

AT-Dist: A Distributed Method for Localization with High Accuracy in Sensor Networks

Clément Saad, Abderrahim Benslimane, Jean-Claude König

► **To cite this version:**

Clément Saad, Abderrahim Benslimane, Jean-Claude König. AT-Dist: A Distributed Method for Localization with High Accuracy in Sensor Networks. *Studia Informatica Universalis*, Hermann, 2008, Special Issue on "Wireless Ad Hoc and Sensor Networks", 6 (1), pp.14-39. lirmm-00270283

HAL Id: lirmm-00270283

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00270283>

Submitted on 4 Apr 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

AT-Dist: A Distributed Method for Localization with High Accuracy in Sensor Networks

Clément Saad* — Abderrahim Benslimane* — Jean-Claude König**

* LIA / CERI, 339 chemin des Meinajaries, BP 1228-84911 Avignon, France

saad@lirmm.fr

benslimane@lia.univ-avignon.fr

** University Montpellier 2 - LIRMM, 161 Rue Ada, F-34392 Montpellier, France

konig@lirmm.fr

RÉSUMÉ. *Beaucoup d'applications dans les réseaux de capteurs sont basées sur la connaissance des positions des capteurs. A priori, quelques capteurs connaissent leurs positions (via GPS). L'objectif de ce papier est de permettre le positionnement des capteurs restants avec précision. Les technologies existantes (e.g. ToA - Time of Arrival) permettent de calculer les distances entre deux nœuds voisins. Une nouvelle technique distribuée basée sur les distances, appelée AT-Dist, est présentée. Elle utilise deux propriétés importantes : premièrement, elle permet de localiser des capteurs tout en fournissant la précision de ces positions reformulées. Ainsi, un capteur peut déduire si sa position estimée est proche de sa position réelle et contribuer au positionnement des autres capteurs. Deuxièmement, AT-Dist gère l'introduction des erreurs de mesure qui est le principal défaut des méthodes basées sur les distances. En conséquence, AT-Dist localise avec des positions exactes ou estimées les capteurs. Les simulations montrent l'efficacité de notre méthode par rapport à d'autres.*

ABSTRACT. *Many sensor network applications are based on the knowledge of sensor locations. A priori, some sensors know their position (via GPS). The objective of this paper is to allow positioning of remaining sensors with high accuracy. Existing technologies (e.g. ToA - Time of Arrival) allow sensors to calculate ranges with their neighbor nodes. A new and original range-based distributed technique, called AT-Dist, is presented. It uses two important properties: first, it allows to locate sensor nodes while giving them the accuracy of their position. Thus, a sensor node can deduce if its estimated position is close to its real position and contribute to the positioning of others nodes. Second, AT-Dist manages introduction of measure errors which is the main drawback of range-based methods. As a result, AT-Dist locates with an exact or*

accuracy position many sensors. Simulations show the efficiency of our method in comparison to others.

MOTS-CLÉS : Réseaux de capteurs, localisation, mesure de distances, algorithme distribué.

KEYWORDS: Sensor networks, localization, distance measure, distributed algorithm.

1. Introduction

Ad-hoc wireless sensor networks have been proposed in many applications. Target tracking, intrusion detection, medical applications, climate control or disaster management are examples of applications. The localization of nodes can be used for routing or others location based services (Ko *et al.*, 1998, Li *et al.*, 2000). The localization problem in wireless sensor networks focused the interest of researchers during the ten last years. Sensors are small devices able to detect an event. Each sensor has a perception radius and if another sensor is in its perception then the two sensors are neighbors. In some case, sensors are equipped with a little battery. Therefore, only some operations (computations and especially communications) have to be performed. An example often used of ad-hoc wireless sensor networks is the aircraft deployment of sensors in a given area. In this network, only some nodes know their localizations (e.g. via GPS or other). These nodes are called *anchors*. A maximum number of remaining nodes have to deduce their positions according to anchor positions. Nevertheless, the number of anchors has to be as small as possible because nodes which are equipped GPS consume more energy. The network has to be self-organizing, in other words, it does not depend on global infrastructure. Proposed solutions must take into account all sensor characteristics.

Many methods have been proposed in order to resolve the problem of positionment in wireless sensor networks. Among these techniques, we can find two categories : Range-Free Localization Schemes and Range-Based Localization Schemes. The first category contains all methods only using anchor positions in order to locate all sensors. The second category contains all methods which use techniques allowing node to calculate either distances or angles with its neighbors. Measures obtained by these techniques can be perturbed by errors due to the network environment. These errors are called *measure errors* or *range errors*. They represent the most important drawback for methods based on distances.

Among these rang-based methods, the most popular techniques are described in (Niculescu *et al.*, 2001, Savarese *et al.*, 2002, Savvides *et al.*, 2002). They are divided into three steps : Each node estimates its distances to anchors, computes an estimation of its position, and then performs a refinement process in order to improve the accuracy of estimation. These methods are described in section 3.

This paper presents a new range-based method called AT-Dist. 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, each sensor node defines a restricted zone 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 zone. AT-Dist proposes **two 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 zones. These properties allow to obtain very good simulation results related to the methods described in (Niculescu *et al.*, 2001, Savarese *et al.*, 2002, Savvides *et al.*, 2002, Niculescu *et al.*, 2003a), even if measure errors are introduced.

The remainder of the paper is organized as follows : Section 2 introduces basic notions for this problem. Section 3 discusses previous works in sensor localization problem. Section 4 explains the approximation technique AT-Dist and presents its two main properties. Section 5 presents the rules of the approximation technique. Section 6 discusses simulation results where our method is compared to the three others methods (Niculescu *et al.*, 2001, Savarese *et al.*, 2002, Savvides *et al.*, 2002). Finally, section 7 gives the conclusion of the paper.

2. Model

Many localization algorithms have been proposed for static wireless ad hoc or sensor networks even if in some applications nodes may be mobile (Saad *et al.*, 2007, Bulusu *et al.*, 2001, Priyantha *et al.*, 2005). This paper focuses 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 $N(u)$.

A priori, some anchors have knowledge of their own position with respect to some global coordinate system. The set of anchors is noted Δ . The set of neighbor anchors for a node u is noted $N_{\Delta}(u)$ ($N_{\Delta}(u) = N(u) \cap \Delta$) and the set of non-neighbor anchors is noted $\overline{N_{\Delta}}(u)$ ($\overline{N_{\Delta}}(u) = \Delta \setminus N_{\Delta}(u)$). Note that all identical nodes (anchors or others nodes) have the same capabilities (energy, processing, communication, ...). The coordinate of a position p_u of node u is noted (x_u, y_u) . \mathcal{P} is the set of all possible positions in a network.

This paper assumes that each node can compute its distances to its neighbors when it receives signals. 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.

AT-Dist defines zones containing nodes. The zone of node u , noted Z_u , is the set of coordinates such as real coordinate of u belongs to Z_u .

Figure 1 represents a network with 12 nodes : 3 anchors (black nodes) and 9 not positioned nodes (white nodes).

