Secure Localization of Nodes in Wireless Sensor Networks with Limited Number of Truth Tellers

J. G. Alfaro[†,‡], M. Barbeau[†], and E. Kranakis[†]

[†]Carleton University, School of Computer Science
5302 Herzberg Building, 1125 Colonel By Drive
Ottawa, Ontario, K1S 5B6, Canada

[‡] Open University of Catalonia,
Computer Science and Multimedia Studies
Rambla Poble Nou 156, 08018 Barcelona, Spain

E-mail: *joaquin.garcia-alfaro@acm.org,*
barbeau@scs.carleton.ca, kranakis@scs.carleton.ca

Abstract

We provide in this paper three algorithms that enable the sensor nodes of a Wireless Sensor Network (WSN) to determine their location in presence of neighbor sensors that may lie about their position. Our algorithms minimize the number of trusted nodes required by regular nodes to complete their process of localization. The algorithms always work for a given number of neighbors provided that the number of liars is below a certain threshold value, which is also determined.

Key words: Network Security; Wireless Security; Sensor Networks; Secure Localization.

1 Introduction

Wireless Sensor Networks (WSNs) are a specific kind of ad hoc networks, highly decentralized, and without infrastructure. They are build up by deploying multiple microtransceivers, also called sensor nodes, that allow end users to gather and transmit environmental data from areas which might be inaccessible or hostile to human beings. The transmission of data is done independently by each node, using a wireless medium. The energy of each node is limited to the capacity of its battery. The consumption of energy for both communication and information processing must be minimized.

Sensor networks use multi-hop communications. The objective is to route collected data to end users using the collaboration of all the nodes. Data must be directed toward the end users of the WSN. Routing algorithms in WSNs can be classified by different criteria (e.g., depending on the network structure or the protocol operations [1]). There are routing protocols that use sensor node positions to route the data in the network. Positions are normally not determined a priori. Indeed, the sensors are deployed into the geographical area where the data must be collected, then they work together to locate their position.

The localization phase is a very critical step that must be secured in order to ensure the integrity of the WSN and its associated services. Firstly, this process allows the sensor to set up the necessary parameters to establish the paths that will lead their data towards end users. The knowledge of their position is also an essential prerequisite for the final application that processes the data collected by sensors, i.e., the user needs to know the origin of collected data before using it. Finally, the end users might want to query some nodes by sending the position where information needs to be collected. The localization process is therefore crucial.

The existence of misbehaving nodes can significantly degrade the effectiveness of WSNs. For instance, an attacker can lead to the calculation of false positions and distances. An attacker can provide wrong routing paths to sensors in order to exhaust their battery life [13]. It may lead to reporting false information on the geography of the phenomenon studied by sensors. Substantial progress has been made to secure the localization process of WSNs [11, 12, 5, 6, 7, 10].

Most of the work is based on the use of trust models, where a few dedicated nodes called *anchors* (e.g., more powerful sensors with GPS), trusted by the other nodes of a WSN, provide information to localization processes. For instance, localization processes may perform triangulation using several GPS-based anchors [3]. Anchors may in fact be defective. Trusted but defective anchors must be detected and isolated. A solution to this problem is the introduction of a new class of sensors, also trusted by the other nodes — often referred in literature as auditors or verifiers. They regularly review information provided by the anchors to decide on the validity of the information they provide.

The necessity of a trust model by these approaches is often too expensive and not always realistic. Firstly, the distribution of anchors and verifiers must be established a priori, to ensure coverage of the network. Since the cost of these special nodes is considerably higher than the cost of regular nodes, their representation in the network is likely to be inferior. It is thus fair to assume that an attacker can easily locate and compromise anchors or verifiers to mislead, for instance, the location process in a WSN. On the other hand, current approaches to deploy trust on WSNs require cryptographic operations supported by sensors. This has impact on their battery life, which can degrade the performance of a WSN. The localization process executed by regular sensors should have a minimal impact in terms of energy consumption. Finally, too much trust may reduce the autonomy of a WSN, since trusted nodes must be monitored to ensure their integrity. This can be a real problem, for example, for military applications in hostile environments where the localization phase must be managed by sensors without any external intervention.

We present in this paper three algorithms that enable the regular sensors of a WSN to determine their location in the presence of neighboring sensors that may lie about their position. We assume that liars are either enhanced or regular sensors that are aware of their position; but that, for any reason, announce false location to their neighbors. Their intent could either be *malicious* (i.e., to mislead regular sensors into wrong calculations) or *unintentional* (i.e., due to the presence of obstacles or other physical circumstances that prevent them from announcing correct positions). The three algorithms that we present guarantee that regular nodes in the WSN always obtain their position provided that the number of liars in the neighborhood of each regular node is below a certain threshold value, which we determine for each algorithm. Our three algorithms allow the regular nodes to identify and isolate nodes that are providing false information about their position. Moreover, our algorithms minimize the necessary number of trusted nodes required by regular sensors to complete their process of localization. They also guarantee a small exchange of data between nodes, minimizing in this manner the impact that the localization process has in terms of energy and battery life of sensors.

