WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. 2014; **14**:1627–1646
Published online 30 October 2012 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/wcm.2303

RESEARCH ARTICLE

Cooperative localization techniques for wireless sensor networks: free, signal and angle based techniques

Abderrahim Benslimane[1*], Clement Saad[1], Jean-Claude Konig[2] and Mohammed Boulmalf[3]

[1] LIA/CERI, University of Avignon, 339 chemin des Meinajaries, BP 1228-84911 Avignon, France
[2] LIRMM, University of Montpellier 2, 161 Rue Ada, F-34392 Montpellier, France
[3] International University of Rabat, Morocco

ABSTRACT

This paper addresses the problem of localization in sensor networks where, initially, a certain number of sensors are aware of their positions (either by using GPS or by being hand-placed) and are referred to as anchors. Our goal is to localize all sensors with high accuracy, while using a limited number of anchors. Sensors can be equipped with different technologies for signal and angle measurements. These measures can be altered by some errors because of the network environment that induces position inaccuracies. In this paper, we propose a family (AT-Family) of three new distributed localization techniques in wireless sensor networks: free-measurement (AT-Free) where sensors have no capability of measure, signal-measurement (AT-Dist) where sensors can calculate distances, and angle-measurement (AT-Angle) where sensors can calculate angles. These methods determine the position of each sensor while indicating the accuracy of its position. They have two important properties: first, a sensor node can deduce if its estimated position is close to its real position and contribute to the positioning of others nodes; second, a sensor can eliminate wrong information received about its position. This last property allows to manage measure errors that are the main drawback of measure-based methods such as AT-Dist and AT-Angle techniques. By varying the density and the error rate, simulations show that the three proposed techniques achieve good performances in term of high accuracy of localized nodes and less energy consuming while assuming presence of measure errors and considering low number of anchors. Copyright © 2012 John Wiley & Sons, Ltd.

KEYWORDS
wireless Sensor networks; localization; distributed algorithms

***Correspondence**
Abderrahim Benslimane, LIA/CERI, University of Avignon, 339 chemin des Meinajaries, BP 1228-84911 Avignon, France.
E-mail: abderrahim.benslimane@univ-avignon.fr

1. INTRODUCTION

Recent advancements in microelectromechanical systems and wireless communications have enabled the development of a new kind of networks: *wireless sensor networks* (WSNs). These networks have been proposed in many fields such as target tracking, intrusion detection, medical applications, climate control, and disaster management. Some issues and solutions are presented in [1].

Wireless sensor network nodes are small battery-powered platforms usually equipped with a microcontroller, a radio module, and a sensor device. The sensing module allows gathering events "data" about their environment, the computing module process these gathered data, and finally, sensors can exchange their information via the communication module. To preserve the battery, only a few operations (computations and especially communications) have to be performed.

In some applications, the knowledge about sensor localization is required, but all sensors cannot be equipped by localization module (e.g., GPS [2]) because of cost and energy constraints. A common example of WSNs is the aircraft deployment of sensors in a given area. In this network, only a few nodes know their positions, thanks to a localization module. 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 of cost and energy constraints. The network has to be self-organizing; that is, it should not depend on global infrastructure. Proposed solutions must take all sensor characteristics into account.

Localization schemes can be classified into two categories: range-free localization schemes and range-based localization schemes. The first category contains all methods that use only anchor positions to locate all sensors. The second category contains all methods that use techniques that allow a node to calculate either distances or angles with its neighbors. Measures obtained by these techniques can be corrupted by some errors because of the network environment. These errors are called *measure errors* or *range errors*. They represent the most important drawback for measure-based methods (distances or angles).

Among these methods, the most popular techniques are described in [3–6]. HTRefine [3] belongs to range-free localization schemes; in APS [5] and SumDist+MinMax [4], nodes can calculate distances with their neighbors; and finally, in APS_{AoA} [6], which is an adaptation of APS, nodes can calculate angles between neighbors. 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 to improve the estimation accuracy. These methods are described in Section 3.

As showed in [7], sensor localization algorithms must be designed to accommodate different application requirements in terms of costs, energy consumption, and localization accuracy. With these requirements, we consider that the localization methods focus on one of two important criteria: either they locate some sensors with high accuracy (i.e., exact position) whereas others sensors are not located or they assign an estimated position to all sensors, the estimations being coarse. In this paper, we consider that these two criteria are linked: the localization with the exact position of some sensors can improve estimation of other nodes, and conversely, good estimations of sensor positions can allow other sensors to obtain an exact position. Hence, to resolve the localization problem in WSN, we propose a new family (AT-Family) of three distributed approximation techniques AT-Free, AT-Dist, and AT-Angle that represent methods where nodes have no measurement capability, distance measurement capability between neighbors, and angle measurement capability with neighbors. In some case, techniques AT-Dist and AT-Angle allow to calculate exact positions. Otherwise, they use approximation techniques to assign estimated positions. The approximation techniques are based on the same general principle: each node defines a restricted zone containing itself, according to the anchor positions and distances from it to the anchors. The gravity center of the restricted zone constitutes an estimated position of the node. Each AT-Family method verifies *two important properties*: first, a node can detect when its estimated position is close to its real position. In this case, this node becomes an *estimated anchor* and will be used by other nodes to help them to obtain their positions. Second, some wrong information (e.g., due to measure errors, which is the main drawback of measure-based methods) can be eliminated according to defined sensor zones. These properties allow to obtain good simulation results even if measure errors are introduced.

Contrary to the existing solutions that derive their schemes under the strong assumption of noisy-free range measurements, in this work we do not make such assumption. So, we introduce errors on the estimated position. In [8], authors propose localization and orientation scheme that is derived under the assumption of noisy angle measurements. But the problem is how a node can well estimate this error. As in [6], positioning error is determined relative to the maximum communication range of a node. An error of 1.0 means that the position resulted from the positioning algorithm is one (maximum size) radio hop away from its true position. In our methods, a node that estimates its position can determine the position error bound without any additional hardware component. If the error on this position is bounded, then the sensor propagates this information to its neighbors.

The remainder of the paper is organized as follows: Section 2 introduces the model notion for this problem. Section 3 discusses previous works on sensor localization. Section 4 explains the three approximation techniques AT-Free, AT-Dist, and AT-Angle and their properties. Section 7 discusses simulation results where our methods are compared with the four most popular methods [3–6]. Finally, in section 8, we give the conclusion.

2. MODEL

Many localization algorithms have been proposed for static wireless ad hoc or sensor networks even though in some applications nodes may be mobile. In this paper, we are interested in static sensor networks. Readers that are interested in mobile sensors can refer to our previous work [9]. There are also several works on localization in mobile sensor networks [10–12]. We assume that all sensors have identical transmission range r. As it will be explained in Section 6, it is easy to adapt our methods to sensors having different transmission ranges. A WSN 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, which means distance between u and v is smaller than r.

$$E = \{(u,v) \in V^2 | u \neq v \ et \ d_{uv} \leq r\}$$

In all figures, white nodes represent sensors that do not know their positions, and black nodes represent anchors. The set of neighbors for a node $u \in V$ is denoted $N(u)$.

A priori, anchors have knowledge of their own position with respect to some global coordinate system. Positions can be obtained by a localization system such as GPS [2]. Hence, an anchor can locate itself with position error less than 1 m, but in military applications, this accuracy is measured in millimeters. The set of anchors is denoted Δ. The set of neighbor anchors for a node u is denoted $N_\Delta(u)$

 Wirel. Commun. Mob. Comput. 2014; **14**:1627–1646 © 2012 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm

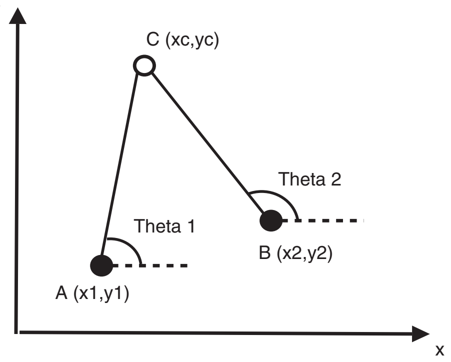

Figure 1. Angle of arrival.

(i.e., $N_\Delta(u) = N(u) \cap \Delta$), and the set of non-neighbor anchors is denoted $\overline{N_\Delta}(u)$ (i.e., $\overline{N_\Delta}(u) = \Delta \setminus N_\Delta(u)$). Note that all nodes (anchors or other nodes) have the same capabilities (energy, processing, communication, etc). The position of node u is denoted (x_u, y_u). $\mathscr{P}$ is the set of all possible positions in a network.

In AT-Family, AT-Dist 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 denoted d_{uv}. Two neighbor nodes u and v know d_{uv} (via ToA, etc). The estimated distance is denoted $\hat{d}_{uv}$. The following section explains how to obtain these estimated distances.

AT-Angle method assumes that each node can only compute its angle (and does not have a capability to compute distances) to its neighbors when it receives a signal. All angles are computed according to one reference axis (north, south, east, west, determined with a compass).

The angle formed by two nodes u and v is denoted $\angle u, v$. When a node receives positions of two anchor neighbors, it deduces its position. For example, in Figure 1, C does not know its position. A and B send to C their positions (x_1, y_1), (x_2, y_2) and angles $\angle A, C = \theta_1$, $\angle B, C = \theta_2$. (x_c, y_c) can be calculated with this system: $(y_c - y_i)/(x_c - x_i) = \tan(\theta_i)$ for $i \in \{1, 2\}$.

Node C becomes an anchor and broadcasts its position.

2.1. Localization problem notation

To homogenize the formulation of all localization problems in WSNs, we propose the following notation:

$$< x, y, z >, \text{with } x, y \in \{S, M\} \text{ and } z \in \{\emptyset, \text{dist}, \text{angle}\}$$

The first (respectively second) field defines if nodes (respectively anchors) are mobile or static (M for mobile, S for static). The last field determines the sensor technology. If a sensor can calculate angles (respectively distances), the value of the last field is assigned to angle (respectively distance). Otherwise, the value is assigned to $\emptyset$. This paper focuses on configurations $< S, S, z >$ with $z \in \{\emptyset, \text{dist}, \text{angle}\}$. In [9], we studied some mobility cases because mobility in mobile sensor networks is still challenging.

3. RELATED WORKS

A large number of existing techniques attempt to solve localization problems, and detailed surveys are provided in [13,14]. These solutions can be organized in three categories: GPS-free methods [15], infrastructure-based systems [16,17], and robot-based systems [11,18].