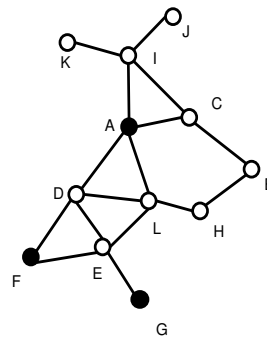


Figure 1. An example of network

2.1. Localization problem notation

In order to organize all localization problems in wireless sensor networks, the following notation is proposed :

$$\langle \{M, S\}, \{M, S\}, \{\emptyset, dist, angle\} \rangle$$

the first (resp. second) field defines if nodes (resp. anchors) are mobile or static (M for mobile, S for static). The last field determinates the capacity of sensors. If a sensor can calculate angles (resp. distances), the value of the last field is assigned to angle (resp. dist). Otherwise, the value is assigned to \emptyset . This paper focuses on configuration $\langle S, S, dist \rangle$.

3. Related works

A large number of existing techniques attempt to solve the localization problem (Nicleseu *et al.*, 2001, Savarese *et al.*, 2002, Savvides *et al.*, 2002, Nicleseu *et al.*,

2003a, Savvides *et al.*, 2001, Saad *et al.*, 2006, Niculescu *et al.*, 2004, He *et al.*, 2005, Chan *et al.*, 2005, Datta *et al.*, 2006).

Detailed surveys are provided in (Hightower *et al.*, 2001, Langendoen *et al.*, 2003). These solutions can be organized in three categories :

- GPS-free methods, that means that a node does not need anchors to locate itself. For example, method in (Čapkun *et al.*, 2001) builds a virtual system of coordinates and the node computes its position in this system.
- Infrastructure-based systems, which need infrastructure like RADAR(Bahl *et al.*, 2000) or Cricket(Priyantha *et al.*, 2000).
- Robot-based systems. In (Bulusu *et al.*, 2001), authors proposed a method which uses robots to locate nodes.

3.1. Anchor-based methods

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 (He *et al.*, 2005, Niculescu *et al.*, 2003b, Bulusu *et al.*, 2000) 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 (Bahl *et al.*, 2000), ToA (et al., 1996), TDoA (Savvides *et al.*, 2001) and AoA (Niculescu *et al.*, 2003a) :

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 theoretical or empirical models are used to translate this loss into distance.

ToA / TDoA (Time of arrival / Time difference of arrival) translates directly the propagation time into distance if the signal propagation speed is known. For example, the most basic localization system using ToA techniques is GPS (et al., 1996).

AoA (Angle of arrival) estimates the angle at which signals are received and uses simple geometric relationships to calculate node positions.

Of course, the accuracy of these measures depends on network's environment. These errors are called *measure errors* or *range errors*. In (Venkatraman *et al.*, 2002, Venkatraman *et al.*, 2003), authors analyze respectively the impact of range and angle errors.

The classical method to compute the node's position is the multilateration : as soon as

a node estimates its distances to at least three anchors, it computes its exact position when anchors are node's neighbors, otherwise, the position is estimated. For example, let X be a node and A, B, C anchors. X wants to compute its position. It knows distances d_{AX}, d_{BX}, d_{CX} and positions of A, B, C which are respectively $(x_A, y_A), (x_B, y_B), (x_C, y_C)$. The following system is solved using a standard least-squares approach in order to give to X its estimated position :

$$\begin{cases} d_{AX}^2 = (x_X - x_A)^2 + (y_X - y_A)^2 \\ d_{BX}^2 = (x_X - x_B)^2 + (y_X - y_B)^2 \\ d_{CX}^2 = (x_X - x_C)^2 + (y_X - y_C)^2 \end{cases}$$

Among localization methods in wireless sensor networks, the most popular are the methods of Niculescu and Nath (APS) (Niculescu *et al.*, 2001), Savvides & al. (Savvides *et al.*, 2002) and Savarese & al. (Savarese *et al.*, 2002). 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 (Langendoen *et al.*, 2003), Langendoen and Reijers provide a detailed comparative survey for each step of these methods. The distance estimation techniques will be described in section 3.2. After the distance estimation step, there are two techniques in order to calculate node position : either multilateration, described above, used by (Niculescu *et al.*, 2001, Savarese *et al.*, 2002), or Min-Max technique, used by (Savvides *et al.*, 2002) : 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. The position of the node is set to the center of the intersection box. The refinement process consists in improving the node positions taking into account informations such as range to node neighbors and their positions. Note that (Niculescu *et al.*, 2001) does not use a refinement process. Section 6 compares our method with these three techniques.

Our technique presents two major properties : first, it detects some wrong informations (due to range errors for example). Second, a node knows if its estimated position is close to its real position. In this case it becomes an estimated anchor. Three rules are defined with this technique in order to locate nodes with exact positions. In our method, each node needs to estimate its distance with anchors. The next subsection describes the distance estimation techniques.

3.2. Distance Estimation Techniques

There are three distance estimation techniques : Sum-Dist(Savvides *et al.*, 2002), DV-Hop(Savarese *et al.*, 2002) and Euclidian(Niculescu *et al.*, 2001). In these three techniques, the anchors start by broadcasting their positions.

3.2.1. Sum-Dist

Description :

This method is the most simple solution for estimating distances to anchors. It adds ranges encountered at each hop during the network flood. Each anchor sends a message including its identity, coordinates and path length initialized to zero. When a node receives this message, it calculates the range from the sender, adds it to the path length and broadcasts the message. Thus, each node obtains a distance estimation and position of anchors. Of course, only the shortest distance will be conserved. For example, in figure 2 the estimated distance between S and D is $d_{SY} + d_{YD}$, and $d_{SD} \leq d_{SY} + d_{YD}$ due to triangular inequality. Let x_1, x_2, \dots, x_q, a be a path from node $x_1 \in V \setminus \Delta$ to anchor $a \in \Delta$. The estimated distance can be defined recursively as follow :

$$\hat{d}_{x_1 a} = d_{x_1 x_2} + \hat{d}_{x_2 a} \quad (1)$$

where \hat{d} represents the estimated distance returned by Sum-Dist.

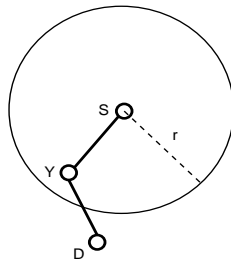


Figure 2. Sum-Dist

Advantages and Drawbacks :

Sum-dist is very simple and fast. Moreover, little computations is required. A drawback of Sum-dist is that range errors are accumulated when distance information is propagated over multiple hops.

3.2.2. DV-Hop

Description :

DV-hop consists of two flood waves. Similarly to Sum-Dist, after first wave, nodes obtained their positions and minimum hop counts to anchors. Second calibration wave allows to convert hop counts into distances. This conversion consists in multiplying the hop count with an average hop distance. As soon as an anchor A receives the position of another anchor B during the first wave, it computes the distance between them, and divides it by the number of hops in order to obtain the average hop distance between A and B . A calibrates its distance when it receives the position of anchor.

