Positioning of Wireless Sensor Nodes in the Presence of Liars*

J. Garcia-Alfaro†,‡, M. Barbeau†, and E. Kranakis†

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

‡ Institut Télécom, Télécom Bretagne
LUSSI Department, 02 rue de la Châtaigneraie,
CS 17607, 35576 Cesson Sévigné, France

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

Abstract

Positioning of nodes in a Wireless Sensor Network (WSN) is a process that allows location-unaware nodes to discover their spatial coordinates. This process requires the cooperation of all the nodes in the system. Ensuring the correctness of the process, especially in the presence of misbehaving nodes, is crucial for ensuring the integrity of the system. We analyze the problem of unaware nodes determining their location in the presence of misbehaving neighboring nodes that provide false data during the execution of the positioning process. We divide and present potential misbehaving nodes in four different adversary models, based on their capacities. We provide algorithms that enable the location-unaware nodes to determine their coordinates in presence of these adversaries. The algorithms always work for a given number of neighbors provided that the number of misbehaving nodes is below a certain threshold value, which is determined for each adversary model.

Key words: Network Security; Wireless Security; Wireless 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 micro-transceivers, 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. Deployment of nodes in a WSN can be planned or it can be done at random. In planned deployments, sensors are placed into pre-determined locations where the data is collected. In random setups, sensors are deployed into the geographical area and they work together in order to determine their mutual coordinates. We assume random deployment of sensor nodes.

Positioning of nodes is a mechanism that allows location-unaware sensors to discover their spatial coordinates in the network. Several approaches in the literature address the design of localization mechanisms. Different assumptions, regarding the energy and computational capabilities of sensors, arise. Energy accuracy and efficiency of positioning mechanisms have been addressed, for example, in [2, 4, 14]. The correctness of the positioning process in random deployments is very critical and it must be secured in order to ensure the integrity of the WSN and its associated services. Firstly, the process must guarantee that all nodes successfully set up the necessary parameters to establish paths that lead their data towards end users. Secondly, when the relative positions of all the nodes in the system are known, they can be used to enforce the protection of the routing services. The knowledge of their position is also an essential prerequisite for the final application that processes the

*This is an expanded and revised version of a paper that appeared in the proceedings of the 7th Annual Communication Networks and Services Research (CNSR) Conference, May 2009, pages 86–93.

data collected by sensors, i.e., the user needs to know the origin of collected data. Finally, the end users might want to query some nodes by sending the position where information needs to be collected. The positioning process is therefore crucial.

Concerns about the security of the process have been arisen only recently [11, 12, 5, 6, 7, 10]. Most of these approaches are based on the use of trust models, where a few dedicated nodes that are aware of their position (e.g., especial nodes equipped with GPS receivers or nodes that have been manually configured with their location), provide information to regular sensors (unaware of their initial coordinates). Then, the localization process uses the information reported by these special nodes to discover the position of unaware nodes (e.g., by applying trilateration of the radio signals of GPS equipped nodes [3]). These special nodes may in fact be defective. Trusted but defective nodes must be detected and isolated. Otherwise, they can lead to the calculation of false positions and distances. A malicious node 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 the sensors nodes.

Security mechanisms to validate the authentication of trusted nodes is often too expensive and not always realistic. Firstly, the deployment of these nodes must be established a priori, to ensure full coverage of the whole network. Since the cost of these special nodes is considerably higher than the cost of regular sensor 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 their security to mislead, for instance, the positioning process. On the other hand, current approaches to deploy trust on WSNs may require cryptographic operations supported by sensors. This has impact on their battery life, which can degrade their performance. Finally, too much trust may reduce the autonomy of the network, since trusted nodes must be monitored to ensure their integrity. This can be a real problem for applications in hostile environments where the localization phase must be managed by sensors without any external intervention.

We analyze in this paper the problem of unaware nodes determining their location in the presence of misbehaving neighboring nodes that provide wrong information during the execution of the positioning process. We divide and present potential misbehaving nodes in four different adversary models, based on their capacities. These misbehaving nodes are either controlled by a malicious adversary or simply nodes that fail providing the appropriate information due. In the first case, we assume that malicious nodes controlled by an adversary aim at leading unaware nodes to the calculation of false positions and distances. In the second case, we assume honest nodes that unintentionally provide wrong distances or positions due, for instance, to physical obstacles or any other unexpected circumstances. We then provide a set of algorithms that enable the location-unaware nodes to determine their coordinates in the presence of the adversary models defined in our work. The whole set of algorithms that we present guarantee that location-unaware 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. The purpose of our algorithms is to provide a formal process that allows the location-unaware nodes to identify and isolate nodes that are providing false information about their position. Our algorithms are resistant to attacks provided that the thresholds that we define are satisfied. They also guarantee a small exchange of data between nodes, minimizing in this manner the impact that the positioning process has in terms of energy and battery life of the sensor nodes.

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

2 Positioning in the Presence of Liars

We assume that the positioning process is based on trilateration [3]. 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 that 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 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

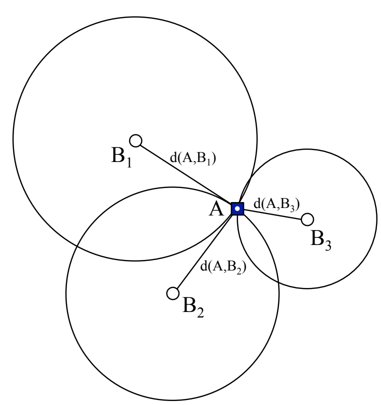


Figure 1. Sensor A wants to determine its location. It receives radiolocation signals from three nodes B_1, B_2, and B_3 that are located in its distance one neighborhood. A determines its position by processing the three signals.

calculated by sensor A when it receives the coordinates $B_1 = (b_{1x}, b_{xy}), B_2 = (b_{2x}, b_{2y}), B_3 = (b_{3x}, b_{3y})$. It measures the distances $d(A,B_1), d(A,B_2), d(A,B_3)$ using radiolocation techniques [2]. The unknown coordinates of $A = (a_x, a_y)$ is obtained as the unique solution of the following system of equations:

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

Consider now that sensor A may receive radiolocation signals from misbehaving nodes that lie by announcing incorrect locations or distances 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?

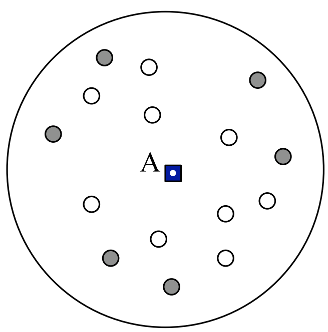


Figure 2. Sensor A is receiving its radiolocation signals from two types of sensors in its distance one neighborhood: liars (gray circles) and truth tellers (blank circles).

