Front. Comput. Sci. China
DOI 10.1007/s11704-008-0018-7

RESEARCH ARTICLE

A robust localization algorithm in wireless sensor networks

Xin LI[1], Bei HUA (✉)[1], Yi SHANG[2], Yan XIONG[1]

1 Department of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China
2 Department of Computer Science, University of Missouri-Columbia, Columbia, MO 65211, USA

Most of the state-of-the-art localization algorithms in wireless sensor networks (WSNs) are vulnerable to various kinds of location attacks, whereas secure localization schemes proposed so far are too complex to apply to power constrained WSNs. This paper provides a distributed robust localization algorithm called Bilateration that employs a unified way to deal with all kinds of location attacks as well as other kinds of information distortion caused by node malfunction or abnormal environmental noise. Bilateration directly calculates two candidate positions for every two heard anchors, and then uses the average of a maximum set of close-by candidate positions as the location estimation. The basic idea behind Bilateration is that candidate positions calculated from reasonable (i.e., error bounded) anchor positions and distance measurements tend to be close to each other, whereas candidate positions calculated from false anchor positions or distance measurements are highly unlikely to be close to each other if false information are not collaborated. By using ilateration instead of classical multilateration to compute location estimation, Bilateration requires much lower computational complexity, yet still retains the same localization accuracy. This paper also evaluates and compares Bilateration with three multilateration-based localization algorithms, and the simulation results show that Bilateration achieves the best comprehensive performance and is more suitable to real wireless sensor networks.

Keywords localization, Multilateration, Bilateration

1 Introduction

Wireless sensor networks (WSNs) are essentially intended to observe spatio-temporal characteristics of the physical world. Locations of sensor nodes are fundamental to providing location stamps, locating and tracking objects, forming clusters, and facilitating routing, etc. However, a priori knowledge of locations is unavailable in large-scale and ad-hoc deployments, and a pure-GPS (Global Positioning System) [1] solution is viable only with costly GPS receivers and good satellite coverage. In a general scenario, only a few nodes (called anchors) are aware of their positions either through manual configuration or equipped with GPS receivers, and the others (called unknown nodes) have to estimate their positions by making use of the positions of anchors.

Localization algorithms in WSNs are broadly divided into range-free approaches and range-based approaches. Range-free approaches normally rely on proximity, near-far information or less accurate distance estimation to infer the locations of unknown nodes [2–6], and range-based approaches require accurate distance or angle measurements to locate the unknown nodes [7–9]. Both approaches must rely on the positions of anchor nodes and some measured/estimated parameters, and the localization accuracy depends on the accuracy of reference positions and relative parameters.

Wireless sensor networks usually run in open environments where attackers may easily intrude. Attackers may disseminate false reference positions in the network, or mislead unknown nodes to get false distance/angle measurements by tricks like modifying distance, jamming communication and creating wormholes [11,12]. In addition to that, a wireless sensor network may be deployed in a hostile environment without attendance, where some of the nodes may fail to function properly due to components or program malfunction and report false information. Environmental noise may also contribute to exceptional measurements in some nodes. Since most of the state-of-the-art localization algorithms just accept the received or measured information as is, they are vulnerable to various location attacks, node faults and exceptional measurements.

In order to defend against location attacks, some secure localization schemes have been proposed recently, among which are location verification [13], distance verification [14,15], distance-bounding [16], received signal strength measurements [18] and ‘‘packet leashes’’ [12], to name a

E-mail: xinxinol@mail.ustc.edu.cn

few. However, most of these methods require powerful computation, precise synchronization, fast transmission, or some training, etc, which are not suitable for tiny, power-constrained sensor nodes. Moreover, these methods seem to lay strong emphasis on the countermeasures against specific attacks rather than the problem of localization itself. Multilateration is a commonly used method for solving the location of an unknown node when given a set of reference positions and corresponding distance measurements to these positions, wherein the Least Squares (LS) is usually used to minimize the estimation error. Due to lack of false information filtering ability, this scheme will cause large location error in hostile environments. In order to get rid of outlier samples to improve the estimation accuracy, Least Median Squares (LMS) is introduced in Ref. [17] to minimize the median of error squares rather than the sum of error squares in LS. LMS achieves higher localization accuracy, however, it requires intensive computation and thus is unsuitable to WSN. Ref. [17] proposes Linear LMS (LLMS) to reduce the computational complexity of LMS by formulating a linearization of the LS estimator; however, this scheme sacrifices the localization accuracy of LMS.

We observe that the goal of all of the location attacks is to cheat the unknown nodes to get false information, which in most of the cases are anchor positions and distance measurements. From the localization point of view, there is no difference among location attacks, node malfunction and exceptional measurements caused by abnormal environmental noise in the sense that they all make false information. Therefore, the goal of a robust localization algorithm is to locate the unknown nodes with acceptable accuracy even in the presence of some false information.

In this paper, we propose a distributed robust localization algorithm called Bilateration, which deals with location attacks, node malfunction and exceptional measurements in a unified way by considering the set of samples consisting of reasonable samples and unreasonable samples and trying to use reasonable samples to locate unknown nodes. By dealing with various cases in a unified way, there is no need to identify what causes the false information, especially what kind of location attack the unknown node is confronted; and there is no need to identify whether the reference positions are from their neighbors or other nodes through a DV-based way. Unlike other distance-based localization algorithms that use trilateration or multilateration, we use Bilateration to greatly reduce the computational complexity of location estimation. Simulation results show that Bilateration achieves the best tradeoff between localization accuracy and computational complexity in hostile environments. Bilateration may further balance the localization accuracy and communication complexity by choosing to use an optimal set of reasonable samples or suboptimal set of reasonable samples to estimate the location of unknown nodes according to the specific environment the WSN is deployed.

To sum up, the main contributions of this paper are as follows. First, we propose a robust localization algorithm Bilateration that can solve the location of unknown nodes in the presence of location attacks, node malfunction or environmental noise. Second, we compare Bilateration with three multilateration-based localization algorithms, i.e., multilateration with LS, LMS and LLMS, in terms of localization accuracy, false position filtering ability and computational complexity. Third, we discuss the tradeoff between localization accuracy and communication complexity when Bilateration runs in an attack free environment, where a suboptimal set of reasonable samples is used to locate the unknown nodes.

The remainder of this paper is organized as follows. Section 2 summarizes related work on secure localization algorithms; section 3 formulates the problem we consider in this paper; section 4 reviews the basic idea of LS, LMS and LLMS; section 5 describes the Bilateration algorithm; section 6 compares the performance of the above four algorithms; section 7 concludes the paper.

2 Related work

Much work has been done on localization algorithms in WSNs; however, most of them are vulnerable to location attacks, node malfunction and excessive environment noise since they do not examine the rationality of the information they get before using them in the location estimation.

Some location related attacks and their countermeasures are described in Ref. [17]. Technically, most of the attacks try to interfere with the measuring of key parameters. For example, to make time-of-flight based localization scheme fail, attackers may remove the direct path between a pair of nodes, delay a response message, or change the difference of propagation speeds by a different medium. To make signal-strength-based localization fail, attackers may bring a different signal propagation model, change the transmission power level, or locally employ ambient channel noise. To make angle-of-arrival based localization fail, attackers typically change the signal arrival angles by using reflective objects, or alter the orientation of the receivers. To attack the geometry constrains schemes, attackers may create worm-holes to enlarge the neighborhood, manipulate the per-hop-distance measurements, or alter the neighborhood by jamming the communication along certain directions. To attack a hop count based scheme, attackers may create wormholes (or jamming) to shorten (or prolong) the route between two nodes, or alter the hop count by manipulating the radio range. To attack neighbor information based schemes, attackers may change radio range by jamming or transmitting at higher power level or creating wormhole, replay/modify message, or change the receiving pattern of the antenna to change the neighbor relationship. In real networks, attackers usually combine several methods to enhance the effect of attack.