Nodes forward calibration messages (only from the first anchor that calibrates them in order to reduce the total number of messages in the network).

Figure 3 represents an example where A estimates the average of hop distance. There are three hops between A and B , and four between A and C . A computes euclidean distance between AB (75m) and AC (125m). The average of hop distance is equal to $\frac{125+75}{3+4} = 28.57m$. Node X estimates distances with B and C as following : $d_{XB} = 2 \times 28.57$ and $d_{XC} = 3 \times 28.57$.

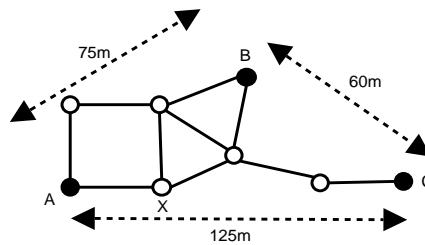


Figure 3. *DV-Hop*

Advantages and Drawbacks :

DV-hop is a stable and predictable method. Since it does not use range measurements, it is completely insensitive to this source of errors. However, DV-hop fails for highly irregular network topologies, the variance in actual hop distances is very large.

3.2.3. *Euclidian*

Description :

Euclidean is based on the local geometry of nodes around an anchor. When a node contains in its neighbourhood two nodes having estimated their distances with an anchor then it uses the *neighbor vote* method or *common neighbor* method in order to estimate its distance to the anchor.

Consider figure 4. Let A, B, C be nodes and D be an anchor. B and C are neighbors to A . B and C estimated their distances to D . A wants to estimate its distance to D . It knows distances ($d_{AB}, d_{AC}, d_{BC}, d_{BD}, d_{CD}$). So, all the sides and one of the diagonals of quadrilateral $ABCD$ are known. The second diagonal corresponding to d_{AD} . But there are two solutions d_1, d_2 . The *neighbor vote* method or *common neighbor* method allows to select the distance d_1 or d_2 . For more explanation cf. (Niculescu *et al.*, 2001).

Advantages and Drawbacks :

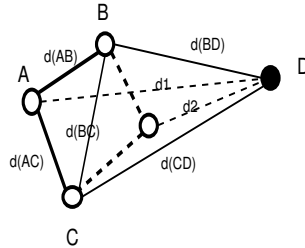


Figure 4. *Euclidean propagation method*

When it is possible, Euclidean provides an exact distance with anchor. But Euclidean is sensible to range errors and is efficient only in highly connected networks. Otherwise, Euclidean's performance rapidly degrades.

This paper uses Sum-Dist in order to estimate distances to anchors because Sum-Dist is not perturbed by the connectivity of the network and the triangular inequality is used by our method. Moreover, the next section explains how to overcome drawbacks of Sum-Dist.

4. Approximation Technique using Distances (AT-Dist)

This section proposes a novel distributed approximation technique, called AT-Dist.

4.1. Approximation Technique

When a node X 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 $A \in N_{\Delta}(X)$, X knows d_{AX} and deduces that it is on the circle of radius equals to d_{AX} and centered in A . If $A \notin N_{\Delta}(X)$ then X knows that it is not inside the circle of center A and radius r otherwise A and X would be neighbors. Moreover, X knows the estimated distance to A (\hat{d}_{AX}) deduced by Sum-Dist. By triangular inequality, $d_{AX} \leq \hat{d}_{AX}$. So, X is inside the circle of center A and radius \hat{d}_{AX} . X applies this technique for each received anchor position. Thus, the intersection of circles defines a zone Z_X containing X . X computes the center of gravity of this zone in order to deduce its estimated position.

To summarize, for each node $u \in V \setminus \Delta$, Z_u is obtained as follow :

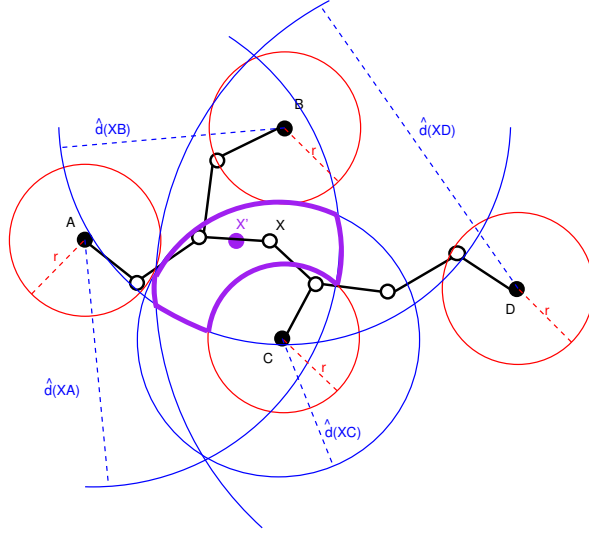


Figure 5. Estimated position for X

$$Z_{N_{\Delta}(u)} = \bigcap_{a \in N_{\Delta}(u)} \{ \forall i \in \mathcal{P}, (x_i, y_i) \mid (x_i - x_a)^2 + (y_i - y_a)^2 = d_{ua}^2 \} \quad (2)$$

$$Z_{\overline{N_{\Delta}(u)}} = \bigcap_{a \in \overline{N_{\Delta}(u)}} \{ \forall i \in \mathcal{P}, (x_i, y_i) \mid r^2 < (x_i - x_a)^2 + (y_i - y_a)^2 \leq \hat{d}_{ua}^2 \} \quad (3)$$

$$Z_u = Z_{N_{\Delta}(u)} \cap Z_{\overline{N_{\Delta}(u)}} \quad (4)$$

An example is illustrated in figure 5. X receives positions of anchors A, B, C and D . It estimates distances $\hat{d}_{AX}, \hat{d}_{BX}, \hat{d}_{CX}, \hat{d}_{DX}$ with Sum-Dist. Since all anchors are not neighbors of X then X is not inside circles centered respectively in A, B, C, D with a radius equals to r but it is inside circles with radius equal to $\hat{d}_{AX}, \hat{d}_{BX}, \hat{d}_{CX}, \hat{d}_{DX}$. The correlation of these informations defines a zone Z_X (represented in figure 5 by large lines). X computes the center of gravity of this zone and estimates its position in X' .

4.2. Implementation

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

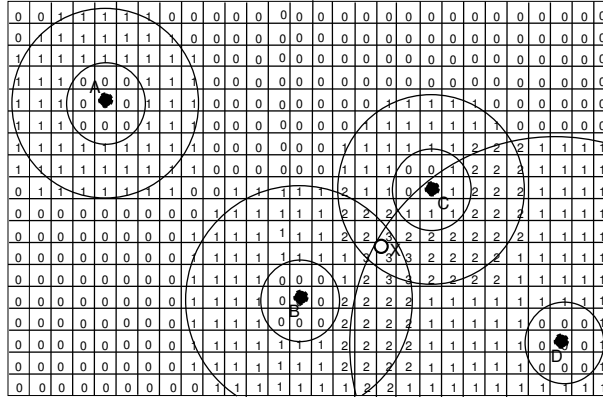


Figure 6. A network represented by a grid

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 6 represents an example of grid : 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.

