A2L: Angle to Landmarks Based Method Positioning for Wireless Sensor Networks

Mustapha Boushaba¹, Abderrahim Benslimane², Abdelhakim Hafid³

¹Network Research Laboratory, Université de Montréal
CP 6128 succ Centre-Ville
Montréal, QC, H3C 3J7, Canada
{boushamu, ahafid}@iro.unmonreal.ca

²Laboratoire Informatique Université d’Avignon
339 chemin des Meinajaries BP 1228-84911
Avignon Cedex 9, France
benslimane@ lia.univ-avignon.fr

Abstract— Thanks to recent technological progress, autonomous wireless sensor networks have experienced considerable development. Currently, they are used in the areas of health care, environment, military etc. For a number of sensor-based applications, the knowledge of the positions of sensors is required or, at least, preferable. In this paper, we propose a new method to locate a large number of nodes in wireless sensor networks where only a subset of them are landmarks (i.e., know their positions). Our method is AOA-based (Angle Of Arrival) and it is called A2L (Angle to Landmark). Compared, via simulations, to previous methods such as APS and AHLoS, A2L considerably increases the number of located nodes with accurate precision while using a smaller node degree.

Keywords- wireless sensor networks, localization, Angle of Arrival, trilateration.

I. INTRODUCTION

Sensor networks are becoming a standard technology in wireless communications. The development of these networks involves many different research areas, such as communication, sensing and computing. This leads to smart disposable micro-sensors that can be deployed anywhere regardless of geographic limitations. Therefore, such sensors may be used widely in military, national security, environment monitoring, traffic surveillance, medical domains. These networks usually use a large number of tiny sensors. Sensor nodes are low-cost, low-power, and communicate in short distances. Together, they communicate in wireless mode and collaborate to provide information for common missions. A sensor node has generally embedded processing capabilities and potentially has a number of sensors dedicated to sensing different features, such as acoustic, seismic, infrared (IR), and magnetic elements; it can also operate as an imager/micro-radar device.

For specific applications, some nodes must know theirs physical positions to determine where events occur. Sensor node location can be found by using extra hardware such as GPS (Global Positioning System); however, equipping each sensor node with GPS is very expensive in terms of energy and cost. A more acceptable solution would require only a subset of nodes equipped with GPS; the positions of their neighbors are computed using techniques, such as trilateration and triangulation. It has been proven that the trilateration cannot be used alone for node localization within a network with a very small density of GPS nodes. Most techniques use recursive algorithms combining triangulation with trilateration or multilateration.

Some of the GPS-based methods use estimated distance between pairs of neighbors. These methods are called Range-Based Localization Schemes (in contrast with Range-Free Localization Schemes [1, 2, 3]). The most popular methods are RSSI (Received Signal Strength Indicator), ToA/TDoA (Time of arrival / Time difference of arrival) and AOA (Angle of arrival). In RSSI, nodes measure the power of the received signals and thus, can calculate the effective propagation loss. Theoretical or empirical models are used to translate this loss into distance. In ToA/TDoA, nodes directly translate the propagation time into distance if the signal propagation speed is known. The most basic localization system using ToA techniques is GPS [4]. In AOA, nodes estimate the angle at which signals are received and use simple geometric relationships to calculate their positions. The accuracy of these measurements is closely related to the network environment; thus, the positions computed by the nodes may contain errors.

In this paper, we propose a localization algorithm called Angle to Landmark (A2L). This algorithm is based on some existing techniques, such as Angle of Arrival (AOA) and distance estimation for computing nodes positions. Our technique is low cost and does not require expensive infrastructure and any compass. With a fraction of the nodes as landmarks, A2L allows each node in the network to calculate its own position. Compared to existing methods, such as APS and AHLoS, A2L considerably increases the number of located nodes with better precision.