Organization of the paper — Section 2 establishes the prerequisites for our approach. Sections 3 and 4 present our algorithms. Section 5 presents results obtained from the simulations of our algorithms. Section 6 points out to some related works.

2 Localization in the Presence of Liars

Let us consider a point $A = (a_x, a_y)$, such that $(a_x, a_y) = \mathscr{F}(B_1, B_2, B_3)$ for any three points B_1, B_2, B_3, and where function $\mathscr{F}$ returns the point obtained as the intersection of the three circles which are centered at B_1, B_2, B_3 and with radii $d(A,B_1), d(A,B_2)$, and $d(A,B_3)$, respectively (cf. Figure 1). $\mathscr{F}(B_1, B_2, B_3)$ is then a unique and well-defined point when the points A, B_1, B_2, B_3 are in general positions. If points are sensors, function $\mathscr{F}$ is calculated by sensor A when it receives the coordinates $B_1 = (b_{1x}, b_{xy}), B_2 = (b_{2x}, b_{2y}), B_3 = (b_{3x}, b_{3y})$ and measures the distances $d(A,B_1), d(A,B_2), d(A,B_3)$ using radiolocation techniques [2].

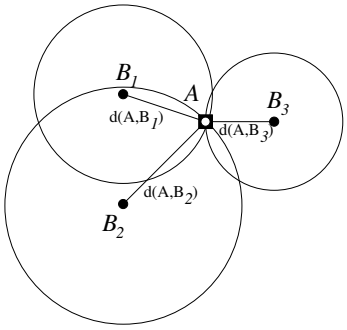

Figure 1. Sensor A wants to determine its location. It receives radiolocation signals from three nodes B_1, B_2, and B_3 that are located in its distance one neighborhood. A determines its position by processing the three signals.

The unknown coordinates of $A = (a_x, a_y)$ might be obtained from the unique solution of the following system of equations:

$$(b_{1x} - a_x)^2 + (b_{1y} - a_y)^2 = d(A,B_1)^2 \quad (1)$$
$$(b_{2x} - a_x)^2 + (b_{2y} - a_y)^2 = d(A,B_2)^2 \quad (2)$$
$$(b_{3x} - a_x)^2 + (b_{3y} - a_y)^2 = d(A,B_3)^2. \quad (3)$$

Consider now that sensor A may receive radiolocation signals from misbehaving nodes that lie about their correct position by announcing incorrect locations to A (cf. Figure 2). Let $N_1(A)$ be the set of sensor nodes at distance one hop away from A and let ℓ (where $\ell \leq \#N_1(A)$) be the number of malicious nodes that lie to A. Can A detect the lie, exclude the incorrect locations, report the liars, and still determine its location?

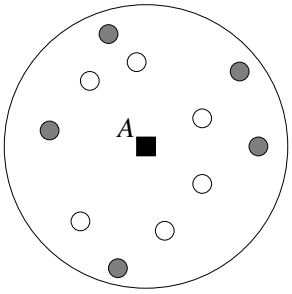

Figure 2. Sensor A is now receiving its radiolocation signals from two types of sensors in its distance one neighborhood: liars (gray circles) and truth tellers (blank circles).

Let us first elaborate on the adversary model that we assume. We suppose that a *liar* is a sensor, say S, aware of its location, and that for any reason announces a false location to A. The intent can be *malicious* (i.e., to mislead A into the wrong calculation of its location) or *unintentional* in the sense that obstacles or other physical circumstances (e.g., multi-path interference) prevent S from announcing its correct location. Whatever, the effect on A is the same, who also must determine its correct location. The algorithms that we analyze in the sequel also depend on how A determines that an announced location is wrong/correct. In particular, for *theoretical convenience* we assume that given two locations $(x,y),(x',y')$ sensor A can determine whether or not they are equal thus rejecting one of the two. This is much simpler than assuming the more realistic scenario that for some constant $\varepsilon > 0$, A identifies the two locations $(x,y),(x',y')$ provided that the Euclidean distance $\sqrt{(x-y)^2+(x'-y')^2} \leq \varepsilon$. Finally, we assume that liars cannot collude. To do so, we consider that the exchange of radiolocation signals between sensors can guarantee this property [10].