4.3. AT-Dist properties

AT-Dist has two importants properties :

- 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 . In others words, if an anchor A is not neighbor of node X then radius of circles become $r - \epsilon$ and $\hat{d}_{AX} + \epsilon$. If X

and A are neighbors then X draws two circles with radius equal to $d_{AX} \pm \epsilon$. Thus, the node deduces that it is between these circles. Therefore, $Z_{N_{\Delta}(u)}$ and $Z_{\overline{N_{\Delta}(u)}}$ become :

$$Z_{N_{\Delta}(u)} = \bigcap_{a \in N_{\Delta}(u)} \{\forall i \in \mathcal{P}, (x_i, y_i) | (d_{ua} - \epsilon_a)^2 \leq (x_i - x_a)^2 + (y_i - y_a)^2 \leq (d_{ua} + \epsilon_a)^2\} \quad (5)$$

$$Z_{\overline{N_{\Delta}(u)}} = \bigcap_{a \in \overline{N_{\Delta}(u)}} \{\forall i \in \mathcal{P}, (x_i, y_i) | (r - \epsilon_a)^2 < (x_i - x_a)^2 + (y_i - y_a)^2 \leq (\hat{d}_{ua} + \epsilon_a)^2\} \quad (6)$$

with $\epsilon_a = 0$ for anchors equipped GPS.

Important note :

This paper assumes that sensors have none informations related to network environment. In some cases, it is possible to know this environment before the deployment of sensors. For example, if sensors are deployed in plat field with any obstacle, then measure errors would be weak. Conversely, if sensors are deployed in a field with many obstacles, then measure errors are high. A statistic analyze can be performed before deployment of sensors in order to obtain a bound of these measures (called ξ). Therefore, each sensor could take into account these bounds. Nodes can manage these errors in the same manner as ϵ . Thus, each circle will be replaced by two circles of radius equal to $radius \pm (\epsilon + \xi)$.

– Second, a node can detect if some informations are wrong. This case is illustrated in figure 6 when X receives the estimated distance by Sum-Dist from anchor A . X is not inside the two circles centered at A , but that is impossible (due to triangular inequality). So X deduces that this information is wrong. More phenomena can cause this situation : Sum-Dist is sensitive to range errors. Estimated distance can be wrong due to these range errors. The anchor may be under the control of an attacker in military context and announces a wrong position. AT-Dist can eliminate some informations. It should know some anchor positions in order to eliminate informations. The number of these positions (called *confidence*) is related to environment of the network and, if the network is strongly perturbed by range errors, then AT-Dist won't be efficient. Section 6 analyzes the value of *confidence* related to measure errors. As these errors are the main drawback of range-based localization schemes, it is interesting to control them.

Note : At the beginning, a sensor conserves the whole grid but as soon as it knows the zone containing itself, it can only conserve this zone and not the whole grid.

The next section presents three rules in order to obtain more anchors providing better estimated positions for each sensor node.

5. Rules to increase position accuracy

This section presents three rules in order to resolve ambiguity when a node can be located at two positions. For example, in figure 7, X does not know its position and B, C are anchors (**not estimated**) such as $B, C \in N_{\Delta}(X)$. X knows the positions of B and C and its distances d_{XB} and d_{XC} . So X can be located at node A or at node A' . When one of the rules can be applied, X will know if it is located in A (ie. (x_A, y_A)) or in A' (ie. $(x_{A'}, y_{A'})$). As described in the previous section, each anchor (estimated or not) broadcasts its position. When a node receives the position of an anchor, it estimates the distance with this anchor thanks to Sum-Dist and applies these rules allowing to resolve the ambiguity when a node can be located at two positions. Hereafter, A is assumed to be the real position of X .

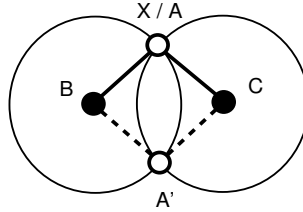


Figure 7. X can be in A or in A'

5.1. Rule 1

This first rule defines a simple born with estimated distance from a node to an anchor calculated by Sum-Dist. Here, the anchor does not belong to the neighborhood of the node looking for its position.

Let X be the node looking for its position and D be an anchor such as $D \notin N_{\Delta}(X)$ (as illustrated in figure 8). In a first time, D is not an estimated anchor. X receives D 's position (ie. (x_D, y_D)) and learns its estimated distance to D (ie. \hat{d}_{XD}). First, X assumes that is in A , so the following items have to be verified :

- $d_{AD} > r$ otherwise A and D will be neighbors (first condition) ;
- $d_{AD} \leq \hat{d}_{XD}$ due to triangular inequality (second condition).

If the two conditions are respected then X may be in A .

Now, X assumes that is in A' : if one of two conditions is not respected then X cannot be in A' and concludes that it is in A . However, if all conditions (for A and A') are respected then X cannot conclude. In conclusion X is in A if :

$$r < d_{AD} \leq \hat{d}_{XD} \wedge (d_{A'D} \leq r \oplus d_{A'D} > \hat{d}_{XD}) \quad (7)$$

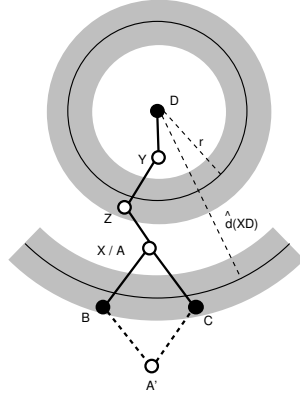


Figure 8. Rule 1

Now D is an estimated anchor. The real position of D is inside the circle centered in its estimated position with a radius equals to ϵ . Therefore, the real circle of D with a radius equals to r is between the two circles centered in estimated position of D with radius $r \pm \epsilon$ (represented in gray in figure 8). Idem for circle centered in D with radius equals to \hat{d}_{XD} . X is assured to be in A if :

$$r + \epsilon < d_{AD} \leq \hat{d}_{XD} - \epsilon \wedge (d_{A'D} \leq r - \epsilon \oplus d_{A'D} > \hat{d}_{XD} + \epsilon) \quad (8)$$

That means A must be between the two gray zones and A' outside.

5.2. Rule 2

Here, the anchor does not belong to the neighborhood of the node looking for its position.

Let X be the node looking for its position and D be an anchor such as $D \notin N_{\Delta}(X)$ (as illustrated in figure 9). In the first time D is not estimated. When X receives D 's position (ie. (x_D, y_D)), it checks : if $d_{A'D} \leq r$ then A' and D would be neighbors. Therefore, X concludes that it is not in A' and then X deduces that it is in A :

$$d_{AD} > r \wedge d_{A'D} \leq r \quad (9)$$

Nevertheless, if $d_{AD} > r$ and $d_{A'D} > r$ then X cannot conclude.

