

A Distributed Method for Localization in Large-Scale Sensor Networks based on Graham's scan

Yassine SABRI, Najib EL KAMOUN

STIC Laboratory,

Chouaib Doukkali University, B.P: 20, El Jadida MOROCCO

Email: {sabriyassino|elkamoun}@gmail.com

Abstract— Localization is an important aspect in the field of wireless sensor networks that has attracted significant research interest recently. The interest in wireless sensor network localization is expected to grow further with the advances in the wireless communication techniques and the sensing techniques, and the consequent proliferation of wireless sensor network applications. This paper presents an improved localization algorithm with high accuracy in large-scale Sensor networks with a large number of sensor nodes based on the Graham's scan, called Slsng. the Graham's scan adapted here for our approximation technique to determining the convex hull of a set of sensors used instead of the Grid-Scan method, to take into account the requirements in memory, to make it scalable and rapidly convergent with small location estimation error. We verify our algorithm in various scenarios and compare it with AT-Dist method. Simulation results show that our proposal is superior to the state-of-the-art localization algorithms for Wireless sensor networks in large-scale.

Index Terms—wireless sensor network (WSN), Localization, and scalability.

I. INTRODUCTION

A wireless sensor network have been discussed for more than 25 years [1], but the vision of wireless sensor networks (WSNs) has been brought into reality only by the recent advances in wireless communications and electronics, which have enabled the development of low-cost, low-power and multi-functional sensors that are small in size and communicate over short distances. Today, cheap, smart sensors, networked through wireless links and deployed in large numbers, provide unprecedented opportunities for monitoring and controlling homes, cities, and the environment. In addition, networked sensors have a broad spectrum of applications in the defence area, generating new capabilities for reconnaissance and surveillance as well as other tactical applications [2]. WSN localization techniques are used to estimate the locations of the sensors with initially unknown positions in a network using the available a priori knowledge of positions of a few specific sensors in the network and inter-sensor measurements such as distance, time difference of arrival, angle of arrival

and connectivity. Sensors with the a priori known location information are called anchors and their locations can be obtained by using a global positioning system (GPS), or by installing anchors at points with known coordinates, etc. In applications requiring a global coordinate system, these anchors will determine the location of the sensor network in the global coordinate system. In applications where a local coordinate system suffices (e.g., in smart homes, hospitals or for inventory management where knowledge like in which room a sensor is located is sufficient), these anchors define the local coordinate system to which all other sensors are referred. Because of constraints on the cost and size of sensors, energy consumption, implementation environment (e.g., GPS is not accessible in some environments) and the deployment of sensors (e.g., sensors may be randomly scattered in the region), most sensors do not know their own locations. These sensors with unknown location information are called non-anchor nodes and their coordinates need to be estimated using a sensor network localization algorithm. In some other applications, e.g., for geographic routing in WSN, where there are no anchor nodes and also knowledge of the physical location of a sensor is unnecessary, people are more interested in knowing the position of a sensor relative to other sensors. In that case, sensor localization algorithms can be used to estimate the relative positions of sensors using inter-sensor measurements. The obtained estimated locations are usually a reflected, rotated and translated version of their global coordinates. Existing researches for sensor localization mainly fall into two categories: range-based approaches and range-free approaches in [3],[4]. Range-free approaches locate nodes using network connectivity information instead of accurate distance measurements between nodes. Range-based approaches ([5], [6], [7],[8]) measure the distance or direction among the ordinary nodes (or target nodes) and seeds to compute the position of each node, 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. This paper presents range-based method called Slsng an improved of AT-Dist [8]. This method proposes a set of three rules and an approximation technique in order to assign either an exact position or an estimated position for each sensor node. The rules and the approximation technique use

the data correlation between anchor positions and distances from nodes to anchors. As soon as a sensor node can apply one of rules, it obtains an exact position. Otherwise, by the approximation technique, it obtains an estimated position. With this approximation technique, using Graham's scan [9] each sensor node defines a convex hull containing itself, according to the anchor positions and distances from it to anchors. To be located, this node computes an estimated position being the center of gravity of this convex hull. three important properties : first, a node can detect when its estimated position is relatively close to its real position. In this case this node becomes an estimated anchor and will be used by others nodes to obtain their positions. Second, some wrong information (e.g. due to measure errors) can be eliminated related to defined sensor convex hull. These properties allow to obtain very good simulation results related to the methods described in [5],[6] , [7] ,[8], even if measure errors are introduced .third, Graham's scan allowed us to reduce the consumption of CPU time (and therefore energy),But also allowed us to optimize including consumption of the memory, focusing not on the overall interpretation of network such as a type algorithm scan-line but only on points of convex hull. Consequently, We get to keep the functional properties of our localization technique despite change in network size with a minimum convergence time.

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

II. MODEL

Extensive approaches have been proposed to locate sensor nodes in WSNs. In this paper ,we focus on static networks. Moreover, it assumes that all sensors have identical reachability radius r . However, it is easy to adapt our method to sensors having different reachability radius. A wireless sensor networks is represented as a bidirectional graph $G(V, E)$ where V is the set of n nodes representing sensors and E is the set of m edges representing communication links. If two nodes $u, v \in V$ are neighbors, then they are linked that means distance between u and v is smaller than r . The set of neighbors for a node $u \in V$ is noted $N(u)$. anchor nodes have knowledge of their location through some other means, such as GPS or simply explicit programming. The set of anchors is noted Λ . The set of neighbor anchors for a node u is noted $N_{\Lambda}(u)(N_{\Lambda}(u) = N(u) \cap \Lambda)$ and the set of non-neighbor anchors is noted $\overline{N_{\Lambda}}(u)(\overline{N_{\Lambda}}(u) = \Lambda / N_{\Lambda}(u))$. Note that all identical nodes (anchors or others nodes) have the same capabilities (energy, processing, communication ...). The coordinate of a position of node u is noted (x_u, y_u) . P

is the set of all possible positions in a network. Our method construct the convex hull of a point cloud S_u for each node u , this convex hull is noted $conv(S_u)$. The localization modules (eg, GPS or Galileo) are expensive and consumers of energy, for this our method seeks to use the least possible anchors with the Nodes can use technology measures distances as ToA, RSSI, AoA.

