in [6] takes the RMS error over the network of n nodes and normalises it using the constant R. In Ahmed et als context, R represents average radio range, meaning the localization results are represented as a percentage of the average distance nodes can communicate over.

$$GDE = \frac{1}{r}\sqrt{\frac{\sum_{i=1}^{n}\sum_{j=i+1}^{n}(\frac{\hat{d}_{ij}-d_{ij}}{d_{ij}})^2}{n(n-1)/2}} \tag{22}$$

avec $\hat{d}_{ij}$ La distance estime entre i et j et d_{ij} La distance relle entre i et j .

Subsequently, the simulations will highlight the influence of following parameters on the performance of our method:

—The density network and dimension of the network;

—Measurement errors δ that will take the values 0%,5%,10% ;

—The percentage of anchors noted α,are selected randomly among the network nodes .

The simulations for $\alpha \in \{2, 4, \cdots, 18, 20\}$ representing density of anchors from 0.12 to 1.23 and δ equals to 0 (the ideal case) and dimensions L = 200. the Corresponding graphs are represented in figures 14 and 15.

The graph of figure 5 represents the performance respectively Slsnj and AT-Dist in a small scale ($L \leq 400$) , when range errors are introduced, the behavior of average error rate MAE of our method related to percentage of anchors. These curves indicate the accuracy of localizations when δ is equal to $\delta = \{0, 5, 10\}\%$. Without surprise, performances of Slsnj decrease when range errors increase as the method AT-Dist (with $L = 200$). But, our method keeps a good estimation of positions. Note also that after 10% of anchors the average error rate decreases slowly.

For underline the capacity of the methods to localize sensors with precision,reference should be made to the graph of Figure 6 and 7. The graph represents the percentage of nodes locatedof Slsnj and AT-Dist for a percentage of anchors varies from $0\% to 20\%$ without errors $\delta = 0\%$. The anchors located by GPS are not taken into account. In others words, the percentage of new exactly located nodes is only considered. For Slsnj, the results are very clear and stable when we move to large scale ($L \geq 400$) with $\alpha = 20\%$: for Slsnj The percentage of nodes located with a position error less than 20% clearly exceeds the 86% but does not exceed 75% for AT-Dist.

Figure 8 shows the impact of density of nodes in small large (($L \geq 400$))on the behavior of average error rate MAE. When the density of nodes increases, the average error rate decreases. In fact, with low density, nodes do not often use rules but only the approximation technique. Therefore, a few number of anchors (estimated or not) are added. The opposite phenomenon occurs when density of nodes increases. Note that after a density of nodes equals to 12, the behavior of average error rate is not significative.

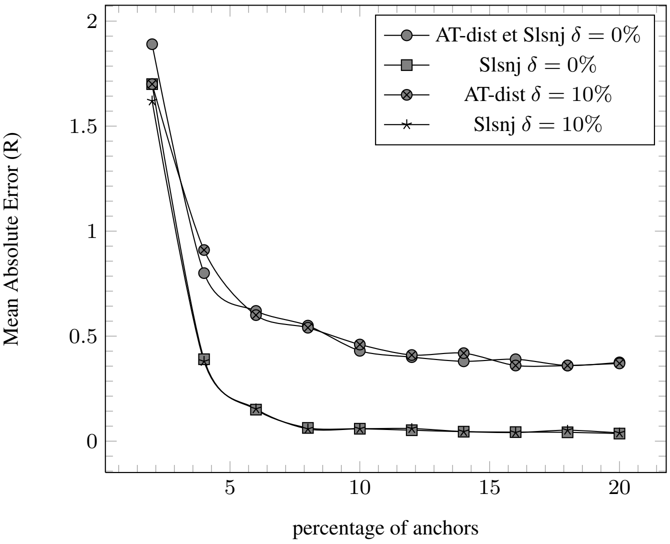


Fig. 5. *Slsnj and AT-Dist rate of the average error (MAE) with a dimension* $L = 200$

