LOCOMO: Distance-based Localization against Malicious Beacon Nodes

Sheng Zhong Murtuza Jadliwala Shambhu Upadhyaya Chunming Qiao
Computer Science and Engineering Department, State University of New York at Buffalo,
Amherst, NY 14260, U. S. A.

***Abstract*—Last few years have seen extensive research being done in the area of localization algorithms for wireless, ad hoc computer and sensor networks. Despite the progress in the area of efficient localization algorithms, the problem of malicious beacon nodes has not received sufficient attention. In this paper, we study the robust localization problem in the presence of such nodes. In particular, we establish necessary and sufficient conditions for distributed distance-based localization in the presence of a given number of malicious nodes. To prove the sufficient condition, we propose LOCOMO, a novel and efficient distance-based localization framework that can provide a guaranteed degree of localization accuracy. This framework can be used with either a polynomial-time algorithm (which has a rigorous analysis of complexity) or a heuristic algorithm (which has practical efficiency). All the above results are extended to the 3-dimensional case. Extensive network simulation experiments demonstrate that our solution has very good localization accuracy and computational efficiency.**

I. Introduction

Recent advances in wireless and computing technology have resulted in the widespread use of highly distributed systems like wireless computer networks, mesh networks, sensor networks etc. for a variety of commercial and military applications. In applications such as health-care, fire fighting, military and other emergency response applications [20], [13], [32], [29], accurate knowledge of self location (and location of other nodes in the network) may be very important. In particular, the information collected by the nodes may be of little or no use without the location of occurrence, whereas, information associated with inaccurate locations can lead to undesirable consequences. Moreover, location information can also be used for efficiently implementing other important services. For example, Ko et al. [16] and Karp et al. [15] have proposed efficient routing protocols for wireless networks that use location information for making routing decisions.

In wireless computer networks, node positions can change continuously because of user movements, while in sensor networks, exact positions of the nodes may be unknown at initial deployment. *Localization* or *location discovery* in such systems refers to the problem where every node, without using a GPS of its own, needs to efficiently and accurately determine its location with respect to some local or global coordinate system. Localization is an important post-deployment service and needs to be carried out periodically and individually by each node in the network. Last few years have seen extensive research being done in the area of localization algorithms for wireless, ad hoc and sensor networks. Majority of the localization algorithms can be classified into two broad categories namely, *beacon-based* and *signature-based* techniques. Beacon-based schemes [11], [30], [1], [23], [3], [22], [9], [28] require the existence of special nodes that know their own locations, called beacon nodes (or anchor nodes), at strategic positions in the network. Remaining nodes in the network estimate their location by computing distance/angle estimates to a fixed set of beacon nodes. On the other hand, signature-based schemes [4], [24], [6], [27], [21], [2], [12] apply optimization techniques like Convex Optimization, Maximum Likelihood Estimation (MLE), Multidimensional Scaling etc. to data such as node distribution information and connectivity (neighborhood) information, so that the location can be estimated. Localization techniques can also be alternatively classified into *range-based (distance-based)* or *range free* depending on whether or not it uses distance estimates between nodes (or nodes and beacons) for location estimation. These schemes have been outlined in brief in Section II.

Despite the progress in the area of efficient localization algorithms, the problem of malicious beacon nodes and localization in the presence of such nodes has not received sufficient attention. Malicious beacon nodes can cheat by broadcasting incorrect self location references or transmitting at a lower power level thereby affecting the resulting distance computations and eventually the localization done based on it. With the increasing usage of wireless and sensor systems in military and emergency monitoring scenarios, the problem of malicious nodes can no longer be overlooked and its effect on localization algorithms need to be studied in greater detail. The problem of network localization in the presence of malicious nodes is not trivial: Eren et al. showed that a subset of the above problem, namely the problem of distance-based network localization under the assumption that all nodes are honest, is itself hard [5]. They prove this by modelling the network as a weighted, undirected graph and reducing the graph rigidity problem (known to be NP-hard) to the localization problem. Other efforts in this direction have also confirmed this result [7], [2],