The paper is organized as follows: Section II presents related work. Section III describes the proposed A2L algorithm. Section IV evaluates the performance of A2L using simulations. Section V concludes the paper and presents future work.
II. RELATED WORK

A large number of existing techniques attempt to solve the localization problem. A detailed survey can be found in [5]. We identify four categories:

- Infrastructure-based systems: They require infrastructures like RADAR [6] or Cricket [7].
- Robot-based systems: They use robots to locate nodes [8].
- GPS-free methods: They do not require landmarks to locate nodes. The authors in [9] propose a method that builds a virtual system of coordinates, and the nodes compute their positions in this virtual system.
- GPS-based methods: They use the positions of landmarks to determine estimated positions of non-landmark nodes.

The authors, in [10, 14, 2], use distance and angle information to compute a node’s position. APS [12] uses the angle-of-arrival technique (AOA) for localization; all nodes have the capability to compute orientation and position. In APS, nodes iteratively obtain position and orientation information starting from landmark nodes. When a non-positioned node knows at least three landmarks, it can apply the trilateration technique for computing its position and its orientation to the landmarks. This information is broadcasted to neighbors for subsequent iterations. To compute its position, a node X needs to have at least two neighbors, Y and Z, which have estimates - angles or ranges for landmark L (Y and Z should be neighbors too).

Let us consider the topology shown in Fig.1; nodes A, B, C are landmarks and D, E are non-positioned nodes (i.e., do not know theirs positions). With APS, the landmarks A, B and C start the localization mechanism. In a recursive way, nodes E and D try to compute their positions and their orientations towards landmarks. Node E needs to have two neighbors that know the coordinates or the orientation towards landmark C; this is not the case in the topology shown in Fig. 1. Thus, no quadrilateral can be formed between node D and the landmark C; the same applies for localizing node E. In this topology, APS cannot locate nodes D and E.

Other techniques are hybrid ones, such as the technique described in [15] where the authors combine APS with two other existing localization methods, namely MDS (Multidimensional Scanning) and SDP (Semidefinite Programming). MDS calculates positions using a set of distances whereas SDP is a relaxation based method.

In [11] the authors propose an AHLoS system that produces high quality positions. It uses ultrasound and RF techniques to deal with the ranging problem. To estimate node locations, AHLoS uses a set of nodes initially configured as landmarks and defines several types of multilateration: atomic, iterative, and collaborative. Atomic multilateration can be applied as a basic multilateration when a node has enough landmarks neighbors. Once at least three distances to three landmarks are known, a node may compute its own location. When a node estimates its position, it becomes a landmark. Therefore, an iterative multilateration continues until no more nodes can be localized. AHLoS does not guarantee that all the nodes will be able to compute theirs positions. In a collaborative fashion, nodes try to estimate their locations using beacons at two hops away.

The main drawback of AHLoS is that it requires high percentage of beacons to achieve high percentage of located nodes. For example, to resolve 90 percent of unknown nodes with an average degree of 6.28, AHLoS requires a density of 45% landmarks.

III. ANGLE TO LANDMARK ALGORITHM

In this Section, we present a new method that allows localizing a high percentage of nodes in wireless sensor networks by using a minimum number of landmarks. It is AOA-based where each node computes the difference between two AOAs: incoming angle from landmark neighbors and from non-positioned neighbors (Fig. 2). These angles are used to compute distances between nodes and landmarks within two hops. We assume that each node (e.g., Medusa node [13]) is able to measure its distance from its immediate neighbors and has an antenna array enabling it to compute the incoming signal angles (AOA).

In Fig. 2, let us suppose that A, D, F and G are the landmarks. B, C, E and N are the non-positioned nodes. Plain lines are the links between two immediate neighbors and dashed line are the links between nodes at two hops away.
As an example, let \( \theta \) be the incoming angle from node \( N \) and \( \theta_F \) the incoming angle from \( F \). When \( N \) knows the angle \( \theta = |\theta_N - \theta_E| \), it becomes easy to compute \( d_{NE} \) (i.e., distance between \( N \) and \( E \)) by using the equation:

\[
    d_{NF} = d_{NE} + d_{EF} - 2 \cdot d_{NE} \cdot d_{EF} \cdot \cos(\theta)
\]