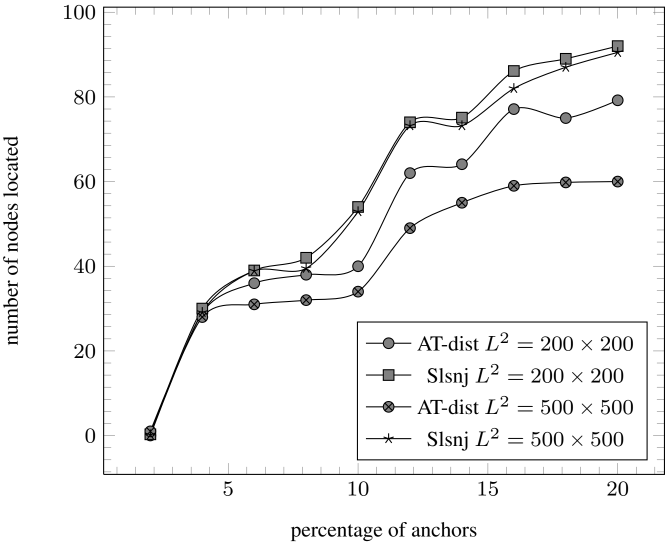


Fig. 6. *Slsnj and AT-Dist: Percentage of localized nodes with a position error less than* 20% *with* $\delta = 0\%$ *and* $L = 200, 400$

Figure 9 shows the impact of the dimension of network on the behavior of average error rate GDE. When the dimension of network increases, the average error GDE remained stable for Slsnj and increases for AT-Dist. In fact, a large scale, the comportemente of our technique based on Jarvis'ss Marchremains stable and capable of operating,but the method AT-Dist based on March-line stabilizes do it that are incapable of representing the network in its memory by the method March-line.

5.2.2 The memory consumption. We also measure the amount of memory allocated before the end of the simulation. The memory

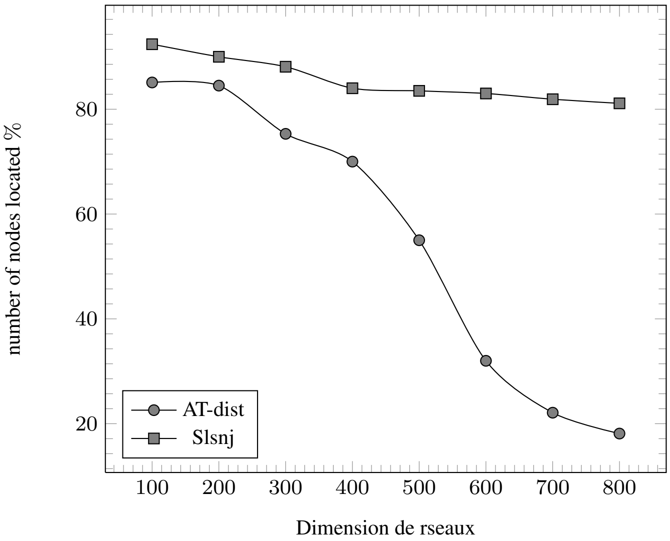


Fig. 7. *Slsnj and AT-Dist: Percentage of localized nodes with a position error less than* 20% *with* $\delta = 0\%$ *with L varied from* 100 *to* 800

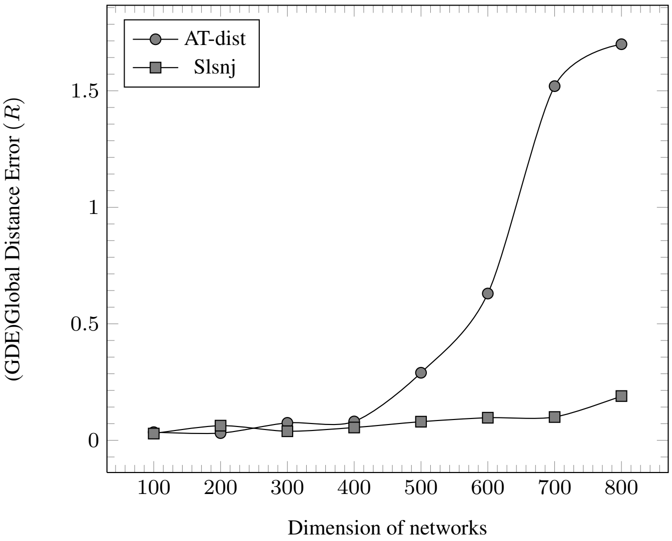


Fig. 9. *Slsnj and AT-Dist:Global Distance Error with* $\delta = 0\%$ *and density of anchors* $\alpha = 15\%$