2.1 Definitions and Assumptions

We define a liar as any node announcing erroneous information (either distances or coordinates) to a target node. The intent can be *malicious* (i.e., to mislead the target node into the wrong calculation of its location) or *unintentional* in the sense that obstacles or other physical circumstances (e.g., multi-path interference) prevent a sensor from announcing its correct location. We assume the use of a two dimension space and euclidean distances without estimation errors. Therefore, given two locations $(x,y), (x',y')$ a node can determine whether or not they are equal, thus rejecting one of the two. The following assumptions also apply: (1) communications channels are bidirectional, i.e., if node A can hear node B, then node B can hear node A; (2) legitimate nodes (i.e., truth tellers) agree on a fixed communication range (e.g., all truth tellers emit using the same signal power); (3) There are sufficient density conditions (e.g., > 10 one-hop neighbors per node) in the system; (4) Nodes can only hold a single identity (i.e., we do not address Sybil attacks) and are in general positions (i.e., no three sensors are collinear).

2.2 Adversary Models

We define the capabilities of the adversaries as follow:

- **(EV2)**: Eavesdropping communications between a target A and, at least, two truth tellers B_1 and B_2, to forge the coordinates of a position A' (that is consistent with A, B_1, and B_2).
- **(EV1)**: Eavesdropping communications between a target A and, at least, one truth teller B, to forge the coordinates of a position X (that is consistent with A and B).
- **(PT)**: Position Tampering whereby an adversary lies about its position.
- **(DT)**: Distance Tampering whereby an adversary lies about its distance.
- **(CL)**: Construction of a covert-channel and collusion, whereby two or more adversaries collude to exchange system data and supply the victim node with wrong information.

Based on these definitions, we classify in the sequel four main categories of liars.

Model 1 (Unconstrained Liars)

A liar node in this model is assumed to be capable of performing (EV2) + (PT) + (DT) + (CL), i.e., it is capable of eavesdropping the communications of a target victim and two truth-tellers (to forge a position that is consistent with the three of them), capable of tampering consistent positions and distances (only one is enough), and capable of building up a covert-channel to collude with other liars.

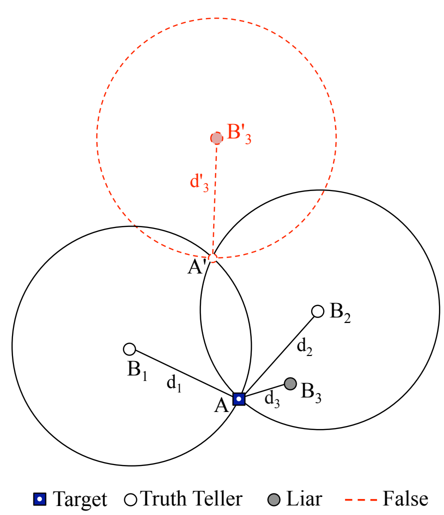
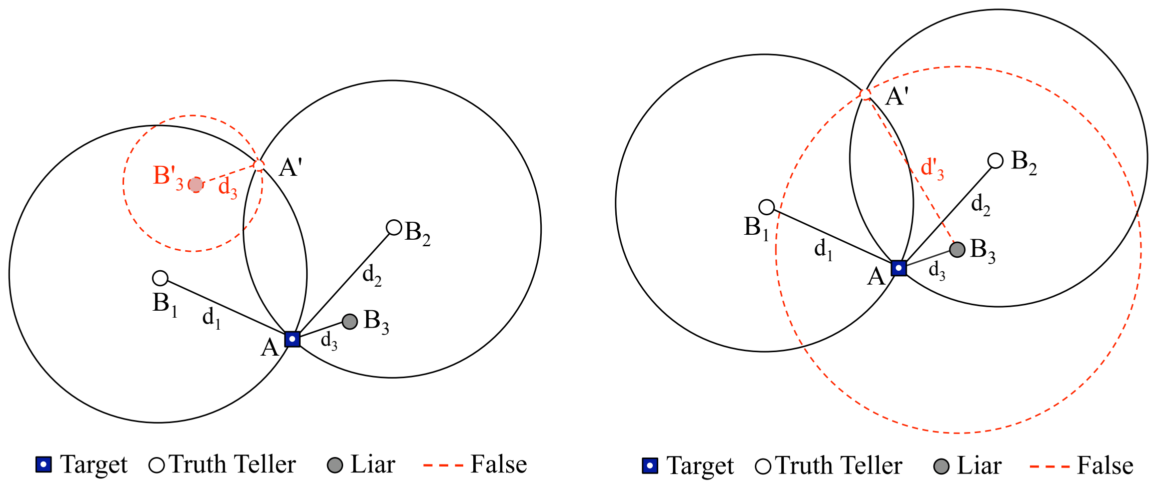
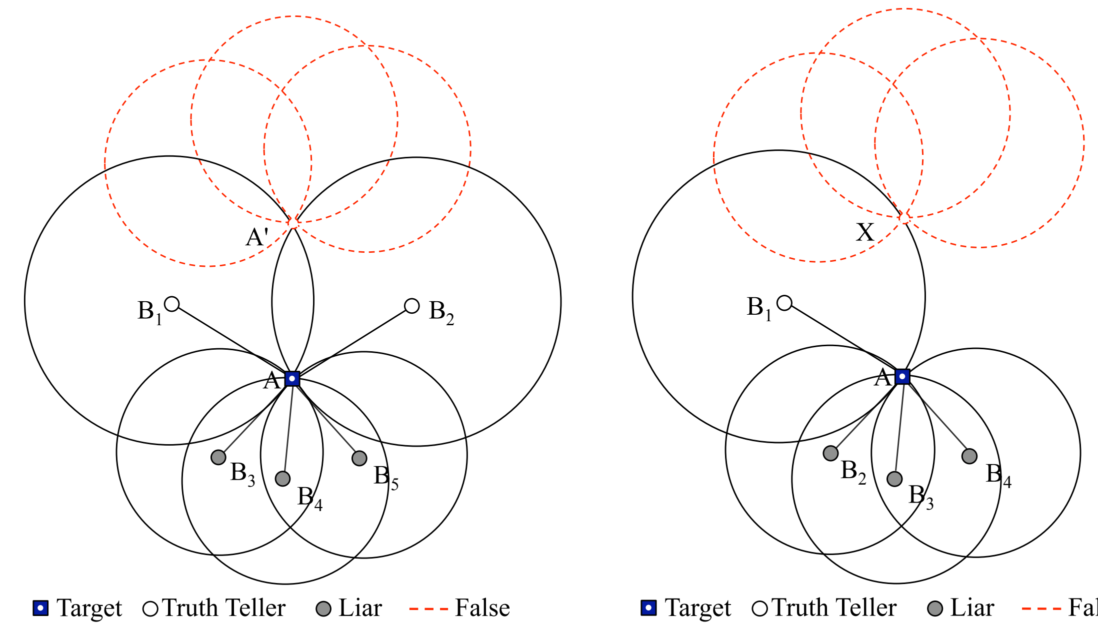


Figure 3. Example of Adversary Model 1.



(a) Node B_3 tampers its position to B_3' (where $B_3' \neq B_3$).