We distinguish two categories among these methods: first, range-free localization schemes that deduce estimated positions for all nodes in the network with only anchor coordinates and second, range-based localization that uses techniques such as ToA and AoA, allowing calculation of either distances or angles between two neighbor sensors.

3.1. Range-free localization schemes

There are methods that deduce estimated positions for all nodes in the network with only anchor coordinates. The techniques described in [19–21] are examples of these methods. For example, if we consider the method described in [19]: let N be the number of anchors in a sensor network. Each node computes all triangles according to all anchor positions. It obtains a set of $\binom{N}{3}$ triangles. For each triangle, the node checks if it is inside or outside of it. The intersection of triangles defines a zone containing the node. The center of gravity of this zone represents the estimated position. By this method, a sensor cannot become an anchor. Otherwise, the number of computations would be very expensive. HTRefine in [3] is the most popular method using range-free localization schemes.

3.2. Range-based localization schemes

The most popular methods to compute the range with two neighbor nodes are RSSI [16], ToA [2], TDoA [22], and AoA [6]:

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) directly translates the propagation time into distance if the signal propagation speed is known.

AoA (Angle of arrival) estimates the angle at which sensors sense the direction from which a signal is received. Simple geometric relationships are used to calculate node positions. AoA sensing requires either antenna array or several ultrasound receivers. Each angle is measured according to an orientation that represents one of the axes – north, south, east, or west.

Of course, the accuracy of these measures depends on the network environment. These errors are called *measure errors* or *range errors*. In [23,24], the authors respectively analyze the impact of range and angle errors.

The classical method for computing the node 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 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 the 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 to give the estimated position of X:

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

The most popular localization methods in WSNs are HTRefine [3], which belongs to range-free localization schemes; APS in [5] and SumDist+MinMax in [4], where nodes are equipped with RSSI, ToA, or TDoA; and finally, APS_{AoA} in [6], which is an adaptation of APS where nodes are equipped with AoA technology. These methods use the same three-step execution schemes: first, anchors broadcast their position; second, each node estimates distances with anchors and derives an estimation of its position from its anchor distances; and finally, a refinement process is performed to improve the estimation accuracy. In [25], Langendoen and Reijers provide a detailed comparative survey for each step of these methods. After estimating distances, two techniques can be used to calculate the node position: either multilateration, described previously, used by APS, HTRefine, and APS_{AoA}, or Min–Max technique, used by SumDist+MinMax: the main idea is to construct, for each node, a bounding box according to anchor positions and estimated distances, and then to determine the intersection of these boxes. The position of the node is set at the center of the intersection box. The refinement process consists of improving the node positions by taking information such as transmission range to neighbors and their positions into account. Note that APS and APS_{AoA} do not use a refinement process. Section 7 compares our methods with these three techniques.

3.3. Angle of arrival estimation techniques

APS_{AoA} [6], *Probabilistic* [8], and *HA-A2L* [26] are angle-based methods. APS_{AoA} is an angle-based version of method APS [5].

In APS_{AoA}, each anchor floods its position. When a sensor X receives a position of an anchor A and regarding to angle with the announcer, it deduces its angle with A. When X knows more than three anchor positions and associated angles, it estimates its position, thanks to the triangulation.

In *HA-A2L*, authors propose a localization method called high accuracy localization based on angle to landmark; it consists of (i) a new protocol that allows nodes to exchange information pertinent to the localization process and (ii) a localization algorithm that uses estimation of distances and incoming angles to locate nodes in sensors networks.

In *probabilistic* method, positions are estimated through probability distribution functions. It defines a pseudo-anchor as a sensor with estimated position probability density function. Anchors and pseudo-anchors broadcast their positions. A sensor that receives these data measures the angle with the announcer and updates its position distribution and probability density function. It becomes a pseudo-anchor and broadcasts its updated probability density function. Authors conclude that their technique is better than APS_{AoA}. However, they use an angle error modeling as being a Gaussian distribution to characterize AoA measurements. Thus, when a sensor sends a signal, it deduces the direction of the line of sight plus or minus an angle error bound. But the efficiency of this model depends on the network's environment. When there are obstructions between transmissions of any two nodes, this modeling cannot be used. In [27], authors use AoA technique combined with antenna arrays. To refine a well-known least-squares algorithm, a new heuristic weighting function is proposed. It enables combining the information from all the anchors more effectively and reducing the location errors. However, no study is given on neither the energy consumption nor the density of nodes.

As APS_{AoA}, AT-Angle does not use any error modeling. The hypothesis about network's environment and sensor knowledge are the same. Therefore, AT-Angle performances are better compared with the ones of APS_{AoA} that have the same assumptions.

3.4. Distance estimation techniques

There are three distance estimation techniques: Sum-Dist [4], DV-Hop [3], and Euclidian [5]. In these three techniques, the anchors start by broadcasting their positions.

3.4.1. Sum-Dist.

Description: This method is the simplest solution for estimating distances to anchors. It adds ranges encountered at each hop during network flooding. Each anchor sends a message, including its identity, coordinates, and path

Wirel. Commun. Mob. Comput. 2014; **14**:1627–1646 © 2012 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm

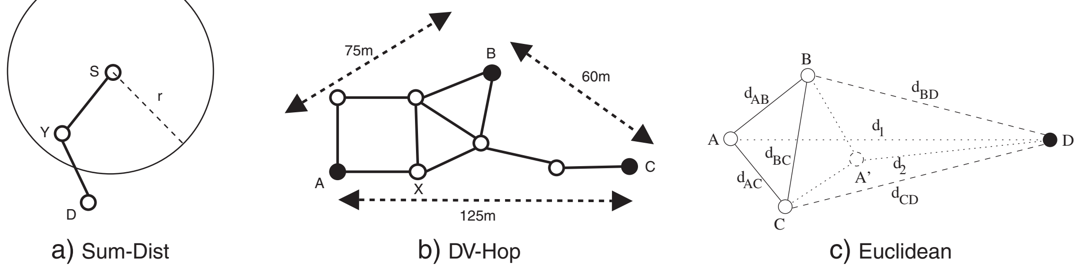

Figure 2. Distance estimation techniques: (a) Sum-Dist, (b) DV-Hop, and (c) Euclidean.

length initialized to zero. When a node receives this message, it calculates the distance to the sender, adds it to the path length, and broadcasts the message. Thus, each node obtains a distance estimation and the position of anchors. Of course, only the shortest distance will be conserved; for example, in Figure 2(a), the estimated distance between S and D is $d_{SY} + d_{YD}$ and $d_{SD} \leq d_{SY} + d_{YD}$ because of triangular inequality. Let $x_1, x_2, \ldots, x_q, a$ be a path from node $x_1 \in V \setminus \Delta$ to anchor $a \in \Delta$. The estimated distance can be defined recursively as follows:

$$\hat{d}_{x_1 a} = d_{x_1 x_2} + \hat{d}_{x_2 a} \tag{1}$$

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

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

3.4.2. DV-Hop.

Description: DV-hop consists of two flood waves. Similar to Sum-Dist, after the first wave, nodes obtain their positions and minimum hop counts to anchors. The second calibration wave allows conversion of hop counts into distances. This conversion consists of 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 to obtain the average hop distance between A and B. A calibrates its distance when it receives the anchor position. Nodes forward calibration messages (only from the first anchor that calibrates them to reduce the total number of messages in the network).

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

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

3.4.3. Euclidian.

Description: Euclidean is based on the local geometry of nodes around an anchor. When a node contains, in its neighborood, two nodes having estimated their distances with an anchor, it uses then the *neighbor vote* method or *common neighbor* method to estimate its distance to the anchor. Consider the example of Figure 2(c). Let A, B, C be nodes and D be an anchor. B and C are neighbors to A. B and C have 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 corresponds to d_{AD}. However, there are two solutions d_1, d_2. The *neighbor vote* method or *common neighbor* method can be used to select the distance d_1 or d_2.

Advantages and drawbacks: When possible, Euclidian provides an exact distance with respect to the anchor. But Euclidean is sensitive to range errors and is efficient only in highly connected networks. Otherwise, the Euclidean performance rapidly degrades.

All of these localization methods described before focus on one of two important criteria: either they locate some sensors with high accuracy (or with exact position) whereas other sensors are not located (particularly in measure-based methods) or they assign an estimated position for all sensors but with coarse estimations (particularly in free-measure methods). Hence, we propose three distributed approximation techniques that consider these two criteria. Our techniques have two major properties: first, they detect some wrong information (e.g., due to range errors). Second, a node knows if its estimated position is close to its real position. In this case, it becomes an estimated anchor. The performances of our methods are compared according to the performances of the methods described in [3–6] in Section 7.

4. FAMILY OF APPROXIMATION TECHNIQUES (AT-FAMILY)

This section explains the three methods AT-Free, AT-Dist, and AT-Angle, which are based on the same general principle. Each node determines its limited geographic zone and calculates the center of gravity of this zone to obtain an estimated position. However, these methods differ in the use of techniques to define these zones according to capabilities of nodes. Moreover, AT-Dist and AT-Angle propose rules to locate some nodes exactly.

4.1. AT-Free technique

This section presents an approximation technique in which nodes have no capability and only use anchor positions to locate themselves. Each node defines a zone to which it belongs according to the anchor positions. The node considers the center of this zone as its estimated position. If the size of its zone is small, the node knows that its estimated position is close to its real position; it becomes an estimated anchor and announces its position to help other sensors to obtain their positions. This section also explain how AT-Free can be used in a preliminary step by other localization methods to eliminate wrong information.

4.1.1. Description.

Initially, anchors broadcast their positions. Then, each sensor deduces the number of hops (only the smallest numbers are considered) between itself and each anchor.

When a node receives a position of an anchor:

- If the node and the anchor are neighbors, then the node deduces that it is inside the disk of radius r and centered on the anchor. Later, we will introduce the uncertainty on the transmission range and see how this measure error can influence on the position of sensors.
- If the node and the anchor are not neighbors, then the node deduces that it is outside the disk of radius r and centered on the anchor, but inside the disk of radius $r \times h$ and centered on the anchor, where h is the number of hops from the anchor to the node.

Thus, the intersection of disks defines a restricted zone, denoted Z_u, containing the node. Formally, Equations (2)–(4) allow to obtain the restricted zone for each node $u \in V \setminus \Delta$.

Figure 3 is an illustration example of AT-Free to estimate the position of a node. Node X receives positions of anchors A, B, C. It calculates the number of hops with these anchors. X is not a neighbor of A, B, C, so X is outside disks centered respectively on A, B, C of radius r. But X is inside disks centered on A (respectively B, C) of radius $r \times h_A$ where h_A represents the number of hops to A (respectively h_B, h_C). Finally, Figure 3 represents the calculated zone with disks and straight lines. X computes the center of this zone and estimates its position in X'.