Recently, much attention has been paid to secure localization algorithms. Distance-bounding technique was first introduced in Ref. [13], which enables a node to determine an upper bound of the Euclidean distance to another node. Two similar schemes for secure distance verification are proposed in Refs. [14] and [15], in which they make use of the ultrasound-based distance bounding technique to determine whether a node is present within a monitored area. Another distance bounding protocol called MAD (Mutually Authenticated Distance-bounding) is introduced in Ref. [16], in which each side of a pair of nodes acts as a claimant and a verifier and mutually authenticates the distance bounding based on RF propagation. Received signal strength measurements [18] is used to detect malicious alteration of signal power level. In the television industry, in order to prevent cloning of set-top boxes, people make use of existing telecommunications infrastructure, such as satellites, paging and cellular networks [22]. Packet leashes [12] are used to prevent wormholes by making use of geographic positions of nodes (called geographic leashes) or packet transmission time between nodes (called temporal leashes). SecRLoc [23] employs a sectored antenna, an encryption mechanism and a transmission protocol to make sure that two sensor nodes that can hear from each other must be within the distance of $2R$, where R is fixed to defend against attacks. Ref. [26] proposes an asymmetric security mechanism for navigation signals. Based on consistency of received beacons, Ref. [27] provides mechanisms for the detection of malicious attacks against beacon-based location discovery in sensor networks. In the presence of range measurement noise, Ref. [28] introduces the probabilistic notion of robust quadrilaterals to avoid flip ambiguities that otherwise corrupt localization computations. Most of the above mechanisms need additional complex and expensive hardware that is not suitable to tiny, cheap and power constrained sensor nodes. Moreover, they are designed specifically for one or two kinds of attacks, and are not sufficient in the real world.

Since the ultimate goal of all location attacks is to provide unknown nodes with incorrect information, and most of the localization algorithms rely on multilateration and Least Squares to achieve global optimization on all samples, Ref. [17] takes a Median based approach to improve the robustness of localization. Median based approaches for data aggregation in sensor networks have already been proposed in Refs. [24] and [25], and use the median as a resilient estimate of the average of aggregated data. Ref. [17] considers the incorrect samples as outliers, and uses Least Median Squares to filter out the outliers first, then employs LS on other samples to get the final location estimation. LMS achieves much higher location accuracy than LS in the presence of attacks. However, it requires intensive computation, and therefore Linear LMS is proposed in Ref. [17] to trade the localization accuracy for lower computational complexity. LMS and LLMS will be reviewed in section 4.

This paper proposes Bilateration, a distributed robust localization algorithm whose localization accuracy is as high as that of LMS yet whose computational complexity is as low as that of LLMS.

3 Problem formulation

We consider a homogeneous wireless sensor network that consists of a set of nodes including anchor nodes and unknown nodes. Each node is equipped with a radio transceiver, and can communicate with another node if the Euclidean distance between them is smaller than a specific radio range. Each node can measure the distance to other nodes via some ranging technique like TDOA, RSSI or DV-HOP. The measured distance is expressed as formula (1), where $N(0,\ V\ D)$ is a white Gaussian noise. Among all the nodes, a few anchor nodes may report false positions, and some unknown nodes may get erroneous distance measurements.

$$d_{measured}=d_{real}+noise, noise\sim N(0,V\ D) \tag{1}$$

Suppose an unknown node located at $(x_0,\ y_0)$ has collected a set of N samples $\{(x_1,y_1,d_1),\ldots,(x_N,y_N,d_N)\}$. In a threat- and noise-free environment, these samples will satisfy the following N equations:

$$\begin{aligned}(x_1-x_0)^2+(y_1-y_0)^2&=d_1^2\\(x_2-x_0)^2+(y_2-y_0)^2&=d_2^2\\&\vdots\\(x_N-x_0)^2+(y_N-y_0)^2&=d_N^2\end{aligned} \tag{2}$$

If $N\geqslant 3$, the coordinates of $(x_0,\ y_0)$ can be determined by solving any three of the equations if the three selected anchors are not in a line. This method is the classical trilateration algorithm, whose solution in a 2D plane is the intersection point of three circles centered at three anchors, respectively (see fig. 1(a)). Actually all N circles intersect at $(x_0,\ y_0)$ (fig. 1(b)). However, in a real environment with position/distance error, these N circles may not intersect at one point, and an objective function described in formula (3) is usually used to minimize the difference between estimated location and real location of unknown node. This method is the classical multilateration algorithm with Least Squares.

Generally speaking, in a noisy environment without threat, multilateration with LS is not a bad choice. However, in an environment with malicious location

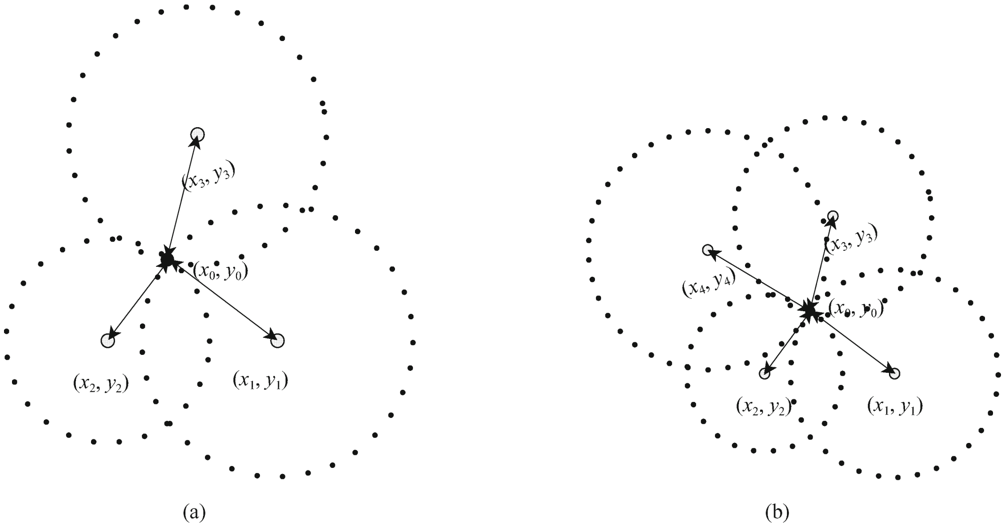


Fig. 1. Trilateration in ideal environments

attacks, the estimated location $(\hat{x}_0,\hat{y}_0)$ may be "removed" far away from the optimal position by some exceptional samples.

4 LS, LMS and LLMS

4.1 Least square

$$(\hat{x}_0,\hat{y}_0)=arg\min_{(x_0,y_0)}\sum_{i=1}^{N}\left[\sqrt{(x_i-x_0)^2+(y_i-y_0)^2}-d_i\right]^2 \quad (3)$$

Multilateration with LS is to minimize the difference between estimated position $(\hat{x}_0,\hat{y}_0)$ and real position (x_0, y_0) of a node, see (3). This method usually involves some iterative searching technique such as gradient descent or Newton method. To avoid local minimum LS must run several times with different initial starting points, which is expensive in terms of computing overhead. Moreover, it is vulnerable to location attacks since it tries to achieve a global optimality on all of the samples including those exceptional ones.

4.2 Least median square

To increase the robustness of multilateration with LS, Least Median Squares is proposed in Ref. [17]. Instead of minimizing the sum of the error squares, LMS tries to minimize the median of the error squares:

$$(\hat{x}_0,\hat{y}_0)=arg\min_{(x_0,y_0)} med_i\left[\sqrt{(x_i-x_0)^2+(y_i-y_0)^2}-d_i\right]^2 \quad (4)$$

According to Ref. [17], the procedure for implementing the robust LMS algorithm is summarized as follows:

(1) Set $n=4$ as the appropriate subset size.