We present in the sequel three algorithms that handle the problem of determining the proper location of regular nodes in the presence of liars. Our algorithms aim not only at determining the proper location but also at excluding the incorrect locations and at isolating the set of liars. We assume the case where A knows a priori the upper bound ℓ of sensor nodes lying in the geographical area where its has been deployed. Our algorithms always work for a given number of neighbors provided that the number of liars is below a certain threshold value, and minimizing the necessary number of neighbors that regular sensors must trust. In turn, they also reduce the necessary number of messages to exchange between nodes, reducing in this manner the impact that the localization process has in terms of energy and battery life of the sensors.

Section 3.1 presents a first algorithm that consists of the following approach. Sensor A, after receiving the radiolocation signals from its one hop neighbors, calculates its possible location using the standard localization technique discussed above (cf. Figure 1), and uses a majority decision rule to accept as correct the location that occurs in the majority. In this case, A computes for each triple of neighbors $B_i,B_j,B_k \in N_1(A)$ the point $\mathscr{F}(B_i,B_j,B_k)$, where $N_1(A)$ is the set of sensor nodes at distance one neighborhood from A. Then, A selects as its position the point which has the majority value. Evidently, if the number of neighbors is sufficiently high, then A should be able to determine its actual location without problems. Indeed, we show in Section 3.1 that if the number of liars among the nodes in $N_1(A)$ is higher than two, then A is able to determine its proper location if $n^3-3(2\ell+1)n^2+2(3\ell^2+6\ell+1)n-(2\ell^3+6\ell^2+4\ell)>0$, where n is the number of nodes in $N_1(A)$ and ℓ is the number of liars among them. This means, for instance, that if ℓ is exactly three, A needs, at least, sixteen one hop neighbors to guarantee the success of the process.

Section 3.2 presents a second algorithm that relies on the use of pairs of signals and a majority rule. It follows the following approach. For every pair of neighbors $B_i,B_j \in N_1(A)$, sensor A computes the pair of points $[(a_x,a_y),(a_{x'},a_{y'})] = \mathscr{F}(B_i,B_j)$ as the resulting two points obtained from the intersection of the two circles centered at B_i,B_j and with radii $d(A,B_i)$ and $d(A,B_j)$, respectively. This is a well-defined two element set when the points A,B_i, and B_j are in general positions. Hence, it might easily be obtained from the unique solution of equations 1 and 2. As before, A uses the majority rule to determine the more plausible position. We show in Section 3.2 that if the number of liars among the nodes in $N_1(A)$ is higher than one, A is able to determine its proper location if $n > \frac{4\ell+1+\sqrt{8\ell^2+1}}{2}$. Hence, if ℓ is exactly three, A needs, at least, eleven one hop neighbors to guarantee the success of the process.

Section 4 presents a third algorithm that improves the previous results by relaxing our initial hypotheses. We now assume that sensor A may trust one of the nodes in its distance one neighborhood, say node B_1. We now rely on the use of frequencies of occurrence instead of a majority rule. Let the neighbors of A be n_A independent and identically distributed random variables $B_1,B_2,\ldots,B_{n_A}$ indicating their positions. Sensor A calculates for every neighbor $B_i \in N_1(A)$ other than B_1, the pair of points $[(a_x,a_y),(a_{x'},a_{y'})] = \mathscr{F}(B_1,B_i)$. The random variables obtained by A when applying this process are also independent. Sensor A uses then these random variables and calculates the frequencies of occurrence of each position. A finally selects as a plausible position the most frequently occurring value. We show in Section 4 that A is able to determine its proper location in the presence of any ℓ liars, if $n-\ell \geq 3$. This means that, for instance, if ℓ is exactly three, A only needs six one hop neighbors to guarantee the success of the process.

We provide in the sequel sufficient conditions for the validity of these three algorithms, all of them depending on the number of one hop neighbors and liars among them.

3 Secure Localization without Trusted Nodes

3.1 Use of Three Neighbor Signals

Algorithm 1 depicts the approach. Following is the analysis.

Algorithm 1 Majority-ThreeNeighborSignals

1: Sensor A requests the location of its neighbors.
2: Every sensor in $N_1(A)$ sends its location to A.
3: For each triple t of neighbors $B_i, B_j, B_k \in N_1(A)$, A computes (x_t, y_t).
// (x_t, y_t) is the point of intersection of the three circles
// centered at B_i, B_i, B_k and with radii $d(A,B_i)$,
// $d(A,B_j)$, and $d(A,B_k)$.
4: A accepts the majority as its location, and reports the nodes lying about the resulting position.
// if there is no consensus, then A aborts the process,
// and declares that it cannot compute its location.