Let us assume a 2-D space, \((x,y)\) is the unknown position, and \( (x_i,y_i) \) are the coordinates of the \( i^{th} \) landmark for \( i = 1, \ldots, n \). The coordinates and the estimated distance \((d_i)\), distance between the \( i^{th} \) landmark and \((x,y)\), are related by the following set of equations:

\[
\begin{bmatrix}
    (x_1-x_n)^2 + (y_1-y_n)^2 \\
    (x_2-x_n)^2 + (y_2-y_n)^2 \\
    \vdots \\
    (x_n-x_n)^2 + (y_n-y_n)^2
\end{bmatrix} = \begin{bmatrix}
    d_1^2 \\
    d_2^2 \\
    \vdots \\
    d_n^2
\end{bmatrix}
\]

To resolve this set of equations, we transform (2) into a linear system of equations by subtracting the \( n^{th} \) equation (the last line in (2)) from each other equation (lines 1 to \( n-1 \) in (2)). The linear system is written in the form \( AX = b \), where

\[
X = \begin{bmatrix} x \\ y \end{bmatrix}
A = \begin{bmatrix}
    2(x_1-x_n) & 2(y_1-y_n) \\
    2(x_2-x_n) & 2(y_2-y_n) \\
    \vdots \\
    2(x_n-x_n) & 2(y_n-y_n)
\end{bmatrix}
\]

\[
b = \begin{bmatrix}
    x_1^2 - x_n^2 + y_1^2 - y_n^2 + d_n^2 - d_1^2 \\
    x_2^2 - x_n^2 + y_2^2 - y_n^2 + d_n^2 - d_2^2 \\
    \vdots \\
    x_n^2 - x_n^2 + y_n^2 - y_n^2 + d_n^2 - d_{n-1}^2
\end{bmatrix}
\]

When range \((d_i)\) in (2) measurements are noisy, resolving different equations (lines in (2)) would not yield the same results. To solve this, we use a least-squares solution (4) which is a technique borrowed from linear algebra that is often used in applications that consist of over-determined systems with noisy measurements.

\[
    X = (A^T A)^{-1} A^T b
\]

**A. Message exchanges by A2L nodes**

We consider an adhoc network \( R \) modeled by a bidirectional graph \( G = (NR, E) \), where \( NR \) represents the set of sensor nodes and \( E \) the set of links between nodes. A link between two nodes exists if each node is within the transmission range of the other. We classify nodes into two sets: (1) \( NL \): the set of landmarks (i.e., know their positions using GPS for example); and (2) \( NnL \): the set of non-positioned nodes. All nodes in \( NL \) and \( NnL \) are randomly placed in a geographic area using a uniform distribution.

The goal of our proposed protocol is to locate a maximum number of nodes in \( NnL \) by using a minimum of landmarks. Our protocol requires that nodes exchange two messages called INIT and POSITION:

- **INIT message**
  It is broadcasted once by \( NL \) nodes (landmarks) to their immediate neighbors (one hop). The INIT message structure is defined by two fields \(<idL, CoordL>\), where \( idL \) is the identifier of the sending node and \( CoordL \) its coordinates \((x,y)\).

- **POSITION Message**
  When the message \( POSITION \) is broadcasted, a receiver node \( K \) can extract information of interest from A2L records; this will help \( K \) (if non-positioned node) to locate itself. This message is broadcast by each node \( I \) in \( NnL \), having at least one neighboring landmark \( J \) and another neighboring node \( K \) which is not a landmark. When node \( K \) receives this message, it can compute its distance towards landmark \( J \) by applying the triangulation mechanism using equation (2). The format of the message \( POSITION \) is defined as:

\[
    <idS, CoordS, A2L_I, A2L_J, ..., A2L_n>
\]

where (a) \( idS \) is the identifier of the node which broadcasts the \( POSITION \) message; (b) \( CoordS \) are the coordinates \((x,y)\) of the sender node broadcasting the \( POSITION \) message; and (c) \( A2L_k \) is defined as \(<idL_k, idL_J, CoordL, DistL, Angle>\), where \( idL_I \) is the identifier of landmark \( J \), \( idL_k \) is the identifier of node \( K \) to which the message A2L is destined, \( CoordL \) are coordinates \((x,y)\) of landmark \( J \), and \( Angle \) is the angle between landmark \( J \), transmitter \( I \) and the receiver \( K \) of A2L message. This angle will be computed by node \( I \); it is the difference between two AOAs (AOA from nodes \( J \) and \( K \)).