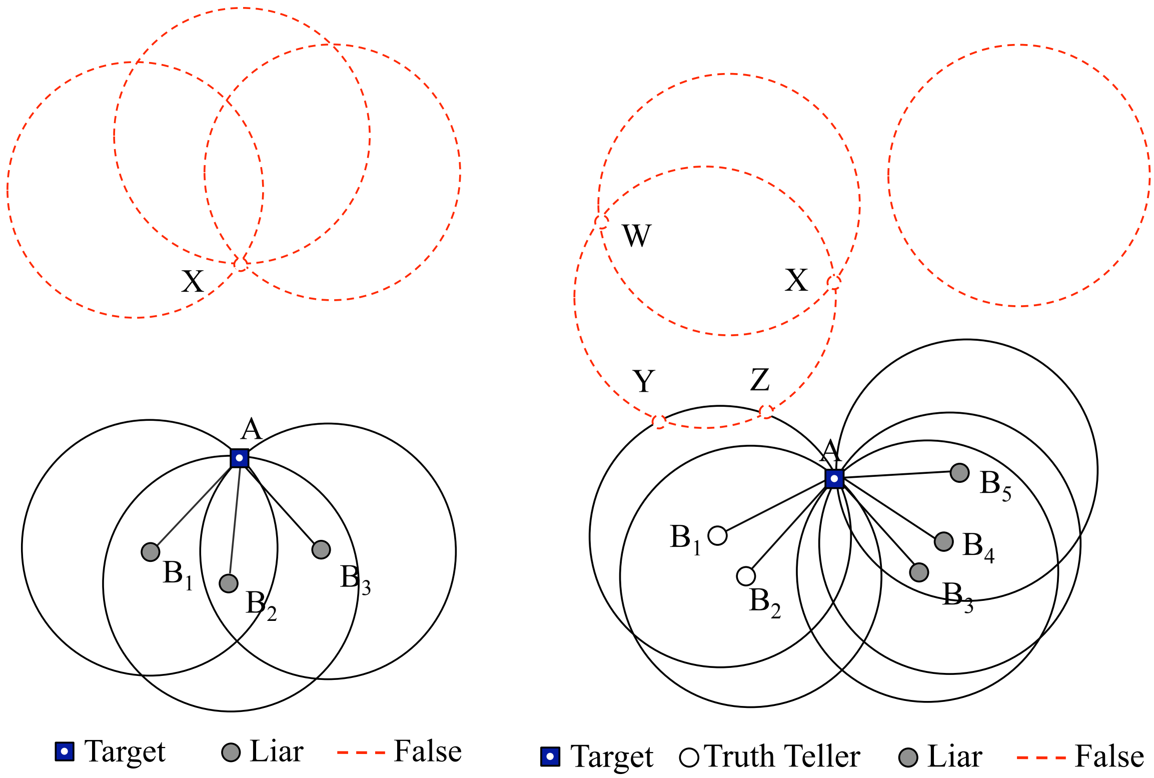
(b) Node B_3 tampers its distance to node A as d_3' (where $d_3' \neq d_3$).



(c) Three liars eavesdrop the communications between target node A and truth tellers B_1 and B_2. With this information, the liars collude and tamper their positions and distances to force the target to conclude that its position is A' instead of A.

(d) Three liars eavesdrop the communications between target node A and truth teller B_1. With this information, the liars collude and tamper their positions and distances to A, to force the target to conclude that it is positioned at location X.

Figure 4. Examples for Adversary Models 1 and 2.



(a) Three liars tamper their positions and distances. They collude and force A to conclude that it is located at X.

(b) Three liars tamper their positions and distances. They do not collude.

Figure 5. Examples for Adversary Models 3 and 4.

Example depicted by Figure 3 shows that if a liar node in this model can eavesdrop the communications between, at least, two truth tellers and the target node, it can then tamper its position and distance, to successfully steal the coordinates of a legitimate position A'. In this sense, we can see first that liar node B_3 eavesdrops the communications between truth teller B_1 and target A, and computes distance d_1. Second, liar node B_3 eavesdrops the communications between truth teller node B_2 and target node A, and computes distance d_2. Using this information, liar node B_3, that is located at a distance d_3 from target A, computes distance d_2' (where $d_3' \neq d_3$) and position B_3' (where $B_3' \neq B_3$). Figure 4 shows that by only tampering its position (cf. Figures 4(a)) or its distance 4(b)), node B_3 can also steal the coordinates of a node to later lead the target noed to conclude that its location is A' instead of A. Finally, we can see in the example depicted by Figure 4(c) that when multiple liar nodes applying this first adversary model in the system successfully collude, e.g., by means of a covert-channel, they can lie consistently to target the node A and lead it to the calculation of its position as A' instead of A.

Model 2 (Partially Constrained Liars)

A liar node in this model is assumed to be capable of performing (EV1) + (PT) + (DT) + (CL), i.e., it can eavesdrop the communications of a target victim and one truth-teller (to forge a position that is consistent with the two of them), tamper consistent positions and distances (only one is enough), and build up a covert-channel to collude with other liars. The example depicted by Figure 4(d) shows that when multiple liar nodes in the system may perform the previous actions, they can eventually collude to lie consistently in order to target A and lead it to the calculation of its position as X instead of A.

Model 3 (Fully Constrained Liars)

A liar in this third model is not assumed to be capable of eavesdropping the communications between the target A and any of the truth tellers in its neighborhood. It is only assumed to be capable of performing (PT) + (DT) + (CL), i.e., it can tamper its position or distance (only one is enough), and collude with other liars (by means of a covert-channel) to lie consistently about a unique bogus position. Example depicted by Figure 5(a) shows an example where multiple liar nodes applying this model in the system can eventually collude to lie consistently to target A and lead it to the calculation of its position as X instead of A.

Model 4 (Unintentional Liars)

A liar in this model is not assumed to be capable of eavesdropping the communications between a target victim and any of the truth tellers in its neighborhood. It is not assumed either to collude with other liars. A liar here is only capable of, probably unintentionally, performing (PT) + (DT), i.e., capable of tampering its position, its distance to the target, or both. Example depicted by Figure 5(b) shows three liar nodes that are unintentionally announcing false distances and coordinates to target node A. They do not collude. The positions derived by A using these three unintentional liars intersects in at most one point (if any).

3 Algorithms and Upper Bounds

We present algorithms that solve the problem of determining the proper location of nodes in the presence of liars according to the adversary models defined in Section 2.2. The algorithms aim not only at determining the proper location but also at excluding the incorrect locations and at isolating the liars. We assume the case where A knows a priori the upper bound ℓ of sensor nodes lying in the geographical area where it 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, while minimizing the necessary number of neighbors that location-unaware sensor nodes must trust.

Section 3.1 presents three algorithms that consist of the following approach. Sensor A, after receiving the radiolocation signals from its one hop neighbors, calculates its position using the localization technique discussed in Section 2.1 (cf. Figure 1), and uses either a majority decision rule (cf. Algorithms 1 and 2) or a most frequent decision rule (cf. Algorithm 3) to derive the position. We provide the conditions for the validity of these three algorithms in the presence of liars applying the adversary models presented in Section 2.2. We present the upper bounds for each case, all of them depending on the number of one hop neighbors and liars among them. Section 3.2 relaxes the initial hypotheses and assumes that a victim may always trust one of the nodes in its distance one neighborhood. We present algorithms, and their bounds, for this second scenario.