Now D is an estimated anchor. As a previous case, the real circle of D with radius equals to r is inside the gray zone in figure 9. X is assured to be in A if : $r + \epsilon < d_{AD}$ and $d_{A'D} \leq r - \epsilon$. In others words, A must be outside the circle centered in D with

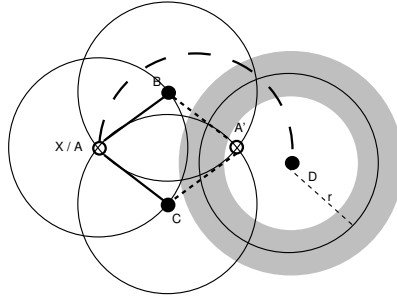


Figure 9. Rule 2

radius equals to $r + \epsilon$ and A' must be inside the circle centered in D with radius equals to $r - \epsilon$:

$$d_{AD} > r + \epsilon \wedge d_{A'D} \leq r - \epsilon \quad (10)$$

5.3. Rule 3

This rule is applied when a node has at least three anchors in its neighborhood.

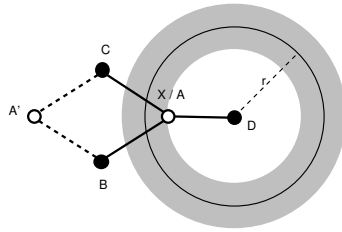


Figure 10. Rule 3

Let X be the node looking for its position and D be an anchor such as $D \in N_{\Delta}(X)$ (as illustrated in figure 10). D is not an estimated anchor. When X receives D 's position (ie. (x_D, y_D)), it checks : if $d_{A'D} > r$ then A' and D would not be neighbors. Therefore, X concludes that it is not in A' , and then X deduces that it is in A :

$$d_{AD} \leq r \wedge d_{A'D} > r \quad (11)$$

Nevertheless, if $d_{AD} \leq r$ and $d_{A'D} \leq r$ then X cannot conclude.

Now assume that D is an estimated anchor. X is assured to be in A if : $d_{AD} \leq r - \epsilon$ and $d_{A'D} > r + \epsilon$. In others words, A must be inside the circle centered in D with radius equals to $r - \epsilon$ and A' must be outside the circle centered in D with radius equals to $r + \epsilon$:

$$d_{AD} \leq r - \epsilon \wedge d_{A'D} > r + \epsilon \quad (12)$$

The multilateration is not recommended, even if a node has at least three anchors in its neighborhood, because it is very sensible to errors. When multilateration is used, the position of node corresponds to the point of intersection of three circles that centers are the neighbors. If one of distances with neighbors is wrong or if one of neighbors is not exactly located, then the node cannot compute its position. The second case is described in figure 11 : the localization of node D is lightly wrong and D is located in D' . But the circle centered in D' does not intersect in the same as the two others circles. It is easy to see that with our rule this problem is resolved. So, when range errors are equal to 0% then the multilateration can be used, but as soon as range errors are introduced, then this rule is better.

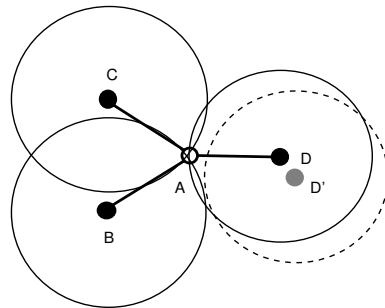


Figure 11. *Multilateration cannot be applied*

The next subsection shows errors of rules due to range errors and describes the voting process allowing to avoid efficiency a lot of errors.

5.4. voting process

In ideal case (ie. without range errors), when a node receives an anchor position and when one of rules can be applied then the node resolves ambiguity and obtains its position. In fact, if a rule can be applied with only one anchor (node D in figures) then the node is located. But, what is the consequence if an information (position, estimated distance,...) related to this anchor is wrong due to measure errors or an attacker who has the control of a node ?

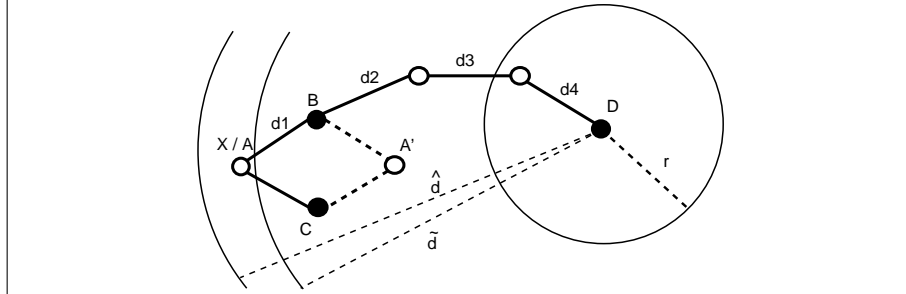


Figure 12. Error of Sum-Dist due to range errors

Figure 12 represents estimated distance error related to range errors. Let $\hat{d} = d_1 + d_2 + d_3 + d_4$ be the result of Sum-Dist between nodes D and X without range errors and $\tilde{d} = \tilde{d}_1 + \tilde{d}_2 + \tilde{d}_3 + \tilde{d}_4$ the result of Sum-Dist with range errors ($\tilde{d}_1, \tilde{d}_2, \tilde{d}_3, \tilde{d}_4$ being distances with range errors) such that $\tilde{d} < \hat{d}$. Figure 12 shows that without range errors, X cannot resolve ambiguity with anchor D because none rules can be applied. If this example is considered with range errors, estimated distance is \tilde{d} and then X deduces that it is located in A' by rule 1 that is false. Although AT-Dist eliminates some wrong informations, X cannot do confidence to only one anchor. The voting process allows to take into account some anchors in order to deduce the localization of a node. When a node can be located at two positions p_1 and p_2 , it checks rules with all anchors that it knows. When a rule can be applied with an anchor allowing to determinate the position of node in p_1 (resp. p_2) then the node increments a counter cp_1 (resp. cp_2). Now, if $cp_1 - cp_2 \geq \text{confidence}$ (resp. $cp_2 - cp_1 \geq \text{confidence}$) then the node is located in p_1 (resp. p_2). Without range errors (in others words, the percentage of range errors is equal to 0%) then *confidence* is equal to 1. In fact, the value of *confidence* is related to the environment of network and it is defined experimentally. Note that the threshold *confidence* is the same that used in order to eliminate some wrong informations in sub-section 4.3.

Important note :

As indicated in sub-section 4.3 this paper assumes that sensors have none informations related to network environment, especially informations about measure errors. If a bound of these errors (ξ) is known, each rule would take into account ξ . For example, let x_1, x_2, \dots, x_q be a path of length equals to q . The estimated distance from x_1 to x_q is bounded :

$$\hat{d}_{x_1 x_q} - q \times \xi \leq d_{x_1 x_q} \leq \hat{d}_{x_1 x_q} + q \times \xi \quad (13)$$

In this case, the voting process would be useless to manage introductions or accumulations of range errors, but stay usefull to manage others error sources. In a futur work, it would be interesting to find an optimized interval for the estimated distance.

