

# GPS-Free Directional Localization via Dual Wireless Radios<sup>☆</sup>

Hüseyin Akcan<sup>a,\*</sup>, Cem Evrendilek<sup>b</sup>

<sup>a</sup> Dept. of Software Engineering, Izmir University of Economics 35330, Balçova, Izmir, Turkey

<sup>b</sup> Dept. of Computer Engineering, Izmir University of Economics 35330, Balçova, Izmir, Turkey

---

## Abstract

Location discovery, especially in mobile environments, has recently become the key component of many applications. Accurate location discovery, particularly in safety critical applications using autonomous robots or unmanned vehicles, however, is still an open problem. Existing popular methods either heavily rely on the use of global positioning systems (GPS) which do not readily lend themselves for use for the majority of applications where precision is of primary concern or are not suitable for ad-hoc deployments. In this paper, we propose a novel *directional* localization algorithm, called *Dual Wireless Radio Localization* (DWRL), which performs accurate node localizations in the plane using only distances between nodes, without the use of a GPS or nodes with known positions (anchors). The main novelty of DWRL is the use of an additional radio per node to support *directional* localization in static networks. To the best of our knowledge, this is the first time dual radios are employed in a localization setting. Existence of the dual radios on board enables DWRL algorithm to perform *directional* localization, which is not possible with existing single radio systems in static networks. We present the practical and theoretical benefits of the use of an additional radio per node in detail, test our algorithm under excessive synthetic and real-world noise scenarios, and show that DWRL algorithm is robust enough to perform *directional* localization even in high noise environments.

*Keywords:* Localization, location estimation, wireless adhoc networks, wireless sensor networks.

---

## 1. Introduction

Location based systems focus on finding the locations of items, cars, data, or people, and have been gaining a lot of interest in the research community lately. The problem is widely known as the localization problem in the wireless network community. Accurate discovery of the real-time locations of objects is the key element to the success of many applications ranging from tracking, surveillance, search-and-rescue missions to traffic control, collision avoidance, and smart cars. Traditionally, position information has been gathered using global positioning systems (GPS). However, these systems in general suffer from two major drawbacks. First of all, the devices require a clear sight of at least 4 satellites in order to work, which is not possible indoors, in downtown city centers with tall buildings, or in geographically obstructed outdoor areas, such as deep valleys. The second drawback is the imprecision of the localization when nodes stay close to each other. If the nodes are closely clustered in a small area, GPS based localization schemes do not provide enough accuracy. Boukerche *et al.* [8] states the limitations of the

global positioning systems in safety critical applications for vehicular ad-hoc networks, such as collision avoidance, in which sub-meter accuracy is needed.

Many applications depend on a level of accuracy for localization that global positioning systems cannot provide, even in places where the satellite signals are available. In search-and-rescue missions, for example, a crew of emergency personnel is deployed inside a building, and real-time tracking of positions of the crew by mission control personnel possibly outside the building is essential. In this scenario, precise adhoc localization is required in both *line-of-sight* (LOS) and *non-line-of-sight* (NLOS) conditions. In [20] a similar scenario is presented in which localization is performed by radios wearable by the emergency crew. Another application area would be the localization of unmanned vehicles to provide coordinated movement of the vehicles among themselves, or among vehicles and personnel in the area. In this scenario, levels of sub-meter accuracy is required to support mobility of multiple vehicles without collision, while still preserving the connection among them.

The limitations of the global positioning systems has led the research community to start looking into GPS-free localization methods [22]. As a result various probabilistic [23] or infrastructure based methods [9] have been proposed. The probabilistic approaches do not guarantee the desired level of accuracy in localization for the applications mentioned above. The infrastructure based ones, on

---

<sup>☆</sup>This study has been supported in the scope of the project numbered A0902003 which was accepted by Izmir University of Economics Scientific Research Projects Committee.

\*Corresponding author. Tel: +90 232 488 8287, Fax: +90 232 488 8475

Email addresses: huseyin.akcan@ieu.edu.tr (Hüseyin Akcan), cem.evrendilek@ieu.edu.tr (Cem Evrendilek)

the other hand, require the existence of reference nodes, or anchors, with known global positions on the field. Clearly, for applications such as emergency response, or collision avoidance, infrastructure based approaches are costly to apply due to long setup times. Such kind of applications require more of an ad-hoc style localization approach, without the need for any external infrastructure to operate and work only with local data collected within the network.

A good alternative for localization that uses only local resources is the ranging based localization algorithms [11, 34]. In this method, nodes only use neighborhood graph and the ranges to their immediate one-hop neighbors to perform localization. However, the problem is harder than it looks and if only ranging is used, turns into a graph realization problem [15], which has been shown to be NP-Hard [46, 49]. When the network has three beacons with known positions and the underlying graph is a trilateration graph, the problem can be solved in polynomial time using trilaterations [15]. However, beacon based methods generally use GPS to obtain the locations, or use pre-set locations, which makes the approach less applicable to ad-hoc deployments. On the other hand, global positions are not always necessary for most applications. In some applications, such as collision avoidance, relative positions of the neighbors in terms of the local coordinate system of each node is sufficient enough. For scenarios where there are no beacons with known global positions, Čapkun *et al.* [11] and Moore *et al.* [30] solve the problem in polynomial time using *relative* positioning. In this case each node assumes itself as the origin, and assigns virtual coordinates to one of its neighbors on the positive  $x$ -axis and another third neighbor on the positive  $y$ -axis. This solves the problem in polynomial time, but virtual node positions do not overlap with the real world positions, so this method is limited to only geographical routing scenarios.

In order to support critical applications, such as collision avoidance, *directional* localization [1, 2] methods are needed. For example, two vehicles in a collision course should detect the real world positions of each other to plan an accurate escape route to avoid collisions. The main distinction between *directional* and *relative* localization is that the positions provided by the former support the above application, contrary to the virtual positions provided by the latter. *Directional* localization specifies both the relative positions of the nodes, and the relative direction as the primary means of being able to approach other nodes. This attribute not only enhances capabilities of the localization towards mobility but also increases the agility of the algorithms in reacting to sharp and sudden changes. In this paper, we propose a novel *directional* localization algorithm, which only uses the ranging information between nodes to perform localization in polynomial time, for a certain class of network topologies, without the use of a global positioning system or infrastructure. Our novel contribution in this paper is that in order to solve the problem in polynomial time we introduce a sec-

ond wireless radio on each node. We also assume that the distance among the two radios on a single node is known a priori. Given a set of  $n$  nodes, what this achieves is, essentially, a set up of an initial configuration with  $2n$  radios where as many as  $n$  of the distances between radio pairs are already known. Such an arrangement, in fact, directly follows from the observation made in [6] that for a general network, if certain conditions such as the underlying network being realizable by a trilateration graph are satisfied, it is sufficient that  $O(n)$  pairs of nodes, among  $\Omega(n^2)$  possible pairs, know their distances to have a unique solution to the localization.

Although using dual radios on each node seems to double the cost at first, we propose to use  $n/2$  nodes in the network with known inter-radio distances, instead of  $n$  nodes with a single radio. This in return allows us to perform accurate *directional* localization in polynomial time, which is not possible by using single radios in static networks. However, using dual radios on a single node has some potential problems, such as interference, collision, energy efficiency, and coverage. [29, 37, 48] address some of these problems for multi radio sensor networks. In this paper, on the other hand, we restrict ourselves to analyzing the potential benefits of using dual radios only in a localization setting, and do not pursue communication related problems any further.

In order to achieve accurate ranging, we use the optional ranging ability of the ultra-wideband radios [17, 43], as part of the IEEE 802.15.4a standard <sup>1</sup>.

Our main contributions in this paper are:

- We introduce a novel *directional* localization algorithm for static wireless networks by using dual wireless radios on each node. We present the theoretical and practical benefits of the additional radio.
- We perform *directional* localization using only ranging information, without the requirement of a global positioning system or infrastructure, which is not possible by existing single radio systems in static networks.
- By introducing dual radios on a single node with a priori known distances in between, we provide a mechanism through which we study the effect of this distance value on the performance of our localization algorithm.
- We test our localization algorithm in various synthetic and real-world environmental noise scenarios, which causes errors on distance measurements, conduct extensive experimental study and simulations, and present methods to control and reduce the effects of noise even in high-noise environments.

---

<sup>1</sup>IEEE Computer Society, (August 31, 2007). IEEE Standard 802.15.4a-2007. New York, NY: IEEE.

- We give answers to the questions of how the node density, wireless range and inter-radio distance of wireless radios effect the accuracy of our localization algorithm.

The rest of the paper is organized as follows. We describe our novel localization algorithm in Section 2. In Section 3 we present the results of our experiments, while in Section 4 we discuss the related work. Finally, we present our concluding remarks in Section 5.

## 2. Dual-Radio Localization

In this section, we present our *directional* localization algorithm, named *Dual Wireless Radio Localization* (DWRL), for static wireless networks. First, the terminology used throughout the rest of the paper and the assumptions made about the DWRL algorithm are presented in Section 2.1. The primary computational steps of the DWRL algorithm, namely, semi-localization and rigid-localization are introduced and described subsequently in Sections 2.2 and 2.3 respectively. Finally, the localization algorithm proposed is described in Section 2.4.

### 2.1. Assumptions and Preliminaries

In this section, all the assumptions made about the DWRL algorithm for static wireless nodes are highlighted and basic terminology and definitions that are essential to the understanding of the paper are introduced.