In the presence of ℓ liars and given n one hop neighbors, consider all possible triples of sensors such that at least one of the sensors in the triple is a liar. Such a triple can have in each case either[1]

1. all three sensors liars, which gives a total of $\binom{\ell}{3}$ triples of liars, or
2. exactly two sensors liars (and the other one truth teller) which gives a total of $\binom{n-\ell}{1} \cdot \binom{\ell}{2}$ triples of liars, or
3. exactly one sensor liar (and the other ones truth tellers) which gives a total of $\binom{n-\ell}{2} \cdot \binom{\ell}{1}$ triples of liars.

A location that is determined by A is correct if it is provided by three truth tellers; otherwise it is (possibly) *incorrect*. The majority rule in Algorithm 1 will succeed if the number of *correct* locations is bigger that the number of *incorrect* locations. This amounts to having the inequality.

$$\binom{n}{3} - \binom{\ell}{3} - \binom{n-\ell}{1} \cdot \binom{\ell}{2} - \binom{n-\ell}{2} \cdot \binom{\ell}{1} > \binom{\ell}{3} + \binom{n-\ell}{1} \cdot \binom{\ell}{2} + \binom{n-\ell}{2} \cdot \binom{\ell}{1},$$

from which we derive

$$\binom{n}{3} > 2\left(\binom{\ell}{3} + \binom{n-\ell}{1} \cdot \binom{\ell}{2} + \binom{n-\ell}{2} \cdot \binom{\ell}{1}\right) \quad (4)$$

as a necessary and sufficient condition for the majority rule decision to succeed at A.

Table 1 depicts the minimum number of neighbors for a given number of liars. The table can be derived as follows. If $\ell = 1$ then $\binom{\ell}{3} = \binom{\ell}{2} = 0$ and Inequality 4 can be simplified to $n > 6$, which means A can determine a correct location in the presence of a liar if it has at least 7 neighbors. If $\ell = 2$ then $\binom{\ell}{3} = 0, \binom{\ell}{2} = 1$ and Inequality 4 can be simplified to $n(n-1)/6 > 2(1+(n-3))$, which in turn is equivalent to $n > \frac{13+\sqrt{75}}{2}$. This means that A can determine a correct location in the presence of two liars if it has at least eleven neighbors. More generally, if $\ell \geq 3$ then cumbersome but elementary calculations show that Inequality 4 can be simplified to the following inequality:

$$n^3 - 3(2\ell+1)n^2 + 2(3\ell^2+6\ell+1)n - (2\ell^3+6\ell^2+4\ell) > 0. \quad (5)$$

Plotting inequality 5 (cf. Figure 3), we obtain that for $\ell = 3$ liars the minimum number of neighbors must be at least 16, and for $\ell = 4$ at least 21.

[1] We use the standard convention for binomial coefficients that $\binom{s}{t} = 0$ when $s < t$.

Number of Liars	Min Number of Neighbors
$\ell = 1$	$n = 7$
$\ell = 2$	$n = 11$
$\ell = 3$	$n = 16$
$\ell = 4$	$n = 21$

Table 1. Minimum number of neighbors required for a node to determine its correct location (using Algorithm 1) in the presence of ℓ liars in its neighborhood.

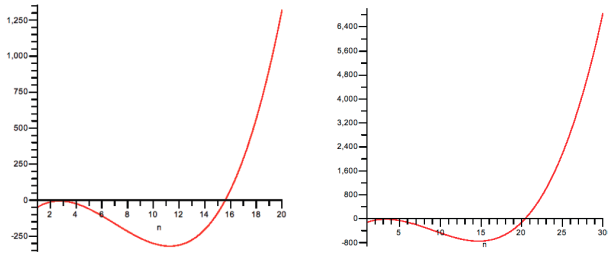

Figure 3. Plotting the minimum neighborhood size n as a function of the number of liars ℓ so as to guarantee that inequality (5) is true for $\ell = 3$ (left diagram) and $\ell = 4$ (right diagram).

3.2 Use of Two Neighbor Signals

Suppose now that sensor A uses only the radiolocation signals of two neighbors and therefore the correct location is one of the two points of intersection of the two circles centered at these two neighbors. The whole process is described in Algorithm 2. By using this algorithm, sensor A computes

Algorithm 2 Majority-TwoNeighborSignals

1: Sensor A requests the location of its neighbors.
2: Every sensor neighbor of A sends its location to A.
3: For each pair p of neighbors $B_i, B_j \in N_1(A)$, A computes two possible locations $(x_p, y_p), (x'_p, y'_p)$.
// The locations computed are the two points of intersec-
// tion of the two circles centered at B_i, B_j and radii
// $d(A,B_i)$ and $d(A,B_j)$, respectively.
4: A accepts as correct the pair of locations determined by the majority rule.
// If there is no such pair among them, then A aborts
// the process, and declares that it cannot compute the
// pair of locations.

for every two neighbors $B_i, B_j \in N_1(A)$ a pair of locations $\{X, X'\}$. The pair $\{X, X'\}$ of locations is obtained from the intersection of the two circles centered at B_i, B_j, and radii $d(A,B_i), d(A,B_j)$, respectively. As depicted in Figure 4, the correct location of sensor A is either X or X'. A may then use again the majority rule to determine the more plausible position and to report those nodes that lied about the proper location.