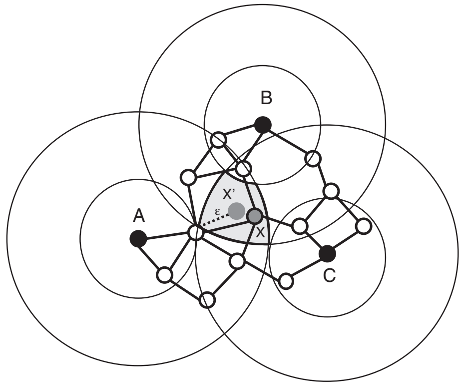

Figure 3. Approximation.

A node knows if its estimated position is close to its real position. Let ϵ be the distance between the center and the point, contained in the zone, furthest away from the center. If we consider the distance d_{err} between the estimated position of the node and its real position representing the estimation error, then the node knows that $d_{err} \leq \epsilon$. Let us consider a $threshold$; if $\epsilon \leq threshold$, then the node has an estimated position close to its real position because $d_{err} \leq \epsilon$. In this case, the node becomes an estimated anchor and broadcasts its position and its ϵ.

When a node applies AT-Free with an estimated anchor, it considers ϵ to calculate disks. If an anchor A is not a neighbor of node X, then the radius of the disks become $r - \epsilon$ and $(r \times h) + \epsilon$. If X and A are neighbors, then X draws a disk with radius $r + \epsilon$.

Therefore, $Z_{N_\Delta(u)}$ and $Z_{\overline{N_\Delta}(u)}$ become Equations (5) and (6) with $\epsilon_a = 0$ for GPS-equipped anchors.

4.1.2. Energy cost and precision tradeoff.

The optimal solution with AT-Free is that each sensor sends its position with its ϵ as soon as it calculates its position, and when it improves its position, it again sends its new position and its new ϵ, and so on. As the sensor energy is limited, this solution is not appropriate because too many messages are sent. This section proposes a technique to manage the number of retransmissions.

In the previous section, a sensor becomes an estimated anchor if its ϵ is lower than a threshold. Only anchors (estimated or not) send their positions. We define two thresholds Γ and ρ such that $\Gamma > \rho > 0$. A node becomes an estimated anchor if its ϵ is lower than threshold Γ. As soon as its ϵ is lower than threshold ρ, it stops calculating and sending its position. The frequency of retransmissions between these two thresholds is defined according to the following principle: *the more a sensor improves its position, the more it should broadcast its position and its* ϵ. Thus, we must define different thresholds from Γ to ρ such that a sensor will broadcast its position and its ϵ when ϵ is lower than one of these thresholds. To define these thresholds and according to the previous principle, we define a

 Wirel. Commun. Mob. Comput. 2014; **14**:1627–1646 © 2012 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm

multiplicative constant δ such that $0 < \delta < 1$. The values of these thresholds from Γ to ρ are equal to

$$Z_{N_\Delta(u)} = \bigcap_{a \in N_\Delta(u)} \left\{(x_i, y_i) \in \mathscr{P} \mid (x_i - x_a)^2 + (y_i - y_a)^2 \le r^2 \right\} \tag{2}$$

$$Z_{\overline{N_\Delta}(u)} = \bigcap_{a \in \overline{N_\Delta}(u)} \left\{(x_i, y_i) \in \mathscr{P} \mid r^2 < (x_i - x_a)^2 + (y_i - y_a)^2 \le (r \times h)^2 \right\} \tag{3}$$

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

$$Z_{N_\Delta(u)} = \bigcap_{a \in N_\Delta(u)} \left\{(x_i, y_i) \in \mathscr{P} \mid (x_i - x_a)^2 + (y_i - y_a)^2 \le (r + \epsilon_a)^2 \right\} \tag{5}$$

$$Z_{\overline{N_\Delta}(u)} = \bigcap_{a \in \overline{N_\Delta}(u)} \left\{(x_i, y_i) \in \mathscr{P} \mid (r - \epsilon_a)^2 < (x_i - x_a)^2 + (y_i - y_a)^2 \le (r \times h + \epsilon_a)^2 \right\} \tag{6}$$

$$\Gamma \ge \delta\Gamma \ge \delta^2\Gamma \ge \cdots \ge \delta^i\Gamma \ge \cdots \ge \rho, \quad i \in \mathbb{N} \tag{7}$$

Thus, when δ is small (respectively large), number of retransmissions decreases (respectively increases). The retransmission frequency increases when approaching ρ. Let k be the maximal number of retransmissions (or iterations) with $k > i > 0$. Hence, broadcasts stop as soon as

$$\delta^k \Gamma \le \rho \tag{8}$$

A maximal bound for k can be deduced as

$$k \le \frac{\log \rho - \log \Gamma}{\log \delta} \tag{9}$$

and δ is fixed at

$$\delta \simeq \left(\frac{\rho}{\Gamma}\right)^{\frac{1}{k}} \tag{10}$$

Thus, values of Γ, ρ, k and δ determine the accuracy of the positions of each sensor.

The number of retransmissions k is a maximal bound: a node sends its position as soon as its ϵ is lower than one of the thresholds $(\Gamma, \delta\Gamma, \delta^2\Gamma, \ldots, \rho)$. It is possible that a node obtains a good estimation of its position quickly without performing exactly k retransmissions, if it receives significant good localization information. In section 7, we analyze values assigned to Γ, ρ, k, and δ to obtain the best result with AT-Free.

4.2. AT-Dist technique

In this section, we propose an approximation technique (AT-Dist) where nodes are equipped with technologies such as ToA/TDoA to calculate distances with respect to their neighbors.

4.2.1. Description.

The principle is the same as AT-Free, but instead of estimating the distance from a node to an anchor as the product of the number of hops and the radius r, it is calculated by Sum-Dist.

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: one circle corresponds to the radius r ($d_{AX} \le r$) and the other to the estimated distance (d_{AX}). In fact, if $A \in N_\Delta(X)$, then X knows d_{AX} and deduces that it is on the circle of radius d_{AX} and centered on A. If $A \notin N_\Delta(X)$, then X knows that it is not inside the disk of center A and radius r; otherwise, A and X would be neighbors. Moreover, X knows the estimated distance to A $\left(\hat{d}_{AX}\right)$ deduced by Sum-Dist. By triangular inequality, $d_{AX} \le \hat{d}_{AX}$. So X is inside the disk of center A and radius $\hat{d}_{AX}$. X applies this technique for each received anchor position. Thus, the intersection of disks defines a zone Z_X containing X. X computes the center of gravity of this zone to deduce its estimated position.

To summarize, for each node $u \in V \setminus \Delta$, Z_u is obtained from Equations (11)–(13).

An example is illustrated in Figure 4. 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. Because all anchors are not neighbors of X, then X is not inside disks centered respectively in A, B, C, D with radius r but is

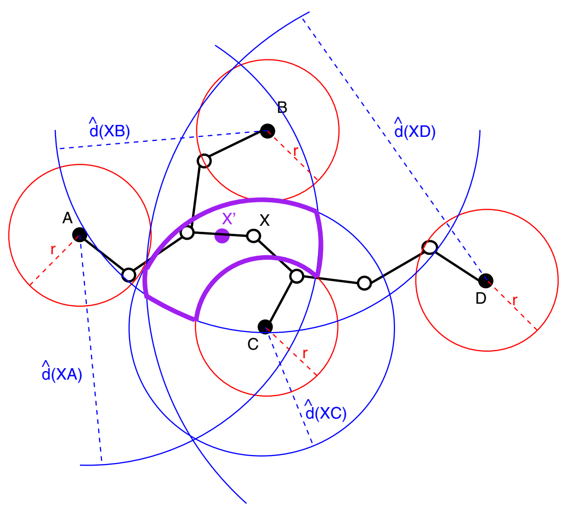

Figure 4. Estimated position of X.

inside disks with radius $\hat{d}_{AX}, \hat{d}_{BX}, \hat{d}_{CX}, \hat{d}_{DX}$. Correlation of these data defines a zone Z_X (delimited in Figure 4 by thick lines). X computes the center of gravity of this zone and estimates its position in X'.

As in AT-Free, a node calculates its estimated position while giving the accuracy of its position (noted ϵ). If ϵ is lower than a predefined threshold Γ, the node becomes an estimated anchor. In AT-Dist, the value of Γ can be very small because the distances are known. For example, in Section 7, this threshold Γ is set at $0.15 \times r$ and $k = 1$. This means that the estimated anchor sends its position only once. Thus, there is quick convergence to obtain accurate positioning and then energy conservation according to communications and computations. When a node applies the approximation technique with an estimated anchor, it takes ϵ into account. In other words, if an anchor A is not a neighbor of node X, then the radius of disks becomes $r - \epsilon$ and $\hat{d}_{AX} + \epsilon$. If X and A are neighbors, then X draws two circles with radius $d_{AX} \pm \epsilon$. Thus, the node deduces that it is between these circles. Therefore, $Z_{N_\Delta(u)}$ and $Z_{\overline{N_\Delta(u)}}$ become Equations (14) and (15) with $\epsilon_a = 0$ for GPS-equipped anchors.

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

$$Z_{N_\Delta(u)} = \bigcap_{a \in N_\Delta(u)} \Big\{(x_i, y_i) \in \mathscr{P} \mid (x_i - x_a)^2 + (y_i - y_a)^2 = d_{ua}^2\Big\} \tag{11}$$

$$Z_{\overline{N_\Delta(u)}} = \bigcap_{a \in \overline{N_\Delta(u)}} \Big\{(x_i, y_i) \in \mathscr{P} \mid r^2 < (x_i - x_a)^2 + (y_i - y_a)^2 \le \hat{d}_{ua}^2\Big\} \tag{12}$$

$$Z_u = Z_{N_\Delta(u)} \cap Z_{\overline{N_\Delta(u)}} \tag{13}$$

$$Z_{N_\Delta(u)} = \bigcap_{a \in N_\Delta(u)} \Big\{(x_i, y_i) \in \mathscr{P} \mid (d_{ua} - \epsilon_a)^2 \le (x_i - x_a)^2 + (y_i - y_a)^2 \le (d_{ua} + \epsilon_a)^2\Big\} \tag{14}$$

$$Z_{\overline{N_\Delta(u)}} = \bigcap_{a \in \overline{N_\Delta(u)}} \Big\{(x_i, y_i) \in \mathscr{P} \mid (r - \epsilon_a)^2 < (x_i - x_a)^2 + (y_i - y_a)^2 \le (\hat{d}_{ua} + \epsilon_a)^2\Big\} \tag{15}$$

4.2.2. Rules to increase position accuracy.

This section presents three rules to resolve ambiguity when a node can be located at two positions. For example, in Figure 5, 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 (i.e., (x_A, y_A)) or in A' (i.e., $(x_{A'}, y_{A'})$). 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 respect to this anchor via Sum-Dist and applies these rules to resolve the ambiguity when a node can be located at two positions. Hereafter, A is assumed to be the real position of X.