So, when it receives a signal from a transmitter, a node deduces that it is located on the circle centered on the transmitter. The exact distance between two nodes u and v is noted d_{uv} . Two neighbor nodes u, v know d_{uv} (via ToA, ...). The estimated distance is noted \hat{d}_{uv} . The following section explains how to obtain these estimated distance. the set of circles built from the knowledge of anchor neighbors is noted $C_{N_{\Lambda}}$, the set of circles built from the knowledge of non-anchor neighbors is noted $C_{\overline{N_{\Lambda}}}$. δ is the distance between the estimated position $(x_{u_{estm_i}}, y_{u_{estm_i}})$ of the sensor u and the summit furthest from convex hull $Conv(S)$. Let d_{err} being the distance between the estimated position of a node and its real position, representing the position error. The node knows that d_{err} . By using a predefined threshold, if $d_{err} \leq threshold$ then the node has an estimation close to its real position. In this case the node becomes an estimated anchor and broadcasts its position.

III. RELATED WORKS

A. Anchor-based methods

Localization of nodes in WSNs can be split up into two parts: First, the process of distance estimation or measurement and second, the localization algorithm.

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 [3], [10],[11] 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 [12], ToA [13], TDoA [14] and AoA [10] : 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 in [13]. 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.

Among localization methods in wireless sensor networks, the most popular are the methods of Niculescu and Nath APS in [5], Savvides, al [7] and Savarese, al [6]. 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 [15], Langendoen and Reijers provide a detailed comparative survey for each step of these methods. The distance estimation techniques will be described in section III-B. After the distance estimation step, there are two techniques in order to calculate node position : either multilateration, described above, used by [6] et [5], or Min-Max technique, used by [7] : 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 [5] does not use a refinement process. in AT-Dist [8], This method based on the method for estimating distance Sum-Dist used by [7] and on a method based on the intersection of the disc centered by anchors nodes for each sensor that seeks its position, the intersection of these disc provided an area and the center of gravity of this area considered as the estimated position. AT-Dist method exploits the location error when the error is below certain threshold the sensor is also starting to broadcast its position estimated accompanied by localization error as the anchor. The implementation uses by this method to represent the network and areas constructed is Grid-scan described in section III-B4.

B. AT-Dist:

a. Description :

In [8], authors present an interesting localization method. In a first time, nodes determine their positions with a position error bound using anchors positions, and when this position error bound goes below a given threshold on a node, this node is considered as an estimated anchor and other nodes uses this information to improve the knowledge of their positions. Resulting localization information are provided with a position error bound, which is interesting as it can be used for geographical routing for example [15]. Simulation results show that AT-dist method performs accurate localization of the nodes when distance measurement errors are small results are

clearly less accurate in large scale . Each node represents the network by a grid. The length of a grid side is set of $0.1r$ in order to guarantee that estimation accuracy is not noticeably compromised. When a node receives an anchor position, it increments the cases in the grid that may be its position :

- if the node and the anchor are not neighbors : all cases between the two circles : one with radius equals to r and the other with radius equals to estimated distance returned by Sum-Dist.
- if the node and the anchor are neighbors : all cases on the circle having as center the anchor of radius equals to the range.
-

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

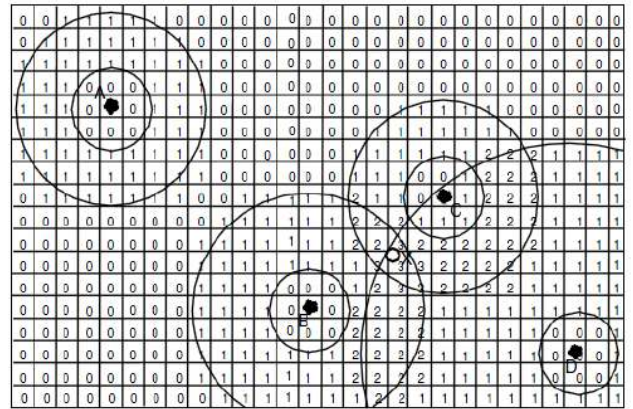


Fig. 1. Grid-scan

b. Advantages and Drawbacks :

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

IV. SCALABLE LOCALIZATION SENSOR NETWORK BASED ON GRAHAM SCAN

A. Approximation technique

a. Description :

Initially, each anchor broadcasts its position. A node can therefore be deduced the distance between each of the anchors We use the technique SumDist (Savvides et al., 2002) for estimating distances adding the distances between separated

sensor nodes of an anchor. Upon receiving the position of an anchor, a node considers the following cases:

- if it receives directly the position of the anchor, he deduces they are neighbors and therefore it located on the circle centered at the anchor or radius of a circle is r (this Circle belongs to).
- if it receives the position by an intermediate node, it concluded that it is not neighbor of the anchor and therefore it is not inside the circle of radius r centered in anchor (this Circle belongs to).

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

So, u is inside the circle C_{Au} of center A and radius \hat{d}_{Au} . Thus, the intersection of circles defines a cloud of points S_u . The center of gravity of the convex hull of this cloud $conv(S_u)$ obtained by Graham's scan in \cite{Grah} represents the estimated position of u .

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

Initialization of the algorithm:

$$S_0 = P$$

$$C_{N_{\Lambda}(u)_0} = C_{\overline{N_{\Lambda}(u)_0}} = \{\emptyset\}$$

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

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

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

$$C_{ua_i} = \{(x_i, y_i) \in P \mid (x_i - x_a)^2 + (y_i - y_a)^2 = d_{ua_i}^2\}$$

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

$$W_{u_i} = \{(x_i, y_i) \in (C_{(u)_{i-1}} \cap C_{ua_i}) \mid (x_i - x_{u_{estm_i}})^2 + (y_i - y_{u_{estm_i}})^2 \leq \hat{d}_{u_{i-1}}^2\}$$

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

$$Z_{u_i} = \{(x_i, y_i) \in S_{i-1} \mid (x_i - x_a)^2 + (y_i - y_a)^2 \leq d_{ua_i}^2\}$$