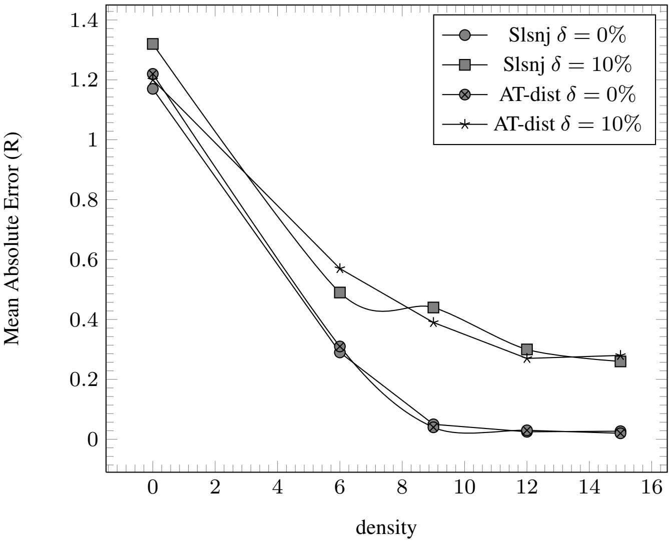


Fig. 8. *Slsnj and AT-Dist : impact of the density with a dimension L=200*

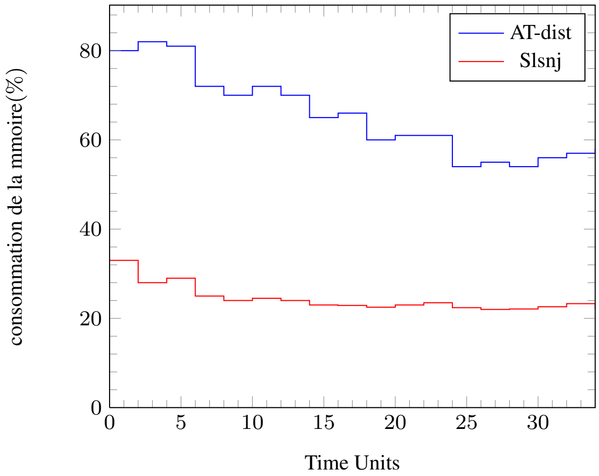


Fig. 10. *Slsnj and AT-Dist: rate of consumption of the memory with* $\delta = 0\%$ *and density of anchors* $\alpha = 20\%$ *and dimension L=400*

usage before the end of the simulation represents the amount of memory allocated to complete the 300 s simulation. As shown in figure 10 and figure 11, Slsnj use less memory than AT-Dist in large-scal. This demonstrates that the data structures are used in a more scalable manner in Slsnj to represent different classes and their interaction in the WSN framework. and the Jarvis's method used by our technique Slsnj to reduce the memory used.

5.2.3 Algorithmic Complexity. Standard notions of computational complexity in time and space (i.e. big O notation) can be used as comparison metrics for the relative cost of localization algorithms. For example, as a network increases in size, a localization algorithm with $O(n^3)$ complexity is going to take a longer time to converge than an $O(n^2)$ algorithm. The same is true for space complexity as the number of nodes increases, the amount of RAM needed (either per node, or centrally) is going to increase at a particular rate; algorithms which require less memory (comparatively) at a given scale may be preferable.

Which makes the protocol Slang converges faster than AT-dist it is the use of Jarvis' March that it has a complexity of order $O(n \log(n))$ with n is the number of points in the cloud,instead of using the grid Marchmethod used by AT-Dist of complexity of order $O(n^2)$ with n the number of subdivisions of the network.

Figures 13 and 12 show the evolution the location accuracy convergence. depending on the size of networks. in first graph, the convergence time increases linearly with the dimension, and in the second graph represents the evolution of convergence time that is the time when the Metric MAE is stabilized over time. convergence time with our method in a dimension 400 × 400 corresponds to $65s$ and $190s$ with at-dist . In fact, the main particularity of our

