ADNL: ACCURATE DISTRIBUTED NODE LOCALIZATION ALGORITHM IN WIRELESS SENSOR NETWORKS

Julien Champ, Vincent Boudet
LIRMM - University of Montpellier II - CNRS
161 rue Ada
34392 Montpellier Cedex 5, France
Email: {champ,boudet}@lirmm.fr

ABSTRACT
This paper deals with the multi-hop localization problem in static Wireless Sensor Networks. The knowledge of geographical positions of nodes is useful in such networks, as it can be used in communication protocols or to provide geographical information of detected events. We present in this paper a new and original method to locate sensor nodes, named ADNL. In our method, each sensor knows anchor nodes positions in its k-hop neighborhood and also distance between neighbor nodes thanks to a technology like TDoA. Nodes with enough anchors in their neighborhood locally run a force-based algorithm so as to deduce their position, and then forward it so as to enable other nodes to locate themselves. We provide extensive simulations under various network topologies, densities, number of anchor nodes and distance estimation errors, so as to present results showing the accuracy of our localization algorithm. ADNL algorithm is shown to be particularly accurate and robust to both distance estimation errors and irregular topologies.

1. INTRODUCTION
As a consequence of recent advancements in miniaturization and wireless communications, a new kind of network has come to the fore: Wireless Sensor Networks (WSN). In those networks, nodes (sensors) can gather information from their environment, such as temperature, gas leak, etc. They can also communicate, thanks to their wireless communication device, with other nodes in their transmission range. WSN recently attracted a lot of attention because of its wide range of applications. They can be used in a many different fields, monitoring tasks either for the military, or the environment, security, health-care, and habitat automation [1].

In WSN, many applications are monitoring tasks, so it is often needed to annotate sensed data with geographical information. Geographical positions of nodes can also be used to communicate in such networks, for example in geographic routing [2]. Thus each sensor has to know or assess its own position: this is the localization problem. The easiest way to solve this problem is to equip all nodes with a GPS device. But a GPS comes with many drawbacks: it often does not work in indoor environment, it is also expensive to equip all sensors with such a device, and as most of the time sensors are static, it is clearly not cost-efficient to provide such equipment for only a one-time localization.

In the localization problem, the goal consists in determining the positions of a maximum number of sensor nodes with the highest accuracy. Because of to their limited battery power, sensors are only able to communicate with their neighbors, thus they need to collaborate so as to estimate their positions. Due to the potentially large number of sensors, and the limited resources of sensor nodes, distributed algorithms are more suitable regarding to scalability. Many algorithms have been presented in the literature according to various hypothesis, such as the presence of anchor nodes, distance estimation methods, density, irregular networks. . .

In this paper we focus on the anchor and range-based distributed multi-hop localization problem in static WSN. It considers a small set of nodes which know their positions (anchor-based), and nodes with the ability to estimate distances with their neighbors (range-based). We propose ADNL algorithm, whose main principle is to use a force-based algorithm in sensors neighborhood so as to locate sensors. Thanks to the localization process in two steps using a force-based algorithm, and thanks to the waiting mechanism before executing the localization process, our proposed localization scheme produces very accurate localization results.

This paper is organized as follows: Section 2 presents the localization problem and previous works. Section 3 is dedicated to the description of the different steps of ADNL method. In section 4 we describe parameters used to run simulations, and we present and discuss simulation results. Section 5 concludes this paper and describes our plans for future work.

2. RELATED WORK

2.1 The Localization Problem
Let’s consider n sensor nodes deployed in a given physical region, for example a square area. Each node is able to communicate with other nodes inside its communication range: its neighbors. Using their wireless communication device, sensor nodes can collaborate to perform chosen tasks. The localization problem consists in finding geographical coordinates for all nodes, as accurately as possible. However, various hypothesis can be made. In the single-hop localization problem, each node is adjacent to at least three anchor nodes, contrary to the multi-hop localization problem where nodes which need to be localized can be several hops distant to anchor nodes. We focus in this paper only on the multi-hop localization problem.

Among localization algorithms which can be found in the literature, we can find two main categories: range-based and range-free approaches. In range-based algorithms, a specific hardware is needed to provide for example distance or angle measurements between neighbor nodes. Technologies such as RSSI (Received Signal Strength Indicator), ToA (Time Of Arrival) or TDoA (Time-Difference of Arrival) provide
an estimation of the distance between neighbor nodes, and AoA (Angle of Arrival) provides angular information between neighbor nodes, [3]. Range-free approaches only consider the connectivity knowledge of the network (or of the neighborhood) to resolve the localization problem. Obviously range-based approaches perform better regarding to localization accuracy but require extra hardware and so extra cost.