(2) Set $M=\begin{cases}20,\ if\ N>6\\ \binom{N}{4},\ otherwise\end{cases}$ as the appropriate total number of subsets.

(3) Randomly draw M subsets of size n from the set of heard anchors $\{(x_1,y_1),\ldots,(x_N,y_N)\}$. For each subset j, estimate $(\hat{x}_0,\hat{y}_0)_j$ using LS and calculate the median of estimation residuals r_{ij}^2. Here $i=1,2,\ldots,N$ is the index for heard anchors, while $j=1,2,\ldots,M$ is the index for the subsets.

(4) Set $m=\arg\min_j med_i\left\{r_{ij}^2\right\}$, then $(\hat{x}_0,\hat{y}_0)_m$ is the location estimation with the least median of errors among all subsets, and $\{r_{im}\}$ is the corresponding residue.

(5) Calculate $s_0=1.4826\left(1+\frac{5}{N-2}\right)\sqrt{med_i r_{im}^2}$.

(6) Assign weight ω_i to each heard positions with equation

$$\omega_i=\begin{cases}1,\ \left|\dfrac{r_i}{s_0}\right|\leqslant\lambda\\ 0,\ otherwise\end{cases},$$

$$r_i=\sqrt{(x_i-\hat{x}_0)^2+(y_i-\hat{y}_0)^2}-d_i.$$

(7) Do LS on all heard positions with weights $\{\omega_i\}$ to get the final estimation $(\hat{x}_0,\hat{y}_0)$.

It can be seen from the above procedure that to make LMS work, there should be enough heard anchors for each unknown node and the percentage of compromised nodes that give exceptional samples should be less than 50%. In the simulation in Ref. [17], the number of anchor nodes heard by each unknown node is 30, which however is not always possible in the real world. Another disadvantage of LMS is high computing overhead, since LMS must run LS M times in step 3 and once in step 7.

4.3 Linear LMS

Linear LMS [17] transforms nonlinear LS into linear LS in location estimation to lower the computational complexity, which is a suboptimal solution but is efficient in computing. The transforming process is as follows:

(1) Average all the left parts and right parts of (2) to get:

$$\frac{1}{N}\sum_{i=1}^{N}\left[(x_i-x_0)^2+(y_i-y_0)^2\right]=\frac{1}{N}\sum_{i=1}^{N}d_i^2 \tag{5}$$

(2) Subtract each side of (5) from (2), and linearizes to get N new equations:

$$\begin{aligned}\left(x_1-\frac{1}{n}\sum_{i=1}^{n}x_i\right)x_0+\left(y_1-\frac{1}{n}\sum_{i=1}^{n}y_i\right)y_0&=\frac{1}{2}\left(x_1^2+y_1^2-d_1^2-\frac{1}{n}\sum_{i=1}^{n}\left(x_i^2+y_i^2-d_i^2\right)\right)\\ &\vdots\\ \left(x_n-\frac{1}{n}\sum_{i=1}^{n}x_i\right)x_0+\left(y_n-\frac{1}{n}\sum_{i=1}^{n}y_i\right)y_0&=\frac{1}{2}\left(x_n^2+y_n^2-d_n^2-\frac{1}{n}\sum_{i=1}^{n}\left(x_i^2+y_i^2-d_i^2\right)\right)\end{aligned} \tag{6}$$

(3) Estimate $(\hat{x}_0,\hat{y}_0)$ using linear LS.

Transforming nonlinear LS into linear LS saves much computation time, since the solution can be calculated directly from (6) without iterative searching and repeating. Furthermore, the solution of linear LS can be used as the starting point of nonlinear LS to prevent nonlinear LS from getting trapped in a local minimum. In section 6, we use this starting point to do nonlinear LS in the simulation.

However, due to the subtraction, the optimal solution of linear equations in (6) is not exactly the same as that of nonlinear LS in (2), which means much accuracy is lost, especially when the number of heard anchors is small, e.g., the number of herd anchors is less than 7.

Experiments in Ref. [17] show that when the number of anchors heard by each unknown node is set to 30 and the percentage of compromised nodes is less than 50%, performance of linear LS is very good. However, 30 heard anchors per unknown node is almost impossible in real wireless sensor networks, and we try to find a solution that requires fewer anchor nodes while keeping the computational complexity as low as possible.

5 Bilateration

We take another strategy to solve the equations in (2). To avoid using an LS estimator, we choose to evaluate two equations at a time. Use the first two equations as an example. We first subtract the second equation from the first equation to get

$$\begin{aligned}x_0&=\frac{-(mn-ny_1-x_1)}{1+n^2}\pm\frac{\sqrt{2(nx_1+m)y_1-y_1^2-n^2x_1^2-2mnx_1-m^2+(1+n^2)d_1^2}}{1+n^2}\\ y_0&=m+nx_0\end{aligned} \tag{7}$$

and set

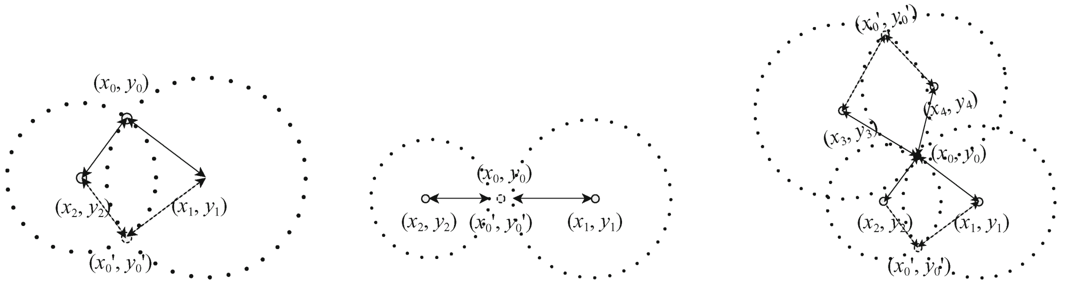
$$\begin{aligned}m&=\frac{1}{2}\frac{\left(x_1^2-x_2^2\right)+\left(y_1^2-y_2^2\right)-\left(d_1^2-d_2^2\right)}{y_1-y_2}\\ n&=-\frac{x_1-x_2}{y_1-y_2}\end{aligned}$$

The real solutions of the two equations are called candidate positions, which in a 2D plane are the points of intersection of two circles (see fig. 2(a)); the complex solutions are not considered in this paper (see fig. 2(b)). Evaluation of (7) is very fast given (x_1,y_1,d_1) and (x_2,y_2,d_2). If another two anchors (at least one of the anchors does not belong to $\{(x_1,y_1),(x_2,y_2)\}$) are selected, another two candidate points can be found for (x_0,y_0). Among the four candidate points, at least two points overlap if no noise exists, and this point is the correct position of (x_0,y_0) (see fig. 2(c)). If more anchors are available, more overlapped points can be found for (x_0,y_0).

In a real noisy environment, there may be no overlapped points due to position/distance error. However, there is reason to believe that reasonable positions should be close to each other if the error is bounded. The basic idea of Bilateration is to find out the reasonable positions and take the average of reasonable positions as the final estimation position.

Since each sample binds a reference position with a distance value, which means a sample is unusable as long as one of them gives false information, for the simplicity of description, we will assume hereafter that some of the anchor nodes are compromised and report false reference positions, whereas all the distance errors are bounded.

We first define two terms as follows:



(a) Two real solutions (b) No real solutions (c) Overlapped solutions

Fig. 2. Solutions of Bilateration
(a) Two real solutions; (b) No real solutions; (c) Overlapped solutions

Reasonable candidate positions: a group of candidate positions among which there is at least one position whose distances to the other members are less than a threshold δ.