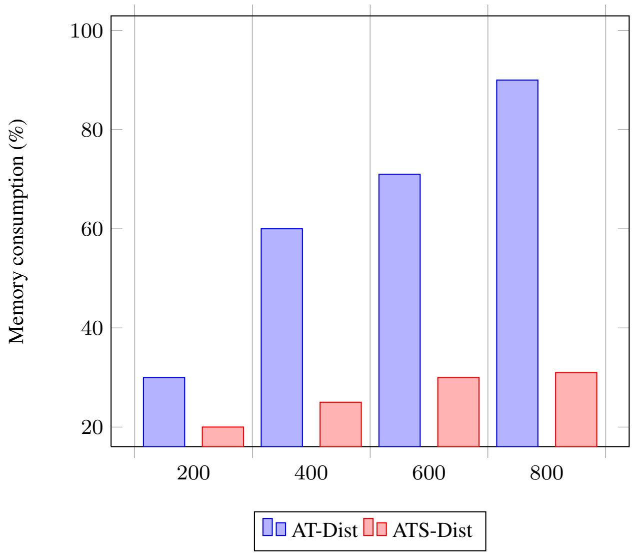


Fig. 11. *Slsnj and AT-Dist: rate of consumption of the memory with* $\delta = 0\%$ *and density of anchors* $\alpha = 20\%$ *and L varied from* 100 *to* 800

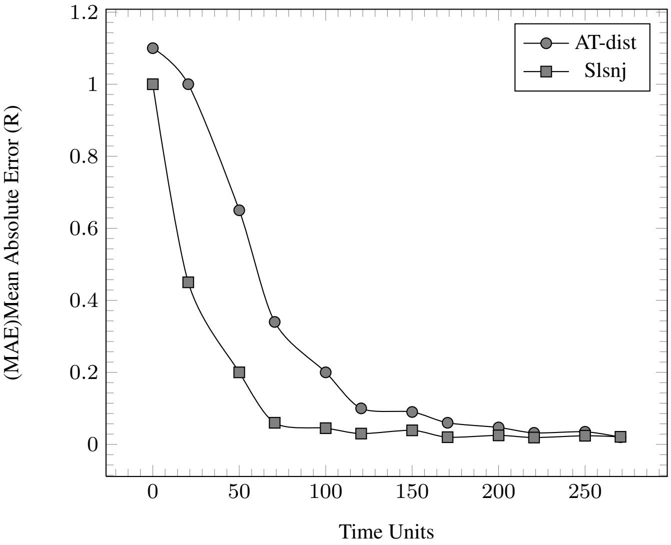


Fig. 12. *Mean Absolute Error with* $\delta = 0\%$ *,* $\alpha = 20\%$ *and L=200*

protocol is that the complexity does not depend on the dimension of networks,but the number of nodes constructing the convex hull. .

6. CONCLUSION

In this work, we proposed novel distributed algorithm to solve the RSS/AoA localization problem for known in large scale transmit powers based on Jarvi's march relaxation technique . The proposed SLSNJ algorithm provides exceptional localization accuracy in just a few iterations. Our algorithm based on GTRS framework is solved via a simple bisection procedure, and it represents an excellent alternative to our SLSNJ algorithm, since its somewhat lower accuracy is the simulation results confirmed the robustness of the proposed algorithms to the imperfect knowledge , which is a very important practical scenario. SLSNJ presents three important advantages : first, this technique eliminates some wrong propagated informations. These wrong informations are due to range errors or attackers who have the control of sensors. Second, a node knows if its estimated position is close to its real position and in this case, it becomes an estimated anchor. Third, Jarvis' March allowed us to reduce the consumption of CPU time (and therefore energy,memory).

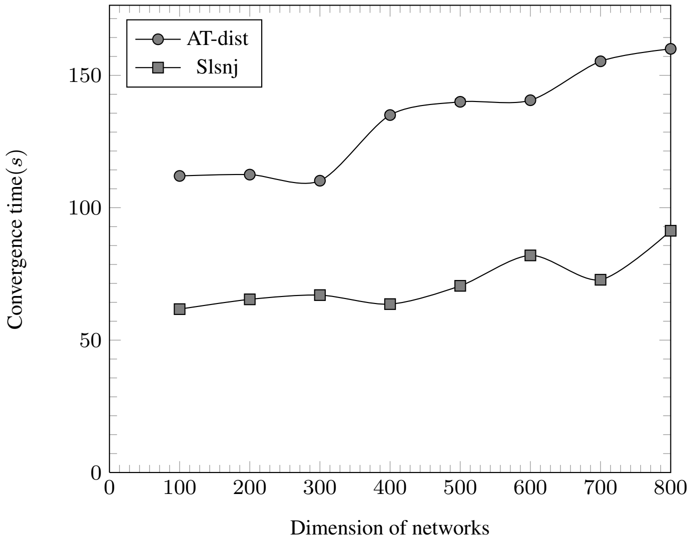


Fig. 13. *Slsnj and AT-Dist:Convergence time with* $\delta = 0\%$ *and density of anchors* $\alpha = 20\%$