3.1 Positioning without Trusted Nodes

Algorithm 1 enables a location-unaware node to determine its position in presence of neighbors applying any adversary model. Following is the analysis.

Theorem 1 *Let n be the number of distance one neighbors nodes of a location-unaware sensor A, the execution of the majority rule in Algorithm 1 by A always gives its correct position in the presence of ℓ liars if inequality $n^3 - 3(2\ell+1)n^2 + 2(3\ell^2+6\ell+1)n - (2\ell^3+6\ell^2+4\ell) > 0$ is satisfied.*

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_j, 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 fails compute its location.

Proof Given n one hop neighbors and the presence of ℓ liars applying any of the models defined in Section 2.2, 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 2 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 is simplified

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

Number of Liars	Min Number of Neighbors
$\ell = 1$	$n = 7$
$\ell = 2$	$n = 11$
$\ell = 3$	$n = 16$
$\ell = 4$	$n = 21$
$\ell = 5$	$n = 26$
$\ell = 10$	$n = 31$
$\ell = 15$	$n = 74$
$\ell = 20$	$n = 98$

Table 1. Minimum number of location-aware neighbor nodes required for a location-unaware node to determine a correct pair of locations (using Algorithm 1) in the presence of ℓ liars applying any of the adversary models defined in Section 2.2.

to $n > 6$, then 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 $n > \frac{13+\sqrt{73}}{2}$. This means that A can determine a correct location in the presence of two liars if it has at least eleven neighbors. When $\ell \geq 3$, 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 we can obtain the rest of values depicted in Table 1. Figure 6 shows the minimum number of neighbors for $\ell = 3$ and $\ell = 4$.

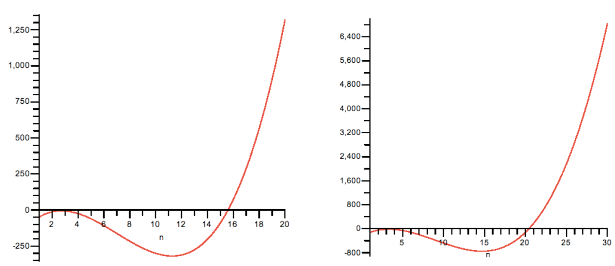


Figure 6. Plotting the minimum neighborhood size n as a function of the number of liars ℓ so as to guarantee that inequality (5) is true for $\ell = 3$ (left diagram) and $\ell = 4$ (right diagram).

We can, therefore, affirm that inequality (5) gives the necessary and sufficient upper bound on the number n of neighbors of a location-unaware node so that it can compute a correct and unique position despite the presence of ℓ liars of any model call in its neighborhood. □

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 $(x_p, y_p), (x'_p, y'_p)$.
// The locations computed are the two points of
// intersection of the two circles centered at B_i, B_j
// with radii $d(A, B_i)$ and $d(A, B_j)$, respectively.
4: A calculates the frequencies of occurrence of each position and accepts the position that has majority. It reports the nodes lying about the resulting position.
// If there is no consensus, then A aborts the process, and
// declares that it fails to compute its location.

Improving the Previous Approach

Algorithm 2 describes a process in which a sensor A uses only the radiolocation signals of two neighbors to derive its position. The correct location is one of the two points of intersection of two circles centered at these two neighbors. To handle the existence of neighboring liars, sensor A computes 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, with radii $d(A, B_i), d(A, B_j)$, respectively. As depicted in Figure 7, the correct location of sensor A is either X or X'. A uses the majority rule to determine the most plausible position and to report nodes that lied about their location or distances.

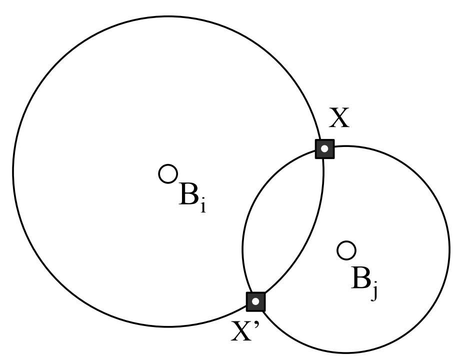


Figure 7. Sensor A applying Algorithm 2.

Theorem 2 *The execution of the majority rule in Algorithm 2 by a location-unaware sensor node always gives the correct position in the presence of any ℓ liars if the number of its distance one neighbors exceeds $\frac{4\ell+1+\sqrt{8\ell^2+17}}{2}$.*

Proof In the presence of ℓ liars applying any of the adversary models defined in Section 2.2, and given n one hop neighbors, the majority rule in Algorithm 2 succeeds if the number of *correct* pairs of locations is bigger than the number of *incorrect* pairs of locations. We assume the strongest adversary model (i.e., Unconstrained Liars), in which liars can eavesdrop the communications from, at least, two truth tellers — say nodes B_1 and B_2. Therefore, a pair of locations is correct if it is determined by any two truth tellers other than B_1 and B_2; otherwise, it is (possibly) *incorrect*. Consider all pairs of (possibly) incorrect locations. Such pairs can have either

1. exactly the two sensor nodes whose communications were eavesdropped, or
2. both sensors are liars, for a total of $\binom{\ell}{2}$ pairs, or
3. exactly one sensor is a liar, for a total of $\binom{n-\ell}{1} \cdot \binom{\ell}{1}$ pairs.

The majority rule in Algorithm 2 therefore succeeds if the following inequality is satisfied

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

Number of Liars	Min Number of Neighbors
$\ell = 1$	$n = 6$
$\ell = 2$	$n = 9$
$\ell = 3$	$n = 12$
$\ell = 4$	$n = 15$
$\ell = 5$	$n = 18$
$\ell = 10$	$n = 35$
$\ell = 15$	$n = 52$
$\ell = 20$	$n = 69$

Table 2. Minimum number of location-aware neighbor nodes required for a location-unaware node to determine a correct pair of locations (using Algorithm 2) in the presence of ℓ liars applying any of the adversary models defined in Section 2.2.