Candidate neighbors: two candidate positions between which the distance is within δ.

For an unknown node μ, the procedure for implementing our Bilateration algorithm is summarized as follows:

(1) If $n \leqslant 3$, set μ as un-localized and terminate the algorithm. This situation will not be considered in our performance comparison, because there is no way to distinguish which position is false.

(2) For each pair of anchor nodes, a_i and a_j, and corresponding distance measurements, d_i and d_j, evaluate (7). Suppose M candidate positions $\{c_1,\ldots,c_M\}$ are solved from all the $\binom{n}{2}$ sample combinations.

(3) For each candidate position c_i, calculate $\{D_{i1},\ldots, D_{ii-1},D_{ii+1},\ldots,D_{iM}\}$, where D_{ij} is the distance between c_i and c_j, and $i,j = 1,2,\ldots,M$ is the index to candidate positions.

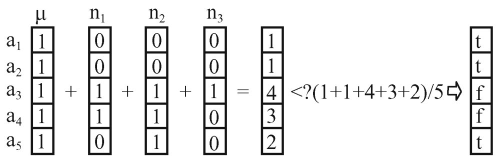
(4) For each c_i, find out all the distances shorter than threshold δ to get $\{D_{ip},\ldots,D_{it}|D_{ip} < \delta \wedge \ldots \wedge D_{it} < \delta, D_{ip},\ldots,D_{it} \in \{D_{i1},\ldots,D_{ii-1},D_{ii+1} ,\ldots,D_{iM} |D_{ip},\ldots,D_{it}|$. ($|\cdot|$denotes the cardinality of a set).

(5) Find out $m = argmax_i\{n_i\}$; suppose $\{D_{mp},\ldots,D_{mt}\}$ are the distances between c_m and its candidate neighbors $\{c_p,\ldots,c_t\}$; find out the corresponding anchors $\{a_l,\ldots,a_q\} \subseteq \{a_1,\ldots,a_n\}$ from which $\{c_m,c_p,\ldots,c_t\}$ are solved; set the weights of $\{a_l,\ldots,a_q\}$ as 1; set the weights of the other heard anchors as -1.

(6) Exchange the weight table with its neighbors.

(7) Collect all the weight tables from its neighbors; pick out the common heard anchors; add their weights together; set the anchors whose weight is less than the average weight as the compromised nodes. (see fig. 3)

(8) Delete the candidate positions caused by compromised nodes from $\{c_1,\ldots,c_M\}$; take the average of all the remaining candidate positions as the final estimated position e_μ.



$$\begin{array}{cccccccccccc} & \mu & & n_1 & & n_2 & & n_3 & & & & \\ a_1 & \boxed{1} & & \boxed{0} & & \boxed{0} & & \boxed{0} & & \boxed{1} & & \boxed{t} \\ a_2 & \boxed{1} & & \boxed{0} & & \boxed{0} & & \boxed{0} & & \boxed{1} & & \boxed{t} \\ a_3 & \boxed{1} & + & \boxed{1} & + & \boxed{1} & + & \boxed{1} & = & \boxed{4} & <?(1+1+4+3+2)/5 \Rightarrow & \boxed{f} \\ a_4 & \boxed{1} & & \boxed{1} & & \boxed{1} & & \boxed{0} & & \boxed{3} & & \boxed{f} \\ a_5 & \boxed{1} & & \boxed{0} & & \boxed{1} & & \boxed{0} & & \boxed{2} & & \boxed{t} \end{array}$$

μ collects three weight tables from its neighbors n_i (i=1, 2, 3), each of the tables records the weights of 5 anchors a_j (j=1, ···, 5) ; t or f represents true or false compromised node.

Fig. 3. Who is the compromised node

If the unknown node hears 4 different positions including 1 false position, LMS and LLMS are unable to deal with this situation, whereas our scheme can find out the reasonable positions if the distance between the correct candidate positions is shorter than δ.

6 Simulation

To evaluate Bilateration, we simulated it and multilateration with LS, LMS, and LLMS on Matlab, and compared them in terms of estimation error, ability of false position filtering, and computational complexity in a simulation environment. Estimation error is the average variance between estimated locations and real locations. Ability of false position filtering is reflected by the average number of false positions (i.e., not filtered out) in the location estimation of each unknown node. Each data point represents the average value of 500 trials with different random seeds. We use ideal LS as a benchmark in the performance comparisons, which can filter out all the compromised anchors before estimation.

In our simulation settings, we have the following definitions and assumptions.

— Anchors and unknown nodes are uniformly distributed in an area of 200 × 200 m^2.

— The coordinates of false positions, x and y, are independently and identically follow normal distribution $N(100, V\ P)$, where VP varies from 20 to 200 m.

— The noise of measured distance obeys normal distribution $N(0, V\ D)$, where VD varies from 0 to 50 m.

— R is the radio range of node, and is fixed to 50 *m* in our experiments.

— NA is the average number of anchors heard by each unknown node.

— NU is the average number of neighboring unknown nodes of each unknown node.

— CP is the percentage of compromised anchors, and varies from 0 to 1.

In the following experiments, if without specification, the default environment settings are: $V\ P = 20\ m$, $V\ D = 5\ m$, $N\ A = 7.5$, $N\ U = 7.5$ and $C\ P = 0.2$.

6.1 Thresholds

In this section, we choose the appropriate thresholds for Bilateration, LMS and LLMS according to the result of performance comparison. We use different *δs, and λs* to do the comparison and find the appropriate thresholds for all schemes.

In Bilateration, δ affects the strictness of the definition of reasonable candidate positions. If δ is too small, it needs more iterations to search candidate positions; on the contrary, it brings more compromised positions into localization. As a matter of fact, without any attacks the closeness of reasonable candidate positions is determined by the variance of measurement noise ($V\ D$). If there is enough experiential data before attacks, we can choose δ as the experiential estimation error; or else, we can estimate the variance of ranging noise, and then choose $2 \times V\ D$ as the default value of δ. For example, if there isn't any measurement noise ($V\ D = 0$) we can choose $\delta = 0$. In the following sections, we set $\delta = 2 \times V\ D$.

In the first experiment, we run Bilateration with different thresholds. According to fig. 4(a), lower estimation error is achieved with smaller δ; however, the difference is not so obvious when VD is over 15. In fig. 4(b), smaller δ improves false position filtering ability, but smaller δ (e. g., $\delta = 5$) requires more iteration to search reasonable candidate positions. As in the following subsections, the default value of $V\ D$ is 5; we choose $\delta = 10$ as the default value to balance the estimation error and computational complexity.