[21]. Clearly, localization in the presence of malicious nodes is even harder than the localization problem with all honest nodes. Research efforts to overcome the problem of malicious beacon nodes in localization algorithms have focused on removing the (over)dependence on such specialized beacon nodes by using intelligent statistical tools and coding theory [6], [24], [12], [31], [4]. Such schemes, called signature-based (beaconless) schemes, are too complex and computationally intensive for low-power devices and do not scale very well in case of large networks. Other efforts in this direction have used redundancy, voting schemes, error minimization functions etc. to minimize the effect of malicious nodes on the localization process [18], [19], [14]. Although these schemes provide some improvement, they make simplifying assumptions, like special monitoring nodes, that may not be possible or applicable in all scenarios. In general, the problem of localization in the presence of malicious nodes is far from solved. Specifically, past research has not made any attempt to systematically study the hardness and feasibility of the localization problem in hostile environments. Moreover, none of the past techniques provide any provable guarantee on the localization accuracy, which makes it difficult to compare these schemes against other localization techniques. In this paper, we attempt to solve the robust localization problem by establishing necessary and sufficient conditions for distributed distance-based localization in the presence of a given number of malicious nodes. In particular, we propose *LOCOMO*, a novel and efficient distance-based LOCalizatiOn framework for MObile devices that can provide a guaranteed degree of accuracy on localization.

In this work, we make the following contributions. First, we prove an important necessary condition, called the *Lower Bound Theorem*, for localization againt malicious beacon nodes. This theorem states that, assuming a reasonable network model, if information from only $2k+2$ beacon nodes (or fewer) is available, where k of the nodes are malicious, then no algorithm can provide any degree of localization accuracy. Next, we prove a theorem on sufficient condition. This theorem states that there exist algorithms that provide a guaranteed degree of localization accuracy, if distances from at least $2k+3$ beacon nodes are available (where k is, again, the number of malicious beacon nodes). To prove this result, we propose LOCOMO, a robust distance-based localization framework that determines the location of a node by computing an area in the intersection of at least $k+3$ rings (each ring is centered at each beacon node). The final location of the node is a randomly selected point from this area. There are many ways to determine the area in the intersection of $k+3$ or more rings. We present two different strategies, namely the polynomial-time algorithm and the heuristic algorithm, that compute such an intersection area. The first algorithm is guaranteed to finish computing the location in polynomial time in the worst case, while the second algorithm has much better efficiency in practice. Regardless of which algorithm is used, we have an upper bound on the localization error, which is proportional to the margin of the measurement error. An important point to note here is that, either malicious nodes or distance measurement error, while past efforts have concentrated on only one of the above factors at a time, our localization framework takes both these factors into account before computing location. In addition to the above theorems and algorithms, we extend our work to the 3-dimensional case, where the location of every node is represented by points in the three-dimensional coordinate system. Finally, we verify the localization accuracy and computational efficiency of our LOCOMO framework through extensive network simulation experimentation done using the ns2 [8] network simulator tool.

The rest of the paper is organized as follows. We discuss the background and related work in Section II and present our network model in Section III. In Section IV, we prove the lower bound theorem; in Section V, we design LOCOMO and prove the sufficient condition for robust localization. The algorithms for finding the intersection of rings are given in Section VI, while the extension to 3-dimensional localization is given in Section VII. Experimental evaluations are in Section VIII. We conclude in Section IX.