We have made the following assumptions for the DWRL algorithm:

- Each node will have two ultra-wideband (UWB) wireless radios attached to apriori known positions aboard.
- Radios on each node are tagged as  $Radio_1$  and  $Radio_2$ . Both  $Radio_1$  and  $Radio_2$  can accordingly identify themselves as such.
- Nodes can measure the distance to other nodes within their proximity by using the ranging feature of UWB radios [43]. A measurement between any two nodes that are within range involves figuring out all four distances between radios of both nodes.
- Each node has a reasonably large size on board<sup>2</sup> to allow for a sufficient distance between the two radios, such that the individual radios aboard can be distinguished as required during the localization process. The distance between the two radios on a node is called *inter-radio distance*.

<sup>2</sup>In Section 3.1, we empirically show that in a typical wireless network with range set to 50 meters, inter-radio distance of 60 centimeters would be sufficient for successful localization. We argue that it is feasible for radios of this size to be carried around by humans or mounted on small vehicles such as unmanned vehicles.

- One of the nodes is designated as the *sink* to initiate the localization process. The position of  $Radio_1$  on the sink node is chosen to coincide with the origin of the coordinate system of the network, and  $Radio_2$  is assumed to be positioned in the direction of the positive  $x$ -axis. Thus, a line drawn from  $Radio_1$  to  $Radio_2$  points in the local East direction.
- Other than the UWB radio ranging, no additional positioning device or infrastructure is used.
- We assume the presence of environmental noise, which causes errors in the distance measurements performed by the radios.

Before we get into the details of the DWRL algorithm, we briefly state some of the terminology used. Two nodes are said to be *co-linear* if there exists a line passing through all four radios of the nodes. The line which passes through the line segment joining the points corresponding to the two radio positions of a node is said to be the *axis of the node*.

A graph with nodes corresponding to a set of point labels in the plane and edges denoting the Euclidean distance between points is *globally rigid* [13, 15] if for any two different mappings of nodes to actual positions in the plane in such a way that the distances in the graph are all preserved, then, the distances between every pair of positions under these mappings are also preserved. If every sufficiently small perturbation of the points in a mapping of a globally rigid graph creates also a globally rigid graph obtained by updating the existing distances in the graph in conformance with the perturbed mapping, then we say that the graph is *generically globally rigid* [13, 15].

We present the DWRL algorithm in three parts. We first present the semi-localization step in Section 2.2. We discuss the rigid-localization in Section 2.3, and finally we analyze the details of the DWRL algorithm in Section 2.4.

### 2.2. Semi-localization

*Semi-localization* is defined to be the set of computations performed to localize an unlocalized node with respect to an already localized node. Figure 1 depicts a configuration where node  $n_2$  is semi-localized against  $n_1$ . It should be noted at this point that a localized node may be characterized by either a 2-tuple of the coordinates of its both radio positions or a 3-tuple consisting of  $Radio_1$  position, inter-radio distance, and the orientation given in terms of the slope of the line passing through  $Radio_1$  and  $Radio_2$ . The two radios,  $Radio_1$  and  $Radio_2$ , that each node has are shown in blue and red respectively in Figure 1. The position of  $Radio_1$  is, in general, assumed to coincide with the origin of the local coordinate system associated with each node. Node  $n_1$  which is assumed to have already been localized has its  $Radio_1$  and  $Radio_2$  at positions given by  $(x_0, y_0)$  and  $(x_1, y_1)$  respectively. The coordinates of  $Radio_1$  and  $Radio_2$  on node  $n_2$  to be calculated through semi-localization are denoted by  $(x_2, y_2)$  and

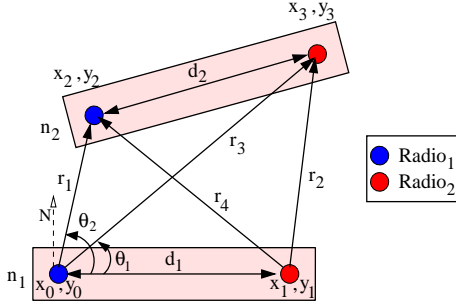


Figure 1: Semi-localization between nodes  $n_1$  and  $n_2$ .

$(x_3, y_3)$  respectively. The semi-localization of  $n_2$  against  $n_1$ , hence, involves the computations of  $(x_2, y_2)$  and  $(x_3, y_3)$ . As the inter-radio distances  $d_1$  and  $d_2$ , and the positions of the two radios associated with  $n_1$  are already known, all that remains is to measure the distances  $r_1$ ,  $r_2$ ,  $r_3$ , and  $r_4$  shown in Figure 1 between the corresponding radios using ultra-wideband ranging. Having learned the values  $(x_0, y_0)$ ,  $(x_1, y_1)$  and  $d_1$  after a communication with  $n_1$ , node  $n_2$  can now initiate the process of *ranging* to carry out the measurements.

Once the measurements are finished, all the data required become available at node  $n_2$ . It may now locally perform the computations involved to derive the possible coordinates resolving the relative position and orientation of node  $n_2$ .

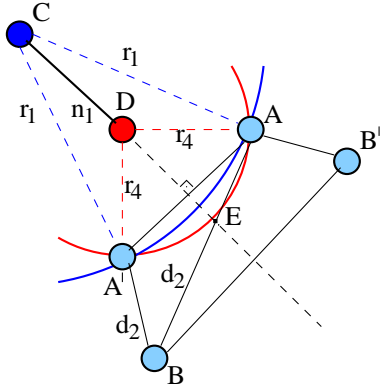


Figure 2: When localizing node  $n_2$  ( $[AB]$ ) based on the position of node  $n_1$  ( $[CD]$ ), there exist at most two solutions.

Using law of cosines (see Figure 1) we can write:

$$r_2^2 = d_1^2 + r_3^2 - 2d_1r_3 \cos \theta_1, \quad (1)$$

$$r_4^2 = d_1^2 + r_1^2 - 2d_1r_1 \cos \theta_2 \quad (2)$$

from which  $\theta_1$  and  $\theta_2$ , both assumed to be in the range  $[0, \pi]$ , can be solved for as:

$$\theta_1 = \pm \arccos \left( \frac{d_1^2 + r_3^2 - r_2^2}{2d_1r_3} \right), \quad (3)$$

$$\theta_2 = \pm \arccos \left( \frac{d_1^2 + r_1^2 - r_4^2}{2d_1r_1} \right) \quad (4)$$

In order to simplify the computations, node  $n_1$  is, at this stage, assumed to be translated and rotated so that

the global positions of its radios overlap with its local coordinate system, namely,  $(0, 0)$  and  $(d_1, 0)$  extending along the positive  $x$ -axis from *Radio*<sub>1</sub> to *Radio*<sub>2</sub>. We can, now, easily calculate the radio positions of node  $n_2$  relative to the new transformed position of node  $n_1$  to obtain:

$$x'_2 = r_1 \cos(\theta_2), \quad (5)$$

$$y'_2 = \pm r_1 \sin(\theta_2), \quad (6)$$

$$x'_3 = r_3 \cos \theta_1, \quad (7)$$

$$y'_3 = \pm r_3 \sin \theta_1. \quad (8)$$

It is finally possible to compare the four possible values of  $(x'_2 - x'_3)^2 + (y'_2 - y'_3)^2$  corresponding to four possible combinations of  $\pm\theta_1$  and  $\pm\theta_2$  with the square of inter-radio distance  $d_2^2$  to get rid of the two unwanted combinations. After obtaining the two feasible solution sets for  $\theta_1$  and  $\theta_2$ , we can conclude the semi-localization process by transforming back the radio positions  $(x'_2, y'_2)$  and  $(x'_3, y'_3)$  of  $n_2$  to get the actual positions  $(x_2, y_2)$  and  $(x_3, y_3)$  respectively by a suitable amount of translation and rotation which is actually what is needed to rotate and translate  $n_1$  back to its original position. The critical assumption made during the computations that semi-localization of  $n_2$  with respect to  $n_1$  always yields at most two solutions, apparently, needs to be verified. This is, thus, investigated next below.

As to the orientations  $\theta_1$  and  $\theta_2$  of the two radios of node  $n_2$  relative to  $n_1$ , there are a total of four possible combinations which depends on whether  $\theta_1$  and  $\theta_2$  are both positive, both negative, or one positive and the other negative. These four combinations are easily justified by the following observation (see Figure 2): The possible positions for  $(x_2, y_2)$  denoted by  $A$  and  $A'$  are simply those corresponding to at most two intersections between the two circles centered around *Radio*<sub>1</sub> (denoted by  $C$ ) and *Radio*<sub>2</sub> ( $D$ ) of  $n_1$  with radii  $r_1$  and  $r_4$  respectively. The same holds also for  $(x_3, y_3)$  which might be located at either  $B$  or  $B'$  corresponding to the intersection of circles of radius  $r_3$  and  $r_2$  each respectively this time. There are, thus, a total of at most four possible configurations for  $(x_2, y_2)$  and  $(x_3, y_3)$  which can be given as  $\{(A, B), (A', B), (A, B'), (A', B')\}$ . Two of these can be eliminated simply by enforcing the distance between them to be  $d_2$ . In order to see that, let's choose  $A$  and  $B$  arbitrarily to be the actual radio positions corresponding to *Radio*<sub>1</sub> and *Radio*<sub>2</sub> of  $n_2$  respectively.  $|AB|$  is, hence,  $d_2$ . We need to show, then, that  $|A'B|$  cannot be of length  $d_2$ . Let's assume, by contradiction, that  $|A'B| = d_2$  which in turn requires that the base angles  $\angle A'AB$  and  $\angle AA'B$  of the isosceles triangle  $\triangle ABA'$  to be equal. This is easily seen to be impossible unless  $B$  is located at  $E$  which is the point formed by the intersection of the axis of node  $n_1$  and the line segment  $[AB]$ . If  $B$  and  $E$  are not co-located,  $\angle A'AB$  is strictly less than  $\angle AA'B$  since  $\angle AA'B$  is  $\angle A'AB + \angle EA'B$ . If  $B$  and  $E$  are co-located, however, there remains a total of at most two candidate configurations given as  $\{(A, B), (A', B)\}$  with

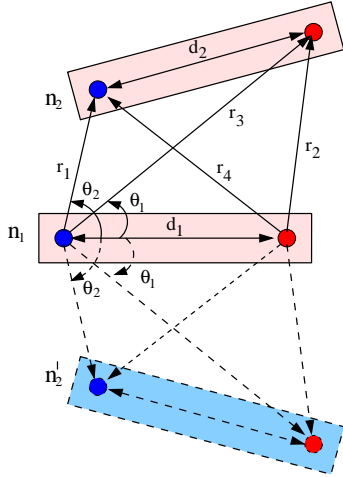


Figure 3: Two possible positions for node  $n_2$ : one actual, the other located symmetrically on a flip around the axis of  $n_1$ .

the position of at least one radio fixed on the axis of symmetry. This confirms that any radio after a flip cannot be at an equal distance to the other radio known not to be on the axis of symmetry. This discussion justifies the assumption that semi-localization always yields at most two solutions.