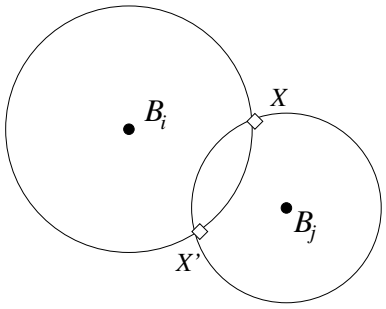

Figure 4. Sensor A applying Algorithm 2.

Correctness — Consider all possible pairs of sensors at least one of whom is a liar. Such a pair can have either

1. both sensors liars, for a total of $\binom{\ell}{2}$ pairs, or
2. exactly one sensor liar for a total of $\binom{n-\ell}{1} \cdot \binom{\ell}{1}$ pairs

such sensors in each case. A pair of locations that is determined by A is correct if it is determined by two truth tellers; otherwise, it is (possibly) *incorrect*. The majority rule in Algorithm 2 will succeed if the number of *correct* pairs of locations is bigger that the number of *incorrect* pairs of locations. This amounts to having the inequality.

$$\binom{n}{2} - \binom{\ell}{2} - \binom{n-\ell}{1} \cdot \binom{\ell}{1} > \binom{\ell}{2} + \binom{n-\ell}{1} \cdot \binom{\ell}{1},$$

from which we derive the inequality

$$\binom{n}{2} > 2\left(\binom{\ell}{2} + \binom{n-\ell}{1} \cdot \binom{\ell}{1}\right) \tag{6}$$

as a necessary and sufficient condition for the majority rule to succeed at A.

Number of Liars	Min Number of Neighbors
$\ell = 1$	$n = 5$
$\ell = 2$	$n = 8$
$\ell = 3$	$n = 11$
$\ell = 4$	$n = 14$

Table 2. Minimum number of neighbors required for a node to determine a correct pair of locations (using Algorithm 2) in the presence of ℓ liars in its neighborhood.

Table 2 depicts the minimum number of neighbors for a given number of ℓ liars. The table is derived as follows. If $\ell = 1$ then $\binom{\ell}{2} = 0$ and Inequality 6 becomes $n > 4$, which means A can determine a correct pair of locations if it has at least 5 neighbors. If $\ell = 2$ then $\binom{\ell}{2} = 1$ and Inequality 6 becomes $n > \frac{9+\sqrt{33}}{2}$. Unlike the process proposed for Algorithm 1, in this case it is much simpler to solve the inequality and find an exact formula for the minimum number of neighbors required. More generally, when $\ell \geq 2$ then Inequality 6 can be simplified to the inequality

$$n^2 - (4\ell+1)n + 2\ell^2 + 2\ell > 0.$$

Solving the corresponding quadratic we see that

$$n > \frac{4\ell + 1 + \sqrt{8\ell^2+1}}{2} \tag{7}$$

is a necessary and sufficient condition on the number n of neighbors of A so that it can compute a correct pair of locations despite the presence of ℓ liars in its neighborhood.

Resolving the ambiguity — Unlike the process executed in Algorithm 1, the new procedure executed by A does not determine a single location but rather a pair of potential locations for A. We show in the sequel how to guarantee that sensor A will resolve the ambiguity in the pair of locations computed by Algorithm 2.

Assume that A knows there is exactly one liar among its n neighbors. Assuming that $n = 5$, we can use Algorithm 2 to determine a correct pair of locations, say $\{X, X'\}$. Then, the next step is to identify the correct location which must be either X or X'. Since A has exactly 5 neighbors, from which only one is a liar, the remaining four must be truth tellers. However, already two sensors contributed to the correct pair $\{X, X'\}$. Let us assume that they are the first and second nodes, i.e., nodes B_1 and B_2. This leaves us the three sensors B_3, B_4, B_5, out of which the liar must be excluded (cf. Figure 5). Among these three sensors only one is a liar, while the other two point to the correct answer. Therefore using a majority rule among the remaining sensors we can

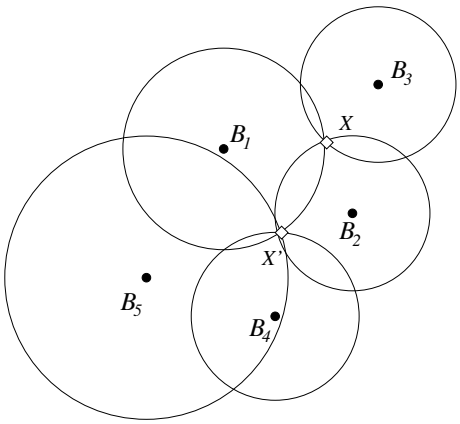

Figure 5. Resolving the ambiguity in the pair of locations computed by Algorithm 2

exclude the liar's location and identify the correct location of sensor A between X and X'.