Table 2 depicts the required minimum number of neighbors for a given number of any ℓ liars. The table is derived as follows. If $\ell = 1$ then $\binom{\ell}{2} = 0$ and Inequality 6 becomes $n > 5$, which means A can determine a correct pair of locations if it has at least 6 neighbors. If $\ell = 2$ then $\binom{\ell}{2} = 1$ and Inequality 6 becomes $n > \frac{9+\sqrt{49}}{2}$. More generally, when $\ell \geq 3$ then Inequality 6 can be simplified as the following inequality

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

Solving the corresponding quadratic equation, we see that

$$n > \frac{4\ell + 1 + \sqrt{8\ell^2 + 17}}{2} \qquad (7)$$

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. □

Theorem 3 *A location-unaware sensor node always derives a unique position from the execution of Algorithm 2 in the presence of ℓ liars if the number of its distance one neighbors exceeds $2\ell + 2$.*

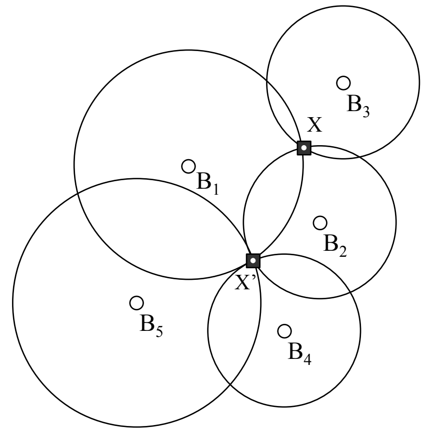


Figure 8. Resolving the ambiguity in the pair of locations computed by Algorithm 2

Proof 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, in 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 a liar must be excluded (cf. Figure 8). 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 exclude the liar's location and identify the correct location of sensor A among 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 the 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 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 identifies 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

$$\frac{4\ell+1+\sqrt{8\ell^2+17}}{2} > 2\ell+2,$$

as the reader can easily check. □

Using Most Frequent Rule

Based on our previous result (cf. Inequality 8), and assuming the use of a most frequent rule instead of the majority rule, we present in Algorithm 3 an alternative process that allows a location-unaware node A to find its correct position with a weaker constraint between the number of neighbors and the number of liars nodes.