Generally in the localization problem, a small set of nodes, named anchor nodes, know their position thanks to a GPS (Global Positioning System) or to manual deployment. As GPS is expensive and as it can not be used in indoor environments it is not possible to equip all nodes with such a device. Some localization algorithms, are called anchor-free: they provide localization information, but with relative positions comparing to absolute positions provided by algorithms using anchor nodes.

Centralized and distributed algorithms can be found in the literature, and they obviously both have advantages and drawbacks. But, regarding to scalability and energy consumption it is generally more suitable to use distributed algorithms, excepted may be when considering small networks.

![Figure 1: A small wireless sensor network.](image)

Figure 1 represents a static wireless sensor network with 12 sensors: 3 anchor nodes represented by black circles and 9 other nodes represented with white circles.

### 2.2 Existing Algorithms

A large number of methods have been proposed to solve the multi-hop localization problem. As it is impossible to describe all of them, we describe the most important methods, and those with similarities with our work. For a more exhaustive description of existing algorithms, references can be found in [4].

**APS:** In [5], one of the most popular family of methods, named APS (Ad Hoc Positioning System), is presented. After a flooding step made by all the anchors, each node deduces its distances to anchor nodes, using the average hop size if there is no device to estimate distance between neighbor nodes (DV-Hop), otherwise using the shortest distance (DV-Distance). Using multilateration, nodes positions are computed, as accurately as possible. The main advantage of this method is its simplicity, and its low need of computations. However, this method is not the most accurate, especially in irregular networks. We can note that APS with DV-Hop propagation method is not sensitive to range measurement errors because it does not use this information, while distance measurement errors highly affect localization accuracy provided by APS when using DV-Distance. Another drawback of this method is the important number of communications needed as each anchor node needs to broadcast its position to the whole network.

**AT-DIST:** In [6], 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 [7]. Simulation results show that AT-DIST method performs accurate localization of the nodes when distance measurement errors are small but results are clearly less accurate with higher ones. Moreover, as with APS algorithm, in an initial phase, 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.

**MDS-MAP:** One accurate localization method, the distributed MDS-MAP method, is presented in [8]. Based on multidimensional scaling, it uses topological information and can additionally use distance estimation between nodes. The distributed method presented in [8], is based on the centralized one proposed sooner in [9]. The main principle of MDS-MAP(P) is that each node build a local map of its neighborhood, and then local maps are merged to form a global map. MDS-MAP(PR) is an improvement with an additional refinement step which is expensive regarding to computational cost. Extensive simulations showed that the distributed MDS-MAP has good performances even in irregularly shaped network. However an important exchange of additional messages is needed to compute positions for each node. Moreover, as exchanged messages contain local map information, their size is important and this may lead to many collisions in a wireless network and to an important energy consumption.

Among all localization algorithms in the literature, some of them use force-based algorithms. Force-based algorithms are often used in the graph drawing field in order to represent in a pleasant way a given graph.

**MSDR:** In [10], authors used such an algorithm in the centralized, anchor-free localization problem. Moreover they consider that sensor nodes are equipped with devices to obtain both range and angular information between neighbors. They study the capacity of their algorithm, the Multi-Scale Dead Reckoning (MSDR) algorithm, to compute relative positions in sensor networks with a complex shape. However because of previous hypothesis (centralized, distance and angle information) their method is difficult to use in practice.

**AFL:** AFL method, described in [11], is another anchor-free algorithm but in a distributed way contrary to MSDR. Each node starts with a random initial estimated position, and collaborates with its neighbors to reach its final position. Authors proposed a method which consists in a first part in configuring the graph representing the network into a topology
which looks like an unfolded view of the real configuration; then they use a distributed force-based algorithm: messages are periodically exchanged between neighbor nodes and interpreted as repulsive or attractive forces. The main drawback of AFL is the very large number of exchanged messages until sensors positions converge.

3. ADNL

Let’s consider a static wireless sensor network, with a set of nodes with unknown geographical positions and a (smaller) set of anchor nodes whose positions are known. We assume that sensor nodes have the same transmission range, however our method also works in a wireless sensor network with different transmission ranges. Sensors are also equipped with a technology such as ToA which enables them to estimate distances with their neighbors. Thus, we are working on the anchor and range based distributed multi-hop localization problem. For ease of explanation, we choose to focus on 2-Dimension localization, but ADNL can be easily modified to work in 3-Dimensions.