A similar argument would work for any number ℓ of liars provided that the number of A's neighbors is sufficiently high. The previous argument indicates that sensor A can resolve the ambiguity and exclude the liars by adding the following steps at the end of Algorithm 2:

5: A selects any two sensors that give a correct pair of locations in step 4.

6: A identifies its correct location using majority rule among the sensors remaining after removing the two correct neighbors identified in step 5.

7: A reports the nodes that did not correlate the proper location.

It is easy to show the correctness of the new procedure. Indeed, sensor A identifies a pair of sensors among the ones that give the correct pair of locations after the execution of Algorithm 2. After removing these two neighbors A is left with the remaining $n-2$. Clearly, the ℓ liars must be among these $n-2$ sensors. Therefore, if there is majority of truth tellers among these $n-2$ nodes, then the majority rule will identify the correct location for A between X and X', i.e., if

$$n-2 > 2\ell. \tag{8}$$

However, if n satisfies Inequality 7 then it must also satisfy Inequality 8. The reason is that

$$\frac{4\ell+1+\sqrt{8\ell^2+1}}{2} > 2+2\ell,$$

as the reader can easily check.

4 Secure Localization Using One Trusted Node in the Distance One Neighborhood

Algorithm 3 presents a third algorithm that improves the previous results by relaxing our initial hypotheses. We now assume that sensor A may trust one of the nodes in its distance one neighborhood, say node B_1. Let the neighbors of A in $N_1(A)$ be independent and identically distributed random variables $B_1, B_2, \ldots, B_{n(A)}$ indicating their positions. Then, sensor A calculates for every neighbor $B_i \in N_1(A)$ other than B_1, the pair of points $[(a_x, a_y), (a_{x'}, a_{y'})] = \mathscr{F}(B_1, B_i)$. The random variables obtained by A when applying this process are also independent. Sensor A uses then these random variables and calculates the frequencies of occurrence of each position. A finally selects as a plausible position the most frequently occurring value.

Algorithm 3 Tabulated-TwoNeighborSignals

1: Sensor A requests the location of its neighbors.
2: Every neighbor of A sends its location to A.
// This algorithm is executed by all the neighbors of A.
3: For every neighbor B_i other than B_1, A computes the pair of points $\{X, X'\}$.
// The locations computed are the two points of intersec-
// tion of the two circles centered at B_1, B_i and radii
// $d(A,B_1)$ and $d(A,B_i)$, respectively.
4: A calculates the frequencies of occurrence of each position, accepts as correct the most frequently occurring value, and reports the nodes that did not correlate such a position.
// If there is no any position whose frequency of
// occurrence is, at least, twice the frequency of
// occurrence of the second most frequent position,
// then A declares that it cannot compute its location.

Table 3 depicts the minimum number of neighbors for a given number of liars. The table can be derived as follows. Assume that A knows there is exactly one liar among its n neighbors. Assuming that $n = 4$, we can use Algorithm 3 to determine the most frequently occurring position. Since A has exactly 4 neighbors, from which only one is a liar, the remaining three must be truth tellers. The first truth teller is node B_1, to whom sensor A trusts. Let us as-

Number of Liars	Min Number of Neighbors
$\ell = 1$	$n = 4$
$\ell = 2$	$n = 5$
$\ell = 3$	$n = 6$
$\ell = 4$	$n = 7$

Table 3. Minimum number of neighbors required for a node to determine a correct pair of locations (using Algorithm 2) in the presence of ℓ liars in its neighborhood.

sume that the other two truth tellers are nodes B_2 and B_3. Then, the two pairs of locations provided by the intersection of circles centered at B_1, B_2 and B_1, B_3, respectively, return twice the proper location of A. Since we assume that variables B_1, B_2, B_3, and B_4 are independent and identically distributed random variables, any other position rather than the real location of A will be counted more than once. Therefore A can report that B_4 is the liar and identify the correct location from the set of pairs.