II. Background and Related Work

Localization in distributed systems like mobile ad-hoc and sensor networks has been a highly researched problem. A taxonomy of the various algorithms for localization can be found in the survey by Hightower et al. [10]. Initial research in this direction was aimed at developing a good theoretical understanding of this problem by using efficient mathematical models. Savvides et al. [26] derived the Cramér-Rao lower bound (CRLB) for network localization and concluded that the error introduced by the algorithm is just as important as the measurement error in assessing end-to-end localization accuracy. Eren et al. [5] provided a theoretical foundation for the problem of distance-based network localization by modelling the problem as a (grounded) graph problem. They studied the computational complexity of the network localization problem and showed that a network has a unique localization if and only if its underlying grounded graph is generically globally rigid. Goldenberg et al. [7] take this a step further by studying partially localizable networks, i.e., networks in which there exist nodes whose positions cannot be uniquely determined and demonstrate the relevance of networks that may not be fully localizable. Bruck et al. [2] showed that it is NP-hard to find a valid embedding in the plane such that neighboring nodes are within distance 1 from each other and non-neighboring nodes are at least distance 1 away. In summary, the above results are very interesting; however, they are based on the assumption that all nodes are honest.

Researchers have followed two approaches towards overcoming the problem of malicious nodes in localization algorithms. The first approach is to detect malicious nodes by observing the inconsistencies in the communication from such nodes and efficiently eliminating them (from consideration) before localization. Liu et al. [18] proposed two methods for robust localization in the presence of malicious beacon nodes. The first method filters out malicious beacon signals on the basis of inconsistency among multiple beacon signals, while the second method tolerates malicious beacon signals by adopting an iteratively refined voting scheme. Sastry et al. [25] proposed a location verification technique to verify the relative distance between a verifying node and a beacon node while Pires et al. [14] gives protocols to detect malicious nodes in range-based localization approaches by detecting malicious message transmissions. Liu et al. [19] also proposed methods to detect malicious beacon nodes in beacon-based localization approaches by deploying special detector nodes that capture malicious message transmissions by the beacons.

Another approach towards robust localization is to efficiently perform localization in the presence of errors in distance measurements. These errors can be a result of external factors like random noise, measurement errors etc. or due to malicious nodes. Moore et al. [21] formulated the localization problem as a two-dimensional graph realization problem and described a beaconless (anchor-free), distributed, linear-time algorithm for localizing nodes in the presence of range measurement noise. Doherty et al. [4] described a robust localization technique using connectivity constraints and convex optimization when some nodes are initialized with known positions. Yi et al. [27] and Ji et al. [12] use efficient data analysis techniques like Multi-Dimensional Scaling (MDS) to perform robust distributed localization using connectivity information and distances to neighboring nodes. Priyantha et al. [23] eliminated the dependence on beacon nodes by using communication hops to estimate the network's global layout and then used force-based relaxation to optimize this layout. Fang et al. [6] model the localization problem as a statistical estimation problem and use maximum likelihood estimation method to estimate the location. Although, the above techniques are efficient, they do not completely address the problem of localization in the presence of malicious nodes. The main aim of the above techniques is to maximize localization accuracy by minimizing the effect of errors. Recently, researchers have also applied ideas from other domains like coding theory to achieve robustness in localization algorithms [24], [31]. The irreducibility property of codes makes these techniques robust against node failures but not against malicious nodes. A very unique idea proposed by Lazos et al. [17] uses sectored antennas for robust localization. The algorithm, called SeRLoc, does not require any communication amongst nodes and is robust against malicious attacks like the wormhole attack, sybil attack and compromised sensor attack.

III. Network Model

In this section, we describe the network model for the problem of distance-based localization (using beacon nodes) of a mobile device M in hostile environments. In other words, M wants to compute its own location using distance estimates to beacon nodes that know their own locations and these beacon nodes may or may not behave maliciously. Suppose that there are n beacons available for localization: $B_1, \ldots, B_n$; k of these n beacons are dishonest, while the remaining $n-k$ are assumed to be honest. The mobile device M is assumed to be honest throughout the localization process. Regardless of being honest or dishonest, each beacon B_i provides M with a measurement $\tilde{d}_i$ of the distance between B_i and M. More specifically, each beacon B_i provides M with some information from which the distance $\tilde{d}_i$ can be computed efficiently by M. The precise distance between B_i and M is the Euclidean distance between the position coordinates of B_i and M and is denoted by $dst(B_i, M)$. Let H be the set containing only the honest beacons amongst a total of n beacon nodes. Then, for each beacon $B_i \in H$, $\tilde{d}_i$ is assumed to follow some probability distribution, denoted as $msr(dst(B_i, M))$, such that