New cloud of points S_i :

$$S_i = Z_{u_i} \cup W_{u_i}, i \geq 3$$

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

$$C_{N_{\Lambda}(u)_i} = C_{ua_i} \cup C_{N_{\Lambda}(u)_{i-1}}$$

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

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

$$C_{\overline{ua_i}} = \{(x_i, y_i) \in P \mid (x_i - x_a)^2 + (y_i - y_a)^2 = \hat{d}_{ua_i}^2\}$$

$$W_{u_i} = \{(x_i, y_i) \in (C_{(u)_{i-1}} \cap C_{\overline{ua_i}}) \mid (x_i - x_{u_{estm_i}})^2 + (y_i - y_{u_{estm_i}})^2 \leq \hat{d}_{u_{i-1}}^2\}$$

$$Z_{u_i} = \{(x_i, y_i) \in S_{i-1} \mid r^2 \leq (x_i - x_a)^2 + (y_i - y_a)^2 \leq \hat{d}_{ua_i}^2\}$$

$$S_i = Z_{u_i} \cup W_{u_i}, i \geq 3$$

$$C_{\overline{N_{\Lambda}(u)_i}} = C_{\overline{ua_i}} \cup C_{\overline{N_{\Lambda}(u)_{i-1}}}$$

$$C_{(u)_i} = C_{N_{\Lambda}(u)_i} \cup C_{\overline{N_{\Lambda}(u)_i}}$$

The end for each node we will have a set of points S_u of the cloud:

$$S_u = \{p_1, p_2, p_3, \dots, p_n\}$$

Calculate the convex hull S_u based on Graham's scan:\

$$conv(S_u) = \left\{ \sum_{n=0}^n \alpha_i p_i \mid \alpha_i \geq 0, \sum_i \alpha_i = 1 \right\}$$

The new estimation error \hat{d}_{u_i} :

$$\hat{d}_{u_i} = \max_{p \in conv(S)} d(p, u_{estm_i})$$

The main design of the Slsng, which is a simple finite state chine. As shown in figure 2 a node running Slsng is in one of four states at any time: (i) Sensor not estimated, (ii) Sensor estimated, (iii) estimated Anchor, and (iv) improve the accuracy. Transitions between the states are triggered by events.

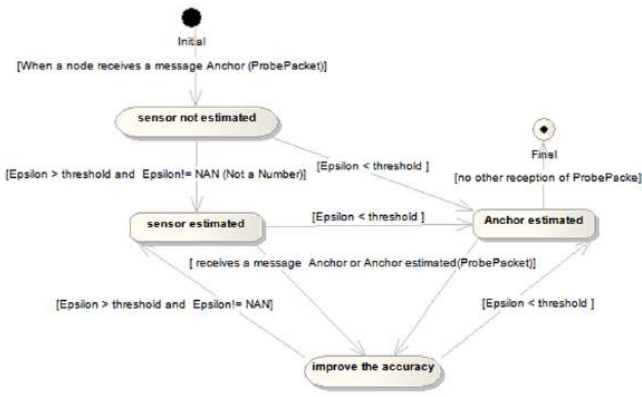


Fig. 2. State machine diagram for Sensor node not estimated

After the Slsng protocol is initiated, the node enters the Sensor not estimated state. Whenever the node receives a broadcasting *ProbePacket* packet, the node enters the Sensor not estimated state and uses this packet to estimate its position, after this stage of estimation the node switches to another state depending on the value of the estimation error found, if $epsilon < threshold$ the node enters in estimated Anchor state else it enters in Sensor estimated state. In the latter two states a node is still waiting of probpacket packet from anchor or estimated Anchor nodes to enter in improve the accuracy state and improve its accuracy. when there will be no more ProbePacket, the node switches to the state final and considered as estimated with an error of precision.

An example is illustrated in figure 3. X Receives positions of anchors A, B and C . It estimates distances $\hat{d}_{AX}, \hat{d}_{BX}, \hat{d}_{CX}$ with Sum-Dist. Since all anchors are not neighbors of X then X is not inside circles centered respectively in A, B, C with a radius equals to r but it is inside circles with radius equal to $\hat{d}_{AX}, \hat{d}_{BX}, \hat{d}_{CX}$. The intersection of these circles defines the cloud points S_X for a node X . X computes the center of gravity of the convex hull $conv(S_X)$ of this cloud and estimates its position in G_2 .

b. Pseudo-code:

The pseudo-code for the Slsng is shown in figure 4. Each anchor exact (equipped with GPS or Galileo) or estimated broadcasts its position through the control message P, and depending on number of hops traveled by the packet P we check its validity, if the number of hops is less than a certain threshold called *ThresholdHopcount* it is considered con-

firmed otherwise a packet is rejected. after we apply our method as described previously.

B. Slsng properties:

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

-First, a node knows if its estimated position is close to its real position. Let δ 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 $d_{err} \leq \delta$. By using a predefined *threshold* if $\delta \leq threshold$ 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 δ . When a node applies the approximation technique with an estimated anchor radius, it takes into account δ . Consider a sensor X calculating its position with an estimated anchor A . If they are neighbors, X trace two circles (belongs to C_{N_A}) centered in A of radius $d_{AX} \pm \delta$ and deduce that it is between these two circles. If they are not neighbors, X deduces that it is not inside the circles centered at A of radius $r - \delta$ and belongs to a circle of radius $d_{AX} + \delta$, the definitions (4),(6),(9) and (11) become :

si $a_i \in N_{\Lambda}(u)$:

$$Z_{u_i} = \{(x_i, y_i) \in S_{i-1} \mid (x_i - x_a)^2 + (y_i - y_a)^2 \leq (d_{ua_i} \pm \delta_{u_i})^2\}$$

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

$$C_{ua_i} = \{(x_i, y_i) \in P \mid (x_i - x_a)^2 + (y_i - y_a)^2 = (\hat{d}_{ua_i} \pm \delta)^2\}$$

$$Z_{u_i} = \{(x_i, y_i) \in S_{i-1} \mid (r - \delta)^2 \leq (x_i - x_a)^2 + (y_i - y_a)^2 \leq (\hat{d}_{ua_i} + \delta)^2\}$$

-Second, a node can detect if some informations are wrong. This case is illustrated in expression W_{u_i} . With its bound error δ , nodes reject the cloud points that are outside of circle centered at its estimated position and of radius δ . for example, when a node u detects a point of its cloud S_u it outside in the circle centered at u of radius δ will not take it into account. This property is defined by the expression W_{u_i} .