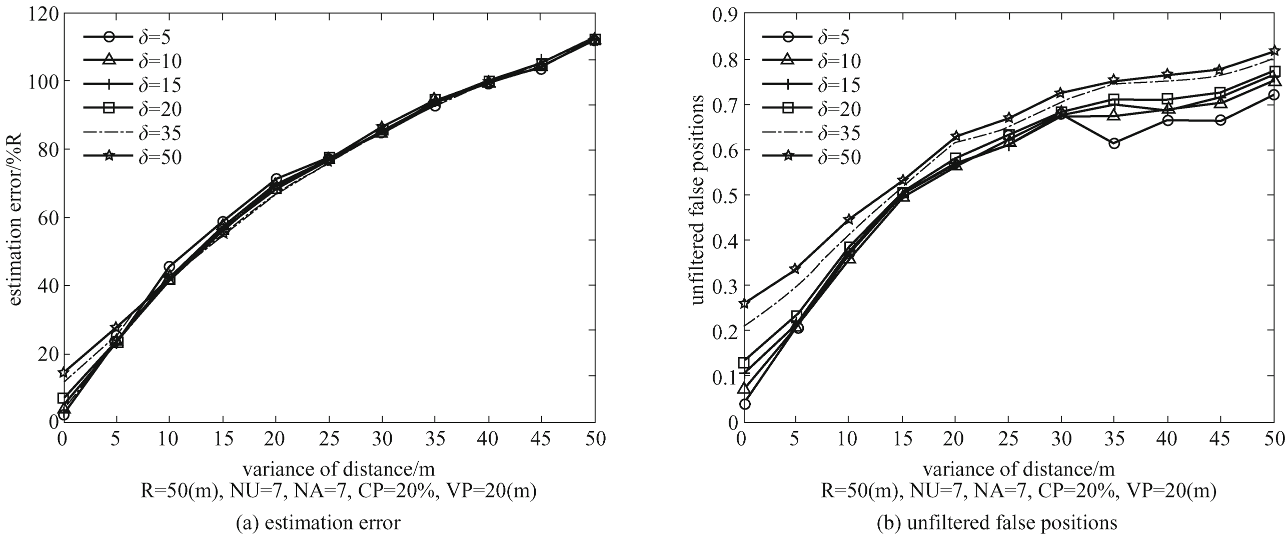
To be fair in performance comparison, the threshold λ of LMS must be chosen properly as well. From fig. 5, we can see that estimation error and the average number of unfiltered false positions do not change much with λ, and bigger λ seems lowering the filtering ability a little. Since in most of the cases λ has little influence on the comprehensive performance of LMS, we choose a moderate value of 1.5 as the default setting, which may save much calculation in step 5 of LMS.

Since in most of the cases the performance of LLMS are hardly affected by λ, we simply set λ to 1.5 to achieve the best performance.

6.2 Influence of average number of anchors

In this experiment, we investigate the influence of average number of heard anchors (NA) on the performance of the four localization algorithms.

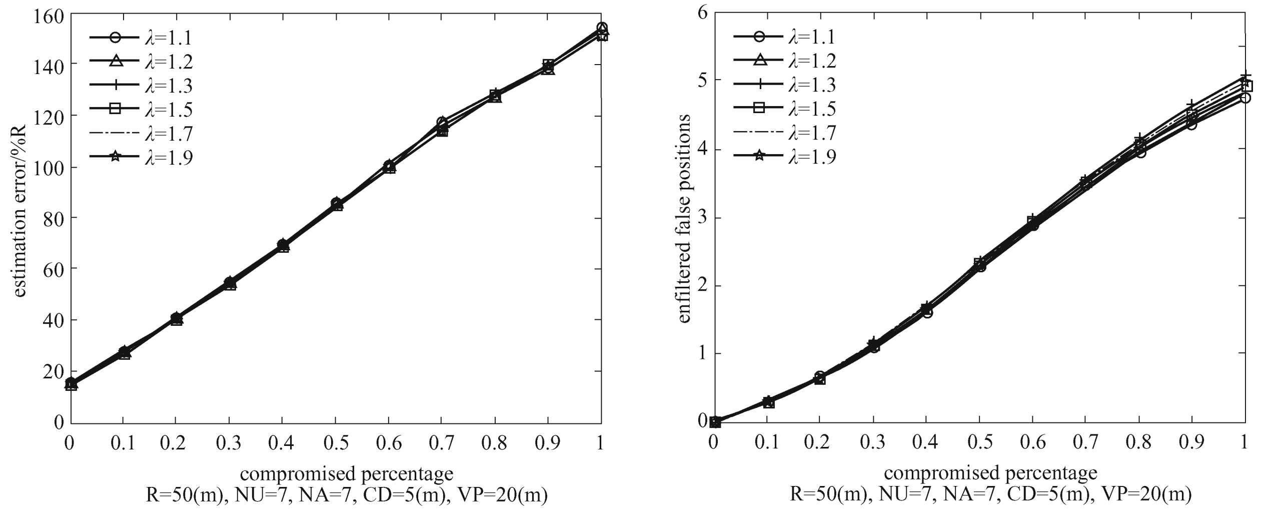
In fig. 6(a), except for LS whose estimation error increases about 5% when NA increases from 5 to 25 due



(a) estimation error

(b) unfiltered false positions

Fig. 4. The influence of δ
(a) Estimation error; (b) Unfiltered false positions



(a) estimation error (b) unfiltered false positions

Fig. 5. The influence of λ
(a) Estimation error; (b) Unfiltered false positions

to its lack of filtering ability, the estimation error of the other four algorithms (including Ideal LS) decreases. Bilateration has lower estimation error than LMS and LLMS, but their gap shrinks when NA increases. Meanwhile, the estimation error of Bilateration is close to that of ideal LS all the time, whereas LMS and LLMS require many more anchors to get the same accuracy.

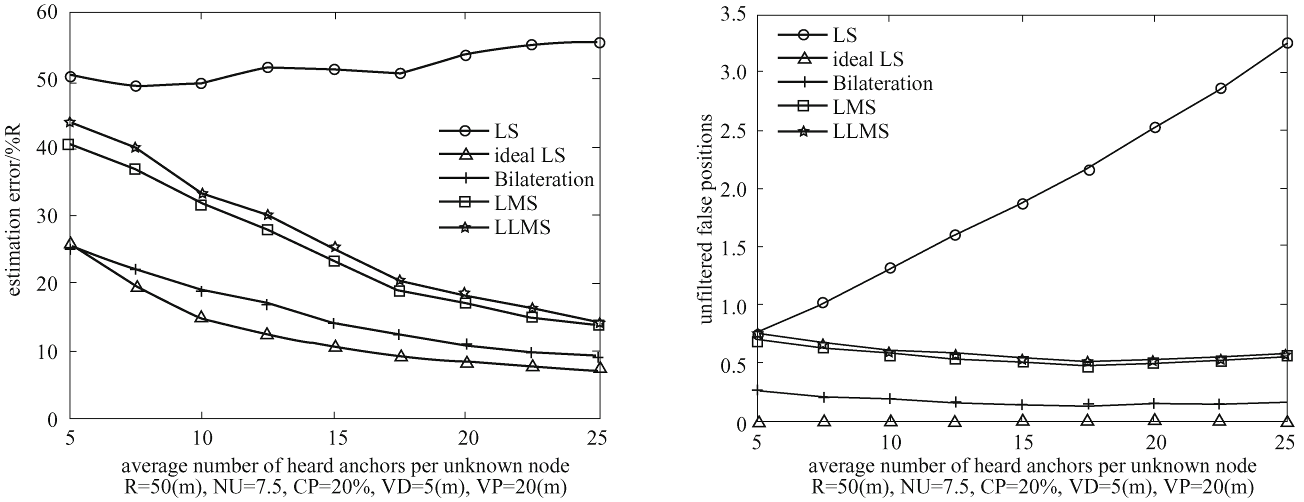
Figure 6(b) compares the filtering ability of all four algorithms. Since LS does not filter out outliers and CP is fixed, the number of false positions used by LS increases with NA. The number of unfiltered false positions used by Bilateration is much smaller than that used by LMS and LLMS; that is to say, Bilateration has stronger filtering ability than LMS and LLMS.

As a localization algorithm without any attacks, Bilateration is also a suboptimal estimation method which is by no means better than LMS or LLMS. However, in the presence of attacks the stronger filtering ability of Bilateration compensates for the suboptimal estimation accuracy. This explains why Bilateration has lower estimation error than LMS in a hostile environment.

Since usually there are only a few anchors in a real wireless sensor network, this result show that Bilateration is more suitable to real settings.