Algorithm 3 MostFrequent-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 ∈ N_1(A), A computes
      (x_p, y_p), (x'_p, y'_p).
      // The locations computed are the two points of
      // intersection of the two circles centered at B_i, B_j
      // with radii d(A, B_i) and d(A, B_j), respectively.
4: A calculates the frequencies of occurrence of each po-
      sition, accepts as correct the most frequently occur-
      ring value, and reports the nodes lying about it.
      // 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 aborts the process, and declares failure to
      // compute its location.
```

Table 3 compares the minimum number of neighbors and number of liars to satisfy the most frequent rule in Algorithm 3 for each adversary model.

In the sequel, we provide sufficient conditions to derive the values contained in the table.

Theorem 4 *The execution of the most frequent rule in Algorithm 3 by a location-unaware sensor node always gives the correct position in the presence of ℓ liars applying the first model (Unconstrained Liars) if the number of its distance one neighbors exceeds $2\ell+2$.*

# of Liars	Min # of Neighbors			
	Model 1	Model 2	Model 3	Model 4
$\ell=1$	$n=5$	$n=4$	$n=4$	$n=4$
$\ell=2$	$n=7$	$n=6$	$n=6$	$n=5$
$\ell=3$	$n=9$	$n=8$	$n=7$	$n=6$
$\ell=4$	$n=11$	$n=10$	$n=9$	$n=7$
$\ell=5$	$n=13$	$n=12$	$n=11$	$n=8$
$\ell=10$	$n=23$	$n=22$	$n=21$	$n=13$
$\ell=15$	$n=33$	$n=32$	$n=31$	$n=18$
$\ell=20$	$n=43$	$n=42$	$n=41$	$n=23$

Table 3. Comparison of minimum number of neighbors required for a node to determine a correct location (using the most frequent rule defined in Algorithm 3) in the presence of ℓ liars applying the set of adversary models defined in Section 2.2.

Proof In the presence of ℓ liars applying the first model (*Unconstrained Liars*), the most frequent rule in Algorithm 3 succeeds if the number of pairs pointing to the correct location (i.e., the $\binom{n-\ell}{2}$ pairs where both nodes are truth tellers) is bigger than the number of incorrect pairs pointing to the most frequent false position. The most frequent false position can be derived from those pairs that have either

1. exactly the two truth tellers whose communications are eavesdropped by the ℓ liars, for a total of one pair, or
2. exactly one liar and one of the two truth tellers whose communications are eavesdropped, for a total of 2ℓ pairs, or
3. exactly two liars, for a total of $\binom{\ell}{2}$ pairs.

This amounts to having

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

as a necessary and sufficient condition for the most frequent rule to succeed at A. Solving the corresponding quadratic equation, the previous inequality can be simplified as

$$n > \frac{2\ell+1+\sqrt{(2\ell+3)^2}}{2} = 2\ell+2$$

as a necessary and sufficient condition for the most frequent rule to succeed at the correct position. □

Theorem 5 *The execution of the most frequent rule in Algorithm 3 by a location-unaware sensor node always gives the correct position in the presence of ℓ liars applying the second adversary model (Partially Constrained Liars) if the number of its distance one neighbors exceeds $2\ell+1$.*

Proof In the presence of ℓ liars applying the second model (*Partially Constrained Liars*), the most frequent rule in Algorithm 3 succeeds if the number of correct pairs of locations (i.e., the $\binom{n-\ell}{2}$ pairs where both nodes are truth tellers) is bigger than the pairs that have either

1. exactly one liar and the truth teller whose communications are eavesdropped, which gives a total of ℓ pairs, or
2. both sensors liars, which gives a total of $\binom{\ell}{2}$ pairs.

Algorithm 3, therefore, succeeds if

$$\binom{n-\ell}{2} > \ell + \binom{\ell}{2}$$

is satisfied. It can be simplified as

$$n > \frac{2\ell+1+\sqrt{(2\ell+1)^2}}{2} = 2\ell+1$$

as a necessary and sufficient condition for the most frequent rule to succeed at the correct position. □

Theorem 6 *Given n one hop neighbors and ℓ liars applying the third adversary model (Fully Constrained Liars). The execution of the most frequent rule in Algorithm 3 by a location-unaware sensor requires $n > \ell+2$ distance one hop neighbors when $\ell = 1$; and $n > 2\ell$ distance one neighbors when $\ell > 1$.*

Proof In the presence of ℓ liars applying the third model, the most frequent rule in Algorithm 3 succeeds if the number of correct pairs is bigger than the number of incorrect pairs where exactly both nodes are liars, i.e., if the following inequality is satisfied

$$\binom{n-\ell}{2} > \binom{\ell}{2} \tag{9}$$

The case of $\ell = 1$, and so $\binom{\ell}{2} = 0$ represents an exception, since even in the case of a single liar, the number of correct pairs must bigger than one. In this case, we assume that Inequality 9 must be replaced by

$$\binom{n-\ell}{2} > 1$$

which can be simplified as

$$n > \frac{2\ell+1+\sqrt{9}}{2} = \ell+2$$

as a necessary and sufficient condition for the most frequent rule to succeed at the correct position when $\ell = 1$.

Otherwise, when $\ell > 1$, Inequality 9 is just simplified as

$$n > \frac{2\ell+1+\sqrt{(2\ell-1)^2}}{2} = 2\ell$$

as a necessary and sufficient condition for the most frequent rule to succeed at the correct position. □

Theorem 7 *The execution of the most frequent rule in Algorithm 3 by a location-unaware sensor node always gives the correct position in the presence of ℓ liars according to the fourth adversary model (Unintentional Liars) if the number of its distance one neighbors exceeds $\ell+2$.*

Proof In the presence of ℓ liars applying the fourth model (*Unintentional Liars*), the most frequent rule in Algorithm 3 always succeeds in computing the correct location if the number of correct pairs is, at least, twice the frequency of occurrence of the second most frequent position. Since liars modeling this last case scenario do not collude, it suffices to satisfy the following inequality:

$$\binom{n-\ell}{2} > 1$$

Solving the corresponding quadratic equation, the previous inequality can be simplified as

$$n > \frac{2\ell+1+\sqrt{9}}{2} = \ell+2$$

as a necessary and sufficient condition for the most frequent rule to succeed at the correct position. □

3.2 Positioning with One Trusted Node

We relax now our initial hypotheses. We suppose, in addition to the assumptions defined in Section 2.1, that any target A in the system may always trust exactly one of the nodes in its distance one neighborhood, say node B_1. We adapt Algorithms 1, 2, and 3 to the positioning processes defined in Algorithms 4, 5, and 6. Following is the analysis.

Majority Rule plus One Trusted Node

Algorithms 4 and 5 define the use of a majority rule to enable location-unaware nodes to determine their position in presence of liars. The upper bounds of these two algorithms for all the adversary models is analyzed in the sequel.

Theorem 8 *The execution of the majority rule in Algorithm 4 by a location-unaware sensor node always gives the correct position in the presence of any ℓ liars if the number of its distance one neighbors exceeds $\frac{4\ell+3+\sqrt{8\ell^2+1}}{2}$.*

Proof Given n one hop neighbors and the presence of ℓ liars applying any adversary model, consider from all possible triples of sensors for every two neighbors B_i, B_j plus the trusted node B_1 (i.e., a total of $\binom{n-1}{2}$ triples) such that at least one of the sensors in the triple is a liar. Such a triple can have in each case either

1. exactly two liars, which gives a total of $\binom{\ell}{2}$ triples, or
2. exactly one liar (and the other two, say B_1 plus B_i are truth tellers), which gives a total of $\binom{n-1-\ell}{1} \cdot \binom{\ell}{1}$ triples.