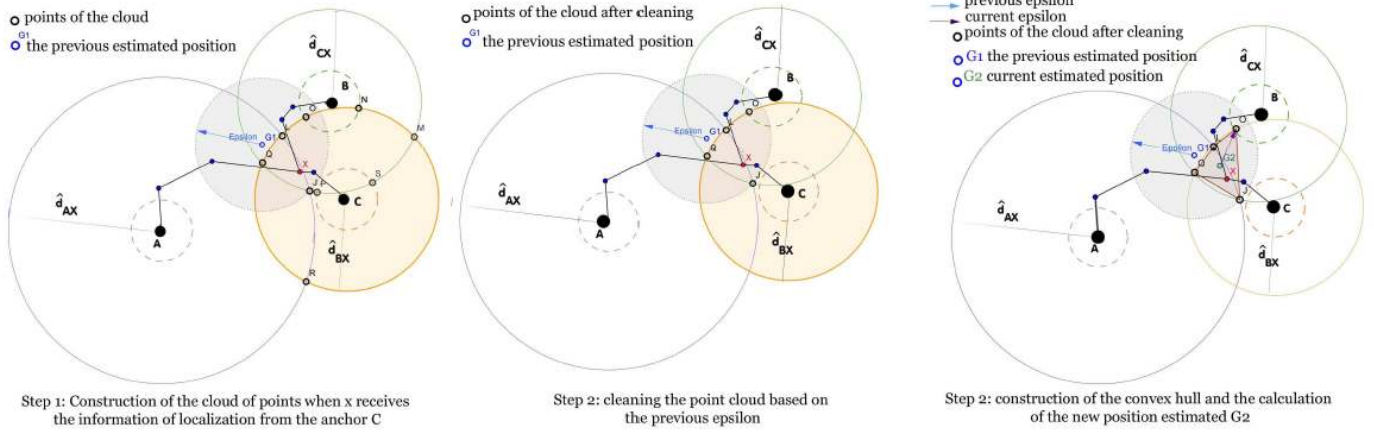


Fig. 3. Estimated position for X at the reception of localization information from C anchor

Algorithm IV.1: When a node n receives a message controle (P)

```

global List_Anchors, comment: structure contains all the received anchor
List_Pts_Intersection, comment: structure contains the set of points of the cloud  $S_u$ 
MyCercl_epsilon, comment: circle centered at the estimated position and of radius  $\epsilon_u$ 
local rcved_Anchor, comment: received anchor
cercl_epsilon_tmp, comment: temporary variable to calculate the estimated position and the estimation error
if ( $P.getHopcount() \leq \text{Threshold\_hopcount}$ )
   $P.setHopcount(P.hopcount+1)$ 
  send ( $ProbePacket(P)$ )
  if ( $Isneighbour(P)$ )
    then {
       $rcved\_Anchor.exterior\_cercl \leftarrow cercl(P.getX(), P.getY(), distance(myself, P.neighbour))$ 
       $rcved\_Anchor.interior\_cercl \leftarrow cercl(P.getX(), P.getY(), 0)$ 
    }
    else {
       $rcved\_Anchor.exterior\_cercl \leftarrow cercl(P.getX(), P.getY(), P.Dist + P.epsilon)$ 
       $rcved\_Anchor.interior\_cercl \leftarrow cercl(P.getX(), P.getY(), radius - P.epsilon)$ 
    }
  if ( $List\_Anchors.IsEmpty()$ )
    then {
       $List\_Anchors.add(rcved\_Anchor)$ 
       $List\_Pts\_Intersection \leftarrow Intersection(List\_Anchors, rcved\_Anchor)$ 
       $List\_Pts\_Intersection \leftarrow Clean\_Failed\_Pts(MyCercl\_epsilon, rcved\_Anchor)$ 
       $List\_Pts\_Intersection \leftarrow Graham\_Scan(List\_Pts\_Intersection)$ 
       $cercl\_epsilon\_tmp.centr \leftarrow Centre\_of\_gravity(List\_Pts\_Intersection)$ 
       $List\_Anchors.add(rcved\_Anchor)$ 
    }
    else {
       $cercl\_epsilon\_tmp.radius \leftarrow calc\_epsilon(cercl\_epsilon\_tmp.centr, List\_Pts\_Intersection)$ 
      if ( $cercl\_epsilon\_tmp.radius \leq MyCercl\_epsilon.radius$ )
         $MyCercl\_epsilon.radius \leftarrow cercl\_epsilon\_tmp.radius$ 
         $MyCercl\_epsilon.centr \leftarrow cercl\_epsilon\_tmp.centr$ 
        if ( $cercl\_epsilon\_tmp.radius \leq \text{Threshold\_AnchorEstimated}$ )
          then {
             $setImAnchorEstimated(true)$ 
            send ( $ProbePacket(myID, BroadcastID, 2, MyCercl\_epsilon, pktSequence+1)$ )
          }
  }

```

Fig. 4. Description of Slsng algorithm

-Third, we used the Graham's scan method instead of Grid scan method used by AT-Dist to calculate the convex hull $conv(S)$ a cloud of points with a very optimum complexity,

of order $O(n \log(n))$ with n the number of points of the cloud, which allowed us to reduce consumption of CPU time (and therefore energy), but also allowed us to optimize particularly the consumption of memory storage, focusing not on global interpretation of the network as in an algorithm of type

Grid-scan presented in III-B1 making algorithm incapable of following the size of networks when we pass a large scale, but only on points of the cloud. The improvement made allowed us to retain the properties functional Our localization technique despite the change in network size, and efficiently localize the nodes (continuously) and with a certain level of quality in different scales.

C. Structure of the control message exchanged:

Our approach Slsnj requires the exchange of Specific Information. For this, a specific control message is designed. The fields in this message, called *ProbePacket*, exchanged during the execution of the localization algorithm are shown in Figure 5, tow possible values for the packet subject are used in the algorithm: *Anchor*, *Anchor estimated*. Note that when a node broadcasts or sends a message in a wireless network, all nodes in its scope communication receive this message. The validation of a control message is limited by a threshold of validation, called *Threshold_hopcount*.

SrcId	global identifier ID of the transmitter node of the packet
DestId	global identifier ID the receiving node
NodeType	type of the transmitter node of the packet (Anchor, Anchor estimated)
MyCercle_epsilon	circle centered at the estimated position or exact position of the transmitter , and of radius ϵ (0 for transmitter Anchor)
Hopcount	the number of hops between the anchor transmitter and the sensor receiver
pktSequence	to avoid the redundantly reception of the same message
Dist	the sum of the distances intermediate between the anchor transmitter and the receiver sensor

Fig. 5. Fields of the message *ProbePacket*

V. EXPERIMENT AND RESULTS

A. Simulation environment :

Experiments were built upon the J-Sim simulator [9] dedicated to WSN simulations. It is a compositional, component-based simulation environment. It is built upon the concept of autonomous component programming model. J-Sim is developed entirely in Java. The signal attenuation due to obstacles or other factors (e.g. use of unidirectional antennas) is simulated in J-Sim. Therefore, the vicinity of a node in terms of transmission range is not necessarily spherical. Note that there several simulators in the literature such as GlomoSim[18] , OMNET++[19] , OPNET[20] , NS-2[21] . The MAC layer is considered perfect and the transmission of messages are without loss in our simulation.