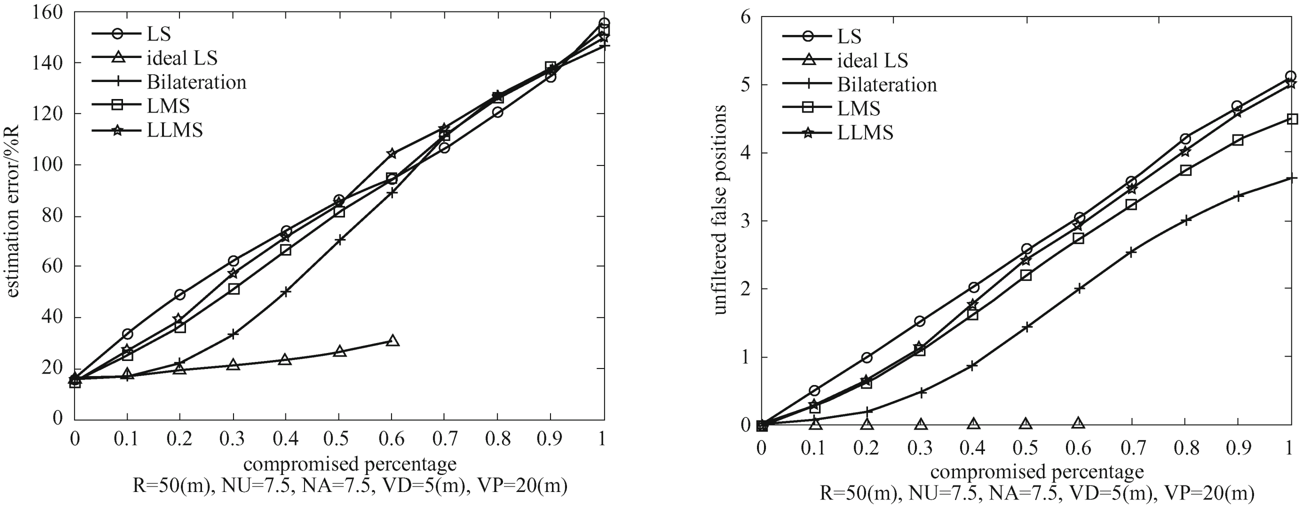
6.3 Influence of percentage of compromised nodes

In this experiment, we investigate the influence of compromised percentage (CP) on the performance of algo-



(a) estimation error (b) unfiltered false positions

Fig. 6. The influence of average number of anchors
(a) Estimation error; (b) Unfiltered false positions



(a) estimation error

(b) unfiltered false positions

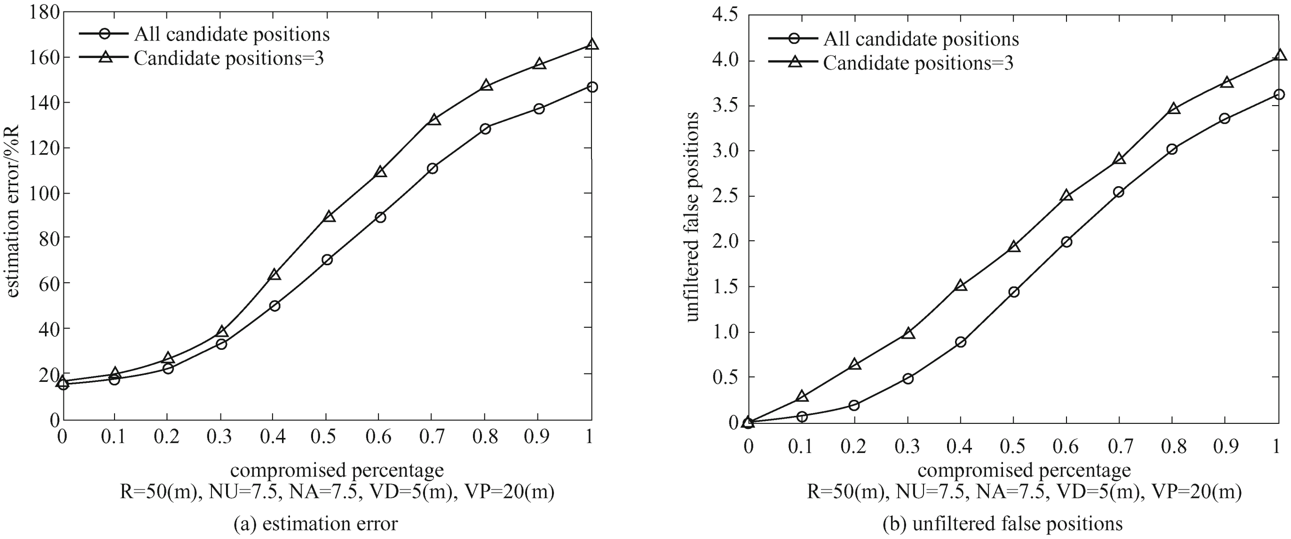
Fig. 7. The influence of percentage of compromised nodes
(a) Estimation error; (b) Unfiltered false positions

rithms. It is interesting to observe that ideal LS terminates when CP reaches 0.6, this is because the number of uncompromised anchors heard by each unknown node is smaller than 3 when $N\ A = 7.5$. Therefore, we will not discuss the performance for CP larger than 0.6.

In fig. 7(a), the estimation error of Bilateration is lower than that of LS, LMS and LLMS all the time when CP is smaller than 0.6, which shows that Bilateration is less affected by CP. However, the four curves tend to approach when CP increases, since there is no difference among them when no right position is available.

In fig. 7(b), the number of unfiltered false positions used by Bilateration is smaller than that used by LS, LMS and LLMS, which shows that Bilateration has the strongest filtering ability.

Figure 8 shows the performance discrepancy of Bilateration when optimal and suboptimal set of reasonable candidate positions are used. If suboptimal set is used, another threshold θ is defined so that when θ reasonable candidate positions are found in step (3), Bilateration stops to find more candidate positions and takes the average of these θ candidate positions as the location estimation. We set $\theta = 3$ in fig. 8. It can be seen that when CP is small (e.g., CP = 10–30%), the performance improvement of using optimal set of reasonable candidate positions is not so obvious. Therefore, in an attack-free environment where CP is usually very small, we can use suboptimal set of reasonable candidate positions to estimate the location of unknown node and omit the step of weight table exchange, which will



(a) estimation error

(b) unfiltered false positions

Fig. 8. Performance improvement of Bilateration using optimal set of reasonable candidate positions
(a) Estimation error; (b) Unfiltered false positions

greatly reduce the communication complexity and computational complexity.

6.4 Influence of distance measurement error

In this experiment, we investigate the influence of distance measurement error on the performance of algorithms.

In fig. 9(a), the estimation error of Bilateration increases rapidly as the variance of distance (VD) increases. The estimation error of Bilateration is lower than that of LS, LMS and LLMS when VD is less than 13, then it exceeds them quickly. We observe that the estimation error of Bilateration does not reach 0 even when VD is 0, since $\delta = 10$ allows some false positions to participate in the location estimation (fig. 9(b)). If δ is set to 0, then Bilateration can filter out all the false positions when VD is 0. LMS and LLMS outperform LS when VD is less than 22 and 15 respectively, and then lost their advantage as well.

In fig. 9(b), the number of unfiltered false positions used by each of the four algorithms increases with VD, since large distance error makes it more difficult to distinguish between correct position and false position. Therefore, if the distance error cannot be well bounded, there is no meaning to discuss the filtering ability of an algorithm.

This experiment shows that Bilateration is more suitable to work in an environment with moderate noise that is less than 24% of radio range.

6.5 Tradeoff between performance and communication complexity

Bilateration is the only algorithm that needs to communicate with neighboring unknown nodes to identify compromised nodes. The performance of Bilateration is not sensitive to the average number of neighboring unknown nodes (NU) when NA is small, while the performance of other three algorithms are not sensitive to NU as well (fig. 10(a)(b)). in this experiment we only evaluate Bilateration with big NA, e.g., $NA = 25$, and different CPs.