6. Simulations

6.1. Simulation environment

Our solutions are performed with the simulator created by Langendoen and Reijers in (Langendoen *et al.*, 2003) based on OMNET++, a discrete event simulator (Varga, 2001). This simulator allows to use the three distance estimation techniques, but in our case, nodes are configured in order to use only Sum-Dist. Concurrent transmissions are allowed if the transmission areas (circles) do not overlap. When a node wants to broadcast a message while another message in its area is in progress, it must wait until that transmission are completed. The simulator uses CSMA policy. Message corruption are not considered, so all messages sent during our simulations are delivered.

A random network topology is generated according to the number of nodes and the number of anchors. The nodes are randomly positioned, with an uniform distribution, within a square area. The anchors are selected randomly. The transmission range used between connected nodes is blurred by drawing a random value from a normal distribution having a parameterized standard deviation and having the right range as the average. The simulator selected this error model based on the work of Whitehouse and Culler (Whitehouse *et al.*, 2002).

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 .

The connectivity (average number of neighbors) is controlled by specifying the radio range. By default, scenarios use networks with 150 nodes. These nodes are distributed in a square 100×100 . Here the radio range is set to 14. Thus, the density of sensors is equal to 9,24. The percentage of anchors, noted α , varies from 0% to 20% representing density of anchors from 0.12 to 1.23.

Different scenarios are used while changing the percentage of measure errors δ respectively equals to 0%, 5%, 10% and 15%. Moreover, a node becomes an estimated anchor if its maximum position error is lower than 15% (ie. $\epsilon \leq r \times 0.15$).

AT-Dist is proposed in order to resolve the localization problem in wireless sensor networks. Therefore, simulations focus on two criterious allowing to evaluate performance of AT-Dist for this problem : first, the average error rate (i.e., the sum of position errors divided by the number of nodes minus the number of anchor equipped with GPS). Second, the percentage of nodes located with very good positions without considering anchors located by GPS. These criterious are analyzed related to percentage of anchors, percentage of measure errors and density of nodes.

In our analysis, each scenario is performed 100 times. Thus, a relatively small variance is obtained. Graphs represent means and confidence intervals for each analyzed parameters. Here there is 95% of chance that the real value belong to this interval.

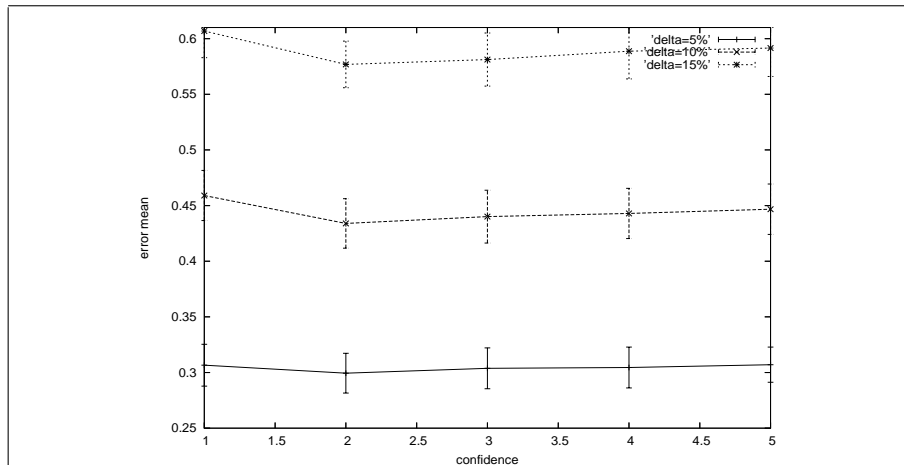


Figure 13. Variation of confidence with $\alpha=10\%$

6.2. Results

In a first time, an analyze is given related to the threshold noted *confidence* used, for example, in order to eliminate wrong informations due to measure errors. This threshold allows a sensor to consider a set of anchors in order to deduce its position in voting process and eliminate some wrong informations in AT-Dist. Figure 13 represents error mean in a network containing percentage of anchors equals to 10% and range errors respectively equals to 5%, 10% and 15%, related to value of the threshold *confidence*.

When the value of *confidence* is equal to 2, the obtained error mean is the best. In fact, when the value of *confidence* is higher than 2, the voting process is very strict and nodes cannot deduce their positions. Conversely, when the value of *confidence* is lower than 2, 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 δ 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 2.

Now, simulations focus on efficiency of our method and consider a scenario with 150 nodes, with α varies from 0% to 20% representing density of anchors from 0.12 to 1.23 and δ equals to 0 (the ideal case). Corresponding graphs are represented in figures 14 and 15. The first graph represents the percentage of nodes located without errors. The anchors located by GPS are not taken into account. In others words, the percentage of new exactly located nodes is only considered. With 8% of anchors our method gives 86% of nodes exactly positioned.

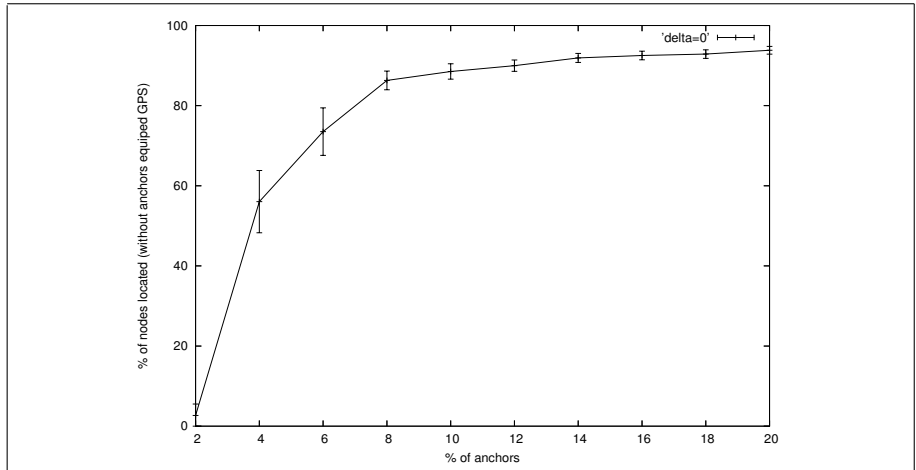


Figure 14. Performance of AT-Dist with $\delta=0\%$

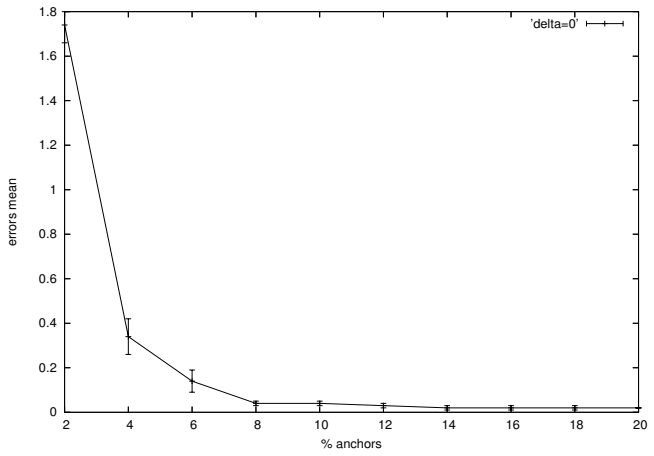


Figure 15. Average error rate for $\delta=0\%$

The second graph represents the average error rate without taking into account anchors located by GPS. With 8% of anchors the average error rate is equal to 0.04. Therefore, with $\alpha = 8\%$ all nodes obtain a position with high accuracy. It is interesting to note that after $\alpha = 10\%$, the percentage of node exactly located and the average error rate are lightly improved.