In the field of localization in of wireless sensors networks and services, The scalability was analyzed as a problem of performance where enough variety of metrics were considered. These metrics are concentrated around the measurement of response time, Consumption of resources and the number of messages exchanged between nodes. The factor scale most considered in the literature is the number of nodes. This section analyzes the performance of our method slsnj following three metrics: accuracy, storage space, complexity, in order for

us show the good performance of our protocol in large networks.

In order to allow easy comparison between different scenarios, range errors as well as errors on estimated positions are normalized to the radio range. For example, 50% of position error means a distance of half the range of the radio between the real and estimated positions. The percentage of range errors is noted δ .

B. the Results :

In figure 6 when the value of confidence is equal to 3, the obtained error mean is the best. In fact, when the value of confidence is higher than 3, the voting process is very strict and nodes cannot deduce their positions. Conversely, when the value of confidence is lower than 3, the voting process assigns in some times bad positions to sensors because it uses a few number of anchor positions and some wrong informations can be used. This comment is confirmed when increases. But, it is possible that this value increases when the percentage of range errors is higher than 15. In the next experiences the value of confidence is equal to 3.

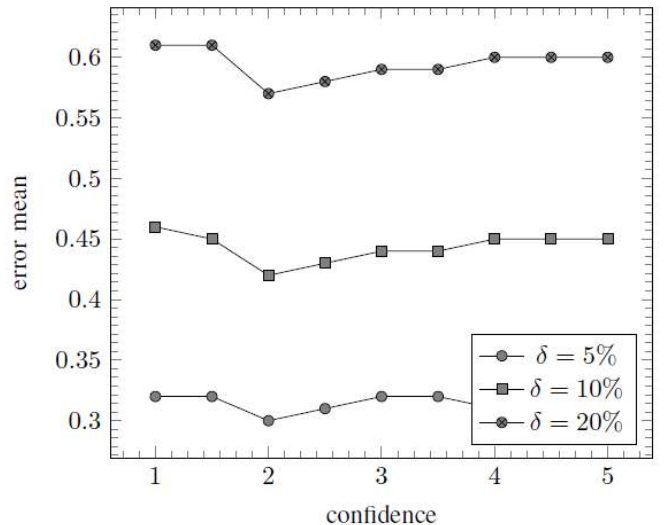


Fig. 6. *Slsnj* : impact du seuil confiance

1. The accuracy :

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

Globally, the positions determined by a localization algorithm represent a geometrical layout of the physical positions of the sensors. This layout must be compared to the ground truth, or known layout of the sensors. It is important therefore that not only the error between the estimated and real position of each

node is minimized, but also that the geometric layout determined by the algorithm matches well the original geometric layout. In order to have a unified approach for evaluate the accuracy of our technique and a solid frame for analysis of the scalability, we propose to use two metrics.

- *MAE (Mean Absolute Error):*

The simplest way to describe localization performance is to determine the residual error between the estimated and actual node positions for every node in the network, sum them and average the result. Broxton et al in [22] do this using the mean absolute error metric (MAE), which, for each of n nodes in the network, calculates the residual between the estimated nodes and actual coordinates.

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

With (x_i, y_i) the real position and (\hat{x}_i, \hat{y}_i) the estimated positions.

- *GDE (Global Distance Error):*

As discussed briefly at the start, it is important for the accuracy metric to reflect not only the positional error in terms of distance, but also in terms of the geometry of the network localization result. GDE in [23] takes the RMS error over the network of n nodes and normalizes it using the constant R . R represents average radio range, meaning the localization results are represented as a percentage of the average distance nodes can communicate over.

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

With \hat{d}_{ij} The estimated distance between i and j and d_{ij} The actual distance between i and j .

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

- The density of network and dimension of the network;
- Measurement errors δ that will take the values 0%, 5%, 10% ;
- The percentage of anchors noted α , are selected randomly among the network nodes .

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

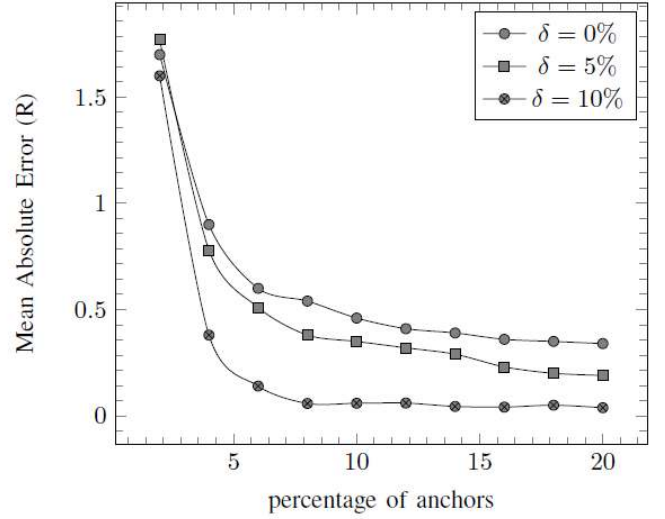


Fig. 7. Slsng rate of the average error (MAE) with a dimension $L = 300$

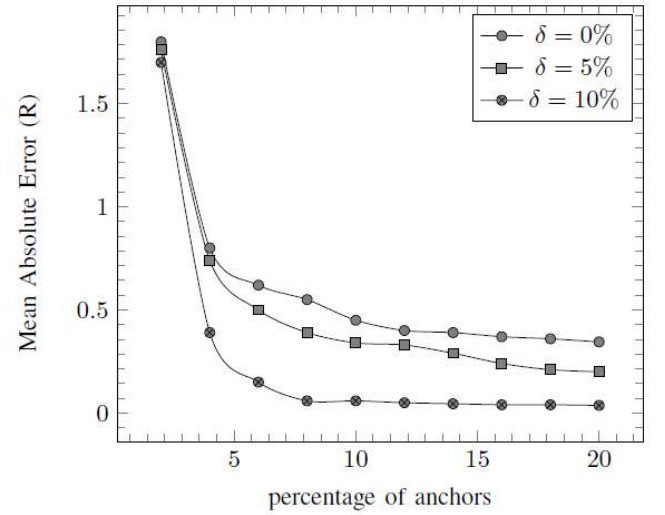


Fig. 8. AT-Dist rate of the average error (MAE) with a dimension $L = 300$

7,8 and 9 represents the performance respectively Slsng and AT-Dist in a small scale ($L \leq 400$), when range errors are introduced, the behavior of average error rate MAE of our method related to percentage of anchors. These curves indicate the accuracy of localizations when δ is equal to $\delta = \{0, 5, 10\}\%$. Without surprise, performances of Slsng decrease when range errors increase as the method AT-Dist (with $L = 200$). But, our method keeps a good estimation of positions.