**B. A2L algorithm**

Initially, every landmark initializes A2L positioning algorithm by broadcasting INIT. For each node, the MAC Layer can provide information for building Neighbors table, called \( TN \), shown in Fig. 3. For each node \( I \), each entry of \( TN \) includes: (1) \( id \): node’s identifier from incoming signal; (2) \( AOA \): incoming angle from node \( id \); and (3) \( Distance \): the distance between node \( id \) and node \( I \).

<table>
<thead>
<tr>
<th>id</th>
<th>AOA</th>
<th>Distance</th>
</tr>
</thead>
</table>

Fig 3. Neighbors table structure

When a landmark’s neighbor \( I \) receives INIT messages, it updates its landmarks table, called \( TL \) (Fig. 4), and tries to resolve the corresponding trilateration system by using equation (4). Node \( I \) must have at least three non-aligned landmarks in its \( TL \) table. For each node \( I \), each entry of \( TL \) includes: (1) \( idL \): the landmark identifier; (2) \( Coord \): coordinates \((x,y)\) of landmark \( idL \); (3) \( Distance \): the distance between landmark \( idL \) and node \( I \); it is retrieved from \( TN \) or computed by triangulation; and (4) \( nextHop \): the sender of the message \( POSITION \). If its value is equal to \( idL \), then the landmark \( idL \) is neighbor; otherwise, they are two hops away.

Node \( I \) builds a list of A2L by combining the \( TL \) and \( TN \) records; the Angle in each element of the list corresponds to the difference between the AOA from landmark \( J \) and the AOA from node \( K \). This information allows node \( I \) to build the \( POSITION \) message to be broadcasted toward its neighbors. As an example, let us consider node \( I \) with identifier equal to 3, that maintains two tables: (1) \( TN \) with 3 entries <4, 2.3, 10; 5, 1.1, 12; 6, 3.2, 10>; and (2) \( TL \) with one entry <5, (850,200), 12, 5>. The information in \( TL \) means that node 5 is a landmark...
and it is one hop away from node 3. The message POSITION, to be broadcasted by 3, will include two A2L fields:

\[ A2L_d = \langle 5, 4, (850, 200), 12, 1.2 \rangle \]
\[ A2L_o = \langle 5, 6, (850, 200), 12, 2.1 \rangle \]

Thus, POSITION = \langle 3, A2L_d, A2L_o \rangle. In this example, the field CoordS is null; this means that node 3 (I) cannot be localized; it broadcasts its message POSITION to help nodes 4 and 6 to compute their location.

<table>
<thead>
<tr>
<th>idL</th>
<th>Coord</th>
<th>distance</th>
<th>nextHop</th>
</tr>
</thead>
</table>

Fig 4. Landmarks table structure

Upon receipt of a POSITION message, a non-positioned node J checks whether the message contains A2L; if the response is yes, it applies triangulation to compute the distances towards other landmarks at two hop neighbors and updates its table TL. Node J consults its table TL to compute its position. It must have at least three landmarks from its immediate neighbors. Therefore, to compute its position, J takes into consideration landmarks at one hop then landmarks at two hops. This technique allows node J to compute its position with a better degree of accuracy by reducing errors caused by AOA measurement. If the least-squares system is resolved, node J becomes a landmark and notifies its neighbors by broadcasting the corresponding POSITION message. Otherwise, it simply broadcasts POSITION messages to help other nodes to be localized. The coordinates \((x, y)\) sent by node J will help nodes that are \(n (n>2)\) hops away computing their positions.

Algorithm 1. Angle to Landmark Algorithm: process when node i receives a message POSITION.