Since liars cannot collude when lying about their position (as it has been established as one of the prerequisites defined in Section 2), a similar argument would work for any number ℓ of liars provided that the number of nodes in $N_1(A)$ is sufficiently high to allow, at least, three truth tellers, i.e., when $n - \ell \geq 3$.

5 Simulations

We conducted simulations to confirm that our algorithms increase the percentage of nodes that can derive their own location in an arbitrary WSN under the presence of liars. We assume that n sensors are located in a random setting whereby they were dropped randomly and independently with the uniform distribution in the interior of a unit square. We also assume that the communication range of each sensor is a circle centered at its position and of radius $r = \sqrt{\frac{\ln n + k \ln\ln n + \ln(k!) + c}{n\pi}}$ as proposed in [3]. Parameter k parametrizes the network connectivity. A network is $k+1$-connected if it remains connected when at most k notes are deleted (i.e., connected corresponds to $k = 0$). The constant c is used to quantify the probability that the network is $k+1$ connected with probability depending on c (cf. [3] and citations thereof). The network is therefore $(k+1)$-connected for any integer $k \geq 0$ and real number constant c. Our simulations assume that both k and c are set to value 1. Figure 6 pictures the average results and the 95% confidence intervals of the simulation of twenty to 100-sensor WSNs. An average of 30% of the sensors, from which a 3% lie about their location, are GPS equipped and can determine their position independently of other sensors. The rest of the sensors execute our set of algorithms as follows. Each node unaware of its location requests the location of its neighbors. It then applies Algorithm 1 using the received information. If the process does not derive a proper location, it then executes Algorithm 2 using the same information. If it fails to derive a proper location, it then seeks a trusted sensor in its distance one neighborhood, executes Algorithm 3, and tabulates the positions collected so far in the previous two steps. If that fails again, it then repeats the whole process later, expecting that the number of neighbors aware of their location increases. The simulation was run for 100 times for each network size. Similarly, Figure 7 pictures the results of the simulation of twenty to 100-sensor networks with confidence levels set to 95%. An average of 30% of the sensors, from which a 10% lie about their location, are GPS equipped and can determine their position independently of other sensors. The simulation was run for 100 times for each network size.

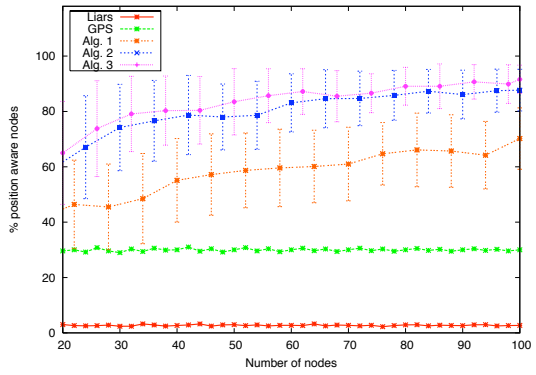

Figure 6. 30% of the sensors are GPS equipped. 3% of the sensors liars.

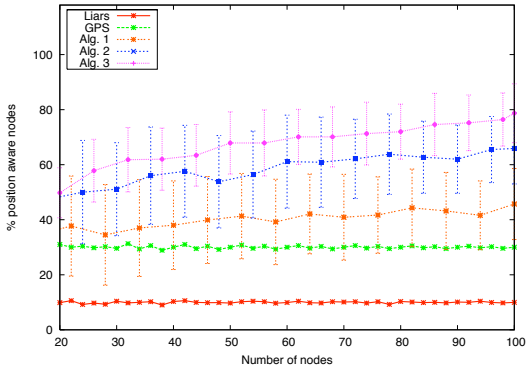

Figure 7. 30% of the sensors are GPS equipped. 10% of the sensors are liars.

We can observe from the simulations depicted in Figures 6 and 7 that both Algorithm 2 and Algorithm 3 increase significantly the number of sensors aware of their position when the average of liars is low. However, as soon as we start increasing the presence of liars in the network, the effectiveness of Algorithm 2 reduces significantly, and leaves the results of Algorithm 3 as the most effective.

6 Related Work

Research in the field of the security of wireless sensor networks is very active at this moment. We can structure the current research lines according to the following themes: (1) security of network services (2) reliability and fault tolerance; (3) security of the infrastructure; (4) distribution and

exchange of keys; and (5) aggregation of data. The contributions presented in this paper are related to the category *security of network services* and, particularly, to issues related to the *routing, location and synchronization of WSN nodes*. The problem of localization in the absence of misbehaving nodes has already been studied in [14, 4, 9, 3]. Most of these approaches base their discovery process on the use and evaluation of distances techniques such as *Received Signal Strength* (RSS) and *Time of Flight* (ToF) [2].