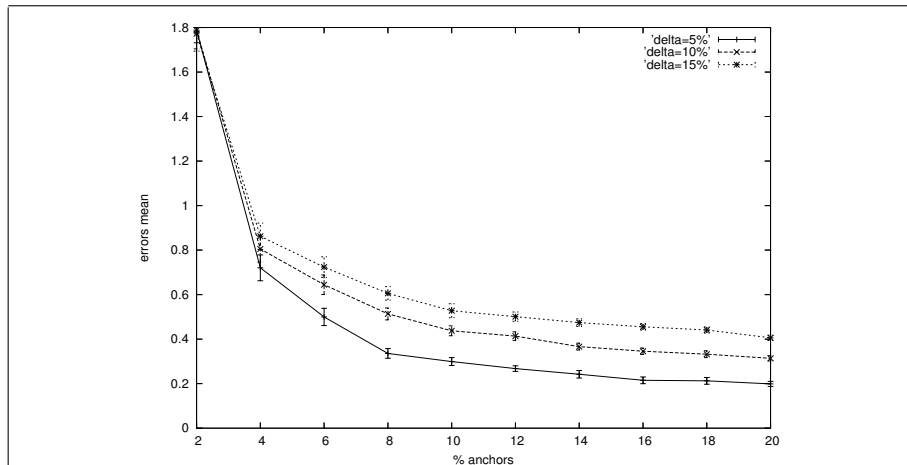


Figure 16. Average error rate for $\delta = \{5,10,15\}\%$ related to α

Figure 16 represents, when range errors are introduced, the behavior of average error rate of our method related to percentage of anchors. These curves indicate the accuracy of localizations when δ is equal to 5%, 10% and 15%. Without surprise, performances of AT-Dist decrease when range errors increase. But, our method keeps a good estimation of positions. Note also that after 10% of anchors the average error rate decreases slowly.

Figure 17 shows impacts of the range error on position mean error. There are three curves representing respectively the position mean error when the percentage of anchors equipped GPS is equal to 5%, 10% and 20% related to range errors. On the horizontal axis the percentage of range error is varied from 0% to 40%. This graph shows the performances of AT-Dist in managing introductions or accumulations of range errors. Related to values of range error, the average error rate stays reasonable.

Figure 18 shows the impact of density of nodes on the behavior of average error rate. 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.

Figure 19 represents the percentage of located nodes with a position error lower than 20% using our method and methods described in (Niculescu *et al.*, 2001, Savarese *et al.*, 2002, Savvides *et al.*, 2002). These methods are respectively called *APS*, *HTRefine* and *SumDist+MinMax*. Here, δ is set to 5%. The efficiency of our method is clearly shown. For example, with $\alpha = 10\%$, our methods locates 55% of nodes with

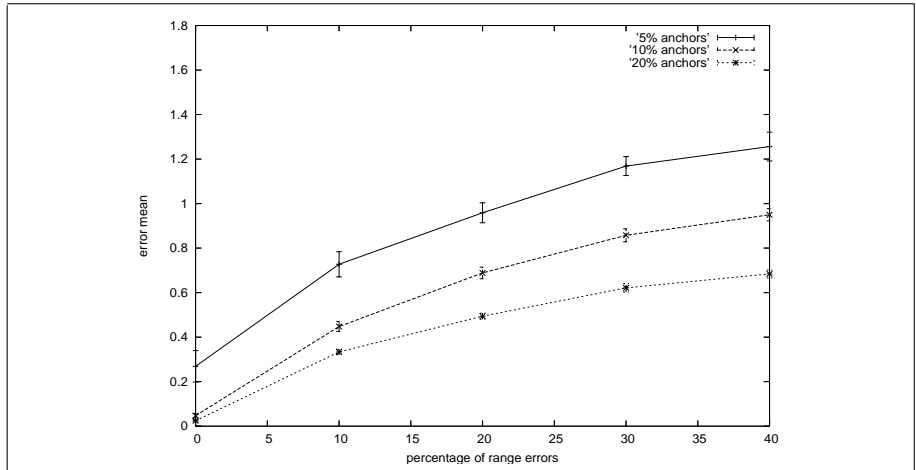


Figure 17. Average error rate with range errors from 0% to 40%

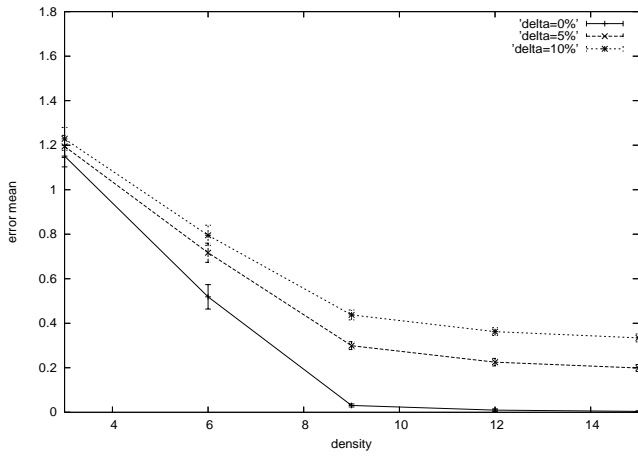


Figure 18. Average error rate related to density of nodes with $\alpha = 10\%$

an error lower than 20% and the others methods locate less than 17% of nodes with an error lower than 20%.