It can easily be observed also that the semi-localization process can only produce (generically) globally rigid configurations between any two given nodes. Besides, the only solutions which can be realized, once the position of the localized node is fixed, are those that are mirror-reflexive around the axis of the already localized node. The process, therefore, finds both the position and the orientation of the semi-localized node up to a single flip around the localized node. This is depicted in Figure 3 and the reasoning leads to the following Lemma 2.1.

**Lemma 2.1.** (Semi-localization) *The semi-localization of a node  $n_2$  with respect to an already localized node  $n_1$  will specify exactly two (possibly equal) positions in the plane which are always symmetrical with respect to the axis of node  $n_1$  so long as the distances  $r_1$ ,  $r_2$ ,  $r_3$ , and  $r_4$  between radios of the two nodes can be correctly measured and a feasible solution with respect to equations (1) through (8) is known to exist.*

*Proof.* Equations (1) through (8) and the accompanying reasoning above leave us with two cases to consider: In the first case where there are no solutions at all, the lemma is already correct. In the second case where a feasible solution exists as to the final location of node  $n_2$  being semi-localized, let one solution be denoted by a 2-tuple  $P(n_2.Radio_1, n_2.Radio_2)$  of its radio positions. Let now  $P'(n'_2.Radio_1, n'_2.Radio_2)$  be the new location obtained by flipping  $P$  around the axis of node  $n_1$ . By the similarity of all the triangles involved, then, the distances  $r_1$ ,  $r_2$ ,  $r_3$ , and  $r_4$  measured are all preserved.  $P'$  is, thus, qualified as another solution. If  $P$  and  $P'$  are different, the whole solution set must precisely be these two distinct solutions corresponding to  $P$  and  $P'$ . If they, however, happen to be the same which is only possible when node  $n_2$  is on the

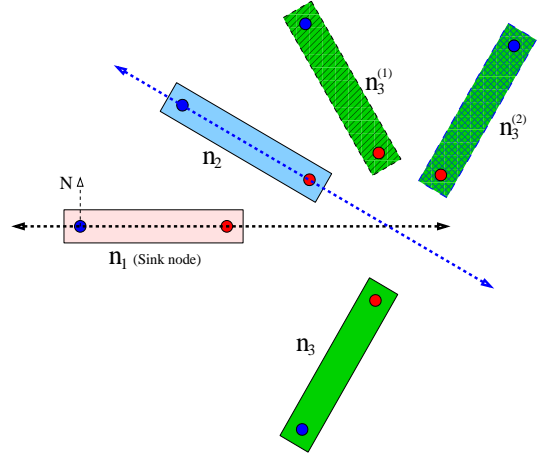


Figure 4: Rigid-localization of node  $n_3$  using localized nodes  $n_1$  and  $n_2$ , based on the distance measurements between radios of the nodes.

same line as the axis of symmetry, namely, the axis of node  $n_1$ , there exists just one solution.  $\square$

### 2.3. Rigid-localization

In order to perform *directional* localization, we ensure that the graph obtained is globally-rigid throughout localization. This is addressed by introducing an additional localized node into what we call as the *rigid-localization* process. Each rigid-localization consists of two semi-localization steps for an unlocalized node with respect to two already localized nodes in an effort to come up with a unique relative position. It is assumed throughout the process that the localized nodes are not co-linear which guarantees that some three out of four radios are always in general position. The rigid-localization process is illustrated in Figure 4. Let's initially assume that the nodes  $n_1$  and  $n_2$  are somehow rigid-localized. The figure shows the localized positions for  $n_1$  and  $n_2$  filled with solid colors. Let a solid color once more denote the actual position of node  $n_3$  which is to be localized. In the first step of the rigid-localization, unlocalized node  $n_3$  gets semi-localized against one of the localized nodes, say  $n_1$ . The two possible configurations after the computations performed locally are marked by  $n_3$  and  $n_3^{(1)}$  in Figure 4. In the second step, the semi-localization is carried out, this time, with respect to the other localized node,  $n_2$ , the outcome of which are characterized by the possible positions marked  $n_3$  and  $n_3^{(2)}$  in Figure 4. A simple comparison in the final step of possible configurations obtained through these semi-localizations reveals a unique solution, hence, rigid-localizing node  $n_3$ . It should be noted that the results of the two semi-localizations can uniquely localize  $n_3$  only when  $n_1$  and  $n_2$  are not co-linear. The preceding discussion can be summarized in the following lemma.

**Lemma 2.2.** (Rigid-localization) *Rigid-localization of a node  $i$  with respect to two already localized nodes  $j$  and  $k \neq j$  whose radios are not co-linearly located uniquely identifies the position of  $i$  relative to these nodes.*

*Proof.* Node  $i$  during a single rigid-localization gets semi-localized twice, one for each of nodes  $j$  and  $k$ . Either radio of node  $i$  is, therefore, relatively positioned with respect to a total of four other radios on  $j$  and  $k$ . By the non co-linearity assumption, some three out of four radios on  $j$  and  $k$  are known to exist in general position. Each radio on  $i$  is, hence, positioned with respect to at least three known locations in general position which, in turn, implies global rigidity. This establishes the proof of the lemma.  $\square$

It can easily be observed that a directional localization algorithm making repeated use of rigid-localizations always constructs a globally-rigid graph of interconnected nodes. One nice property characterizing such localizations is the fact that the presence of even a single localized node in the network with known real-world radio coordinates will suffice to produce a solution that will be as far to the real-world coordinates as at most a single flip around the axis of that specific node. In order to see this, the globally rigid graph can be translated and rotated as necessary so that the two radios of the sink overlap with the known coordinates. All the other nodes now either are at their real-world coordinates or need to be flipped once around the line passing through the known radio positions of sink node (see Figure 5). We discuss this issue more in the section below.

#### 2.4. The DWRL algorithm

In designing a directional localization algorithm that tries to minimize the overhead of communication in terms of the number of rigid-localizations performed, one of the nodes is first designated as the sink node whose positions for  $Radio_1$  and  $Radio_2$  are assumed to coincide with  $(0, 0)$  and  $(d_1, 0)$  respectively. The sink, next, picks within its wireless range a node to be semi-localized. One of the two positions which is on the positive  $y$  direction is, then, chosen for the semi-localized node and it is assumed to be rigidly localized thereafter. The nodes, throughout the rest of the algorithm, keeps listening until they hear from any two rigidly localized nodes. This is exactly when they first become ready to localize themselves through rigid-localization. We choose, in the DWRL algorithm, to start the localization process with the node closest to the sink. It should, however, be noted at this point that selecting the closest node to the sink does not always lead to a complete localization as the edge between them is not guaranteed to belong to the set of edges that allow for the initiation of a sequence of rigid-localizations resulting in a complete localization. The decision to use the closest node to the sink for the very initial semi-localization, on the other hand, is reasonable in that it helps minimize the error propagation in the network. If the distribution of nodes and their respective wireless ranges allow a sequence of rigid-localizations that span all the nodes, continuing iteratively, the nodes in the network can be all directionally localized. Based on which of the two possible positions of the semi-localized node has initially been selected, and assuming that the sink node is positioned correctly, the positions of

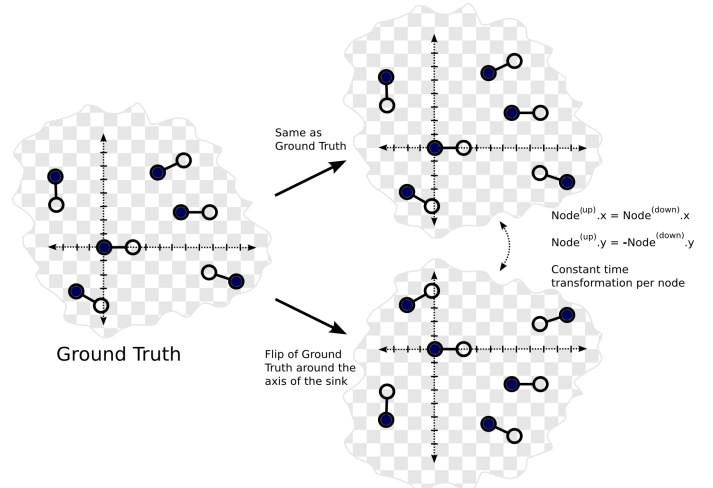


Figure 5: Based on the initial selection of the closest node’s position, the positions of the rest of the nodes can either be the correct positions, or positions flipped around the axis of the sink node. Correcting the position information is a constant time operation per node.

all the other nodes may be adjusted by at most a single flip around the sink node.