7. REFERENCES

[1] *Localization algorithms and strategies for wireless sensor networks*, pages 2–30. Information Science Reference (an imprint of IGI Global), thirteenth edition, 2009.

[2] The network simulator. `http://www.isi.edu/nsnam/ns/`, cited July 2010.

[3] Opnet technologies. `http://www.opnet.com/`, cited July 2010.

[4] About glomosim. `http://pcl.cs.ucla.edu/projects/glomosim/`, cited July 2011.

[5] Omnet++ community site. `http://www.omnetpp.org/`, cited July 2011.

[6] Ahmed A. Ahmed, Hongchi Shi, and Yi Shang. Sharp: A new approach to relative localization in wireless sensor networks. In *Proceedings of the Second International Workshop on Wireless Ad Hoc Networking - Volume 09*, ICDCSW '05, pages 892–898, Washington, DC, USA, 2005. IEEE Computer Society.

[7] Mustapha Boushaba, Abdelhakim Hafid, and Abderrahim Benslimane. High accuracy localization method using aoa in sensor networks. *Comput. Netw.*, 53(18):3076–3088, December 2009.

[8] Michael Broxton, Joshua Lifton, and Joseph A. Paradiso. Localization on the pushpin computing sensor network using spectral graph drawing and mesh relaxation. *SIGMOBILE Mob. Comput. Commun. Rev.*, 10:1–12, January 2006.

[9] Nirupama Bulusu, John Heidemann, and Deborah Estrin. Gps-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, 7(5):28–34, October 2000.

[10] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher. Range-free localization schemes in large scale sensor networks. In *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, 2003.

[11] R. A. Jarvis. On the identification of the convex hull of a finite set of points in the plane. *Inf. Process. Lett.*, 2(1):18–21, 1973.

[12] Koen Langendoen and Niels Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Comput. Netw.*, 43:499–518, November 2003.

[13] Mo Li and Yunhao Liu. Rendered path: range-free localization in anisotropic sensor networks with holes. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, MobiCom '07, pages 51–62, New York, NY, USA, 2007. ACM.

[14] D Niculescu. Ad hoc positioning system (aps) using aoa. *IEEE INFOCOM 2003 Twentysecond Annual Joint Conference of the IEEE Computer and Communications Societies IEEE Cat No03CH37428*, 00(C):1734–1743, 2003.

[15] Dragos Niculescu and Badri Nath. Ad hoc positioning system (aps). In *IN GLOBECOM*, pages 2926–2931, 2001.

[16] Clément Saad, Abderrahim Benslimane, Julien Champ, and Jean-Claude König. Ellipse routing: A geographic routing protocol for mobile sensor networks with uncertain positions. In *GLOBECOM*, pages 78–82, 2008.

[17] Clément Saad, Abderrahim Benslimane, and Jean-Claude König. AT-Dist: A Distributed Method for Localization with High Accuracy in Sensor Networks. *International journal Studia Informatica Universalis, Special Issue on "Wireless Ad Hoc and Sensor Networks"*, 6(1):N/A, 2008.

[18] Chris Savarese, Jan M. Rabaey, and Koen Langendoen. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In *Proceedings of the General Track of the annual conference on USENIX Annual Technical Conference*, pages 317–327, Berkeley, CA, USA, 2002. USENIX Association.

[19] Andreas Savvides, Chih-Chieh Han, and Mani B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, MobiCom '01, pages 166–179, New York, NY, USA, 2001. ACM.

[20] Andreas Savvides, Heemin Park, and Mani B. Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, WSNA '02, pages 112–121, New York, NY, USA, 2002. ACM.

[21] Kamin Whitehouse and David Culler. Calibration as parameter estimation in sensor networks. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, WSNA '02, pages 59–67, New York, NY, USA, 2002. ACM.