$$E[\tilde{d}_i] = dst(B_i, M),$$

i.e., the expected (mean) value of the estimated distance $\tilde{d}_i$ for each beacon B_i in H, is the precise distance between the beacon B_i and the node M. Also, in the case when B_i is honest, the difference between the estimated and the true distance is assumed to be very small, i.e.,

$$|\tilde{d}_i - dst(B_i, M)| < \epsilon,$$

where ϵ is a small constant. Ideally, this difference should be zero when the beacon is honest, but such discrepancies in distance estimates can occur due to factors like *measurement errors* either at the source or target. For each beacon $B_i \notin H$, $\tilde{d}_i$ is a value selected arbitrarily by the adversary. Note that we implicitly allow colluding attack here: In our model, we consider a single adversary who controls all malicious beacon nodes and decides $\tilde{d}_i$ for all $B_i \notin H$. This is a very strong adversary model that covers all possibility of collusion among malicious beacon nodes.

Since we assume a distance-based localization strategy, the output O of a localization algorithm can be defined by a function F of the measured distances ($\tilde{d}_i$) from the mobile device M to every beacon node in the network as shown below.

$$O = F(\tilde{d}_1, \ldots, \tilde{d}_n).$$

The error e of the localization algorithm is defined as the Euclidean distance between the actual position of the mobile device and the one output by the algorithm.

$$e = E[dst(M, O)].$$

Our next aim is, given the above network model, to derive the necessary and sufficient conditions for

efficient distance-based localization (both in terms of localization accuracy and time of execution) in the presence of malicious beacon nodes. These conditions determine the feasibility of the localization process in situations where some beacon nodes behave maliciously and a maximum number of such malicious nodes is known.

IV. Lower Bound Theorem

In this section, we present an important result, called the *Lower Bound Theoerem*, that gives a lower bound on the number of (honest) beacons required to compute the location of M using distance information. Assuming the network model discussed in Section III, this theorem says that if at most $k+2$ honest beacon nodes are available for localization, where k is the number of malicious nodes, then no distance-based localization algorithm will be able to provide *any* guaranteed degree of localization accuracy. This theorem is a necessary condition for localization in the presence of malicious nodes and can be formally stated as follows:

Theorem 1: Suppose that $n \leq 2k+2$. Then, for any distance-based localization algorithm, for any locations of the beacons, there exists a scenario in which $e \rightarrow +\infty$.

Proof: Without loss of generality, we assume that $n = 2k+2$ (because having more honest beacons clearly won't hurt). We give the proof for the above theorem by a contradiction argument. Suppose that, in all scenarios, the output error $e < a$, where a is a constant. We shall see that this supposition leads to a contradiction.

We consider two scenarios S_1 and S_2, as shown in Figure 1. The locations of all the beacons are same in both the scenarios. Select an arbitrary point P in the line segment B_1B_2 and draw a line L through P such that L is perpendicular to B_1B_2. Choose an arbitrary number $a' > a$. Then there are two points P_1 and P_2 on the line L such that

$$dst(P_1, P) = dst(P_2, P) = \frac{1}{2} dst(P_1, P_2) = a' \geq a.$$

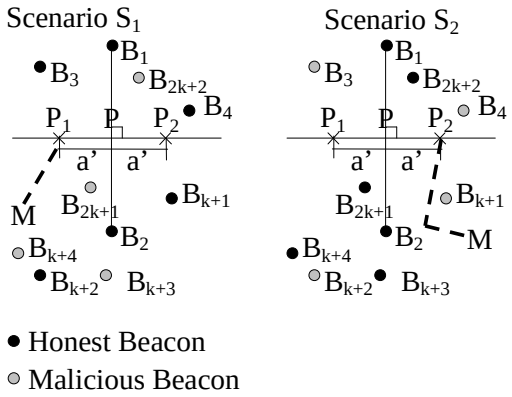

Fig. 1. Two Scenarios for Lower Bound Theorem

In scenario S_1, M is at location P_1 and the set of honest beacons is $H_1 = \{B_1, B_2, B_3, \ldots, B_{k+2}\}$. Denote by $\tilde{d}_{i,1}$ the measurement $\tilde{d}_i$ in scenario S_1. So, for each $B_i \in H_1$,