We leave the step to verify the correctness of the initial guess to a third party outside of the network, for example to an operator of the system. Once the inaccuracy of the initial guess is observed, the actual positions of all the nodes in the network can subsequently be adjusted by a single flip easily (see Figure 5).

We can now highlight the important parts of the DWRL algorithm shown in Figure 6 as follows:

- All the nodes are initially unlocalized. Sink node initiates localization by using the *InitialLocalization* algorithm shown in Figure 6. After finding the closest neighbor  $n_{cl}$ , the sink sends a message so that  $n_{cl}$  semi-localizes itself with respect to the sink. Both the sink and the semi-localized node, then, declare themselves rigid-localized, at that time, and start broadcasting their positions thereafter. We use the term *broadcast* in this paper to represent the local communication from a node to its immediate neighbors within the wireless range, using the broadcast medium of the wireless radios.
- Whenever a non-localized node receives positions of localized neighbors, it performs rigid-localization with respect to two of its already localized neighbors using the *RigidLocalization* algorithm in Figure 6.
- Each node looks for at least two rigid-localized nodes within its range that are verified to be not co-linear for performing a rigid-localization. Each rigid-localization issues two calls for the corresponding semi-localizations.
- Each node starts broadcasting its own position relative to the sink node right after it gets localized.
- Localization process continues until all the nodes are localized.

<pre> INITIALLOCALIZATION(NODE <math>N</math>) 1: <math>Localized \leftarrow FALSE</math> 2: <math>LocalPosition \leftarrow \emptyset</math> 3: <b>if</b> <math>N</math> is Sink <b>then</b> 4:   Select <math>Radio_1</math> position as origin 5:   Find the closest neighbor, <math>n_{cl}</math> 6:   Send a message to <math>n_{cl}</math> to get semi-localized 7:   set <math>Localized \leftarrow TRUE</math> 8:   set <math>LocalPosition \leftarrow \{(0,0), (d_1, 0)\}</math> 9:   Broadcast <math>\{Localized, LocalPosition\}</math> 10: <b>else if</b> <math>N</math> gets initial semi-localization message <b>then</b> 11:   /* <math>n_{cl}</math> computes its positions as <math>Pos_1</math> and <math>Pos_2</math> */ 12:   <math>\{Pos_1, Pos_2\} \leftarrow</math> Call <math>SemiLocalization(N, Sink)</math> 13:   /* Among <math>Pos_1</math> and <math>Pos_2</math>, select the one 14:     on the positive <math>y</math> axis */ 15:   <math>Pos_{n_{cl}} = PositiveYAxis(Pos_1, Pos_2)</math> 16:   set <math>Localized \leftarrow TRUE</math> 17:   set <math>LocalPosition \leftarrow \{Pos_{n_{cl}}\}</math> 18:   Broadcast <math>\{Localized, LocalPosition\}</math> </pre>
<pre> RIGIDLOCALIZATION(NODE <math>N</math>) 1: <b>if</b> <math>N</math> is not <math>Localized</math> <b>then</b> 2:   /* <math>NL \leftarrow Localized</math> neighbors list */ 3:   <b>if</b> Received position info from neighbor <math>n_{new}</math> <b>then</b> 4:     Add <math>n_{new}</math> into <math>NL</math> 5:     <b>for</b> Each neighbor <math>n \neq n_{new}</math> in <math>NL</math> <b>do</b> 6:       <b>if</b> <math>(n_{new}, n)</math> are not co-linear <b>then</b> 7:         Call <math>SemiLocalization(N, n_{new})</math> 8:         Call <math>SemiLocalization(N, n)</math> 9:         <math>Pos \leftarrow</math> resolve final position of <math>N</math> using <math>d_N</math> 10:        set <math>Localized \leftarrow TRUE</math> 11:        set <math>LocalPosition \leftarrow Pos</math> 12:        Broadcast <math>\{Localized, LocalPosition\}</math> 13:        /* exit for loop */ 14:      break </pre>
<pre> SEMILOCALIZATION(NODE <math>N_{semi}</math>, NODE <math>N_{rigid}</math>) 1: Compute positions of <math>N_{semi}</math> with respect to <math>N_{rigid}</math> 2: Resolve the final feasible positions of <math>N_{semi}</math> using <math>d_{semi}</math> 3: <b>Return</b> <math>\{Pos_1, Pos_2\}</math> associated with <math>N_{semi}</math> </pre>

Figure 6: The DWRL Algorithm.

**Theorem 2.3.** (Single Flip Theorem) *The Localization algorithm presented in Figure 6 localizes, if possible at all through a sequence of rigid-localizations, all the  $n > 1$  nodes up to a flip around the sink.*

*Proof.* The proof is by induction on the number of nodes. The base case when  $n = 2$  is seen to be capable of performing only a single semi-localization with respect to the other node, namely, the sink. The theorem certainly holds in this case by Lemma 2.1. Let's assume by the inductive hypothesis that the theorem holds for all instances with as many nodes as  $i$  less than some constant  $n > 2$ . When the algorithm is run with  $n$  nodes this time, the first  $n - 1 > 1$  nodes must have been successfully localized by the inductive hypothesis up to a single flip around the sink. Let  $\beta$  denote the node that is considered the last by the algorithm. Node  $\beta$ , then, gets rigid-localized with respect to two amongst the first  $n - 1$  that have already been localized. By Lemma 2.2, the position of  $\beta$  is, thus, unique.

But it is known by the inductive hypothesis that those two nodes with respect to which  $\beta$  is rigid-localized are both either at their correct positions or need to be flipped around axis of the sink. In the former case, the unique position found must correspond to the real-world coordinates of  $\beta$  as all four radios of the localizing nodes are at known positions which are non-co-linear. In the latter,  $\beta$  gets rigid-localized against two nodes whose actual positions can be obtained by a flip around the sink. If all the nodes including  $\beta$  in the network are flipped once around the sink, it is known that the first  $n - 1$  nodes are all correctly placed regarding their actual positions. Besides, the new position of  $\beta$  after the flip has all the distances preserved with the other two lying at their actual positions now since the rigid-localization process always specifies a globally rigid position for  $\beta$  relative to the others. This new position for  $\beta$  after a flip around the axis of the sink must, then, certainly be the actual position.  $\square$

The DWRL algorithm presented is distributed in nature and the total number of messages propagated in the network when the sink node has already been chosen is proportional to only a low degree polynomial in the number of nodes. More precisely, with the assumption that receiving a single message is enough for the receiver radio to measure the distance to the sender, the total communication cost of the DWRL algorithm can be calculated for the broadcasting and the receiving components as follows: During the course of localization, each node only broadcasts the  $\{Localized, LocalPosition\}$  tuple, where the *LocalPosition* is composed of the positions of the two radios. By repackaging the tuple to fit into two exact messages, namely  $\{Localized, LocalPosition\}$  of  $Radio_1$  sent by  $Radio_1$  and  $\{LocalPosition\}$  of  $Radio_2$  sent by  $Radio_2$ , the localization is performed by each radio sending a single message. The total communication cost for the broadcasting component for  $n$  nodes is then  $2n$  messages, which asymptotically amounts to  $O(n)$  messages. The total communication cost incurred by the receiving component is also *linear* in the number of edges. Therefore, the total communication cost of the DWRL algorithm is linear at most in the number of edges in the network. It should, however, be noted that the number of edges is  $O(n)$  since multi-hop wireless adhoc networks are in general not dense.

Optionally, by taking advantage of the fully distributed nature of the algorithm, running time can be traded for improved accuracy by performing multiple rigid-localizations in parallel. Since the communication costs dominate the CPU runtime costs in wireless networks, the cost of the DWRL algorithm hereby is reported only in terms of the communication costs. The CPU runtime cost of the DWRL algorithm, on the other hand, is also low assuming that a floating point unit is present within the CPU to perform the trigonometric calculations. Otherwise, if there are no floating point units available, it is safe to assume that the computations could be efficiently performed using approximations. The following lemma states an upper

bound on the running time of DWRL for multiple parallel rigid-localizations.

**Lemma 2.4.** *If the nodes in the network are all in realizable positions through a sequence of rigid-localizations, then the algorithm depicted in Figure 6 localizes all nodes in polynomial number of semi-localization steps.*

*Proof.* The algorithm after executing an initial semi-localization between the sink and another node (not necessarily the closest node to it), performs two semi-localizations for each rigid-localization step throughout the rest of the algorithm. Each of the remaining  $n - 2$  nodes will, thus, go through a single rigid-localization. The total number of semi-localizations once an edge leading to a complete localization is correctly identified will, therefore, be given by  $1 + 2(n - 2) = 2n - 3$ . Since this many semi-localizations can be carried out with each choice of a different edge, the total number of semi-localizations can be, at most,  $\binom{n}{2} * (2n - 3)$ .  $\square$

In this section, we presented our DWRL algorithm that performs *directional* localization using only ranging data with the help of dual wireless radios. We presented the DWRL algorithm in three parts; semi-localization, rigid-localization, and the detailed analysis of the DWRL algorithm.