We believe that broadcasting anchor nodes positions in the whole network is not needed unless the number of anchor is very low. Indeed, a very distant anchor does not provide a lot of information to a sensor node.

Our method consists in using topological knowledge in the k-hop neighborhood with measured distances information between neighbor nodes. In this paper, provided simulation results will show that 2-hop knowledge is generally enough: and this is more suitable because of the needed exchange of messages to obtain this information. Moreover, 2-hop knowledge is often needed by communication algorithms, such as Connected Dominating Sets construction algorithms for example, [12]. To enable ADNL algorithm to work properly, and in order to limit the number of deployed anchor nodes, we propose to forward anchor-nodes positions at (k+1)-hops. Indeed this increases the average number of known anchor for each sensor.

ADNL works as follows: after the neighborhood discovery step, each node locally computes the shortest paths between all pairs of nodes in its neighborhood. Then, each node with enough anchors in its neighborhood runs the force-based algorithm to deduce its position and then became an estimated anchor. Newly located nodes communicate their positions to their neighbor, perhaps enabling them to execute the localization process. To improve the localization process a delay is added before running the force-based algorithm so as to enable some nodes to collect more information before computing their position.

3.1 Computing Positions Using the Force-Based Algorithm

Let’s see more precisely the behaviour of ADNL: each node with at least 3 anchors or estimated anchors\(^1\) in its neighborhood, can run the force-based algorithm; other nodes wait until some of their neighbors become estimated anchors.

Let’s consider node \(I\), a node with enough anchor nodes in its neighborhood. \(I\) has a partial knowledge of the whole network, and locally runs the following algorithm on its sub-graph:

\[ \vec{F}_{I,J} = (vdist_{I,J} - dist_{I,J}) \times \vec{v}_{I,J} \]  

(1)

As said before, the resultant force on node \(I\) is given by the sum of forces applied on \(I\) by its neighbors:

\[ \vec{F}_{I} = \sum_{K \in \text{Neighbor}(I)} \vec{F}_{I,K} \]  

(2)

\(^1\)Nodes which are initially at unknown positions and which deduce their positions thanks to ADNL.

- In a first time, \(I\) computes the shortest paths between all pairs of known nodes in its neighborhood regarding to measured distance gathered for example thanks to ToA. We denote \(dist_{I,J}\) the shortest distance between nodes \(I\) and \(J\).
- Initially each anchor node is positioned at its real position; other nodes are positioned at the same position that their nearest anchor node.
- Until the nodes no longer virtually move, a force-based algorithm is applied using previously computed shortest distances between nodes. This leads to provide an interesting starting point before the final step of the force-based algorithm, strongly reducing the probability that sensors positions converge to local minima.
- The force-based algorithm is executed again but now using only measured distances between neighbor nodes, and until nodes reach their final position.
- Node \(I\) retrieves its position, then becomes an estimated anchor and communicates its position to its neighbors.

The force-based algorithm used to determine nodes position tries to shift nodes so as to satisfy provided distances (shortest distances between all known nodes or one hop distance measurements). It’s principle is the following:

- during each step of the algorithm, virtual forces (or springs) are applied on nodes:
  - a repulsive force is applied between two nodes whose distance is less than the targeted distance.
  - conversely, an attractive force is applied between two nodes whose distance is greater than the targeted distance.
- after each step of the algorithm, each node is virtually moved in the local graph regarding to the resultant force given by the sum of the applied forces on this node.

Forces applied on nodes depend on the differences between the virtual distances (in the local graph) denoted \(vdist\), and the shortest distances computed and related to the TDoA or RSSI measurements for example. Let \(\vec{v}_{I,J}\) be the unit vector from node \(I\) to node \(J\). \(\vec{F}_{I,J}\), the force applied on \(I\) by \(J\) is given by:

\[ \vec{F}_{I,J} = (vdist_{I,J} - dist_{I,J}) \times \vec{v}_{I,J} \]  

(1)

Figure 2 is a small example displaying virtual forces applied on node \(I\) by 3 neighbor nodes. The resultant force \(\vec{F}_{I}\), also displayed, virtually moves node \(I\) closer to its real position.