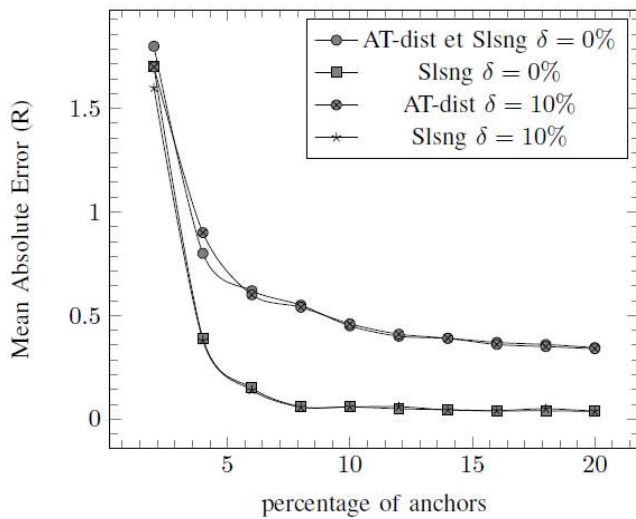


Fig. 9. Slsng and AT-Dist rate of the average error (MAE) with a dimension $L = 200$

Note also that after 10% of anchors the average error rate decreases slowly. for underline the capacity of the methods to localize sensors with precision, reference should be made to the graph of Figure 10 and 11 The graph represents the percentage of nodes located of Slsng and AT-Dist for a percentage of anchors varies from 0% \rightarrow to \rightarrow 20% without errors $\delta = 0\%$. The anchors located by GPS are not taken into account. In others words, the percentage of new exactly located nodes is only considered. For Slsng, the results are very clear and stable when we move to large scale ($L \geq 400$) with $\alpha = 20\%$: for slsng the percentage of nodes located with a position error less than 20% clearly exceeds the 86% but does not exceed 75% for AT-Dist.

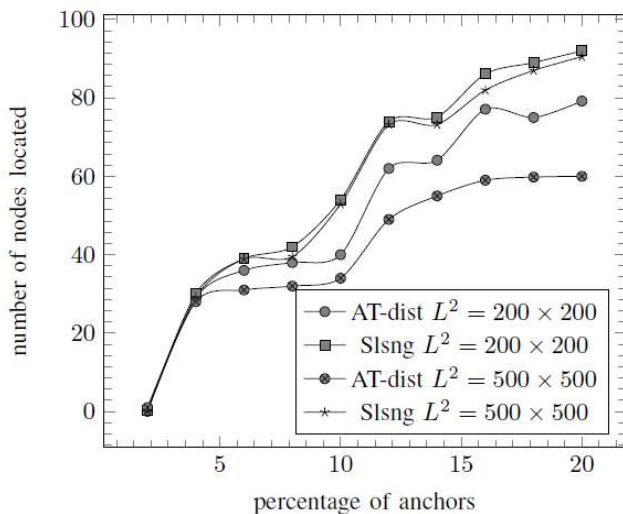


Fig. 10. Slsng and AT-Dist: Percentage of localized nodes with a position error less than 20% with $\delta = 0\%$ and $L = 200, 400$

Figure 12 shows the impact of density of nodes in small large ($L \geq 400$) on the behavior of average error rate MAE. When the density of nodes increases, the average error rate decreases. In fact, with low density, nodes do not often use rules but only the approximation technique.

Therefore, a few number of anchors (estimated or not) are added. The opposite phenomenon occurs when density of nodes increases. Note that after a density of nodes equals to 12, the behavior of average error rate is not significative.

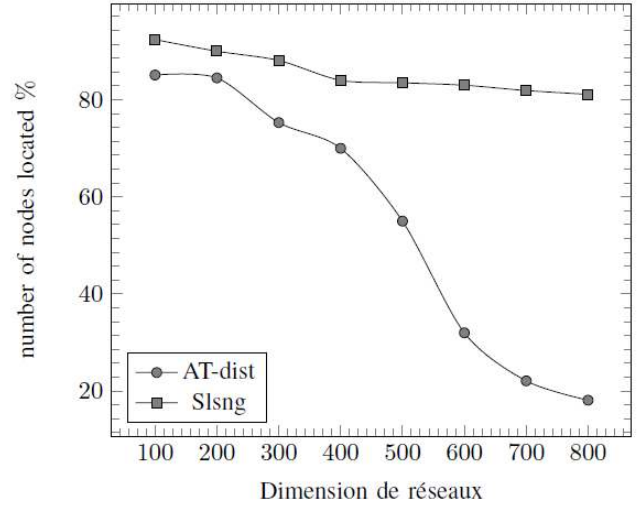


Fig. 11. Slsng and AT-Dist: Percentage of localized nodes with a position error less than 20% with $\delta = 0\%$ with L varied from 100 to 800