$$\tilde{d}_{i,1} \sim msr(dst(B_i, P_1)).$$

In scenario S_2, M is at location P_2 and the set of honest beacons is $H_2 = \{B_1, B_2, B_{k+3}, \ldots, B_{2k+2}\}$. Denote by $\tilde{d}_{i,2}$ the measurement $\tilde{d}_i$ in scenario S_2. So, for each $B_i \in H_2$,

$$\tilde{d}_{i,2} \sim msr(dst(B_i, P_2)).$$

Assume that in scenario S_1, the adversary chooses $\tilde{d}_{k+3,1}, \ldots, \tilde{d}_{2k+2,1}$ such that

$$\forall i \in \{k+3, \ldots, 2k+2\}, \tilde{d}_{i,1} \sim msr(dst(B_i, P_2)).$$

Similarly, assume that in scenario S_2, the adversary chooses $\tilde{d}_{3,2}, \ldots, \tilde{d}_{k+2,2}$ such that

$$\forall i \in \{3, \ldots, k+2\}, \tilde{d}_{i,2} \sim msr(dst(B_i, P_1)).$$

Since B_1 and B_2 are in the perpendicular bisector of line segment P_1P_2, we have

$$dst(B_1, P_1) = dst(B_1, P_2);$$
$$dst(B_2, P_1) = dst(B_2, P_2).$$

Therefore, we have two pairs of identical distributions:

$$msr(dst(B_1, P_1)) \cong msr(dst(B_1, P_2));$$
$$msr(dst(B_2, P_1)) \cong msr(dst(B_2, P_2)).$$

Now, it is easy to see that $(\tilde{d}_{1,1}, \tilde{d}_{2,1}, \tilde{d}_{3,1}, \ldots, \tilde{d}_{2k+2,1})$ and $(\tilde{d}_{1,2}, \tilde{d}_{2,2}, \tilde{d}_{3,2}, \ldots, \tilde{d}_{2k+2,2})$ are identically distributed. Consequently, the two outputs

$$O_1 = F(\tilde{d}_{1,1}, \tilde{d}_{2,1}, \tilde{d}_{3,1}, \ldots, \tilde{d}_{2k+2,1})$$

and

$$O_2 = F(\tilde{d}_{1,2}, \tilde{d}_{2,2}, \tilde{d}_{3,2}, \ldots, \tilde{d}_{2k+2,2})$$

are also identically distributed. This implies that

$$E[dst(P_2, O_1)] = E[dst(P_2, O_2)].$$

On the other hand, by our assumption, the output errors in both scenarios are less than a:

$$e_1 = E[dst(P_1, O_1)] < a,$$
$$e_2 = E[dst(P_2, O_2)] < a.$$

Consequently,

$$\begin{aligned} dst(P_1, P_2) &= E[dst(P_1, P_2)] \\ &\leq E[dst(P_1, O_1)] + E[dst(P_2, O_1)] \\ &= E[dst(P_1, O_1)] + E[dst(P_2, O_2)] \\ &< a + a \\ &= 2a. \end{aligned}$$

This is contradictory to the fact that $dst(P_1, P_2) = 2a' \geq 2a$. ■

The above result implies that if at most $k+2$ honest nodes are available for localization, then for every configuration of beacon node positions, an adversary can select distance estimates ($\tilde{d}_i$) for all the malicious beacons such that no algorithm, given only distance estimates ($\tilde{d}_i$) from all the beacons, can compute the location of the mobile device M with a bounded error. This brings us to our next result in which we prove that, given the network model as explained in Section III, distance estimates from $k+3$ honest beacon nodes are sufficient to compute the location of a mobile device M with an error bound proportional to ϵ.