When applying the force-based algorithm, anchor nodes and estimated anchors are not affected by virtual forces, they remain static at their estimated or accurate position. Obviously, anchor nodes do not need to apply the localization algorithm as they already know their position.
3.2 Localization Propagation

To execute the localization algorithm in 2-Dimensions sensors need at least 3 (preferably non collinear) anchor nodes in their neighborhood. To obtain accurate results, it is possible to delay the time until the localization process starts: this enables nodes to collect more information on newly estimated anchors. Several hypothesis can be made so as to set this delay: here are some criterions which can be taken into account when delaying the localization process:

- It is obviously better for non-localized sensors to wait to have a high number of (estimated or not) anchor nodes inside their neighborhood.
- Moreover, sensors which are close to anchor nodes will probably be localized in a more efficient way than distant ones.

We have studied the impact of these (and other) criterions on simulation results and here is one satisfying formula to define the delay needed to wait until a node, denoted $I$, with at least 3 anchor nodes in its neighborhood starts the localization process:

$$TTW = C \times \frac{\sum_{A \in \text{Anchors}} \text{dist}_{I,A}}{\#\text{Anchors}}$$

with $\text{dist}_{X,Y}$ the shortest distance computed previously between nodes $X$ and $Y$, and a constant $C$.

As soon as a node, for example node $I$, has at least 3 anchor nodes in its neighborhood, it computes its $TTW$ and starts a timer $T$. While $T < TTW$ that is to say if the delay before the start of the localization process is not reached, node $I$ waits in order to get more information; if $I$ receives a new message providing the position of a new estimated anchor, $I$ recomputes its $TTW$. As soon as $T \geq TTW$ node $I$ runs the force-based algorithm, retrieves its position and communicates it to its neighbor. This strategy significantly improves the results regarding to accuracy at the expense of delay.

4. PERFORMANCE ANALYSIS

In order to evaluate performances of ADNL algorithm, we present simulation results in this section. These simulations have been made using WSNET simulator [13]. Most of experiments made in this section are made with nearly similar parameters than those in [8], in order to compare performances of ADNL algorithm with those of MDS-MAP(P,R).

4.1 Simulation Parameters

We consider four different kind of network topologies, all of them inside a $1000 \times 1000$ area. These topologies are all the possible combinations between: deployment in a square or in a C-Shaped area, and random uniform deployment or grid placement. This leads to the four kind of networks shown in figure 3. The number of nodes is fixed for each considered topology and we adjust the transmission range to obtain different connectivity levels (the average number of one-hop neighbor nodes). The transmission range $R$ is chosen from 125 to 250 with an increment equal to 25.

As in [4] or [8], and so as to model distance measurement errors, Gaussian noise is added on measured distances; that is to say, if the real distance between two sensor nodes is $\text{rdist}_{I,J}$, then the measured distance is provided thanks to a normal distribution:

$$\text{mdist}_{I,J} = \text{rdist}_{I,J} \times (1 + N(0, e_r))$$

with $e_r$, the distance measurement error.

Gaussian noise is also added to model error in grid placements, using the same method than the one in [8], and so as to obtain a more realistic deployment.

4.2 Simulation Results

One hundred simulations have been made for each data points. We have considered the four kind of network topologies, various number of anchors (from 8 to 14) and also various distance measurement errors ($e_r$ between 0 and 20%).
To obtain comparable results, the localization errors are normalized using chosen transmission ranges.

4.2.1 Results Using Random Uniform Distribution

In a first time, let’s study results obtained using random uniform deployment of sensors in square and in C-Shaped areas. In square areas chosen transmission range leads to connectivity levels between 8.8 and 31.1, while in C-Shaped areas connectivity levels are between 8.6 and 27.1.

![Figure 4: Localization error as a function of connectivity levels, using ADNL algorithm in a square area with 200 nodes deployed using a random uniform distribution. 10 random sensors are anchor nodes, and distance measurement errors between 5% and 20%.](image1)

Figure 4 displays results obtained using 200 nodes deployed in a square area and using such a distribution. 10 of these sensors are anchor nodes. As expected in this kind of networks (as well as in all other kind of studied networks) when the average connectivity is increased the localization error is reduced; the localization error goes from about 17% in sparse networks to less than 5% with the highest studied connectivity levels and with $e_r$ equals to 5%.