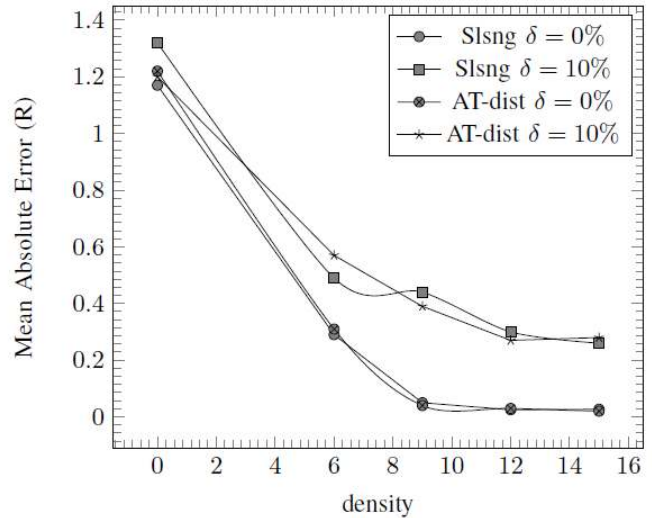


Fig. 12. Slsng et AT-Dist : impact de la densité avec une dimension $L=200$

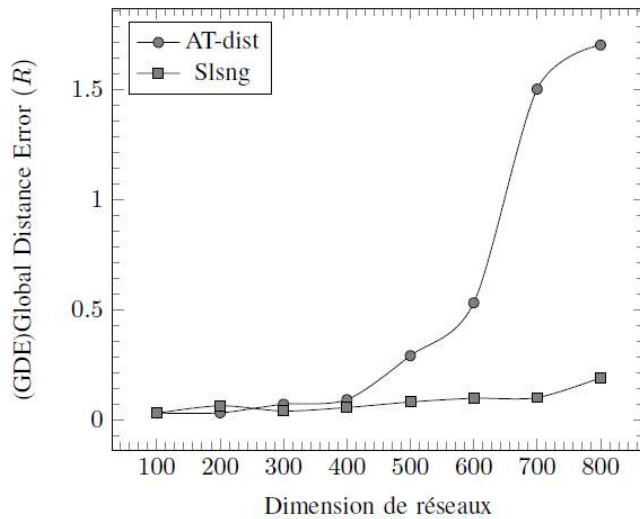


Fig. 13. *Slsng et AT-Dist:Global Distance Error with $\delta = 0\%$ and density of anchors $\alpha = 15\%$*

Figure 13 shows the impact of the dimension of network on the behavior of average error rate GDE. When the dimension of network increases, the average error GDE remained stable for Slsng and increases for AT-Dist. In fact, a large scale, the comportment of our technique based on graham's scan remains stable and capable of operating, but the method AT-Dist based on scan-line stabilizes do it that are incapable of representing the network in its memory by the method Scan-line.

2. the complexity:

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

3. the consumption of memory:

We also measure the amount of memory allocated before the end of the simulation. The memory usage before the end of the simulation represents the amount of memory allocated to complete the 300 s simulation. As shown in figure 14 and figure 15, Slsng use less memory than AT-Dist in large-scal.

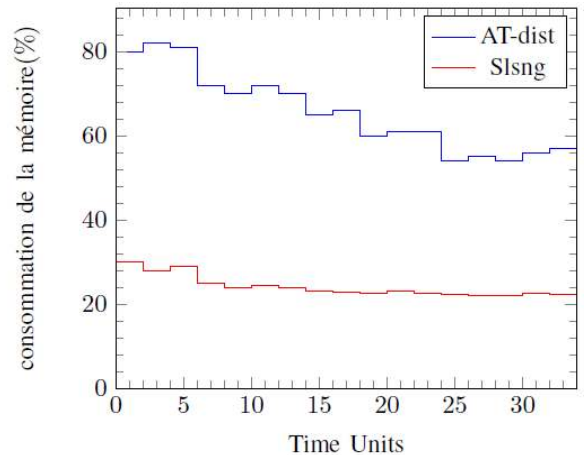


Fig. 14. *Slsng et AT-Dist: rate of consumption of the memory with $\delta = 0\%$ and density of anchors $\alpha = 20\%$ and dimension $L=400$*

This demonstrates that the data structures are used in a more scalable manner in Slsng to represent different classes and their interaction in the WSN framework. and the Graham's method used by our technique Slsng to reduce the memory used.

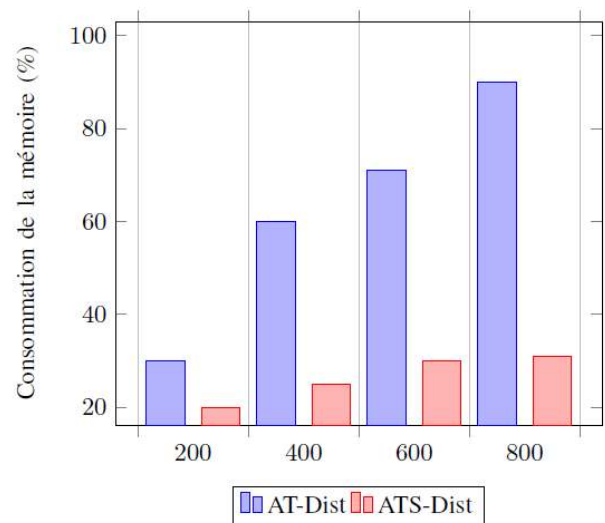


Fig. 15. *Slsng and AT-Dist: rate of consumption of the memory with $\delta = 0\%$ and density of anchors $\alpha = 20\%$ and L varied from 100 to 800*

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

Figures 16 and 17 show the evolution the location accuracy convergence. Depending on the size of networks. In first graph, the convergence time increases linearly with the dimension, and in the second graph represents the evolution of con-

vergence time that is the time when the Metric MAE is stabilized over time. Convergence time with our method in a dimension 400×400 corresponds to 65s and 190s with at-dist. In fact, the main particularity of our protocol is that the

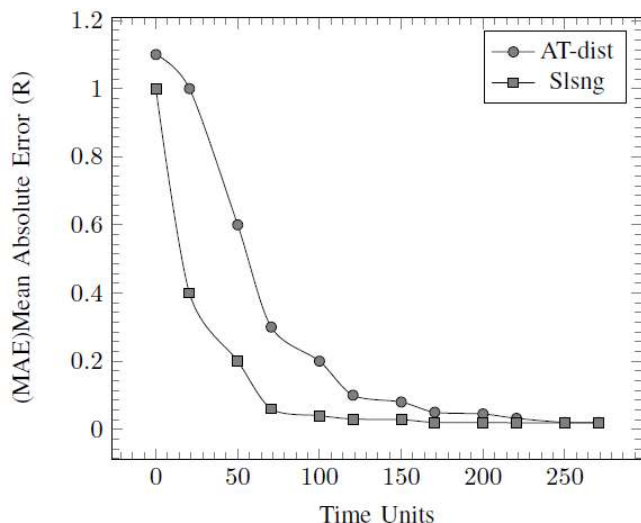


Fig. 16. Accuracy comparison avec $\delta = 0\%$ et densité d'ancres $\alpha = 20\%$ et une dimension $L=200$