A location that is determined by A is correct if it is provided by three truth tellers; otherwise it is (possibly) *incorrect*. Therefore, the majority rule in Algorithm 4 will succeed at A if

$$\binom{n-1}{2} > 2\left[\binom{\ell}{2} + \binom{n-1-\ell}{1} \cdot \binom{\ell}{1}\right] \tag{10}$$

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

Inequality 10 can be simplified as the following inequality

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

Solving the corresponding quadratic equation, we see that

$$n > \frac{4\ell + 3 + \sqrt{8\ell^2 + 1}}{2} \tag{11}$$

is a necessary and sufficient condition on the number of neighbors of A so that it can compute a correct location despite the presence of any ℓ liars in its neighborhood. $\square$

Algorithm 4 Majority-ThreeNeighborSignals-plus-One-Trusted-Neighbor

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 each triple t of neighbors $B_1, B_i, B_j \in N_1(A)$, A computes (x_t, y_t).
// (x_t, y_t) is the point of intersection of the three circles
// centered at B_1, B_i, B_j and with radii $d(A, B_1)$,
// $d(A, B_i)$, and $d(A, B_j)$.
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 fails compute its location.

Theorem 9 *The execution of the majority rule in Algorithm 5 by a location-unaware sensor node always gives the correct position in the presence of ℓ liars applying any adversary model if the number of its distance one neighbors exceeds $2\ell + 3$.*

Algorithm 5 Majority-TwoNeighborSignals-plus-One-Trusted-Neighbor

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
// intersection of the two circles centered at B_1, B_i
// with 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 position that has majority, and reports the nodes that did not correlate such a position.
// If there is no consensus, then A aborts the process,
// and declares that it fails to compute its location.

Proof Algorithm 5 only computes one pair of positions for every neighbor B_i other than the trusted node B_1. This amounts to having $\binom{n-1}{1}$ pairs of locations, from which $\binom{\ell}{1}$, are (possibly) incorrect. Algorithm 5 therefore succeeds at A if

$$\binom{n-1}{1} - \binom{\ell}{1} > \binom{\ell}{1}$$

from which we derive

$$\binom{n-1}{1} > 2\left[\binom{\ell}{1}\right] \tag{12}$$

Inequality 12 can be simplified as

$$n > 2\ell + 1$$

Notice, however, that this upper bound is inferior to the bound obtained in Section 3.1, Theorem 4, in which we proved that in the worst case scenario of liars applying the adversary model 1, there are exactly $2\ell + 2$ potential false positions. We should, therefore, consider here again that liars are capable of eavesdropping the communications from B_1 and, at least, another truth teller, say B_2. In this case, from all $\binom{n-1}{1}$ pairs of positions, we must also discard the pair containing nodes B_1 and B_2. If so, the majority rule in Algorithm 5 therefore succeeds if

$$\binom{n-1}{1} - 1 - \binom{\ell}{1} > 1 + \binom{\ell}{1}$$

from which we derive

$$n > 2\ell + 3$$

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

Most Frequent Rule plus One Trusted Node

Algorithm 6 defines the use of a most frequent rule to enable location-unaware nodes to determine their position in presence of liars. The upper bounds for each adversary model differ. Following is the analysis.

Theorem 10 *The execution of the most frequent rule in Algorithm 6 by a location-unaware sensor node always gives the correct position in the presence of ℓ liars applying in the system adversary models 1, 2, 3, and 4, if the number of its distance one neighbors exceeds, respectively,* $2\ell+2$, $2\ell+1$, $\ell+2$, *and* $\ell+2$.

Proof The most frequent rule in Algorithm 6 always succeeds in the first adversary model (Unconstrained Liars) if the number of *correct* pairs of locations (i.e., $n-1-\ell$) is greater than the number of *incorrect* pairs of locations. We assume in this adversary model that liars are capable of eavesdropping the communications between the trusted node B_1 and, at least, another truth teller, say node B_2. They can, therefore, collude to lead A to compute $\ell+1$ incorrect, but consistent, pairs of locations: the false position is contained, at least, in the pair $\{B_1, B_2\}$; and in the ℓ pairs composed by B_1 and each of the ℓ liars. We can, therefore, derive the following upper bound

$$n-1-\ell > \ell+1$$

which can be simplified as $n > 2\ell+2$.

In the second adversary model (Partially Constrained), liars can only eavesdrop, at most, the communications between the trusted node and the target. Liars colluding can only successfully lead A to compute ℓ times a false position that is, however, consistent with node B_1. The most frequent rule in Algorithm 6 always succeeds in these two cases if inequality $n-1-\ell > \ell$, i.e., $n > 2\ell+1$, is satisfied.

Liars applying the third adversary model (Fully Constrained Liars) cannot eavesdrop communications. They cannot collude either, since no two liars can now appear together in any pair of positions. Therefore, the upper bound of Algorithm 6 in the presence of liars applying the third model is equivalent to the upper bound of Algorithm 6 in the presence of liars applying the fourth model (Unintentional Liars), i.e., liars that neither collude nor eavesdrop the communications with the trusted node. The most frequent rule in these two cases succeeds if $n-1-\ell > 1$, i.e., $n > \ell+2$ is satisfied. □

Algorithm 6 MostFrequent-TwoNeighborSignals-plus-One-Trusted-Neighbor

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
   // intersection of the two circles centered at B_1, B_i
   // with radii $d(A,B_1)andd(A,B_i)$, 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 aborts the process, and declares failure
   // to compute its location.

3.3 Comparison of Results

The scenario presented in Section 3.2 only improves the bounds for satisfying the majority rule in Algorithms 4 and 5 that, compared with the ones of Algorithms 1 and 2, get lower. Table 4 compares the minimum number of neighbors to satisfy the majority rule in Algorithms 1, 2, 4, and 5 to succeed in the presence of ℓ liars applying any of the adversary models defined in Section 2.2.

Majority Rule in Algorithms. 1, 2, 4, and 5				
# of Liars	Min # of Neighbors			
	Alg. 1	Alg. 4	Alg. 2	Alg. 5
$\ell=1$	$n=7$	$n=6$	$n=6$	$n=6$
$\ell=2$	$n=11$	$n=9$	$n=9$	$n=8$
$\ell=3$	$n=16$	$n=12$	$n=12$	$n=10$
$\ell=4$	$n=21$	$n=16$	$n=15$	$n=12$
$\ell=5$	$n=26$	$n=19$	$n=18$	$n=14$
$\ell=10$	$n=31$	$n=36$	$n=35$	$n=24$
$\ell=15$	$n=74$	$n=53$	$n=52$	$n=34$
$\ell=20$	$n=98$	$n=70$	$n=69$	$n=44$

Table 4. Comparison of the minimum number of neighbors required for the majority rule in Algorithms 1, 2, 4, and 5 to succeed in the presence of ℓ liars applying any of the adversary models defined in Section 2.2.

Notice, however, that the rest of bounds for satisfying the most frequent rule in Algorithm 6 remain exactly the same as that for Algorithm 3. Only the case of the third adversary model (Fully Constrained Liars) changes. In fact, liars applying the third adversary model in this new scenario lose their capability of colluding with other liars, and their upper bound gets reduced to the same limit that also applies to the fourth adversary model (Unconditional Liars). We show in Table 5 a comparison between the minimum number of neighbors to satisfy the most frequent rule in Algorithms 3 and 6 to succeed in the presence of ℓ liars applying any of the adversary models defined in Section 2.2.

Most Frequent Rule in Algorithm 3				
# of Liars	Min # of Neighbors			
	Model 1	Model 2	Model 3	Model 4