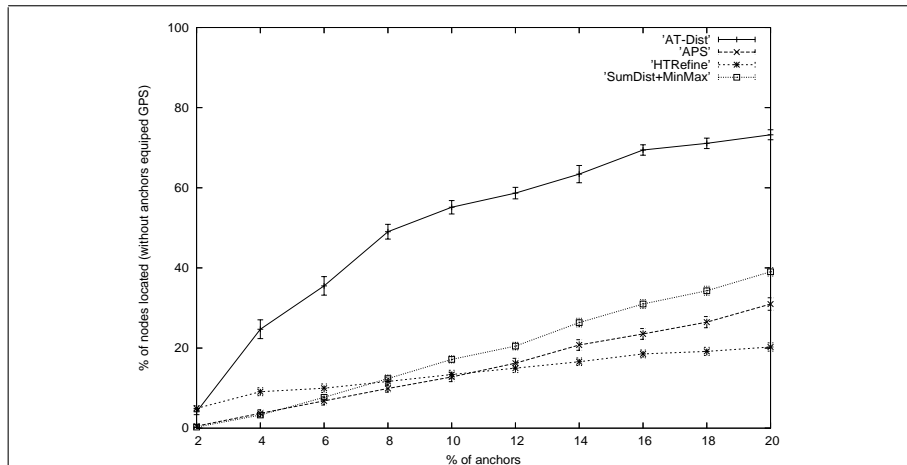


Figure 19. Comparison of percentages of nodes located with an error lower than 20% with $\alpha = 5\%$

7. Conclusions

This paper considers a new method in order to locate sensors with high accuracy. It proposes a original distributed approximation technique AT-Dist in order to estimate the position of nodes. Each node restricts the zone where it can be localized. AT-Dist presents two 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. Three rules are introduced in order to take into account these estimated anchors. Thus, simulations show the efficiency of our method in comparison to the methods in (Niculescu *et al.*, 2001, Savvides *et al.*, 2002, Savarese *et al.*, 2002). Our simulations cannot take into account all real conditions and it would be interesting to check the efficiency of our method in a real environment. Moreover, this paper focuses on performances to locate sensors with high accuracy but does not take into account the energy consumption or the position convergence times. The optimization of these two criterious represents two others major problems in wireless sensor networks. They mainly depend on the broadcast strategy of messages. Some techniques have been proposed for these problems. Futur works will consist in analyze these criterious in AT-Dist either by using these methods or by a novel method adapted to AT-Dist. Finally, this paper assumes that sensors have none informations related to network environment, especially informations about error measures. It proposes some ways to improve AT-Dist when a bound can be calculated for measure errors. But, an in-depth analyze should to be achieved.

8. Bibliographie

Bahl P., Padmanabhan V. N., "RADAR : An In-Building RF-based User Location and Tracking System", *INFOCOM*, 2000.

Bulusu N., Heidemann J., Estrin D., "GPS-less Low Cost Outdoor Localization For Very Small Devices", *IEEE Personal Communications Magazine*, October, 2000.

Bulusu N., Heidemann J., Estrin D., "Adaptative Beacon Placement", *The Twenty First International Conference on Distributed Computing Systems, Phoenix*, 2001.

Chan H., Luk M., Perrig A., "Using Clustering Information for Sensor Network Localization", *In The International Conference on Distributed Computing in Sensor Systems (DCOSS 2005)*, 2005.

Datta S., Klinowski C., Rudafshani M., Khaleque S., "Distributed localization in static and mobile sensor networks", *2nd IEEE International conference on wireless and mobile computing, networking and communications (WiMob)*, 2006.

et al. B. P., "Global Positioning System : Theory and Application", *Progress in Astronomics and Aeronotics*, 1996.

He T., Huang C., Blum B. M., Stankovic J. A., Abdelzaher T., "Range-Free Localization and its Impact on Large Scale Sensor Networks", *IEEE Personal Communications Magazine*, 2005.

Hightower J., Borriello G., "Location Systems for Ubiquitous Computing", *IEEE Computer* 34(8) pp. 57-66, August, 2001.

Ko Y.-B., Vaidya N. H., "Location-Aided Routing (LAR) in mobile ad hoc networks", *in Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom)*, pp. 66-75, 1998.

Langendoen K., Reijers N., "Distributed localization in wireless networks : A quantitative comparison", *Computer Networks*, no 43, pp. 500-518, 2003.

Li J., Jannotti J., Couto D. S. J. D., Karger D. R., Morris R., "A Scalable Location Service for Geographic Ad Hoc Routing", *Proceedings of the 6th ACM International Conference on Mobile Computing and Networking (MobiCom '00)*, pp. 120-130, 2000.

Niculescu D., Nath B., "Ad Hoc Positioning System (APS)", *in Proceedings of GLOBECOM, San Antonio*, November, 2001.

Niculescu D., Nath B., "Ad Hoc Positioning System (APS) Using AoA", *IEEE INFOCOM 2003, San Francisco, CA*, 2003a.

Niculescu D., Nath B., "DV Based Positioning in Ad Hoc Networks", *Journal of Telecommunication Systems*, 2003b.

Niculescu D., Nath B., "Error Characteristics of Ad Hoc Positioning Systems (APS)", *Mobicom*, 2004.

Priyantha N. B., Balakrishnan H., Demaine E. D., Teller S., "Mobile-Assisted Localization in Wireless Sensor Networks", *IEEE Infocom*, March, 2005.

Priyantha N. B., Chakraborty A., Balakrishnan H., "The Cricket Location-Support System", *International Conference on Mobile Computing and Networking, Boston*, 2000.

Saad C., Benslimane A., König J.-C., "MuR : A Distributed Preliminary Method For Location Techniques in Sensor Networks", *2nd IEEE International conference on wireless and mobile computing, networking and communications (WiMob)*, June, 2006.

Saad C., Benslimane A., König J.-C., "A Distributed Method to Localization for Mobile Sensor Networks", *IEEE Wireless Communications and Networking Conference*, March 11-15, Hong Kong, China, 2007.

Savarese C., Rabaey J., "Robust Positioning Algorithms for Distributed Ad-Hoc Sensors Networks", in *USENIX technical annual conference, (Monterey, CA)*, pp. 317-328, June, 2002.

Savvides A., Han C.-C., Srivastava M. B., "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors", *ACM SIGMOBILE*, July, 2001.

Savvides A., Park H., Srivastava M., "The bits and flops of the N-hop multilateration primitive for node localization problems", *First ACM International WSNA, Atlanta*, 2002.

Varga A., "The OMNET++ discrete event simulation system", *European Simulation Multi-conference (ESM'2001), Prague, Czech Republic*, 2001.

Čapkun S., Hamdi M., Hubaux J.-P., "GPS-free positioning in mobile Ad-Hoc networks", in *Proceedings of the 34th Hawaii International Conference on System Sciences*, 2001.

Venkatraman S., J. Caffery J., "Multipath-aided Location Estimation Using Angles-of-Arrival", in *Proceedings of SPIE*, 2003.

Venkatraman S., J. Caffery J., You H. R., "Location Using LOS Range Estimation in NLOS Environments", in *Proc. IEEE Vehicular Technology Conference*, vol. 2, pp. 856-860, Spring 2002, 2002.

Whitehouse K., Culler D., "Calibration as parameter estimation in sensor networks", *First ACM International Workshop on Wireless Sensor Networks and Application (WSNA), Atlanta, GA* pp. 59-67, 2002.

SERVICE ÉDITORIAL – STUDIA UNIVERSALIS
41 rue Gay Lussac, 75005 Paris
Tél : 01-44-10-84/83
mel : contact@complexica.net
Serveur web : <http://studia.complexica.net>