### 3. Experimental Evaluation

In this section, we conduct experiments to analyze the effects of various parameters, including the range measurement errors caused by environmental noise, on the performance of the DWRL algorithm. We evaluate the properties of the DWRL algorithm in four main subsections. First, we discuss the experimental setup in Section 3.1. In order to properly evaluate the features of the DWRL algorithm, we conduct the experiments in Section 3.2 under ideal conditions, without environmental noise. However, range measurement errors due to environmental noise in the real world operations are inevitable. The DWRL algorithm depends only on range measurements performed among nodes to operate, therefore the only external source of error for the DWRL algorithm is the range measurement errors. As a result, in Section 3.3 we introduce noise to range measurements and conduct experiments under various real-world and simulated noise scenarios to evaluate the robustness of the DWRL algorithm in noisy environments. Finally in Section 3.4, based on our observations from the experiments conducted, we summarize the impact of the selected parameters on reducing the negative effects of the environmental noise on the performance of the DWRL algorithm.

#### 3.1. Experimental setup

We developed an in-house simulator written in C++ to carry out the simulations. We simulate static nodes placed with a uniform random distribution in an 100x100 unit

square area. The location of the sink node is also selected randomly as other regular nodes. Unless stated otherwise, each simulation is run for 100 times, and the average values are reported. The parameters for simulated values of the wireless range, node density, inter-radio distance, and range measurement error are specified separately for each experiment. We report two types of localization errors in our experiments; (1) The *precision* error of the localization, which shows the accuracy of the localization for each node, and is reported only for nodes that are localized. (2) The *recall* of the localization, which represents the percent of the nodes that are localized. When reporting the recall value, we include the nodes that are localized with 100% accuracy as well as the nodes localized with precision errors. In our experiments without environmental noise (Section 3.2), the precision of the DWRL algorithm is always 100% as expected. Therefore for that section we only report the recall of the DWRL algorithm. However, for the experiments with environmental noise (Section 3.3), we report both the precision error and the recall of the DWRL algorithm.

In Section 3.3, we represent the magnitude of the environmental noise in the network as  $P$ , which varies between 1 – 10. In our experiments we use two types of environmental noise based on the  $P$  parameter:

- **Synthetic Noise:** Up to  $\pm P\%$  of the wireless range  $R$  selected from a uniform random distribution, which we call as *synthetic* noise. Unless stated otherwise, we use  $P = 1$  in our simulations.
- **Real-world Noise:** Empirically gathered noise data based on the real world characteristics of the UWB radios, as reported in [3, 34], which we call as the *real-world* noise. In line with [34], the *real-world* noise is selected as the summation of a high probability small noise and a low probability large noise. The small noise is modeled as a Gaussian random process with parameters  $N(f(R), P/100)$  where

$$f(R) = 0.022 \ln(1 + R) - 0.038.$$

In our experiments we use  $R$  as the wireless range of the nodes. The large noise value is selected with a uniform random process between 0 and 10 units with probability 0.05.

Among these two noise scenarios, the magnitude of the *synthetic* noise is generally much larger than the *real-world* noise, which is due to the fact that the wireless range contributes linearly in the *synthetic* noise scenario, and logarithmically in the *real-world* noise scenario. In order to better analyze the behaviour of the DWRL algorithm in noisy environments, we choose to conduct the experiments with both *real-world* and *synthetic* noise scenarios.

Before getting into the details of the experiments, we would like to stress out a fact that some of the parameters in the experiments might look impractical, such as inter-radio distances as large as wireless ranges. This is due to the fact that the experiments in this section are conducted to demonstrate the effects of various parameters on



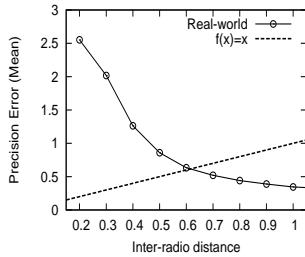


Figure 7: The mean of the precision error in units of the DWRL algorithm in the real-world noise scenario for various inter-radio distances.

the localization accuracy of the DWRL algorithm. Therefore, for the sake of completeness, the DWRL algorithm is evaluated with all theoretically possible parameter sets (especially in Section 3.3). On the other hand, in order to demonstrate the feasibility of the DWRL algorithm for real-world applications, we conduct an experiment to evaluate the precision error of the DWRL algorithm for various inter-radio distances, in *real-world* noise scenario. We select the number of nodes as 100 and the wireless range as 50 units. Figure 7 shows the change of precision error for the inter-radio distances from 0.2 to 1 units. The points below the dotted line represents the inter-radio distances for which the precision error of the DWRL algorithm is less than the inter-radio distance itself. We argue that in order to distinguish individual radios of a node, the precision error should be smaller than the inter-radio distance. As we see from the figure, inter-radio distance close to 0.6 units is the lowerbound of the valid operational range for the current setting. If we select the meter as our unit here, for a typical wireless network with 50 meter wireless range, the inter-radio distance can be selected as low as 60 centimeters. We argue that it is feasible for radios of this size to be easily carried around by humans or mounted on small unmanned vehicles.

### 3.2. Experiments without range measurement errors

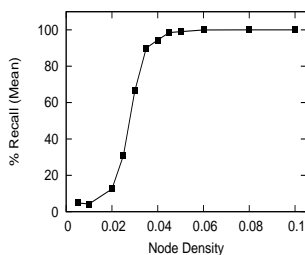


Figure 8: The percentage of the localized nodes in the network for various node densities.

In this experiment we assume that there are no range measurement errors in the network due to environmental noise, thus all range measurements are 100% accurate. We select the inter-radio distance as 0.1 unit, the wireless range as 10 units and carry out the simulation of the nodes for various node densities. Density here represents the average number of nodes in an  $1 \times 1$  unit square area. Figure 8 shows the percentage of the recall value in the DWRL algorithm. As we have stated earlier, the precision of the DWRL algorithm is 100% in this experiment, so we do not

report it separately. We can see in Figure 8 that nodes perform 100% localization for node densities larger than 0.05, for the reported inter-radio distance and wireless range parameters. In order to perform rigid-localization, each node needs to talk to two of its localized neighbors, which is the reason for observing low recall values in lower node densities.

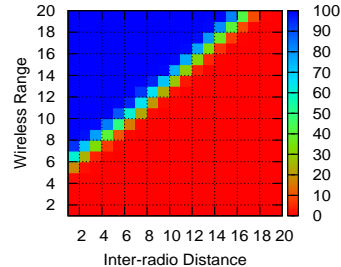


Figure 9: The percentage of recall in the DWRL algorithm for various settings of wireless ranges and inter-radio distances in units. Scale is presented on the right from 0% to 100% recall.

We have also conducted another experiment to observe the effect of the relationship between inter-radio distance and wireless range on the recall of the DWRL algorithm. We have selected the node density as 0.1 (1,000 nodes) for this experiment. Figure 9 shows that there is a linear relation between the inter-radio distance and the wireless range in terms of localization errors. While the *blue* areas (upper left) represent the case where 100% of the nodes are localized, the *red* areas (lower right) represent that none of the nodes (0%) are localized. There is a sharp and clear transition between these two areas. When the inter-radio distance of nodes is increased while keeping the wireless range at a fixed value, the nodes are unable to localize both radios of their neighbors. This, in turn, decreases the ability of the DWRL algorithm to fulfill successful localization, hence results in poor recall values.

### 3.3. Experiments with range measurement errors

In this section, we perform our experiments with the assumption that environmental noise that causes range measurement errors is present in the network. Therefore, in each experiment that we conduct we will investigate methods to limit the effects of the environmental noise on the DWRL algorithm. We select the wireless range as 10 units, the inter-radio distance as 1 unit, and the magnitude of the environmental noise  $P$  as 1. Figure 10 shows the change of the mean and the standard deviation of the precision errors of the DWRL algorithm for various node densities, in both *synthetic* and *real-world* types of noise scenarios. Figure 11 shows the mean and the standard deviation of the recall of the DWRL algorithm for the same experiment. As seen in Figure 10 (left) and Figure 11 (left), the precision error increases with the recall of the DWRL algorithm up until the node density reaches around 0.04. At this point the recall reaches close to 100%, after which any further increase in the density reduces the precision error, thus increases the accuracy of the DWRL algorithm.

Based on the results of this experiment, we can argue that once the recall value reaches close to 100%, any fur-

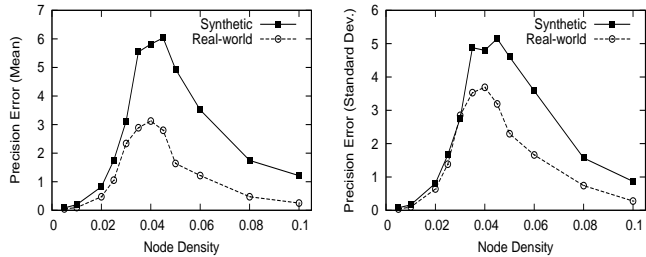


Figure 10: The mean (left) and the standard deviation (right) of the precision error of the DWRL algorithm in units in synthetic and real-world noise scenarios for various node densities.

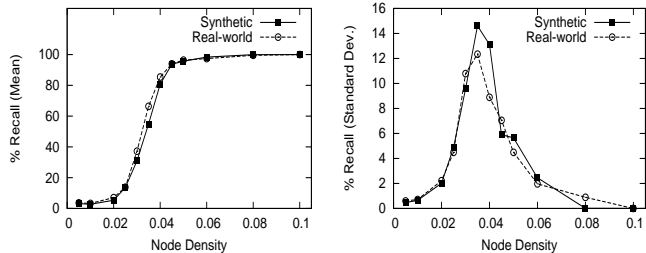


Figure 11: The mean (left) and the standard deviation (right) of the recall of the DWRL algorithm in synthetic and real-world noise scenarios for various node densities.

ther increase in the node density reduces the precision error of the DWRL algorithm. Therefore, adjusting the node density is a way to control the localization errors caused by environmental noise in the network. However, increasing node density has a downside in terms of the cost of hardware and the communication overhead incurred.