$\ell = 1$	$n = 5$	$n = 4$	$n = 4$	$n = 4$
$\ell = 2$	$n = 7$	$n = 6$	$n = 6$	$n = 5$
$\ell = 3$	$n = 9$	$n = 8$	$n = 7$	$n = 6$
$\ell = 4$	$n = 11$	$n = 10$	$n = 9$	$n = 7$
$\ell = 5$	$n = 13$	$n = 12$	$n = 11$	$n = 8$
$\ell = 10$	$n = 23$	$n = 22$	$n = 21$	$n = 13$
$\ell = 15$	$n = 33$	$n = 32$	$n = 31$	$n = 18$
$\ell = 20$	$n = 43$	$n = 42$	$n = 41$	$n = 23$

Most Frequent Rule in Algorithm 6				
# of Liars	Min # of Neighbors			
	Model 1	Model 2	Model 3	Model 4
$\ell = 1$	$n = 5$	$n = 4$	$n = 4$	$n = 4$
$\ell = 2$	$n = 7$	$n = 6$	$n = 5$	$n = 5$
$\ell = 3$	$n = 9$	$n = 8$	$n = 6$	$n = 6$
$\ell = 4$	$n = 11$	$n = 10$	$n = 7$	$n = 7$
$\ell = 5$	$n = 13$	$n = 12$	$n = 8$	$n = 8$
$\ell = 10$	$n = 23$	$n = 22$	$n = 13$	$n = 13$
$\ell = 15$	$n = 33$	$n = 32$	$n = 18$	$n = 18$
$\ell = 20$	$n = 43$	$n = 42$	$n = 23$	$n = 23$

Table 5. Comparison of minimum number of neighbors required for a node to determine a correct location (using Algorithms 3 and 6) in the presence of ℓ liars.

4 Simulations

We conducted simulations to confirm that our algorithms increase the percentage of nodes that can derive their location in an arbitrary WSN under the presence of liars. We assume that m sensors are located in a random setting whereby they were distributed randomly and uniformly within a unit square. We also assume that the communication range of each sensor is a circle centered at its position and of radius $r = \sqrt{\frac{\ln m + k \ln\ln m + \ln(k!) + c}{m\pi}}$ as proposed in [3]. Parameter m determines the number of nodes in the network. Parameter k determines the network connectivity. A network is $k+1$-connected if it remains connected when at most k nodes are deleted (i.e., connected corresponds to $k = 0$). The constant c determines 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.

We run two sets of simulations. The first set represents 50 to 250-sensor WSNs, where an average of 30% of the sensor nodes are GPS equipped and can determine their position independently of other sensors. From these 30% sensor nodes, a 3% lie. The remainder sensors, which are unaware of their position, independently execute on each experiment the set of algorithms defined in Section 3 to derive their positions. For each generated WSN, location-unaware nodes request the locations of their neighbors and apply, depending on each specific simulation, Algorithms 1 to 6. For each simulation, if an unaware nodes fails at deriving its location, it holds its execution, and repeats the same algorithm later, expecting that the number of neighbors aware of their location increases. This process runs for 100 times for each network size. Figures 9(a)—(d) picture the average results and the 95% confidence intervals of executing Algorithms 1— 6 in this first round of experiments. Each Algorithm is identified in the figures by their corresponding boundaries for handling the different adversary models. Table 6 recalls the upper bounds of each algorithm to handle the set of adversary models. The variable n is the number of distance one hop neighbors, and ℓ the number of liars, where $\ell > 2$.

Algorithm	Adv. Model	Upper Bound
1	1–4	$n^3 - 3(2\ell+1)n^2 + 2(3\ell^2 + 6\ell + 1)n - (2\ell^3 + 6\ell^2 + 4\ell) > 0$
2	1–4	$n > (4\ell + (8\ell^2 + 17)^{1/2} + 1)/2$
3 & 6	1	$n > 2\ell + 2$
	2	$n > 2\ell + 1$
	4	$n > \ell + 2$
3	3	$n > 2\ell$
4	1–4	$n > (4\ell + (8\ell^2 + 1)^{1/2} + 3)/2$
5	1–4	$n > 2\ell + 3$
6	3	$n > \ell + 2$

Table 6. Summary of boundaries for each algorithm vs. adversary models.

The results plotted in Figures 9(a)—(d) are presented by ordering the curves in decreasing order of sensors aware of their position after running the algorithms. Notice that the execution of all six algorithms significantly increases the number of sensors aware of their position in this first round of simulations. The execution of the most frequent rule in Algorithms 3 and 6 presents the most relevant results: approximately a 75% of location aware nodes in the 100-sensor networks; more than 80% in the 150 to 200-sensor networks; and almost 90% in the 250-sensor networks. The differences between these results and those obtained by executing the majority rules of Algorithms 1, 2, 4, and 5 are, however, quite low. The execution of the majority rule in all four algorithms results in, approximately, a 70% of location aware nodes in the 100-sensor networks; about 75% in the 150 to 200-sensor networks; and almost 80% in the 250-sensor networks. This low improvement, of about 5%, when executing the majority or the most frequent rule is due to the

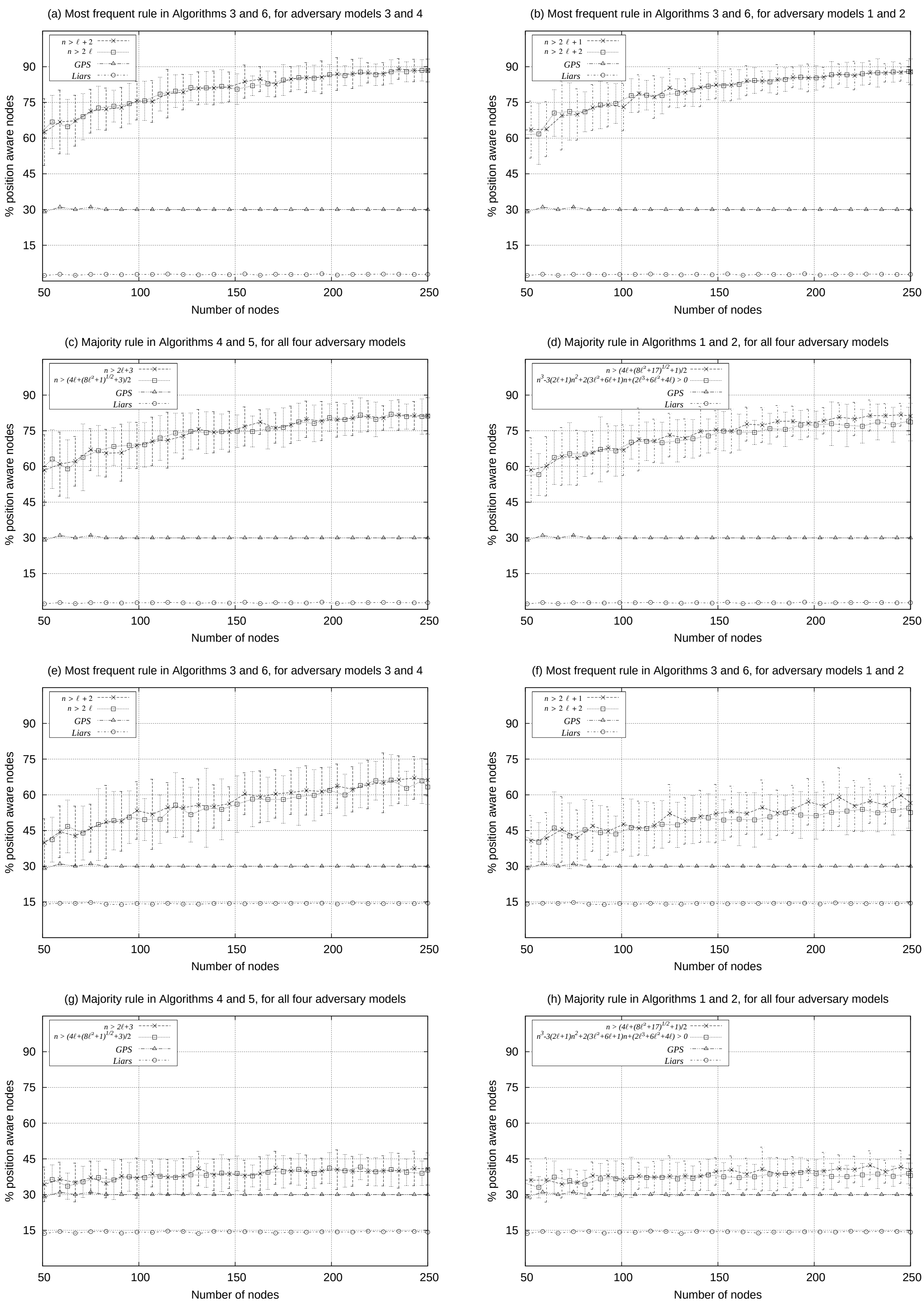


Figure 9. Evaluation of the upper bounds.

low percentage of liars in the neighborhood. The low ratio of liars explains, moreover, the low benefits of using trusted nodes in the neighborhood while comparing the results of Algorithms 1, 2 with those of Algorithms 4, and 5.

In the second set of simulations, the same layout of GPS equipped nodes (i.e., approximately a 30% for each network) applies. The number of liars increases to a 15%. Figures 9(e)—9(h) pictures the average results and the 95% confidence intervals. The resulst are presented by ordering the curves in decreasing order of sensors aware of their position after running the algorithms. Notice that the differences between the application of the majority rule in Algorithms 1, 2, 4, and 5, compared with the application of the most frequent rule in Algorithms 3, and 6, are quite important. While the use of the most frequent argument results in more than 45% of location aware nodes in the 100-sensor networks, and between 50% to 60% in the remainder networks; the use of the majority argument almost remains stable between 35% to 40% for the same setups. And the use of one trusted node in the neighborhood does not seem to provide a very representative increment. By looking at the boundaries shown in Table 6 for Algorithms 3 and 6 we can observe, moreover, that the use of one trusted node in the neighborhood does never have a significant improvement in the use of the most frequent argument. We, therefore, conclude that the use of frequencies of occurrence by Algorithm 3 will always provide the best possible results.

5 Related Works

Research in the field of the security of WSNs 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 position proceeds by correlating the messages received from 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.

6 Conclusions

We presented six algorithms that handle the positioning process of location-unaware 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 three first algorithms allow the localization process without the necessity of a trusted model between sensors. The three last algorithms relax the initial hypothesis, requesting location-unaware sensors to trust one of the nodes in their one hop neighborhood. Just the boundaries of the algorithms based on the majority rule slightly improve the results by assuming the presence of the trustee node. The boundaries of the algorithms based on the most frequent rule remain stable and provide, moreover, the best results.

Acknowledgments — Research supported by the following organizations: Natural Sciences and Engineering Research Council of Canada (NSERC), Mathematics of Information Technology and Complex Systems (MITACS), Institut TELECOM (LOCHNESS grant), and Spanish Ministry of Science (grants TSI2007-65406-C03-03 E-AEGIS and CONSOLIDER-INGENIO 2010 CSD2007-00004 ARES). We would also like to thank Prof. Jean-Marc Robert for fruitful discussions on the topic of this paper.

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.