4.2.2.1. Rule 1. This first rule defines a simple bound with the 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 6). First, D is not an estimated anchor. X receives $D's$ position (i.e., (x_D, y_D)) and learns its estimated distance to D (i.e., $\hat{d}_{XD}$). At the beginning, X assumes that it is in A, so the following conditions have to be verified:

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

If the two conditions are respected, then X is in A.

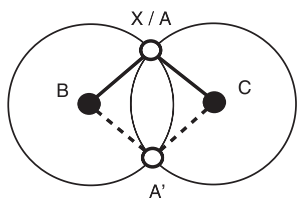

Figure 5. X can be in A or in A'.

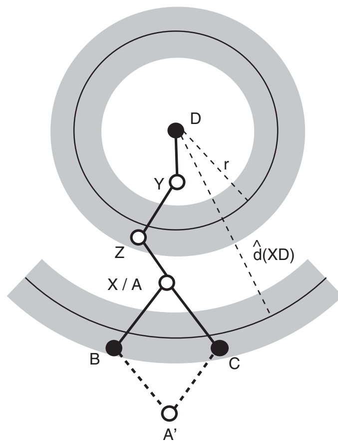

Figure 6. Rule 1.

Now, X assumes that it 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

$$r < d_{AD} \le \hat{d}_{XD} \wedge \left(d_{A'D} \le r \oplus d_{A'D} > \hat{d}_{XD}\right) \quad (\oplus : nor) \tag{16}$$

Second, D is an estimated anchor. The real position of D is inside the disk centered in its estimated position with radius ϵ. Therefore, the real disk of D with radius r is between the two circles centered in the estimated position of D with radius $r \pm \epsilon$ (represented in gray in Figure 6). The same applies for disk centered in D with radius $\hat{d}_{XD}$. X is certain to be in A if

$$r + \epsilon < d_{AD} \le \hat{d}_{XD} - \epsilon \wedge (d_{A'D} \le r - \epsilon \oplus d_{A'D} > \hat{d}_{XD} + \epsilon\Big) \tag{17}$$

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

4.2.2.2. 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 7). First, D is not estimated. When X receives $D's$ position (i.e., (x_D, y_D)), it checks if $d_{A'D} \le r$, and 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:

$$d_{AD} > r \wedge d_{A'D} \le r \tag{18}$$

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

Second, D is an estimated anchor. As in the previous case, the real disk of D with radius r is inside the gray zone in Figure 7. X is certain to be in A if $r + \epsilon < d_{AD}$ and $d_{A'D} \le r - \epsilon$. In others words, A must be outside the disk centered in D with radius $r + \epsilon$, and A' must be inside the disk centered in D with radius $r - \epsilon$:

$$d_{AD} > r + \epsilon \wedge d_{A'D} \le r - \epsilon \tag{19}$$

4.2.2.3. Rule 3. This rule is applied when a node has at least three anchors in its neighborhood.

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 8). First, D is not an estimated anchor. When X receives $D's$ position (i.e., (x_D, y_D)), it checks if $d_{A'D} > r$, and 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:

$$d_{AD} \le r \wedge d_{A'D} > r \tag{20}$$

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

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

$$d_{AD} \le r - \epsilon \wedge d_{A'D} > r + \epsilon \tag{21}$$

4.2.2.4. Note. Rules 2 and 3 are useful when measure errors are introduced in distances. Without measurement errors, rule 2 is included in rule 1. But rule 1 uses Sum-Dist, which can be corrupted by measurement errors. For example, in Figure 9, Sum-Dist returns an estimated

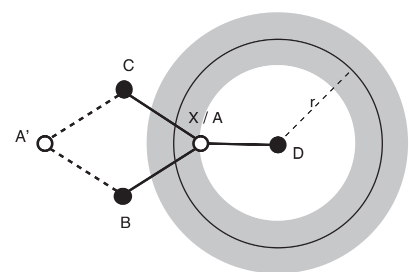

Figure 8. Rule 3.

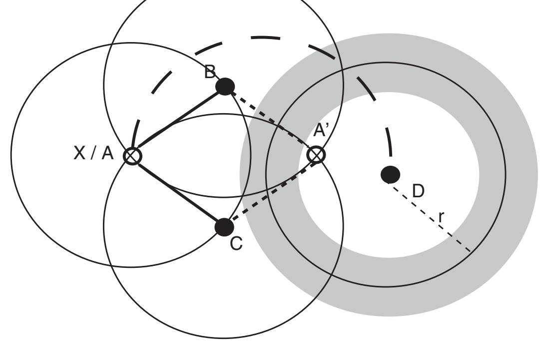

Figure 7. Rule 2.

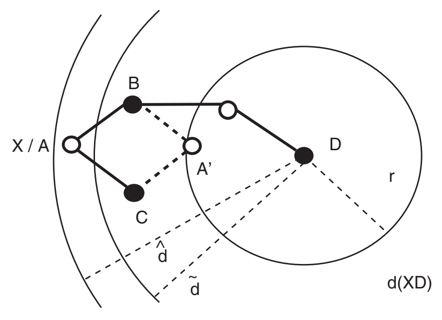

Figure 9. Useful of rule 2.

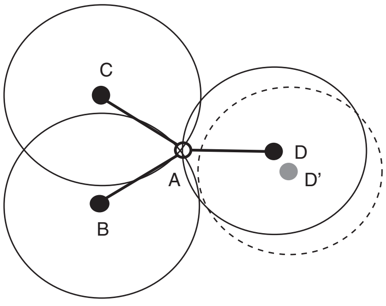

Figure 10. Multilateration cannot be applied.

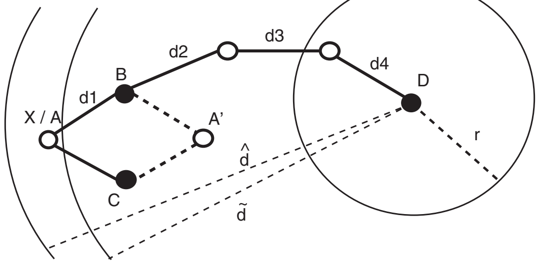

Figure 11. Error of Sum-Dist due to range errors.

distance of $\hat{d}$ without range errors or $\tilde{d}$ with measurement errors such as $\tilde{d} < \hat{d}$. Without range errors, rules 1 and 2 resolve the ambiguity. With measure errors, X cannot conclude in rule 1 and obtains its exact position in rule 2.

For rule 3, multilateration is not recommended, even if a node has at least three anchors in its neighborhood because it is very sensitive to errors. When multilateration is used, the node position corresponds to the point of intersection of three circles whose centers are neighbors. If one of the distances with respect to the 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 10: the localization of node D is slightly wrong, and D is located in D'. But the circle centered in D' does not intersect the same as the two other circles. It is easy to see that this problem is resolved with our rule. So, when range errors are equal to 0%, then multilateration can be used, but as soon as range errors are introduced, then this rule is better.

The next subsection shows rule errors due to range errors and describes the voting process allowing to efficiently avoid a large number of errors.

4.2.2.5. Voting process. In an ideal case (i.e., without range errors), when a node receives an anchor position and when one of the rules can be applied, then the node resolves the ambiguity and obtains its position. In fact, if a rule is applied with only one anchor, such as illustrated in the figures mentioned previously, then the node can be positioned. However, what is the consequence if some information, such as position or estimated distance, from this anchor is wrong because of measurement errors or to an attacker who has taken control of this anchor?

Figure 11 represents the estimated distance error according 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 $\tilde{d} = \tilde{d}_1 + \tilde{d}_2 + \tilde{d}_3 + \tilde{d}_4$ be the result of Sum-Dist with range errors ($\tilde{d}_1, \tilde{d}_2, \tilde{d}_3, \tilde{d}_4$ are distances with range errors) such that $\tilde{d} < \hat{d}$. Figure 11 shows that without range errors, X cannot resolve the ambiguity with anchor D because no rules can be applied. If this example is considered with range errors, the estimated distance is $\tilde{d}$, and then X deduces that it is located in A' by using rule 1, which is false. Although AT-Dist eliminates some wrong information, X cannot achieve confidence with only one anchor. With the voting process, some anchors can be taken into account to deduce the localization of a node. When a node can be located at two positions p_1 and p_2, it checks the rules with all anchors that it knows. When a rule can be applied with an anchor allowing determination of the position of a node in p_1 (respectively p_2), then the node increments a counter cp_1 (respectively cp_2). Now, if $cp_1 - cp_2 \geq confidence$ (respectively $cp_2 - cp_1 \geq confidence$), then the node is located in p_1 (respectively p_2). Without range errors (i.e., the percentage of range errors is equal to 0%), then $confidence$ is 1. In fact, the $confidence$ value is according to the network environment and is defined experimentally. Note that the $confidence$ threshold is the same as the one used to eliminate wrong information in Section 4.4.

4.3. Approximation technique AT-Angle

This section proposes an approximation technique called AT-Angle, where nodes are equipped with the AoA technology to calculate angles with respect to their neighbors.

4.3.1. Algorithmic description.

Initially, anchors broadcast their positions. Then, each sensor deduces the length of the shortest paths to the anchors.

When a node receives a position of an anchor:

- If they are neighbors, then the node deduces that it is inside the disk of radius r and centered on the anchor. Later, we will introduce the uncertainty of the transmission range and see how this measurement error can influence on the position of sensors.
- If they are not neighbors, then the node deduces that it is outside the disk of radius r and centered on the anchor, but inside the disk of radius $r \times h$ and centered on the anchor, where h is the number of hops from the node to the anchor.

Wirel. Commun. Mob. Comput. 2014; **14**:1627–1646 © 2012 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm

Thus, the intersection of disks defines a restricted zone, denoted Z_u^c, containing the node. Formally, Equations (22)–(24) allow to obtain the restricted zone for each node $u \in V \setminus \Delta$.

Let A an anchor, B its neighbor, and d the straight line crossing A's position and B's position.