We carry out an additional experiment to observe the effects of the wireless range of the nodes on the localization performance of the DWRL algorithm. We set the density to 0.01 (100 nodes), the inter-radio distance to 10 units, and the magnitude of the environmental noise  $P$  to 1. Figures 12 and 13 show how the precision error and the recall of the DWRL algorithm vary with the wireless range of the nodes. The impact of the wireless range on the precision error is in particular more significant in the *synthetic* noise scenario. For wireless ranges around 25 units, the precision error increases with the recall value until almost 100% of the nodes are localized. Further increase in the wireless range from 25 to 40 units increases the number of neighbors each node has, and thus reduces the precision error. The more neighbors the nodes have, the more accurate the localization becomes. The improved accuracy in localization here, however, does not come for free. As the energy spent for wireless communication increases with the square of the range of the communication, nodes have to pay the cost of reducing precision error in extra units of valuable energy reserves.

One other interesting observation to make in Figure 12 (left) for *synthetic* noise scenario is the raise of the precision error for wireless range values of more than 40 units. Increasing the wireless range increases the magnitude of the environmental noise in wireless range measurements due to the wireless range factor itself in the noise scenarios. In *synthetic* noise scenario for wireless range

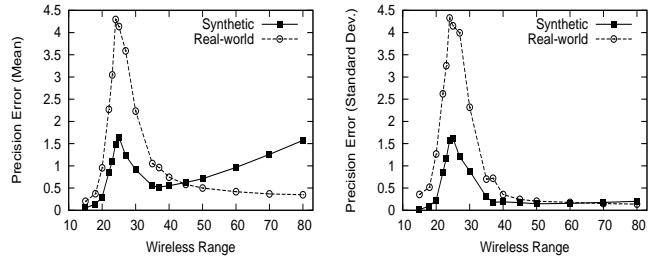


Figure 12: The mean (left) and the standard deviation (right) of the precision error of the DWRL algorithm in units in synthetic and real-world noise scenarios for various wireless ranges.

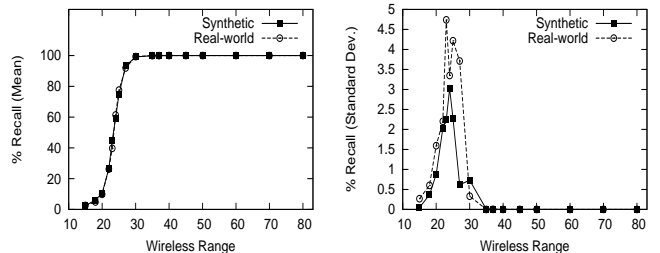


Figure 13: The mean (left) and the standard deviation (right) of the recall of the DWRL algorithm in synthetic and real-world noise scenarios for various wireless ranges.

values larger than 40 units the negative effects of the environmental noise kick in and increase the precision error of the DWRL algorithm. While the net effect of noise is relatively easier to observe in *synthetic* noise scenario, the same reasoning also holds for the *real-world* noise scenario. Due to the logarithmic contribution of the wireless range in the *real-world* noise, however, the overall effect of the environmental noise is less dramatic for small wireless range values. As a result of this experiment we can conclude that in every environmental setting, while keeping all the other parameters fixed, there is an optimum value for the wireless range of the nodes that help reduce the error in our localization algorithm. Therefore, if the characteristics of the environmental noise are known apriori, the wireless range of the nodes can be adjusted to the optimum value for that noise level and the consequences of the environmental noise can be reduced to a minimum.

In order to observe the effects of the magnitude of the environmental noise and the inter-radio distances on the accuracy of the DWRL algorithm, we set the density to 0.01 (100 nodes), the wireless range of the nodes to 40 units and conducted another experiment for both *synthetic* and *real-world* environmental noise scenarios. Figure 14 shows the precision error (left) and recall (right) of the DWRL algorithm for various magnitudes of the *synthetic* noise. Similarly, Figure 15 shows the precision error and recall for different magnitudes of the *real-world* noise. The blue parts in the left sub-figures represent the areas with the lowest precision errors, and the blue parts in the right sub-figures represent the areas with close to 100% recall value. Therefore, we can identify the correct operating range of the DWRL algorithm by overlapping the blue parts in the left and right subfigures, which brings the inter-radio dis-

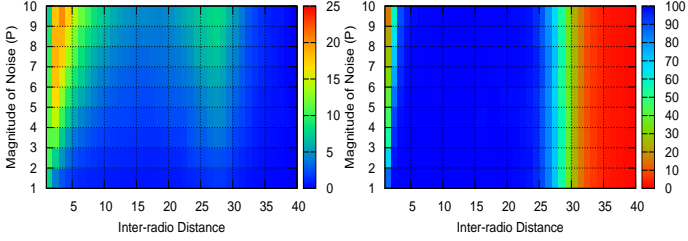


Figure 14: The precision error (left) and the recall (right) of the DWRL algorithm in *synthetic* noise scenario for various noise magnitudes and inter-radio distance units. Scales are given on the right of each figure: (left) 0 [low] to 25 [high] precision error, (right) from 0% to 100% recall.

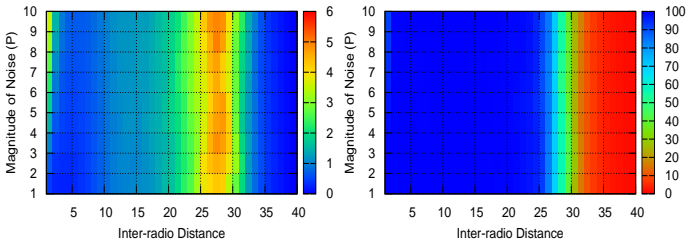


Figure 15: The precision error (left) and the recall (right) of the DWRL algorithm in *real-world* noise scenario for various noise magnitudes and inter-radio distance units. Scales are given on the right of each figure: (left) 0 [low] to 6 [high] precision error, (right) from 0% to 100% recall.

tances to arrange between 1 unit up to around 30 units. We can derive two important results by analyzing these figures. First, the precision error of the DWRL algorithm increases with the magnitude of the environmental noise, for both *synthetic* and *real-world* noise scenarios. Second and more interestingly, the precision error of the DWRL algorithm depends on the inter-radio distance, again for both types of noise scenarios. Figures 14 and 15 show that precision errors for both types of noise scenarios are high for relatively smaller inter-radio distances. A further increase in the inter-radio distances, however, can only decrease the precision errors until an optimum point is reached, which occurs around inter-radio distances of 15 units for the *synthetic* and 5 units for the *real-world* noise scenarios. We observe high precision errors for low inter-radio distances since accurately identifying and localizing the two separate radios of each node in noisy environments is especially harder if the distance among the radios is smaller than the average range measurement errors caused by the environmental noise. Therefore, for small inter-radio distance values, the accuracy of the localization increases with the inter-radio distance of the nodes. The precision errors, on the other hand, start increasing once again for values of inter-radio distances larger than the optimum values reported above. Finally, once the inter-radio distance reaches around 30 units, the recall reported drops sharply, which practically renders the DWRL algorithm useless for inter-radio distances larger than 30 units. The reason for the sudden drop here is the same as the one explained in Section 3.2 for Figure 9. To recap, setting the inter-radio distances to 30 units or more for nodes that have wireless ranges set to 40 units prevents the nodes from localizing both radios of their neighbors at the same time, which in-

terrupts rigid-localization and causes the sudden drop in the recall value.

Selecting the proper inter-radio distance can minimize the effects of the environmental noise on the localization performance of the DWRL algorithm. As seen in Figures 14 and 15, it is not the magnitude but the characteristics of the environmental noise that is driving the optimum value for the inter-radio distance. Therefore, for suitable applications, proper inter-radio distances can be set prior to deployment on the field based on the error characteristics in the environment. By doing so, the unwanted effects of the environmental noise can be reduced without any additional operational cost. This salient feature of the DWRL algorithm is considered as a more efficient tool in controlling the effects of the environmental noise than adjusting the density or the wireless range of the nodes.

In order to better visualize the outcomes of the precision error and the recall value, we present three screenshots in Figure 16. The screenshots are taken during the simulations with *real-world* noise having the characteristics as shown in Figure 15, and help us visualize the correlation of the error values with the ground truth in the network. For ease of presentation, each node is represented as a single dot in Figure 16, where the position of the dot is the mid-point of the radio positions of the node. The same convention is followed to also represent the estimated positions of the nodes. The red dots in the screenshots represent the real positions of the sink nodes. The blue dots represent the real positions of the ordinary nodes. Lines attached to blue dots represent the difference between the real and the estimated positions of the nodes, with line sizes representing the precision error for each node. The circles represent the wireless range of each node, which in this case is presented only for the sink node for comparison purposes. As we have already discussed in Section 2, for each node there are two positions calculated due to the flip ambiguity around the axis of the sink node. However, the screenshots show only a single case of the flip to ease the presentation. The screenshot in Figure 16 (left) is taken from a simulation with inter-radio distance set to 1 unit, the magnitude of noise set to 1, and presents an accurate localization where the real and estimated positions of each node almost match exactly, the precision error is 0.33 units, and the recall is 99.9%. We can argue that the accuracy level in this simulation is suitable enough for localization applications demanding high accuracy. In Figure 16 (middle), the inter-radio distance is again set to 1 unit and the magnitude of noise is increased to 5, which increases the precision error to 2.33 units while the recall drops to 96.7%. Figure 16 (right) presents a simulation with the magnitude of noise set to 10 and the inter-radio distance set to 1 unit. The precision error for this simulation is 5.32 units and the recall is 81.2%. The accuracy levels in the last two simulations are considered medium accurate based on the category given in [8]. We present the screenshots here to visually demonstrate the practical affect of the selected parameters on the ground truth local-