Some more recent approaches propose solutions to the problem of handling secure location of nodes in the presence of misbehaving sensors. Most of these approaches are based on models where there are almost always nodes that must be trusted by the rest of the regular sensors. In [11, 12], we can find some initial work based on this approach. In these proposals, each regular sensor trying to derive its own position proceeds by correlating the messages received by other nodes in the WSN that already are aware of their position (by using, for instance, GPS devices [3]). The use of directional antennas is proposed to improve the security of the localization process. A second solution relies on the use of trust metrics and verifiers [5, 6]. The closest works to ours are the approaches presented in [7, 10]. Both proposals aim at providing a secure location process without the necessity of a priori trust between the nodes of a WSN. The limitation of only giving stochastic guarantees in [10], and the high quantity of messages to exchange in both [7] and [10], of $O(n^2)$ complexity, are the main drawbacks of these approaches.

7 Conclusions

We presented a set of algorithms to handle the localization process of WSN nodes in the presence of liars. The algorithms guarantee the exclusion of incorrect locations, as well as the detection and isolation of the nodes that are lying, if a given threshold of neighbors and liars is met. Otherwise, the algorithms abort the process of deriving the location, and wait to repeat the process again when such parameters can be guaranteed. The two first algorithms allow the localization process without the necessity of a trusted model between sensors. The third algorithm improves the results, but relaxing such an hypothesis, and requesting regular sensors to trust one of the nodes in their one hop neighborhood.

Acknowledgments — The authors graciously acknowledge the financial support received from the following organizations: Natural Sciences and Engineering Research Council of Canada (NSERC), Mathematics of Information Technology and Complex Systems (MITACS), *La Caixa* (Canada awards), the Spanish Ministry of Science and Innovation and the FEDER funds (grants TSI2007-65406-C03-03 E-AEGIS and CONSOLIDER-INGENIO 2010 CSD2007-00004 ARES).

References

[1] J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communications*, 11(6);6–28, 2004.

[2] P. Bahl, V. N. Padmanabhan, and A. Balachandran. Enhancements to the RADAR User Location and Tracking System. *Microsoft Research*, 2000.

[3] M. Barbeau, E. Kranakis, D. Krizanc, and P. Morin. Improving Distance Based Geographic Location Techniques in Sensor Networks. *3rd International Conference on AD-HOC Networks & Wireless (ADHOC-NOW'04)*, pp. 197–210, LNCS 3158, Springer, 2004.

[4] N. Bulusu, J. Heidemann, V. Bychkovskiy, and D. Estrin. Density-adaptive beacon placement algorithms for localization in ad hoc wireless networks. *21th Annual Conference of the IEEE Computer and Communications Societies*, 2002.

[5] S. Capkun and J. P. Hubaux. Secure positioning of wireless devices with application to sensor networks. *24th Annual Conference of the IEEE Computer and Communications Societies*, 2005.

[6] S. Capkun, J. P. Hubaux, and M. Srivastava. Secure Localization with Hidden and Mobile Base Stations. *25th Annual Conference of the IEEE Computer and Communications Societies*, 2006.

[7] S. Delaet, P. Mandal, M. Rokicki, S. Tixeuil. Deterministic secure positioning in wireless sensor networks. *IEEE International Conference on Distributed Computing in Sensor Networks (DCOSS)*, June, 2008.

[8] J. R. Douceur. The Sybil Attack. *Peer-To-Peer Systems: First International Workshop, Iptps 2002, Cambridge, Ma, USA, March 7-8, 2002, Revised Papers*, Springer, 2002.

[9] T. He, C. Huang, B. M. Blum, J. A. Stankovic, T. Abdelzaher. Range-free localization schemes for large scale sensor networks. *9th annual international conference on Mobile computing and networking*, pp. 81–95, ACM, 2003.

[10] J. Hwang, T. He, and Y. Kim. Secure localization with phantom node detection. *Ad Hoc Networks*, 6(7);1031–1050, Elsevier, 2008.

[11] L. Lazos and R. Poovendran. SeRLoc: Robust localization for wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)* 1(1):73–100, 2005.

[12] L. Lazos, R. Poovendran, S. Capkun. ROPE: Robust position estimation in wireless sensor networks. *4th Int'l symposium on Information processing in sensor networks*, 2005.

[13] J. Newsome, E. Shi, D. Song, and A. Perrig. The sybil attack in sensor networks: analysis & defenses. *3rd International Symposium on Information Processing in Sensor Networks*, pp. 259–268, ACM, 2004.

[14] A. Savvides, C. Han, M. Strivastava. Dynamic fine-grained localization in Ad-Hoc networks of sensors. *7th annual international conference on Mobile computing and networking*, pp. 166–179, ACM, 2001.