Then, each node B, which is neighbor of an anchor A, broadcasts the anchor's position and the angle formed between line d and the reference axis (e.g., north, south, east, west, which must be the same for each sensor). All angles are computed according to this axis. In this paper, we consider a plan that will be cut by d into two sides $S1$ and $S2$. Then, a node seeks to deduce if it belongs to this line or to one of the two sides. In the case where a node take a decision, then it broadcasts the collected anchor's information (anchor's position and the angle). Otherwise, it does not broadcast anything.

$$Z_{N_\Delta(u)} = \bigcap_{a \in N_\Delta(u)} \left\{(x_i, y_i) \in \mathscr{P} \mid (x_i - x_a)^2 + (y_i - y_a)^2 \leq r^2\right\} \tag{22}$$

$$Z_{\overline{N_\Delta}(u)} = \bigcap_{a \in \overline{N_\Delta}(u)} \left\{(x_i, y_i) \in \mathscr{P} \mid r^2 < (x_i - x_a)^2 + (y_i - y_a)^2 \leq (r \times h)^2\right\} \tag{23}$$

$$Z_u^c = Z_{N_\Delta(u)} \cap Z_{\overline{N_\Delta}(u)} \tag{24}$$

Let us consider the example described in Figure 12 where A is an anchor and X, Y, and Z are neighbor nodes. When the node X receives the position of anchor A, it deduces the line d (crossing X and A) as well as the angle $\angle A, X = \alpha$ that form this line with the horizontal axis. Because X belongs to the line d, it disseminates this information to its neighbors. At the reception of the message from X, Y deduces the angle $\angle X, Y = \beta$ it forms with X. It plots the straight line d, thanks to the position of A and α, and according to the values of α and β, it deduces the side of d it belongs to: $S1$ or $S2$. If the condition (26) is verified, then Y belongs to $S2$; else, Y belongs to $S1$. Then, Y broadcasts the angle α, its side $S1$, and the position of the anchor A.

When Z receives this message, it measures the angle $\angle Y, Z = \gamma$ that it forms with Y. According to the values of α and γ and side of Y according to d, Z tries to determine the side to which it belongs. Two cases arise: (i) If the angles α and γ verify the condition (26) (Figure 12(a)), then Z deduces that it belongs to the same side $S1$ as Y. Z broadcasts the position of A, and its side $S1$, and so on. (ii) If the angles α and γ does not respect the condition (26) (Figure 12(b)), then Z can deduce nothing. Indeed, without the knowledge of distances, Z does not know if it remains in the same side $S1$ or if it passes on the other side $S2$.

$$\alpha < \beta < \alpha + \pi \tag{26}$$

In other words, a node can conclude if it *moves away* from the straight line formed by the position of A and α, and thus, it stays on the same side of its announcer. Otherwise, a node cannot conclude if it *moves closer* to the straight line and stays on the same side of its announcer. Finally, with the anchor positions and angles, a node computes the zone containing itself.

$$S_d(u) = \begin{cases} S_1 & , \exists v \in N(u), v \in S_1 \mid \angle u, v < \alpha_d \ or \ \angle u, v > \alpha_d + \pi \\ S_2 & , \exists v \in N(u), v \in S_2 \mid \alpha_d < \angle u, v < \alpha_d + \pi \\ S_1 \cap S_2 & , \exists v \in N(u), v \in S_1 \cap S_2 \mid \angle u, v = \alpha_d \ or \ \angle u, v = \alpha_d + \pi \\ \emptyset & , \text{otherwise} \end{cases} \tag{25}$$

Formally, for each node $u \in V \setminus \Delta$, Z_u is obtained as follows:

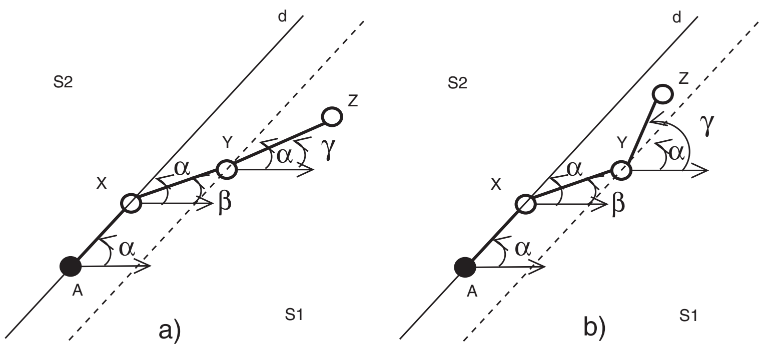

Figure 12. (a) Z deduces its side; (b) Z does not.

From Equation (25), each node u can determine the side $S_d(u)$ it belongs regarding to the straight line d obtained by an anchor and one of its neighbors and the angle α_d.

Let $\mathscr{D}$ be the set of straight lines deduced according to anchor positions and angles:

$$\mathscr{D} = \{d(a, \angle a, v) \ : \ \forall a \in \Delta, \ \forall v \in N(a) \cup \Delta \setminus \{a\}\}$$

Let $\mathscr{S}_u$ be the set of sides for u:

$$\mathscr{S}_u = \bigcap_{d \in \mathscr{D}} S_d(u) \tag{27}$$

The zone Z_u^d of u obtained according to straight lines is defined as follows:

$$Z_u^d = \bigcap_{S \in \mathscr{S}_u} \{(x_i, y_i) \in S\} \tag{28}$$

Zone Z_u of u with straight lines and disks is defined as follows:

$$Z_u = Z_u^d \cap Z_u^c \tag{29}$$

Figure 13 represents the estimation technique. The zone outlined with bold lines defines the area containing node X calculating its position. In Figure 13(a), node X receives positions of anchors A, B, C. It calculates the number of hops with these anchors. X is not a neighbor of A, B, C, so X is outside of disks centered respectively on A, B, C of radius r. But X is inside disks centered on A (respectively B, C) of radius $r \times h_i$ ($i \in \{A, B, C\}$), where h_i represents the number of hops to A (respectively B, C). In Figure 13(b), X also receives angles formed by anchors with their neighbors and other anchors, and X draws straight lines. X deduces sides to which they belong according to the straight lines. Finally, Figure 13(c) represents the calculated zone with disks and straight lines. X computes the gravity center of this zone and estimates its position in X'.

As in AT-Free and AT-Dist, a node calculates its estimated position while giving it the accuracy of its position (noted ϵ). If ϵ is lower than a predefined threshold Γ, the node becomes an estimated anchor. In AT-Angle, the value of Γ can be very small. For example, in simulations (Section 7), this threshold Γ is set to $0.15 \times r$ and k to 1.

4.4. Properties of AT-Family

The three methods in AT-Family have two important properties:

- First, a node knows if its estimated position is close to its real position regarding to its position error bound: this bound denoted as ϵ is the distance between its estimated position (i.e., the gravity center) and the point in its zone, furthest away from the gravity center. Let d_{err} being the distance between its estimated position and its real position that represents the position error. Whatever the real position of the node, it 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. Thus, when a node applies the approximation technique with an estimated anchor radius, it takes ϵ into account: if an anchor A is not neighbor of node X, then radius of disks become $r - \epsilon$ and $r \times h + \epsilon$. If X and A are neighbors, then X belong to disk of radius equal to $r + \epsilon$. Therefore, $Z_{N_\Delta(u)}$ and $Z_{\overline{N_\Delta(u)}}$ become as indicated in Equations (14) and (15) with $\epsilon_a = 0$ for anchors equipped with GPS module.
- Second, according to its zone, a node can detect if some localization information are wrong, particularly when a node receives information locating itself outside of its zone. More phenomena can be responsible for this situation: angle measurements can be wrong because of the network environment and introduce wrong information; the anchor may be under control of an attacker in a military context and announce a wrong position. Thus, when a sensor receives localization information, it checks that these data are consistent regarding to its defined zone; otherwise, it deletes them.

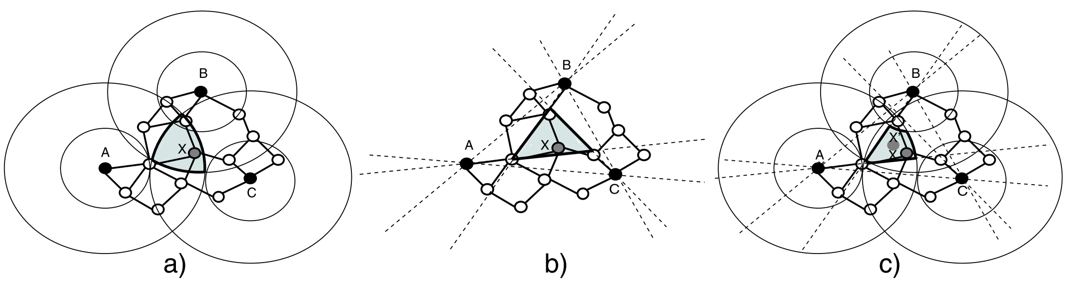

Figure 13. Approximation with (a) disks, (b) straight lines, and (c) both.

Wirel. Commun. Mob. Comput. 2014; **14**:1627–1646 © 2012 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm

5. IMPLEMENTATION

For each method belonging to AT-Family, sensor nodes represent the network by a grid. The length of the cases side in grid is set at $r\sigma$ (where σ is a multiplicative factor). To ensure that the estimation accuracy is not notably compromised, σ is set at 0.01. 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 neighbors, all cases
  - inside the disk having the anchor as center and a radius of r for AT-Free and AT-Angle.
  - on the circle having the anchor as center and a radius equal to the distance returned by ToA or TDoA for AT-Dist.
- If the node and anchor are not neighbors: all cases between the two circles having the anchor as center and a radius, respectively, of $\hat{d}_1$ and $\hat{d}_2$ with
  - $\hat{d}_1 = r$ and $\hat{d}_2 = r \times h$ where h is the minimum hop number from the node and the anchor, for AT-Free and AT-Angle.
  - $\hat{d}_1 = r$ and $\hat{d}_2$ is returned by Sum-Dist.

Moreover, in AT-Angle, each sensor also increments all cases on sides of straight lines containing it. Figure 14 represents an example of grid for AT-Angle: when node X receives the position of A (respectively, B, C), it increments all cases between the two circles centered in A (respectively B, C) and increments cases according to straight lines. The zone containing X is defined by the maximum area in the grid. In Figure 14, this zone is defined by cases equal to 6. X calculates the center of gravity of this zone and obtains an estimated position.

Note that when a node defines the zone where it belongs, it does not conserve the entire grid in its memory but only this zone. The speed of computations in this grid thus increases as the size of the zone decreases.