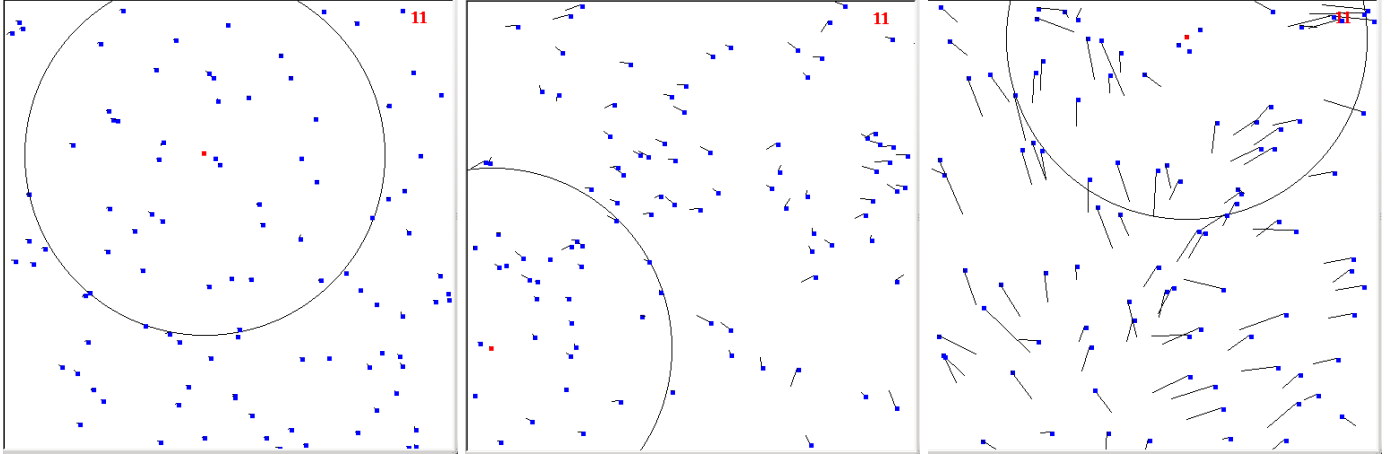


Figure 16: Snapshots from the *real-world* noise scenario simulations in Figure 15. (Left) inter-radio distance is 1 unit, magnitude of noise is 1. (Middle) inter-radio distance is 1 unit, magnitude of noise is 5. (Right) inter-radio distance is 1 unit, magnitude of noise is 10.

ization of the nodes. As we have clearly seen, the accuracy of the DWRL algorithm varies greatly with a proper setting of the parameters.

### 3.4. Discussion of methods to reduce the localization errors

METHOD	ADVANTAGE	COST
<b>Increasing node density</b>	Reduces the precision error and increases recall.	Increases the hardware and communication costs.
<b>Increasing wireless range</b>	Reduces the precision error and increases recall when optimum wireless range is selected.	Increases the cost and amount of communication.
<b>Increasing inter-radio distance</b>	Reduces the precision error and increases recall when optimum inter-radio distance is selected.	Can be done prior to deployment, thus no operational cost.

Figure 17: Comparison of the effects of the parameters over the consequences of environmental noise on the DWRL algorithm.

In real world, it is not possible to avoid the noise present in the environment that affects the range measurements by the nodes. However, for applications that permit adjusting the parameters prior to deployment, the DWRL algorithm can be tuned to be more robust even in noisy environments. Therefore in this section, based on the experience from the previous experiments, the advantages and disadvantages of all possible methods to reduce the unwanted effects of the environmental noise on the performance of the DWRL algorithm are summarized in Figure 17. There is an engineering decision in choosing between robustness and cost when selecting the node density and the wireless ranges. However, changing the inter-radio distance do not introduce any additional operational cost during localization, if done in an optimal way. Therefore, the best option would be to engineer an optimal solution for the selection of these three parameters based on the characteristics of the environmental noise, budget, and the energy requirements of the nodes.

## 4. Related Work

In localization, if only the distances between nodes are known, the problem turns into a graph embedding prob-

lem. In [46] testing the embeddability of weighted graphs in  $k$ -space is shown to be strongly NP-hard. This result is reported for general graphs, but wireless adhoc networks, where the neighborhood size is limited, are better modeled as unit disk graphs [12]. A more precise formulation of the localization problem using unit disk graphs is shown in [5, 6] to be also NP-hard. In [15], the problem is studied in terms of graph rigidity and shown to be NP-hard even when the underlying graph is known to be globally rigid. It is also known that the localization problem has a unique solution only if the underlying network graph is globally rigid. On the other hand, it is shown in [15] that a subclass of graphs known as trilateration graphs can be localized in time linear in the size of the nodes when the edge weights are exactly realizable without errors.

Hightower and Borriello [22] have a survey on localization systems which covers a side by side comparison of the existing methods and their underlying techniques.

Although global positioning systems such as GPS are widely popular in location discovery, the inability of the GPS system to be used indoors, and low precision for close range applications have led the researchers to investigate alternative localization methods. Some of the existing work [9, 14, 28, 31, 44, 45] on node localization in static wireless sensor networks use known reference points, called anchors or seed nodes to perform localization. The position information of the anchors or seed nodes are usually pre-set or gathered via GPS, therefore their use is not practical for adhoc deployments of sensors. In some applications [1, 2, 11, 24, 30, 34, 39, 41] node positions in the local coordinate system of the network is sufficient, therefore localization is possible without an external positioning infrastructure. In order to perform localization, only local information such as neighborhood data and distances between nodes are used. The effects of environmental noise on the ranging accuracy, and the cost of the ranging equipment has led the researchers to propose range-free localization methods [9, 21, 31, 33]. Probabilistic methods are also used to find the location of the nodes [23, 42]. Comparison of various localization methods for different set-

tings can be found in [16, 26]. Data analysis techniques [18, 47] are also used to analyze the neighborhood data of the nodes to perform localization.

Recently, efficient uses of dual or multi wireless radios on sensor nodes have already been considered for various applications (e.g., routing) [29, 37, 48]. Contrary to our approach, one of the radios in these work are short range low power radios, while the other is a long range high power one.

In addition to static networks, localization is also studied in mobile sensor networks with the assumption that only some [27, 40] or all the nodes [1, 2, 34, 50] can be mobile. Localization systems is also studied in vehicular adhoc networks [7, 8, 25, 35].

Recently, [10, 19, 36, 38] have surveyed the distance measurement methods. Ultra-wideband radios have standardized optional accurate ranging ability [17]. More information on the uses of Ultra-wideband positioning systems in node localization is available in [43].

*Directional* localization was introduced in [1, 2], where the nodes calculate the exact positions of their neighbors in their local coordinate system, both in terms of distance and orientation. Our algorithm also performs *directional* localization in static networks by using only ranging, while [1] and [2] require additional hardware, and work only on mobile networks. Localization using only ranging is studied in [11, 24, 30], however, all these methods assume initial virtual coordinates for the nodes, therefore do not perform directional localization. The locations obtained by these methods are suitable for geographical routing purposes, and do not correspond to true positions of the nodes.

In real world, the environmental noise is inevitable. Therefore during the design of the localization algorithms the consequences of the environmental noise should also be considered, and the algorithms should be robust enough to work even in high noise environments. Ni *et al.* [32] presents a tutorial on common errors that affect sensors. [4, 21, 28, 30, 44] manage errors during localization, where Moore *et al.* [30] presents a method to calculate the error bound theoretically. We also assume the presence of the noise in the environment, therefore analyze and present methods to control the effects of the noise on proposed DWRL algorithm. The main difference of our algorithm from the prior work is that we perform directional localization using only ranging data, without the use of any positioning device or infrastructure.

## 5. Conclusion

In this paper, we propose a novel *directional* localization algorithm. To the best of our knowledge, this is the first time dual radios are used to perform node localization. The proposed DWRL algorithm performs the localization in a distributed way, using only ranging information and without the use of any global positioning system or infrastructure. This property of DWRL algorithm makes it a perfect candidate for adhoc deployments, such

as collision avoidance or emergency search-and-rescue systems. We designed DWRL to be robust enough even in noisy environments, tested in excessive real-world and synthetic simulated noise scenarios, and presented empirical evidence that DWRL is feasible to be used in most real-world applications. Our experiments reveal that for nodes of sizes with dimensions greater than the noise granularity, *directional* localization is always possible with the DWRL algorithm. The investigation of theoretical and empirical effects of using variable sized or dynamically changing (if possible) inter-radio distances on the performance of localization is an open research area.