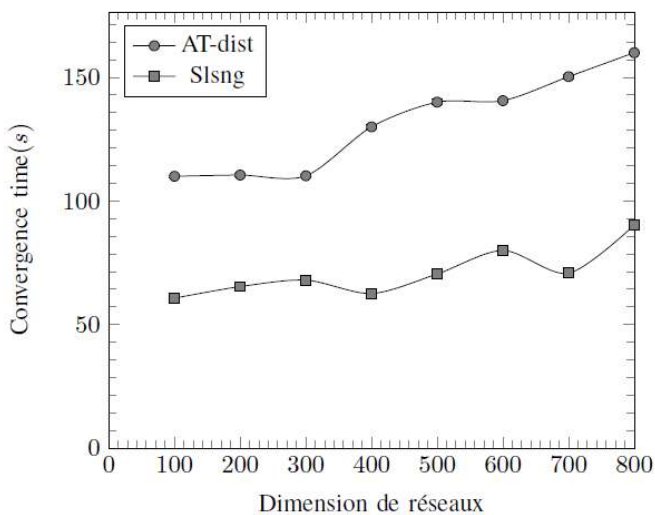


Fig. 17. Slsng et AT-Dist: Temps de convergence avec $\delta = 0\%$ et densité d'ancres $\alpha = 20\%$

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

VI. CONCLUSION

This paper considers an improved method of AT-Dist called Slsng in order to locate sensors with high accuracy and to make it scalable and rapidly convergent and less resource consumption (CPU and memory). It proposes a approximation technique Slsng based on Graham's scan in order to estimate the position of nodes. Each node restricts to determining the convex hull of a set of sensors used instead of the Grid-Scan

method where it can be localized. Slsng presents three important advantages: first, this technique eliminates some wrong propagated informations. These wrong informations are due to range errors or attackers who have the control of sensors. Second, a node knows if its estimated position is close to its real position and in this case, it becomes an estimated anchor.

Third, Graham's scan allowed us to reduce the consumption of CPU time (and therefore energy), But also allowed us to optimize including consumption of the memory, focusing not on the overall interpretation of network such as a type algorithm scan-line but only on points of convex hull. Consequently, We get to keep the functional properties of our localization technique despite change in network size with a minimum convergence time. Thus, simulations show the efficiency of our method in comparison to AT-Dist method in [8] take into consideration the large-scale networks, the requirements in memory and the position convergence times. 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 in large-scale but does not take into account the mobility of sensors or the energy consumption. The optimization of these two criterions 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. Future works will consist in analyze these criterions in Slsng either by using these methods or by a novel method adapted to Slsng. Finally, this paper assumes that sensors have none informations related to network environment, especially informations about error measures. It proposes some ways to improve Slsng when a bound can be calculated for measure errors. But, an in-depth analyze should to be achieved.

REFERENCES

- [1] Localization algorithms and strategies for wireless sensor networks, Thirteenth ed. Information Science Reference (an imprint of IGI Global), 2009, pp. 2–30.
- [2] S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," Proceedings of the IEEE, vol. 91, no. 8, pp. 1247–1256, 2003
- [3] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher, "Range-free localization schemes in large scale sensor networks," in Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, 2003.
- [4] M. Li and Y. Liu, "Rendered path: range-free localization in anisotropic sensor networks with holes," in Proceedings of the 13th annual ACM international conference on Mobile computing and networking, ser. MobiCom '07. New York, NY, USA: ACM, 2007, pp. 51–62.
- [5] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," in IN GLOBECOM, 2001, pp. 2926–2931.
- [6] C. Savarese, J. M. Rabaey, and K. Langendoen, "Robust positioning algorithms for distributed ad-hoc wireless sensor networks," in

Proceedings of the General Track of the annual conference on USENIX Annual Technical Conference. Berkeley, CA, USA: USENIX Association, 2002, pp. 317–327.

[7] A. Savvides, H. Park, and M. B. Srivastava, “The bits and flops of the n-hop multilateration primitive for node localization problems,” in Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, ser. WSNA '02. New York, NY, USA: ACM, 2002, pp. 112–121.

[8] Clément Saad, Abderrahim Benslimane & Jean-Claude König. AT-Dist : A Distributed Method For Localization With High Accuracy in Sensor Networks. Special Issue on "Wireless Ad Hoc and Sensor Networks" in the international journal *Studia Informatica Universalis* (To Appear), 2007.

[9] R. Graham, “An efficient algorithm for determining the convex hull of a finite planar set,” *Information Processing Letters*, pp. 132–133.

[10] D. Niculescu, “Ad hoc positioning system (APS) using aoa,” *IEEE INFOCOM 2003 Twentysecond Annual Joint Conference of the IEEE Computer and Communications Societies IEEE Cat No03CH37428*, vol. 00, no. C, pp. 1734–1743, 2003.

[11] N. Bulusu, J. Heidemann, and D. Estrin, “Gps-less low cost outdoor localization for very small devices,” *IEEE Personal Communications Magazine*, vol. 7, no. 5, pp. 28–34, October 2000.

[12] A. Savvides, C.-C. Han, and M. B. Srivastava, “Dynamic fine-grained localization in ad-hoc networks of sensors,” in Proceedings of the 7th annual international conference on Mobile computing and networking, ser. *MobiCom '01*. New York, NY, USA: ACM, 2001, pp. 166–179.

[13] K. Langendoen and N. Reijers, “Distributed localization in wireless sensor networks: a quantitative comparison,” *Comput.Netw.*, vol. 43, pp. 499–518, November 2003.

[14] J. Champ and V. Boudet, “ADNL: Accurate Distributed Node Localization Algorithm in Wireless Sensor Networks,” in *European Wireless 2010*, Italy, Apr. 2010, p. 8.

[15] C. Saad, A. Benslimane, J. Champ, and J.-C. König, “Ellipse routing: A geographic routing protocol for mobile sensor networks with uncertain positions,” in *GLOBECOM, 2008*, pp. 78–82.

[16] “About glomosim,” <http://pcl.cs.ucla.edu/projects/glomosim/>, cited July 2011.

[17] “Omnet++ community site,” <http://www.omnetpp.org/>, cited July 2011.

[18] “Opnet technologies,” <http://www.opnet.com/>, cited July 2011.

[19] “The network simulator,” <http://www.isi.edu/nsnam/ns/>, cited July 2011.

[20] M. Broxton, J. Lifton, and J. A. Paradiso, “Localization on the pushpin computing sensor network using spectral graph drawing and mesh relaxation,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 10, pp. 1–12, January 2006.

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