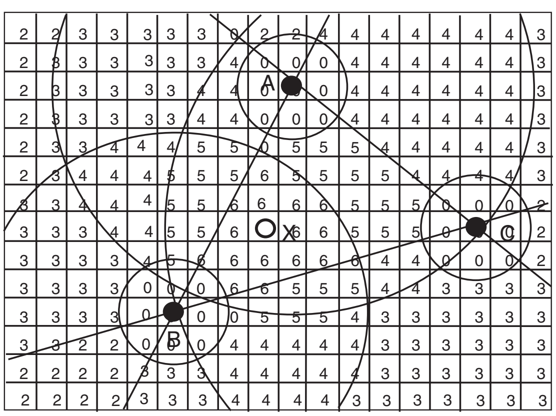

Figure 14. A network represented by a grid.

6. COMMUNICATION MODEL AND ERROR MEASUREMENT

Considering the communication model as a disk is a strong assumption. Indeed, the transmission can be perturbed by obstacles or other elements in the network. The network environment also perturb measures of distances or angles. However, it is possible to analyze the network before the sensor deployment to overcome these errors.

6.1. Communication model

In AT-Family methods, the problem due to perfect disk happens when a sensor receives an anchor position either directly or indirectly. When it receives this position directly, then it is assured to be inside the transmission range of the anchor. But, if it receives this position indirectly, that does not mean that the distance between it and the anchor is high than r (e.g., an obstacle stops communication between them).

On one hand, if it receives this position directly then it is assured to be inside the transmission range of the anchor. On the other hand, if it doesn't receive it directly, this does not necessarily mean that the distance between the sensor and the anchor is greater than r (e.g., it can be just the presence of an obstacle which prevents their communication). So, the first solution is to ignore this property (which is one rule of the implementation seen previously). Hence, the problem would be resolved but AT-Family would lose in precision. The second solution is to perform an analysis of the network environment before deployment.

Figure 15 is an example of two possible covers for a transmission range r.

Thus, we have to take into account the two following items:

- If the distance between two sensors is less than r, then their communication is not assured.
- Communications can be unidirectionals.

However, thanks to the network environment analysis, it is possible to define two bounds b_1 and b_2. These two bounds delineate the transmission range.

Let u and v be two sensors. If v receives a signal from u, it deduces that it is inside the disk centered in u of radius b_2. Conversely, if v does not receive signal from u, it deduces that it is outside the disk centered in u of radius b_1 (i.e., if v is inside this disk, it has to necessarily receive the signal).

It is possible to adapt AT-Family methods to take these data into account. For example, in AT-Free, let u be a sensor that does not know its position and A be an anchor such that $A \notin N_\Delta(u)$. When u estimates its position with A's position, it deduces that it is inside the disk centered in A of radius $r \times h$ and it is outside the disk of radius b_1.

6.2. Measurement of errors

Measurement of errors are also due to network environment. A statistical analysis can be performed before

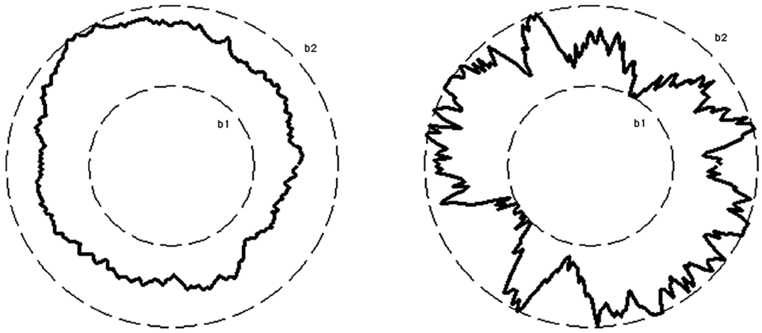

Figure 15. Perturbation example of radio transmission.

sensor deployment to obtain a bound of these measurements (denoted ξ). Therefore, each sensor could take these bounds into account. Nodes can manage these errors in the same manner as ϵ. Thus, each circle is replaced by two circles of radius $radius \pm (\epsilon + \xi)$. In the case of measure errors due to the measurements of angles, these errors can be taken when a node draws a straight line with an angle α into account. The node draws two straight lines with angles $\alpha \pm \xi$ and considers them to deduce its side.

About AT-Dist, if a bound of these errors (ξ) is known, each rule would take ξ into account. For example, let $x_1, x_2, \ldots, x_q$ be a path of length q. The estimated distance from x_1 to x_q is bounded:

$$\hat{d}_{x_1 x_q} - q \times \xi \le d_{x_1 x_q} \le \hat{d}_{x_1 x_q} + q \times \xi \qquad (30)$$

In this case, the voting process would be useless to manage introductions or accumulations of range errors but still be useful for managing other error sources. In future work, it would be interesting to find an optimized interval for the estimated distance.

7. SIMULATIONS

7.1. Simulation environment

AT-Family methods are simulated with OMNET++, a discrete event simulator [28]. Concurrent transmissions are allowed if the transmission areas 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. Simply, the simulator uses CSMA/CA MAC protocol. Message corruption is not considered, so all messages sent during the simulation are delivered.

First, a random network topology is generated according to the number of nodes and the number of anchors. Nodes are randomly placed, with a uniform distribution, within a square area. Also, anchors are randomly selected.

To allow easy comparison between different scenarios, distance errors and errors on the estimated position are normalized according 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. Angle errors are normalized according to π. For example, 10% of angle errors means that the calculated angle can belong to the interval angle $\pm\ \pi \times 0.10$. The percentage of angle errors is called δ.

All nodes are distributed in a square 100×100. The connectivity (average number of neighbors) is controlled by specifying the radio range. By default, scenarios use networks with 150 nodes, and the radio range is set to 14. Thus, sensor density is equal to 9.24. The percentage of anchors varies from 0% to 20% representing density of anchors from 0.12 to 1.23.

Different scenarios are performed while changing the measurements of error percentage δ respectively equal to 0%, 5%, 10%. Moreover, a node becomes an estimated anchor if its maximum position error is lower than 15% (i.e., $\epsilon \le r \times 0.15$).

AT-Family is proposed to resolve the localization problem in WSNs. Therefore, simulations focus on a criterion allowing to evaluate performances of three methods for this problem: 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). This criterion is analyzed according to

- anchor percentage
- measurement error percentage
- node density

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

7.2. Results

This section presents results obtained by our methods according to existing solutions. AT-Free is compared with HTRefine [3], which are two range-free methods. AT-Dist and AT-Angle are compared with APS [5], SumDist+MinMax [4], and APS_{AoA} [6], which are range-based methods.

7.2.1. Range-free methods.

In this section, a percentage of α anchors is randomly selected with $\alpha \in \{5, 10, \ldots, 35, 40\}\%$, giving an anchor density in the square from 0.46 to 3.7. In our AT-Free simulations, the values of Γ, ρ, k, and δ are chosen to obtain the best results with AT-Free. Therefore, the value of ρ is set at $0.01 \times r$, representing a position error threshold of 1%. Below this threshold, a sensor is considered as an anchor with accurate estimation and then stops to calculate its position. The Γ value is defined experimentally, and we claim that the value of this threshold is close to r/d, where d represents the anchor density. It is clear that this threshold is inversely proportional to the anchor density: Γ decreases as the number of anchors in the network increases. Finally, in our simulations, k varies and so δ is calculated by Equation (10). It is clear that the energy consumption is according to the number of broadcasts.

We focus on the performances of AT-Free to locate sensors with high accuracy, and we analyze average errors calculated according to the position of each sensor. The results show the impact of the anchor (GPS equipped) density (α) and the node density.

The graph in Figure 16 represents the average positioning errors according to the percentage of anchors. This graph contains four curves: three curves correspond to AT-Free with $k \in \{1, 5, 10\}$. The fourth curve corresponds to the HTRefine method. Without surprise, the average error decreases as the number of anchors in the network increases, whatever methods or configurations used. The gap between AT-Free and HTRefine significantly increases when the nodes contain more anchors. Note that using different values for k is interesting when α is lower than 25%. After this percentage, we can use $k = 1$. In fact, when the network contains more anchors, information provided by anchors allows a sensor to more quickly deduce a good estimated position, and more than one broadcast is not required. Up to 25% of anchors, the broadcast number has a high impact, particularly when α is low. Note that with a percentage of anchors higher than 30%, the performances are not significant.

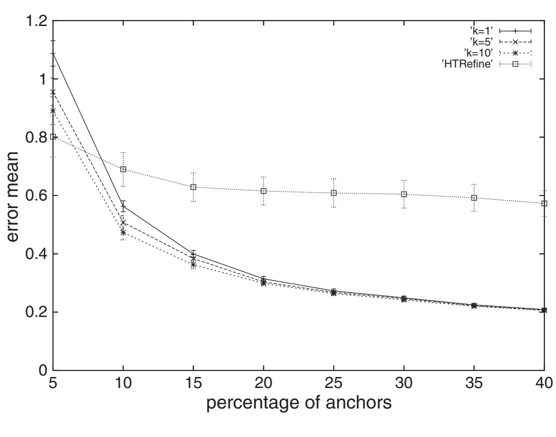

Figure 16. Average error rate of AT-Free for $k = \{1, 5, 10\}$ and HTRefine.

The graph in Figure 17 represents the behavior of average positioning errors according to the density of nodes. The density value varies from 3 (representing a density low) to 15 (representing a high density). These simulations use an anchor percentage equals to 10%. The graph illustrates that the AT-Free performance increases when the node density is high: in fact, constraints imposed by neighborhood relations allow each sensor to obtain a position close to its real position. Note also that the difference between k values is increasingly significant when the density increases. For each node containing more neighbors, if the number of broadcasts increases, then other sensors obtain more localization information to deduce their positions.

Finally, when AT-Free is used in a preliminary step, we can use a number of broadcasts that will provide good result (but not the best result) to obtain small zone sizes while preserving energy. For example, with $\alpha = 10\%$, we can use $k = 5$ rather than $k = 10$ in Figure 16. It is also possible to modify the ρ value to decrease energy consumption.

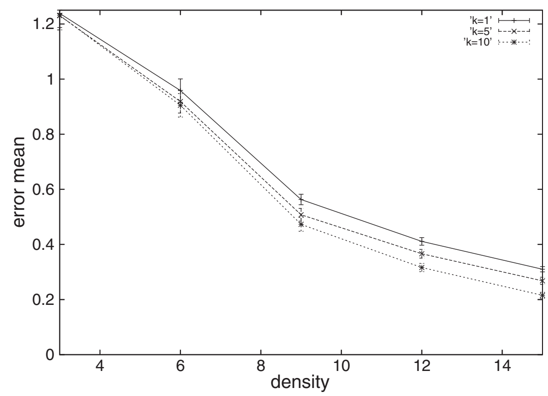

Figure 17. Average error rate according to node density.