![Figure 5: Lets A, B, C and J be 4 estimated or real anchor nodes. The probable localization error on node D will be reflected on nodes E, F, G, H and I. Indeed, this small graph is not rigid, so node D can be localized everywhere on the dashed line.](image2)

In sparse networks, the localization error can be explained by several reasons: some nodes have only one or two 1-hop neighbor so they will probably obtain large errors. Moreover, if this node belongs to an isthmus its poor localization results can affect adjacent nodes which will later run the localization algorithm. This kind of behavior is represented in figure 5: node D will probably locate itself with a large localization error. Moreover, this localization error will be reflected to all nodes on his left.

The four different curves on figure 4 show us the impact of distance measurement errors; obviously large distance measurement errors lead to more inaccurate localization results, however higher distance estimation errors do not affect dramatically localization accuracy.

![Figure 6: Localization error as a function of connectivity levels, using ADNL algorithm in a C-Shaped area with 160 nodes deployed using a random uniform distribution. 10 random sensors are anchor nodes, and distance measurement errors between 5% and 20%.](image3)

Results presented in Fig. 6 are for C-Shaped areas with 160 nodes including 10 anchor nodes. As expected, localization errors are higher than in "ideal" square areas. Such irregular topologies cause higher localization errors mainly because of the "hole" in the area leading to higher distance estimation errors when computing the shortest paths for example. Moreover because of the C-Shaped area, proportionately more sensor are close to empty areas and thus have a few number of neighbors: this leads to more inaccurately localized sensors and to more isthms. The more important errors in low density networks can be explained by the same reason than in the square areas as well as the differences between each chosen value for the distance estimation error parameter $e_r$.

4.2.2 Results Using Grid Placement

Let’s analyze results in square and C-Shaped areas using a noisy grid placement which seems a more realistic placement if nodes are manually deployed. The chosen transmission ranges provide different connectivity levels; the obtained connectivity levels are between 3.2 and 15.8 in square areas, and from 3.8 to 13 in C-shaped ones.

Results presented in Fig. 7 are for square areas and noisy grid placement. Localization results became really accurate when the connectivity level is higher than 4 in such networks: indeed, even when considering large distance measurement errors ($e_r = 20\%$) the average localization results are below 20%. When the connectivity level is the smallest, the initial

---

\footnote{In graph theory, a bridge/an isthmus is an edge whose deletion increases the number of connected components.}
positions of anchor nodes is very important, non-localized nodes have only a few neighbors and so probably a very low number (probably less or equal to 1) of 1-hop neighbors which are anchor nodes. In these cases, even if there are 3 or more anchor nodes in the whole neighborhood of the sensor, the number of neighbors is not sufficient and the local graph is not globally rigid [14]: our force-based localization algorithm can not successfully localize sensor nodes. When connectivity is higher or equal to 6, the localization error nearly no longer moves and principally depends on the distance measurement errors: this leads to obtain average results from less than 5% to nearly 15% when considering distance measurement errors between 0 and 20%.

In Figure 8 we consider networks with noisy grid placement in C-Shaped areas. For the same reason than in square areas, with low connectivity level the localization error can be important; principally in C-Shaped topologies as the localization errors drop from more than 40% to less 20%. When the considered connectivity levels are high (transmission range equals to 200 in Figure 9), the impact on localization errors is noteworthy; even with a few number number of anchor in its neighborhood, the use of the force-based algorithm with a high number of neighbor nodes provide accurate localization results.

4.2.3 Impact of the number of anchor nodes

The number of anchor nodes may obviously have a significant impact on localization algorithms; so we study here its impact on ADNL algorithm.

Fig. 9 shows the impact of the number of anchors on localization errors with a random uniform deployment. When chosen transmission range is 125, and so when connectivity level is low, the impact of the number of anchor is noticeable; principally in C-Shaped topologies as the localization errors drop from more than 40% to less 20%. When the considered connectivity levels are high (transmission range equals to 200 in Figure 9), the impact on localization errors is noteworthy; even with a few number number of anchor in its neighborhood, the use of the force-based algorithm with a high number of neighbor nodes provide accurate localization results.
When considering networks using noisy grid placement, as in Fig. 10, nearly the same conclusion can be made. The impact of anchor nodes in networks with a low connectivity is a little bit more important, but clearly more noticeable than in highly connected networks.

4.3 Comparison and Discussion

We compare in this section results obtained by ADNL algorithm to those obtained by MDS-Map(P,R) which is as far as we know, the most accurate localization method in the literature. However, despite its interesting results regarding accuracy, we believe that this method would be very difficult to use in practice because of the large number of exchanged messages and because of its important computational cost.