Input: message POSITION
Output: the coordinates of node i and the message POSITION
Variables:
- nb2L is the number of landmarks at two hops from node i.
- nbl is the number of landmarks at one hop (immediate neighbors of node i).
- TL table is a landmarks table
Functions:
- Receive (POSITION): it is used to check the A2L fields in the message POSITION which are intended for node i and updates the landmarks table (TL) that node i maintains.
- Positioning (TL): it is used to compute the node’s position by applying the least-squares technique and builds message POSITION.
- Broadcast (POSITION): it broadcasts the message POSITION to the neighbors of node i.

Algorithm 1: Angle to Landmark Algorithm

1. For \((i \in NnL)\) {
2. Receive (POSITION);
3. If \((nb2L + nbl \geq 3)\) {
4. \(POSITION = Positioning(TL)\);
5. Broadcast (POSITION);)
}

Let us consider the topology shown in Fig 2 to describe the execution of the proposed algorithm (Algorithm 1). Initially, A, F, D, and G broadcast their messages INIT. None of the nodes B, C, and E is located after the first iteration. Node E builds and broadcasts the message POSITION including: (a) angle \(\theta\) (where \(\theta = \theta_s - \theta_r\)); (b) position F and (c) distance \(d_{EF}\). At the same time nodes B, C, and N compute the angles to landmarks and broadcast their values with landmarks coordinates and distances (i.e., message POSITION). By receiving the messages POSITION, node N computes distances \(d_{NS}\), \(d_{NM}\) and \(d_{NF}\). Updates its TL and applies trilateration to compute its position. The TTL (Time To Live) value in this case is equal 2. When N is localized, it broadcasts its message POSITION containing its position and A2L fields; nodes B, C, and E can use this message to be located. Thus, all nodes are located with TTL maximum value equal to 3.

IV. EXPERIMENTAL RESULTS

In order to evaluate the performance of A2L, we developed our own JAVA-based simulator. We assume that all messages, broadcasted by nodes during simulation, are reliably delivered to their neighbors. We generate many random sensor network topologies according to the number of nodes and the number of landmarks; we use a square area where nodes are randomly placed using a uniform distribution. Landmarks are selected randomly and nodes’ degrees (average number of neighbors) are controlled by the specification of their radio range. We assume that each node is equipped with an AVR microcontroller [13]. For our simulation, we set the radio transmission power to 0.24mW. The simulation results represent the average of 100 executions.

We studied the effect of TTL and the landmark rate on the rate of nodes being positioned (percentage of non landmarks able to compute their positions) and the corresponding energy consumption. Our results are compared with APS using AOA and AHLoS respectively. We believe that it is preferable to limit the TTL value to 2 for better accuracy localization as localization errors increase with TTL. However, our simulations cover a wide range of TTL values.

In the first set of simulations (Fig 5), we consider a scenario with 300 nodes and 10% of landmarks. We execute A2L and APS algorithms for various values of TTL (from 1 to 10) and various node degrees (4.2, 6.14, 10.27) with TTL value equal to 1; only the nodes with at least three landmarks can apply the trilateration technique to compute theirs positions. With TTL value equal to 6, A2L improves, compared to APS, the localization rate by 16% in the case the node degree is equal to 4.6, and by 21%, in the case the node degree is equal to 6.14. In the case of a large node degree, such as 10.27, A2L locates 12% (resp. 3%) more than APS with TTL equal to 2 (resp. 6). Further the localization improvement, A2L converges faster than APS; indeed, A2L locates all the nodes with TTL value equal to 6 while APS requires TTL value equal to 7. The superiority of A2L is laid in the angles that a node forms with its neighbors and a distant landmark; knowing these angles, even if a node doesn’t have any immediate landmark neighbor, it can use its two-hop landmark neighbors and apply trilateration to compute its position; this is not the case for APS. Hence, A2L is more efficient than APS in uniform topologies with small or large node degree.
In the second set of simulations, we set the TTL value to 2 and vary the landmarks rate in a 300-nodes topology. The radius is set to 14. In this scenario, the average node degree is 6.14. Fig. 6 shows that A2L requires only 45% of landmarks (i.e., 45% of the network nodes are landmarks) to locate 63% of non-localized nodes, whereas APS requires more than 70% of landmarks to achieve the same goal. APS’s problem is that it requires that each node should form a quadrilateral with two neighbors and a landmark. This situation usually occurs in networks with high node degree. We conclude that A2L requires fewer landmarks than APS (using AOA) to locate the same amount of nodes.