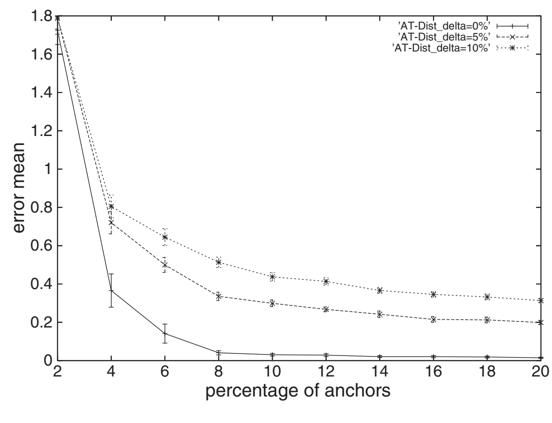

Figure 18. AT-Dist: average error rate for $\delta = \{0, 5, 10\}\%$ according to α.

7.2.2. Range-based methods.

In the next simulations, we consider a *confidence* value equal to 2 for AT-Dist (respectively 3 for AT-Angle). It is the sufficient threshold for a sensor to consider a set of anchors to deduce its position and eliminate some wrong information. The choice of the value 2 (respectively 3) comes from simulations given in Figure 22 (respectively Figure 23) in the Appendix.

Now, we focus on the efficiency of AT-Dist and AT-Angle and consider a scenario with 150 nodes, and when α varies from 0% to 20%, this represents an anchor density ranging from 0.12 to 1.23.

Figures 18 and 19 represent the behavior of the average error rate of AT-Dist and AT-Angle without and with measurements of errors according to the percentage of anchors. These curves indicate the accuracy of localizations when δ is equal to 0%, 5%, or 10%. Without surprise, the performances of AT-Dist and AT-Angle decrease as the measure errors increase. However, our methods provide a good estimation of positions. Note also that after 10% of anchors, the average error rate decreases slowly.

Figures 20 and 21 show the impact of node density on the behavior of the average error rate. The average error rate decreases as the node density increases. Note that after a node density of 12, the behavior of the average error rate is not significant.

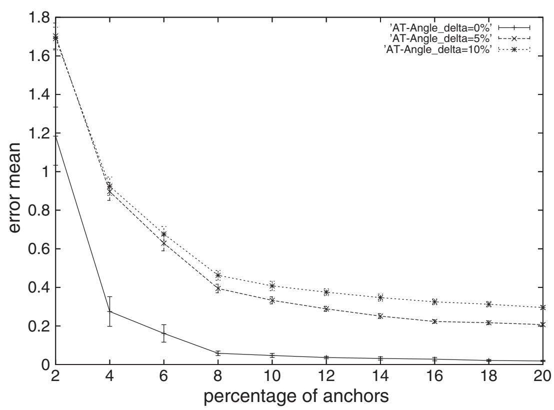

Figure 19. AT-Angle: average error rate for $\delta = \{0, 5, 10\}\%$ according to α.

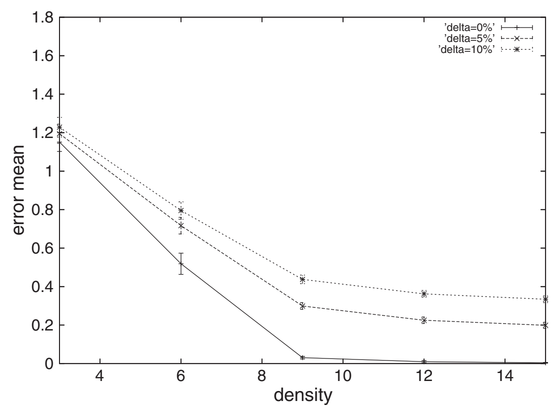

Figure 20. AT-Dist: average error rate according to node density with $\alpha = 10\%$.

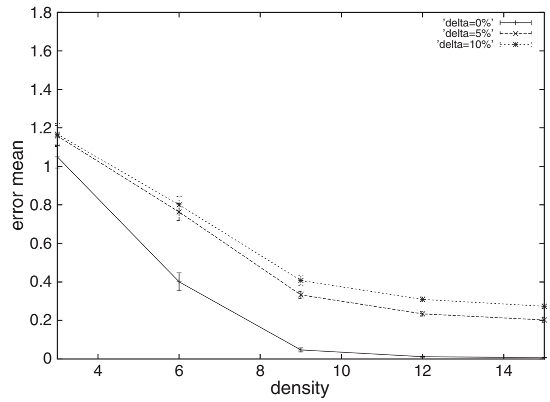

Figure 21. AT-Angle: average error rate according to node density with $\alpha = 10\%$.

In the Appendix, Figures 24 and 25 show the impacts of the range error and the angle measurement error on the position mean error. Figure 26 compares the performances of our methods and methods in [4,5] called *APS* and *SumDist+MinMax*, respectively. This curve represents the percentage of nodes located with a position error less than 20%.

Finally, and unsurprisingly, methods using distance or angle measure techniques (AT-Dist and AT-Angle) are better than the AT-Free range-free method. Note however that according to these graphs, AT-Angle seems better than AT-Dist. In fact, we cannot compare the results obtained by these methods because the percentage of measurement errors is normalized with π and r, respectively.

8. CONCLUSION

We proposed AT-Family, a set of three distributed methods (AT-Free, AT-Dist, AT-Angle) that resolve the localization problem according to sensor capabilities. These methods have the same general principle: each node defines a zone containing it and calculates the center to estimate its position. Nodes become estimated anchors if their positions are close to their real positions. Moreover, some rules are defined to exactly locate some nodes while nodes can calculate either distances or angles. Finally, the defined zones allow sensors to eliminate wrong information as soon as it is in contradiction with the zones. The performances of these methods are shown by simulations in comparison with existing solutions.

Some prospects of this work have to be studied: first, the time of convergence and the energy consumption for each AT-Family method should be studied. However, these two criteria depend on the anchor information broadcasting strategy, and this paper only focuses on the sensor localization problem. Second, in this paper, anchors are positioned randomly in the network. However, anchor positions impact on the performances of our methods. This gives rise to other problems: *Is there an anchor positioning strategy to obtain the best localization for each sensor according to our methods?* or also *Is there an anchor positioning strategy to reduce measure errors according to our methods?*. Problems due to mobility in sensor networks have to be studied while taking mobility characteristics into account: velocity, movement, and so on. Finally, all our proposed methods will be implemented in a real testbed including different technologies of sensors.

APPENDIX

The threshold, noted confidence, allows a sensor to consider a set of anchors to deduce its position in voting process and eliminate some wrong information in AT-Dist and AT-Angle. Figure 22 (respectively Figure 23) represents error mean in a network containing percentage of anchors equal to 10% and range errors respectively equal to 5%, 10%, and 15%, related to value of the threshold confidence in AT-Dist (respectively AT-Angle). When the value of confidence is equal to 2 (respectively 3), the obtained error mean is the best. In fact, when the value of confidence is higher than 2 (respectively 3), the voting process is very strict, and nodes cannot deduce their positions. Conversely, when the value of confidence is lower than 2 (respectively 3), the voting process assigns in some times

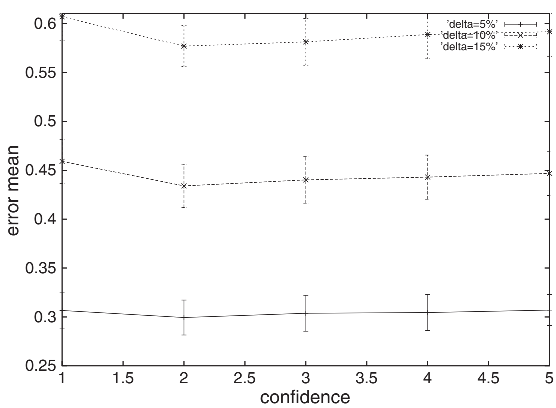

Figure 22. AT-Dist: *confidence* variations with α=10%.

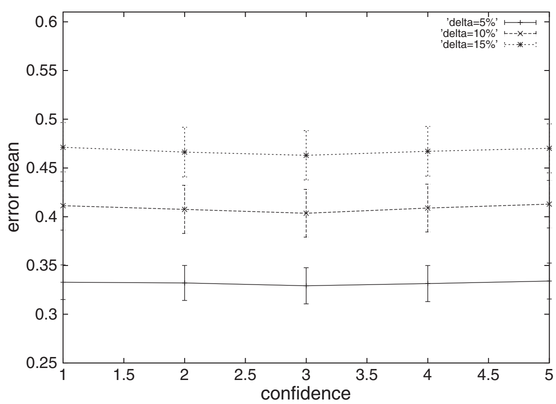

Figure 23. AT-Angle: *confidence* variation with α=10%.

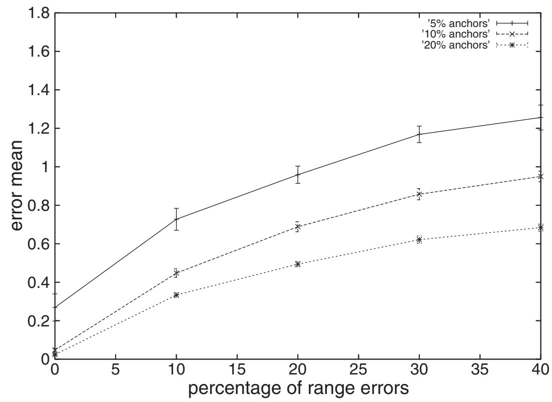

Figure 24. Average error rate with range errors from 0% to 40%.

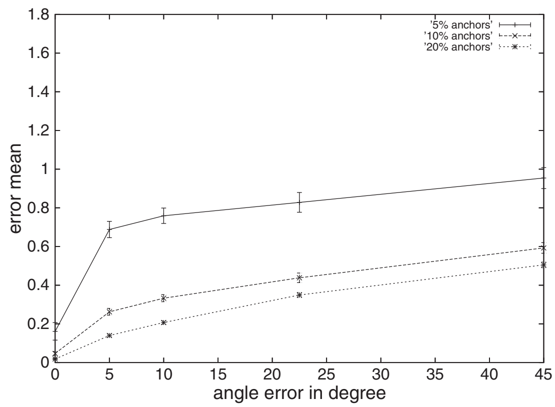

Figure 25. Average error rate with angle errors from 0° to 45°.