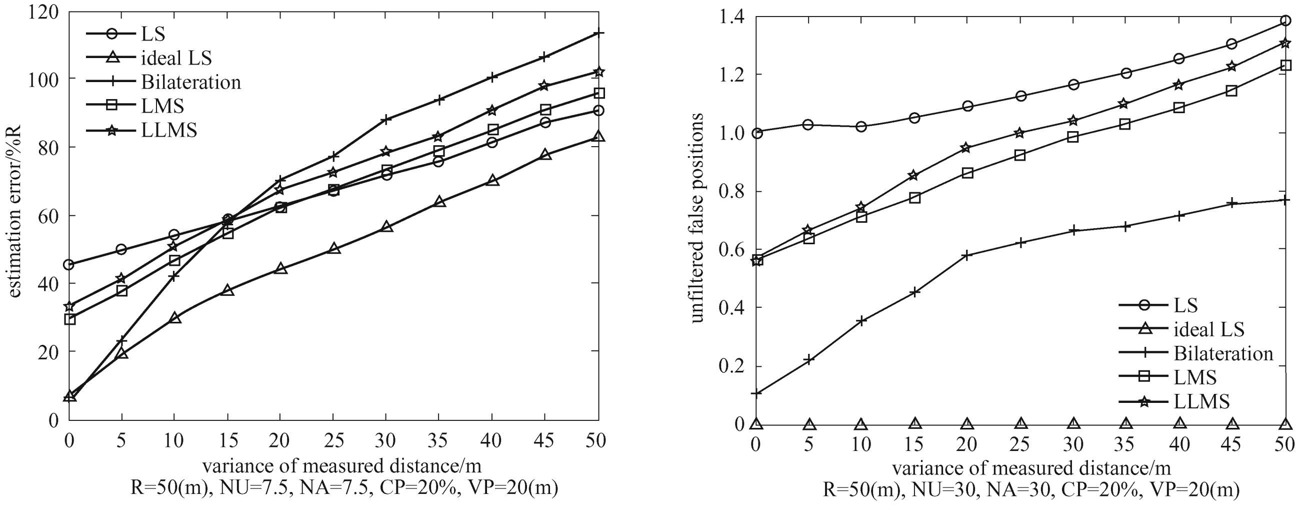
In fig. 10(c) and fig. 10(d), in general for a given CP, both the estimation error and the number of unfiltered false positions decrease when NU increases; this is because more collaboration among unknown nodes enhances the filtering ability of the algorithm. However, the enhancement is obvious only for large CP, i.e., exchanging weight tables in step 6 is effective only for big NA and CP.

Therefore, we can omit step 6 to reduce communication overhead when CP or NA is small, and then the communication overhead of Bilateration is the same as that of LMS and LLMS. Even if sensors have to exchange the weight tables, only one broadcast is enough for each sensor, so step 6 will not introduce much communication cost during localization.

6.6 Computational complexity analysis

Since Bilateration, LMS and LLMS primarily differ in the means of location estimation, we only analyze the amount of computation involved in location estimation in the three algorithms.

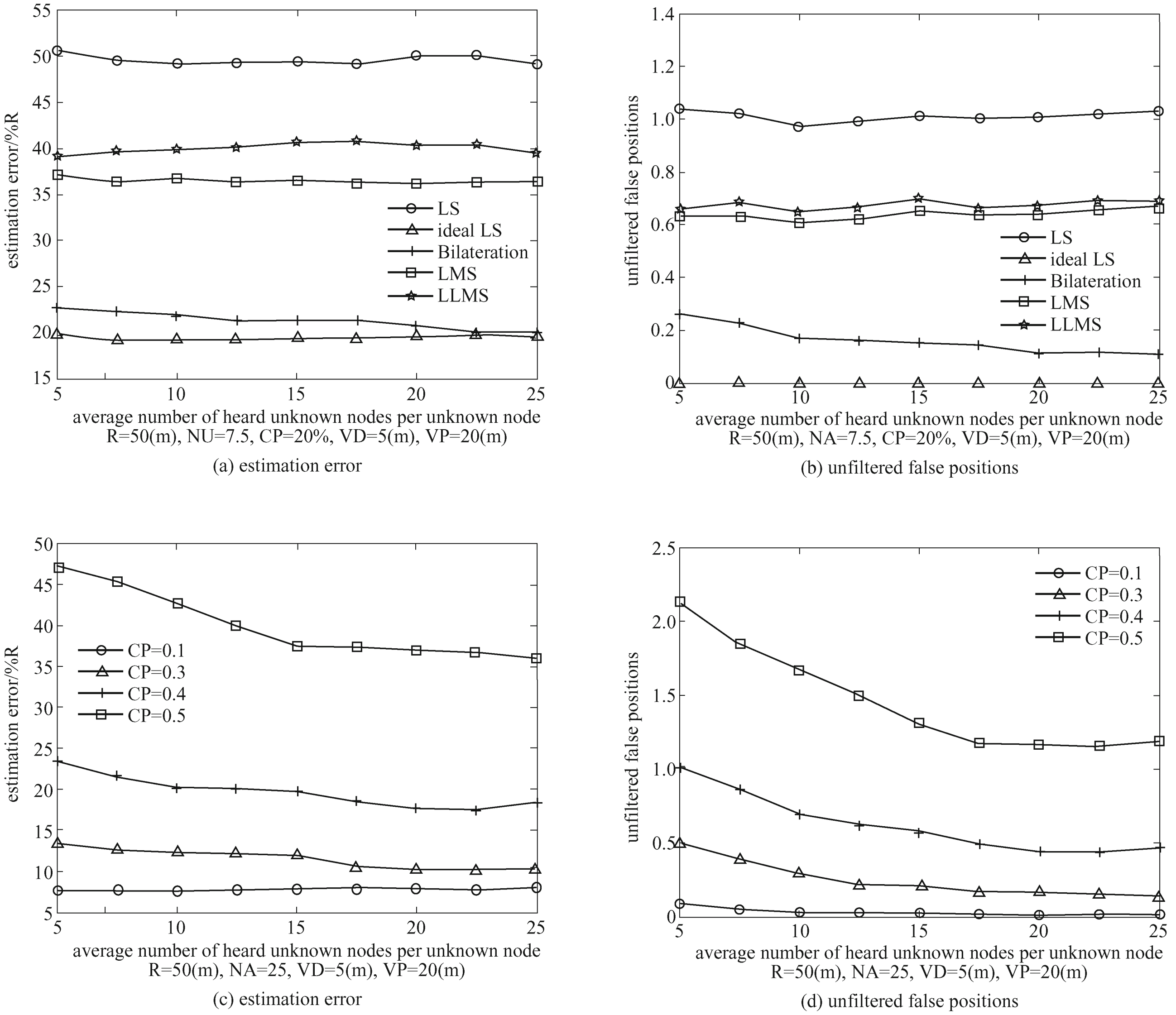
Suppose unknown node μ hears n anchors. In LMS, μ needs to do LS estimation $\binom{n}{4} + 1$ times when $n \leqslant 6$ or 21 times when $n > 6$. In each round of LS estimation except for the last round, four anchor positions are involved in the estimation calculation, and in the last round all the unfiltered positions are involved. In order to avoid local minimum, solution of linear LS is used as a



(a) estimation error

(b) unfiltered false positions

Fig. 9. The influence of distance measurement error
(a) Estimation error; (b) Unfiltered false positions



(a) estimation error

(b) unfiltered false positions

(c) estimation error

(d) unfiltered false positions

Fig. 10. The influence of average number of unknown nodes
(a) Estimation error; (b) Unfiltered false positions; (c) Estimation error; (d) Unfiltered false positions

start point to search the global optimality, which adds another $\binom{n}{4}+1$ or 21 times of linear LS calculation. It is possible to use LS without preliminary linear LS; however, LS may need to search more times for an optimal solution from different start points, and moreover the solution may be trapped into local minimum.