## References

- [1] H. Akcan, V. Kriakov, H. Brönnimann, and A. Delis. GPS-Free node localization in mobile wireless sensor networks. In *Proceedings of the 5th ACM International Workshop on Data Engineering for Wireless and Mobile Access (MobiDE'06)*, pages 35–42, Chicago, Illinois, USA, June 2006.
- [2] H. Akcan, V. Kriakov, H. Brönnimann, and A. Delis. Managing cohort movement of mobile sensors via GPS-free and compass-free node localization. *Journal of Parallel and Distributed Computing*, 70(7):743–757, 2010.
- [3] B. Alavi and K. Pahlavan. Modeling of the TOA-based distance measurement error using UWB indoor radio measurements. *IEEE Communications Letters*, 10(4):275–277, 2006.
- [4] B. D. O. Anderson, I. Shames, G. Mao, and B. Fidan. Formal theory of noisy sensor network localization. *SIAM J. Discrete Math.*, 24(2):684–698, 2010.
- [5] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. O. Anderson, and P. N. Belhumeur. A theory of network localization. *IEEE Transactions on Mobile Computing*, 5(12):1663–1678, 2006.
- [6] J. Aspnes, D. K. Goldenberg, and Y. R. Yang. On the computational complexity of sensor network localization. In *First International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS'04)*, pages 32–44, Turku, Finland, July 2004.
- [7] A. Benslimane. Localization in vehicular ad hoc networks. In *Systems Communications*, pages 19–25, Montréal, Québec, Canada, August 2005.
- [8] A. Boukerche, H. A. B. F. Oliveira, E. F. Nakamura, and A. A. F. Loureiro. Vehicular ad hoc networks: A new challenge for localization-based systems. *Computer Communications*, 31(12):2838–2849, 2008.
- [9] N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, 7(5):28–34, October 2000.
- [10] J. Caffery and G. Stüber. Overview of radiolocation in cdma cellular systems. *IEEE Communications Mag.*, 36(4):38–45, 1998.
- [11] S. Čapkun, M. Hamdi, and J.-P. Hubaux. GPS-free Positioning in Mobile Ad Hoc Networks. *Cluster Computing*, 5(2):157–167, 2002.
- [12] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86(1-3):165–177, 1990.
- [13] R. Connelly. Generic global rigidity. *Discrete & Computational Geometry*, 33(4):549–563, 2005.
- [14] L. Doherty, K. Pister, and L. El Ghaoui. Convex position estimation in wireless sensor networks. In *INFOCOM*, pages 1655–1663, Anchorage, AK, USA, 2001.
- [15] T. Eren, O. Goldenberg, W. Whiteley, Y. Yang, A. Morse, B. Anderson, and P. Belhumeur. Rigidity, computation, and randomization in network localization. In *INFOCOM*, volume 4, pages 2673–2684, Hong Kong, China, March 2004.
- [16] F. Franceschini, M. Galetto, D. Maisano, and L. Mastrogiacono. A review of localization algorithms for distributed wireless sensor networks in manufacturing. *International Journal of Computer Integrated Manufacturing*, pages 1–19, June 2007.

- [17] S. Gezici, Z. Tian, G. Giannakis, H. Kobayashi, A. Molisch, H. Poor, and Z. Şahinoğlu. Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. *Signal Processing Magazine, IEEE*, 22(4):70–84, July 2005.
- [18] G. Giorgetti, S. K. S. Gupta, and G. Manes. Wireless localization using self-organizing maps. In *Proceedings of the 6th International Conference on Information Processing In Sensor Networks (IPSN'07)*, pages 293–302, Cambridge, Massachusetts, USA, April 2007.
- [19] F. Gustafsson and F. Gunnarsson. Mobile positioning using wireless networks. *IEEE Signal Processing Magazine*, pages 41–53, 2005.
- [20] D. Harmer, M. Russell, E. Frazer, T. Bauge, S. Ingram, N. Schmidt, B. Kull, A. Yarovoy, A. Nezirovic, L. Xia, V. Dizdarevic, and K. Witrisal. Europcom: Emergency ultrawideband radio for positioning and communications. In *IEEE International Conference on Ultra-Wideband (ICUWB'08)*, volume 3, pages 85–88, September 2008.
- [21] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher. Range-free localization and its impact on large scale sensor networks. *ACM Trans. Embedded Comput. Syst.*, 4(4):877–906, 2005.
- [22] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66, 2001.
- [23] L. Hu and D. Evans. Localization for mobile sensor networks. In *Proceedings of the International Conference on Mobile Computing and Networking (MOBICOM'04)*, pages 45–57, Philadelphia, PA, USA, 2004.
- [24] R. Iyengar and B. Sikdar. Scalable and distributed GPS free positioning for sensor networks. In *Proceedings of the IEEE International Conference on Communications (ICC'03)*, pages 338–342, Anchorage, AK, USA, May 2003.
- [25] V. Kukshya, H. Krishnan, and C. Kellum. Design of a system solution for relative positioning of vehicles using vehicle-to-vehicle radio communications during gps outages. In *IEEE Vehicular Technology Conference (VTC'05 Fall)*, pages 1313–1317, Dallas, TX, USA, September 2005.
- [26] K. Langendoen and N. Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Computer Networks*, 43(4):499–518, 2003.
- [27] X. Li, N. Mitton, I. Simplot-Ryl, and D. Simplot-Ryl. Mobile-beacon assisted sensor localization with dynamic beacon mobility scheduling. In *Proceedings of the 8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'11)*, Valencia, Spain, October 2011.
- [28] J. Liu, Y. Zhang, and F. Zhao. Robust distributed node localization with error management. In *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'06)*, pages 250–261, Florence, Italy, May 2006.
- [29] D. Lymberopoulos, N. B. Priyantha, M. Goraczko, and F. Zhao. Towards energy efficient design of multi-radio platforms for wireless sensor networks. In *Proceedings of the 7th International Conference on Information Processing In Sensor Networks (IPSN'08)*, pages 15–26, St. Louis, Missouri, USA, April 2008.
- [30] D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys'04)*, pages 50–61, Baltimore, MD, USA, 2004.
- [31] R. Nagpal, H. E. Shrobe, and J. Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In *Proceedings of the 2nd International Conference on Information Processing In Sensor Networks (IPSN'03)*, pages 333–348, Palo Alto, California, USA, 2003.
- [32] K. Ni, N. Ramanathan, M. N. H. Chehade, L. Balzano, S. Nair, S. Zahedi, E. Kohler, G. Pottie, M. Hansen, and M. Srivastava. Sensor network data fault types. *ACM Trans. Sen. Netw.*, 5:25:1–25:29, June 2009.
- [33] D. Niculescu and B. Nath. Dv based positioning in ad hoc networks. *Telecommunication Systems*, 22(1-4):267–280, 2003.
- [34] J. Park, E. D. Demaine, and S. J. Teller. Moving-baseline localization. In *Proceedings of the 7th International Conference on Information Processing In Sensor Networks (IPSN'08)*, pages 15–26, St. Louis, Missouri, USA, April 2008.
- [35] R. Parker and S. Valae. Vehical localization in vehicular networks. In *IEEE Vehicular Technology Conference (VTC'06 Fall)*, pages 1–5, Montréal, Québec, Canada, September 2006.
- [36] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses, and N. S. Correal. Locating the nodes: cooperative localization in wireless sensor networks. *Signal Processing Magazine, IEEE*, 22(4):54–69, July 2005.
- [37] T. Pering, Y. Agarwal, R. K. Gupta, and R. Want. *CoolSpots*: reducing the power consumption of wireless mobile devices with multiple radio interfaces. In *Proceedings of the 4th International Conference on Mobile Systems, Applications, and Services (MobiSys'06)*, pages 220–232, Uppsala, Sweden, June 2006.
- [38] M. Porretta, P. Nepa, G. Manara, and F. Giannetti. Location, location, location. *IEEE Vehicular Technology Magazine*, pages 20–29, 2008.
- [39] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller. Anchor-free distributed localization in sensor networks. In *Proceedings of the First International Conference on Embedded Networked Sensor Systems (SenSys'03)*, pages 340–341, Los Angeles, CA, USA, 2003.
- [40] N. B. Priyantha, H. Balakrishnan, E. D. Demaine, and S. J. Teller. Mobile-assisted localization in wireless sensor networks. In *INFOCOM*, pages 172–183, Miami, FL, USA, March 2005.
- [41] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *Proceedings of the Sixth International Conference on Mobile Computing and Networking (MOBICOM'00)*, pages 32–43, Boston, MA, USA, August 2000.
- [42] M. Rudafshani and S. Datta. Localization in wireless sensor networks. In *Proceedings of the 6th International Conference on Information Processing In Sensor Networks (IPSN'07)*, pages 51–60, Cambridge, Massachusetts, USA, April 2007.
- [43] Z. Şahinoğlu, S. Gezici, and I. Güvenç. *Ultra-wideband Positioning Systems: Theoretical Limits, Ranging Algorithms, and Protocols*. Cambridge University Press, 2008.
- [44] C. Savarese, J. M. Rabaey, and K. Langendoen. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In *USENIX*, pages 317–327, Monterey, CA, USA, 2002.
- [45] A. Savvides, C.-C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th annual International Conference on Mobile Computing and Networking (MobiCom'01)*, pages 166–179, New York, NY, USA, 2001. ACM Press.
- [46] J. B. Saxe. Embeddability of weighted graphs in k-space is strongly np-hard. In *17th Allerton Conf. Commun. Control Comput.*, pages 480–489, 1979.
- [47] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz. Localization from mere connectivity. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc'03)*, pages 201–212, Annapolis, Maryland, USA, 2003.
- [48] T. Stathopoulos, M. Lukac, D. McIntire, J. S. Heidemann, D. Estrin, and W. J. Kaiser. End-to-end routing for dual-radio sensor networks. In *INFOCOM*, pages 2252–2260, Anchorage, Alaska, USA, May 2007.
- [49] Y. Yemini. Some theoretical aspects of position-location problems. In *20th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1–8, San Juan, Puerto Rico, 1979.
- [50] P. Zhang and M. Martonosi. Locale: Collaborative localization estimation for sparse mobile sensor networks. In *IPSN '08: Proceedings of the 7th International Conference on Information Processing In Sensor Networks*, pages 195–206, St. Louis, Missouri, USA, April 2008.