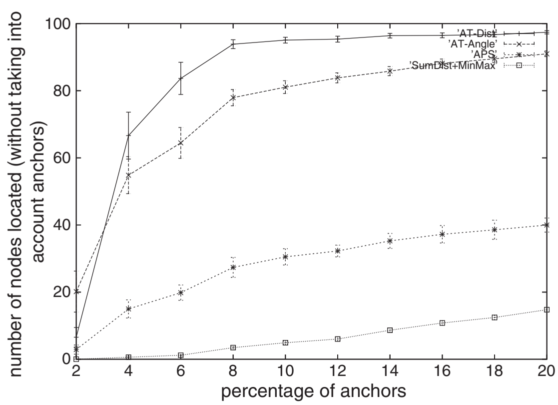

Figure 26. Comparison of percentages of nodes located with an error of less than 20% between our methods and methods in [3–5] with $\delta = 0\%$.

bad positions to sensors because it uses a few number of anchor positions, and some wrong information can be used. It is possible that the confidence value increases when the percentage of range errors is higher than 15%.

Figures 24 and 25 show impacts of the range error and angle error on position mean error. In each figure, 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 and to angle errors. On the horizontal axis, the percentage of range error is varied from 0% to 40% for AT-Dist (respectively from 0° to 45° for AT-Angle). These figures show the performances of AT-Dist and AT-Angle in managing introductions or accumulations of range errors. Related to values of range error and angle error, the average error rate stays reasonable.

Figure 26 represents the percentage of located nodes with a position error less than 20% using our two methods AT-Dist and AT-Angle compared with the methods – APS, HTRefine, and SumDist+MinMax – described in [3–5]. Here, δ is set to 0%. The efficiency of our methods is clearly shown. For example, with $\alpha = 10\%$, AT-Dist locates 90% of nodes with an error less than 20%, AT-Angle locates 80% of nodes with an error less than 20%, and the other methods locate less than 30% of nodes with an error lower than 20%.

REFERENCES

1. Mendes LDP, Rodrigues JJPC. A survey on cross-layer solutions for wireless sensor networks. In *Elsevier Journal of Network and Computer Applications*. Elsevier, March 2011; 523–534.
2. Spilker Jr JJ, Enge PK, Parkinson BW, Axelrad P. Global positioning system: theory and application. *Progress in Astronomics and Aeronotics* 1996; **volume I**.
3. Savarese C, Rabaey J. Robust positioning algorithms for distributed ad-hoc sensors networks, In *USENIX Technical Annual Conference*, Monterey, CA, June 2002; 317–328.
4. Savvides A, Park H, Srivastava M. The bits and flops of the n-hop multilateration primitive for node localization problems, In *WSNA '02 Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, Atlanta, 2002; 112–121.
5. Niculescu D, Nath B. Ad hoc positioning system (APS), In *Proceedings of GLOBECOM*, San Antonio, November 2001.
6. Niculescu D, Nath B. Ad hoc positioning system (APS) using AoA, In *IEEE INFOCOM 2003*, San Francisco, CA, 2003; **3**: 1734–1743.
7. Wang C, Xiao L. Sensor localization under limited measurement capabilities. In *IEEE Network*. IEEE, May/June 2007; **21**(30): 16–23.
8. Rong P, Sichitiu ML. Angle of arrival localization for wireless sensor networks, In *3rd Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, Reston, VA, USA, September 25–28, 2006; 374–382.
9. Saad C, Benslimane A, König JC. A distributed method to localization for mobile sensor networks, In *IEEE Wireless Communications and Networking Conference*, Hong Kong, China, March 11-15, 2007.
10. Tilak S, Kolar V, Abu-Ghazaleh NB, Kang K-D. Dynamic localization control for mobile sensor networks, In *24th IEEE International Performance, Computing, and Communications Conference (IPCCC 2005)*, Phoenix, Arizona, April 7–9, 2005.
11. Priyantha NB, Balakrishnan H, Demaine ED, Teller S. Mobile-assisted localization in wireless sensor networks, In *IEEE Infocom*, March 2005.
12. Li S, Lowe D, Kong X, Braun R. Wireless sensor network localization algorithm using dynamic path of mobile beacon, In *The 17th Asia-Pacific Conference on Communications (APCC)*, Sabah, Malaysia, 2–5 October 2011; 344–349.
13. Wang J, Ghosh RK, Das SK. A survey on sensor localization. *Journal of Control Theory and Applications* 2010; **8**(1): 2–11.
14. Pal A. Localization algorithms in wireless sensor networks: current approaches and future challenges. *Macrothink Institute Online Journals, Networks Protocols and Algorithms* 2010; **2**(1): 45–74.
15. 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*, Island of Maui, January 3–6, 2001; 1–9.
16. Bahl P, Padmanabhan VN. Radar: an in-building RF-based user location and tracking system, In *Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. (INFOCOM 2000)*, Vol. 2, 2000; 775–784.
17. Priyantha NB, Chakraborty A, Balakrishnan H. The cricket location-support system, In *International Conference on Mobile Computing and Networking*, Boston, 2000; 32–43.
18. Bulusu N, Heidemann J, Estrin D. Adaptative beacon placement, In *The Twenty First International Conference on Distributed Computing Systems*, Phoenix, 2001; 489–498.
19. He T, Huang C, Blum BM, Stankovic JA, Abdelzaher T. Range-free localization and its impact on large scale sensor networks. *IEEE Personal Communications Magazine* 2005.
20. Niculescu D, Nath B. DV based positioning in ad koc networks. *Journal of Telecommunication Systems* 2003; **22**(1–4): 267–280.

21. Bulusu N, Heidemann J, Estrin D. GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine* October 2000; **7**(5): 28–34.
22. Savvides A, Han C-C, Strivastava MB. Dynamic fine-grained localization in ad-hoc networks of sensors. *MobiCom '01 Proceedings of the 7th Annual International Conference on Mobile Computing and Networking* 2001; **78**(56): 166–179.
23. Venkatraman S, Caffery J Jr, You HR. Location using los range estimation in nlos environments. *Proceedings of IEEE Vehicular Technology Conference, Spring 2002* 2002; **2**: 856–860.
24. Venkatraman S, Caffery J Jr. Multipath-aided location estimation using angles-of-arrival, In *Proceedings of SPIE*, Orlando FL, USA, April 24, 2003; **5084**: 67–75.
25. Langendoen K, Reijers N. Distributed localization in wireless networks: a quantitative comparison. *Computer Networks: The International Journal of Computer and Telecommunications Networking - Special Issue: Wireless Sensor Networks* 2003; **43**(15): 199–518.
26. Boushaba M, Hafid A, Benslimane A. High accuracy localization method using AoA in sensor networks. In *Elsevier Computer Networks Journal*, Vol. 53, N 18. Elsevier, 24 December 2009; 3076–3088.
27. Kulakowski P, Vales-Alonso J, Egea-Lopez E, Ludwin W, Garcia-Haro J. Angle-of-arrival localization based on antenna arrays for wireless sensor networks. *Computers & Electrical Engineering* November 2010; **36**(6): 1181–1186.
28. Varga A. The OMNET++ discrete event simulation system, In *European Simulation Multiconference (ESM'2001)*, Prague, Czech Republic, 2001.

AUTHORS' BIOGRAPHIES

Abderrahim Benslimane is a full professor of Computer Science and Engineering at the University of Avignon (France) since September 2001. He is attributed the French Award of Scientific Excellency (2011-2014). He has been an associate professor at the University of Technology of Belfort-Montbéliard since September 1994. He obtained the title to supervise researches (HDR 2000) from the University of Cergy-Pontoise, France. He received his PhD degree (1993) and DEA (MS 1989) from the Franche-Comte University of Besançon and his BS (1987) from the University of Nancy, all in computer science.

His research and teaching interests are in wireless ad hoc and sensor networks. Particularly, he works on multicast routing, inter-vehicular communications, quality of service, energy conservation, localization, intrusion detection, and MAC layer performance evaluation. He was also interested in specification and verification of communication protocols, group communication algorithms, and multimedia synchronization. He has several refereed international publications (book, journals, and conferences) in all those domains.

He has served as technical program chair and co-chair and member of a number of international conferences. He has been a reviewer of a great number of journals and of national research projects sponsored by the ANR/Telecom. He is involved in many national and international projects. He is a member of many editorial boards of international journals.

He was the head of Computer Networks and Multimedia Applications group (RAM) of the Computer Laboratory of Avignon from 2001 to 2009. He has several international collaborations for research projects and supervising MSc and PhD students. He is involved in many national and international founded projects. He is a member of several editorial boards of international journals. He serves as a General Chair of the IEEE WiMob since 2008. He chairs many IEEE international conferences and technical symposia such as ICC and Globecom. He was a guest editor of many special issues. He participates to the steering and the program committee of many IEEE international conferences. He is an IEEE senior member, a member of the CA of the IEEE French section, a member of the Technical Committee IEEE ComSoc Communications and Information Security (CISTC), the Vice Chair of the France IEEE student activities, and the Publications Vice Chair of the IEEE COMSOC CISTC 2009-2011. He was a member of the French Council of Universities (CNU section 27) 2003-2007.

Clement Saad received his PhD at the University of Avignon in July 2008. He obtained his MSc degree in computer science from Montpellier 2 University, France, in 2005. His work concerns wireless sensor networks. He focuses especially on positioning and routing problems.

Jean-Claude Konig is a professor of Computer Sciences at the University of Montpelier since 1998. He was an assistant professor at the ENSIA (Massy, Ministry of Agriculture) in 1986, an associate professor at the University Paris XI (Orsay) in 1988, and a professor at the University of Evry-Val-d'Essonne (Evry) in 1992.

He is a member of the team APR (Algorithms and Performances of Networks) within the Laboratory LIRMM since September 1998. He is the head of LIRMM. His works concern the algorithms for network communication and the scheduling theory. He has several international publications. He participated in several research projects.

DOI: 10.1002/wcm

Mohammed Boulmalf received his MSc and PhD degrees both in wireless communications and networking in 1994 and 2001, respectively. From September 1994 to December 1998, he was with INRS-Telecom as a radio communications research engineer. In January 1999, he joined ETS, Quebec University where he worked as a lecturer. In September of the same year, Boulmalf moved to Microcell Telecommunications, GSM Operator in Canada, where he worked as a senior network engineer. From 2000 to 2002, he worked as a principal engineer at the Multi-vendor Integration Department at Ericsson, Montreal, Canada. In 2002, he joined the Arab Emirates University, Al Ain, Abu Dhabi, where he worked as an assistant professor. In 2007, he joined the School of Science and Engineering at Al Akhawayn University in Morocco, and then in 2011, he moved to the International University of Rabat where he is currently an associate professor. He is the author/co-author of about 50 articles in refereed journals and conferences in the areas of wireless networking and communications, mobile computing, and network security.