In LLMS, μ needs to do linear LS $\binom{n}{4}+1$ times when $n \leqslant 6$ or 21 times when $n > 6$. The amount of computation involved in linear LS is much less than that involved in LS.

In Bilateration, μ needs to evaluate (7) $\binom{n}{2}$ times to find all the candidate positions, and then perform $4\binom{n}{2}^2$ times of distance calculation between each pair of candidate position to every other candidate position. All the computations only involve simple algebraic calculation, so Bilateration runs much faster than LMS and comparable to LLMS, which has been verified by our experiments.

7. Conclusion

Robust localization is fundamental to WSNs that run in hostile environments. Instead of designing a specific mechanism to defend against a specific type of location attack, we focus our attention on providing a uniform way to deal with all kinds of location attacks, as well as node malfunction and abnormal environmental noise that commonly occur in real networks. We propose a distributed robust localization algorithm called Bilateration, which tries to find a maximum set of close-by positions from all candidate positions and use the average of these close-by positions as the estimated location. Taking close-by positions as reasonable candidate positions is based on the observation that candidate positions calculated from correct reference positions and distance measurements tend to be close to each other, and the use of maximum (optimal) set of close-by positions is to optimize the localization accuracy as well as defeat collaboration location attack launched by compromised nodes. Bilateration is robust in the sense that it can locate the unknown node with acceptable accuracy even in the presence of some false information.

This paper presents the motivation, design and optimization of Bilateration, evaluates and compares the performance of Bilateration with three multilaterationbased algorithms, i.e., multilateration with LS, LMS and LLMS via simulation. Simulation results show that Bilateration achieves the best trade-off between localization accuracy and computational complexity. In fact, Bilateration outperforms multilateration with LMS and LLMS in environments with small number of anchors and moderate environmental noise, which is closer to the real world. In an attack-free environment, Bilateration may reduce its communication complexity by using suboptimal set of reasonable samples to locate unknown nodes, while hardly losing its localization accuracy.

Acknowledgements This work is supported by the National Natural Science Foundation of China under Grant No.60673173, and the Fund for Foreign Scholars in University Research and Teaching Programs.

References

1. Niculescu D, Nath B. Ad hoc positioning system (APS). In: Proceedings of the IEEE Global Communications Conference of GLOBECOM. San Antonio: IEEE Computer Society, 2001, 2926–2931.
2. Bulusu N, Heidemann J, Estrin D. GPS-less low cost outdoor localization for very small devices. IEEE Personal Communications Magazine, 2000, 7(5): 28–34.
3. He T, Huang C. Range-free localization schemes for large scale sensor networks. In: Proceedings of the 9th annual international conference on Mobile computing and networking. San Diego: ACM, 2003, 81–95.
4. Niculescu D, Nath B. DV based positioning in ad hoc networks. Journal of Telecommunication Systems, 2003, 22(1–4): 267–280.
5. Nagpal R, et al: Organizing a global coordinate system from local information on an ad hoc sensor network. In: Proceedings of 2nd International Workshop on IPSN. Berlin: Springer, 2003, 333–348.
6. Doherty L, Pister K. Convex position estimation in wireless sensor networks. In: Proceedings of 20th Annual Joint Conference of the IEEE Computer and Communications Societies. Anchorage, AK, USA: IEEE, 2001, 1655–1663.
7. Bahl P, Padmanabhan V N. RADAR: an in-building RF-based user location and tracking system. In: Proceedings of 19th Annual Joint Conference of the IEEE Computer and Communications Societies. Tel-Aviv, Israel: IEEE Computer Society, 2000, 775–784.
8. Niculescu D, NatB h. Ad hoc positioning system (APS) using AOA. In: Proceedings of 22 Annual Joint Conference of the IEEE Computer and Communications Societies. San Francisco, CA: IEEE Press, 2003, 1734–1743.
9. Lymberopoulos D, Lindsey Q. An Empirical Analysis of Radio Signal Strength Variability in IEEE 802.15.4 Networks using Monopole Antennas. ENALAB Technical Report 050501. Yale University, 2006.
10. Capkun S, Hubaux J P. Secure positioning in sensor networks. TechnicalReport EPFL/IC/200444. May 2004.
11. Xu W, Wood T, Trappe W, et al. Channel surfing and spatial retreats: defenses against wireless denial of service. In: Proceedings of the ACM workshop on Wireless security. Philadelphia, PA, USA: ACM, 2004, 80–89.
12. Hu Y C, Perrig A, Johnson D. Packet leashes: a defense against wormhole attacks in wireless networks. In: 22 Annual Joint Conference of the IEEE Computer and Communications Societies. San Francisco, CA: IEEE, 2003, 1976–1986.
13. Brands S, Chaum D. Distance-bounding protocols. In: Workshop on the theory and application of cryptographic techniques on Advances in cryptology. New York: Springer-Verlag, 1993, 344–359.
14. Sastry N, Shankar U, Wagner D. Secure verification of location claims. In: Proceedings of the 2nd ACM workshop on Wireless security. San Diego, CA, USA: ACM, 2003, 1–10.
15. Waters B, Felten E. Proving the Location of Tamper Resistant Devices. Technical Report. Princeton University, 2003.
16. Čapkun S, Buttyăn L, Hubaux J P. SECTOR: Secure tracking of node encounters in multi-hop wireless networks. In: Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks. Fairfax, Virginia: ACM, 2003, 21–32.
17. Li Z, Trappe W, Zhang Y Y, et al. Robust statistical methods for securing wireless localization in sensor networks. In: Proceedings of Information Processing in Sensor Networks. Piscataway, NJ, USA: IEEE, 2005, 91–98.
18. Tao P, Rudys A, Ladd A M, et al. Wireless lan location-sensing for security applications. In: Proceedings of the ACM Workshop on Wireless Security. New York, NY, USA: ACM, 2003, 11–20.
19. Kusy B, et al. Node-density independent localization. In: Proceedings of 15th international conference on Information processing in sensor networks. New York, NY, USA: ACM, 2006, 441–448.
20. Zhou G, et al. Impact of radio irregularity on wireless sensor networks. In: Proceedings of 2nd International Conference on Mobile Systems, Applications, and Services. New York, NY, USA: ACM, 2004, 125–138.
21. Rappaport T, et al. Wireless Communications: Principles and Practice. New Jersey: Prentice Hall PRT, 2002.
22. Gabber E, Wool A. How to prove where you are: tracking the location of customer equipment. In: Proceedings of Conference on Computer and Communications Security. New York, NY, USA: ACM, 1998, 142–149.
23. Lazos L, Poovendran R. SeRLoc: Secure range-independent localization for wireless sensor networks. In: Proceedings of the 2004 ACM Workshop on Wireless Security. New York, NY, USA: ACM, 2004, 21–30.
24. Przydatek B, Song D, Perrig A. SIA: Secure information aggregation in sensor networks. In: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems. New York, NY, USA: ACM, 2003, 255–265.
25. Wagner D. Resilient aggregation in sensor networks. In: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks. New York, NY, USA: ACM, 2004, 78–87.
26. Kuhn M G. An asymmetric security mechanism for navigation signals. In: Proceedings of the Information Hiding Workshop. Berlin: Springer, 2004, 239–252.

27. Liu D, Ning P, Du W. Attack-resistant location estimation in sensor networks. In: Proceedings of Information Processing in Sensor Networks. UCLALos Angeles, California, USA: IEEE, 2005, 99–106.

28. Moore D, Leonard J, Rus D, et al. Robust distributed network localization with noisy range measurements. In: Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems. New York, NY, USA: ACM, 2004, 50–61.