In the third set of simulations we study the relationship between the localization rate, the amount of landmarks required and the average node degree.

Fig. 7 shows that A2L, for a 200-nodes network (average node degree is equal to 5.36), uses only 15% of landmarks to localize 98% of the network’s nodes. These results show the superiority of A2L over AHLoS [12], which requires an average node degree equal to 6.28 to localize 90% of the nodes with 45% of landmarks.

In the fourth set of simulations, we study the relationship between the amount of traffic generated - during the localization process - by the network nodes and other attributes, of interest, including network size, energy, etc.

Fig. 8 shows the variation of the average number of transmitted bytes with the network size; the number of transmitted bytes is computed by summing the sizes of all INIT and POSITION messages generated/broadcasted by landmarks and non landmarks to locate the maximum number of nodes. Fig. 9 shows the variation of the energy consumption with the network size; the transmission power of the network’s (Medusa) nodes is set to 0.24mW. The slight energy variations between the network’s sizes are explained by the increase of the node degree and the fast convergence of the nodes toward the greatest rate of localization.

Fig. 10 shows the variation of the amount of traffic generated during the localization process with the percentage of landmarks in the network. The Figure shows that the number of transmitted bytes is inversely proportional to the number of Landmarks. This can be easily explained by the fact that increasing the number of landmarks in the network speeds up the convergence with smaller values of TTL. Indeed, in this case, most of the exchanged messages are of type INIT (rather than POSITION) that has far smaller size than POSITION. Comparing A2L with AHLoS approaches, A2L is still positioned between the distributed and the centralized approaches in term of energy consumption by radio transmission.

Fig. 11 shows the variation of the amount of traffic generated during the localization process with the value of TTL and the percentage of localized nodes; the results are as expected. For smaller values of TTL, the amount of generated traffic is smaller but the percentage of localized nodes is smaller too. For bigger TTL values, the amount of generated traffic is bigger and the percentage of localized nodes is bigger as well.
In the five set of simulations, like in [2, 10], a Gaussian distribution error is applied to AOA (with standard deviation as a parameter) and to distance measurement (with standard deviation and the radio range as parameters). Errors are normalized using the “real” positions of the nodes and the radio range (i.e., error = [Euclidian distance between the estimated position and the “real” position)/(radio range)).

Since A2L uses the Least-Squares technique, we need to control the consistency of the resolution of the equations (4). Thus, we compute the residue [16]:

\[
\text{Residue} = \frac{1}{n} \sum_{i=1}^{n} \sqrt{(x_i - \hat{x})^2 + (y_i - \hat{y})^2 - d_i}
\]

(5)

Where \((\hat{x}, \hat{y})\) is the estimated position. A large residue means that the set of equations used in the least-square is inconsistent. For achieving a high accuracy, we use a small threshold (i.e., if the residue exceeds the threshold, the estimated position is rejected) and we increase the TTL value.

We consider a network of 100 nodes in square area of 100x100; we set the value of the Gaussian distribution parameter to 5% for AOA and 5% and 14 for distances; the rate of landmarks to set to 10%. A2L presents a very good performance compared to APS (Fig. 12). A2L allows locating up to 75% of the nodes with 10% position error contrary to APS which locates 36% using DV-Distance [2] technique and 8% using AOA technique. A2L locates 99% of the nodes with a position error lower than 20%. APS, using DV-Distance, locates 99% of the nodes with a position error lower than 20%. APS which locates 36% using DV-Distance techniques, such as APS and AHLoS, A2L considerably increases the number of located nodes with better precision while using a smaller node degree and fewer landmarks.

Currently, we are finalizing our study of positioning errors introduced by A2L and techniques to minimize them.

ACKNOWLEDGMENT

We would like to thank Dragos Niculescu and Badri Nath for providing us the APS code.

REFERENCES