4.3.1 Comparison Regarding to Computation and Communication Costs

![Figure 11: Total communication cost for ADNL and MDS-Map(P,R) localization algorithms as a function of the number of nodes. 10 anchor nodes are considered and k is equal to 2, leading to 2-hops neighbor discovery.](image)

When using MDS-Map(P,R), in order to work properly and after the computation of local maps on each node, each local map is sent through $\log(n)$ hops, leading to a communication cost equals to $n \log(n)$. When using ADNL algorithm each node sends only its coordinates one time, when this node becomes a newly estimated anchor, so $n$ messages should be sent.

MDS-Map(P,R) and ADNL algorithms are both considered as using k-hops neighborhood. So the minimal number of transmission to gather this information is $k.n$: in a first time, each node sends a HELLO message, and then each node sends other messages including all its previously received information. So as to increase the number of known anchor nodes in ADNL we proposed to propagate anchor nodes at (k+1) hops. Thus, the cost of the neighbor discovery task for MDS-Map is equal to $k.n$ while it is equal to $m + k.n$ for ADNL if $m$ is the number of anchor nodes.

The final communication costs are displayed in Fig. 11 for a number of nodes between 50 and 200. ADNL is clearly more scalable than MDS-Map(P,R) as fewer messages needs to be sent, and as information of a newly localized node is only sent to its neighbor. Moreover, each message transmitted by MDS-Map(P,R) algorithm contains an important amount of information. Each node needs to propagate its local map, and local maps are merged from nodes to nodes: this implies to oversized packets which may lead to many collisions, and to use more than 1 message to send needed information.

Computational cost is also interesting to analyze. With ADNL localization method only nodes in the k-hops neighborhood are used for localization. In MDS-Map, only the initial local maps are computed using MDS on the k-hop neighborhood; other steps are really costly regarding computational time because a larger number of nodes are involved. Finally, MDS-Map also needs to transform the global map to an absolut map using anchors position. For all these reasons, ADNL is also more interesting regarding to the computational cost of the distributed MDS-Map.

4.3.2 Comparison regarding to Accuracy

Regarding to accuracy both algorithm performs very well in square area networks. Fig. 12 shows that results are enough close to say that performances of ADNL and MDS-Map(P,R) in such networks are equals.

![Figure 12: Comparison of ADNL and MDS-Map(P,R) algorithms in Square areas using random uniform deployment, 10 anchor nodes and distance measurement errors equal to 5%.](image)

In C-Shaped networks, as depicted in Fig. 13, ADNL algorithm performs better than MDS-Map(P,R) regarding to accuracy. The difference is important when the connectivity level is low . The gap between both algorithms becomes less important when considering network with a higher connectivity level: nearly 5% when the connectivity level is higher than 14.

5. CONCLUSION

This papers deals with the problem of finding geographical positions of sensor nodes in a given area, as many applications requires this kind of information. We describe in this paper a new distributed and accurate algorithm to locate sensors, using k-hop knowledge and distance estimations between neighbor nodes (using a technology like TDoA).

After the neighborhood discovery, each node with enough anchor nodes inside its neighborhood computes its position thanks to a force-based algorithm which can be divided in two steps: a first step to provide a suitable first estimation of the local graph topology, and then a second step
to "move" virtually nodes towards their real position. Newly located nodes became estimated anchor nodes and communicate their positions to enable other nodes to find their positions. To improve localization accuracy, nodes can delay the time until they start the localization process regarding to the number of (estimated or not) anchor in their neighborhood and regarding to the distance to these nodes.

Simulation results show the accuracy of ADNL algorithm, even with a small number of anchor nodes and in low density networks. In irregularly shaped topologies, such as when we considered the deployment in C-Shaped areas, ADNL still provide very accurate localization information. It is important to note that contrary to many methods, these results remains still accurate when considering important distance measurement errors, which can be observed in practice. Because of its high accuracy, as fewer messages are exchanged, and because ADNL needs less computations, our method seems more suitable to solve the localization problem than MDS-Map.

To conclude, ADNL algorithm is an accurate distributed algorithm, and robust to distance measurement errors. However, to emphasize advantages and drawbacks of our proposed method, we can find several directions for future work. It should be interesting to analyze the behaviour of our method in more realistic networks: in practice, neighbor discovery is a complex task [15]: when considering interferences and collisions, a node is not sure to get information of other sensors inside its transmission range. Studying the impact of a non-perfect neighborhood knowledge, or the presence of obstacles on ADNL method would be interesting. We are also working on an extension of this work using angular information provided by AoA technology.

REFERENCES