V. Solution to Robust Localization

In this section, we give the sufficient condition for localization in the presence of malicious beacon nodes. In particular, we propose LOCOMO, a framework for robust localization that efficiently computes the location of a mobile device in the presence of both malicious nodes and measurement errors. As long as distances from at least $2k+3$ beacon nodes are available (where k is the number of malicious beacons), we can prove an upper bound on the localization error of LOCOMO.

A. LOCOMO: Framework for Robust Localization

For each beacon B_i, define a ring R_i using the following inequality:

$$\tilde{d}_i - \epsilon < dst(B_i, X) < \tilde{d}_i + \epsilon.$$

As mentioned previously in Section III, ϵ is a small constant signifying some small measurement error. Clearly, there are altogether n rings. The boundary of these n rings consists of $2n$ circles—we call these circles the *boundary circles*. In particular, the inner circle of a ring is called an *inner boundary circle*, while the outer circle of a ring is called an *outer boundary circle*.

Definition 1: We say a point is a *critical point* if it is the intersection of at least two boundary circles. We say an arc is a *continuous arc* if it satisfies the following three conditions:

- The arc is part of a boundary cirle.
- If the arc is not a complete circle, then its two ends are both critical points.
- There is no other critical point in the arc.

We say an area is a *continous region* if it satisfies the following two conditions:

- The boundary of this area is one or more continous arcs.
- There is no other critical arc inside the area.

Assuming the network model discussed in Section III, Algorithm 1 above outlines our framework for robust localization. This localization strategy can tolerate the presence of upto k malicious beacon nodes if there are at least $2k+3$ beacon nodes available for localization. In this framework, Step 2 (which computes the continuous region r) needs more clarification: Does there indeed exist an intersection of at least $k+3$ rings? If there exists such an intersection, how can we find it? Our next result (Theorem 2), proves that there does exist an intersection of at least $k+3$ rings. To find this intersection, there are different ways. We discuss a couple of options in detail in Section VI.

1: For each beacon B_i, define a ring R_i using the inequality: $\tilde{d}_i - \epsilon < dst(B_i, X) < \tilde{d}_i + \epsilon$.
2: Find a (non-empty) continuous region r such that r is in the intersection of at least $k+3$ rings.
3: **if** no such continuous region r exists **then**
4: **print** "Localization cannot be done!"
5: Stop the Algorithm
6: **else**
7: Define the output O as a random point inside the continuous region r.
8: **end if**

Algorithm 1: LOCOMO: Framework for Robust Localization in the presence of Malicious Nodes

Theorem 2: For $n \geq 2k+3$, there exists a non-empty continuous region r in the intersection of at least $k+3$ rings.

Proof: Consider the real location of mobile device M. Clearly, for each honest beacon B_i, M must be in the ring R_i:

$$\tilde{d}_i - \epsilon < dst(B_i, M) < \tilde{d}_i + \epsilon.$$

Since $n \geq 2k+3$, there are at least $k+3$ honest beacons. So M must be in the intersection of at least $k+3$ rings. Define r as the continuous region in the intersection of these rings that contains the real location of M. Since M is in r, r must be non-empty.

(Figure 2 gives an illustration.) ■

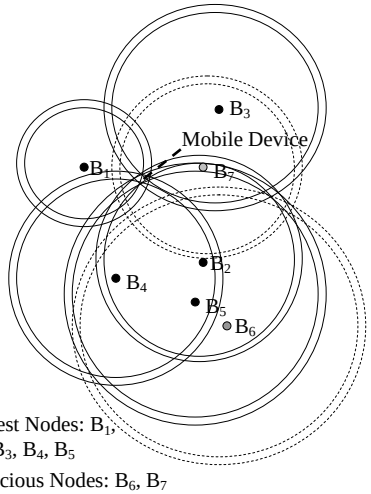

Fig. 2. Existence of Intersection of Rings ($k = 2$)

B. Error Bound Analysis

To analyze the error bound of our framework for robust localization, we need to establish a couple of new definitions.

Definition 2: The *beacon distance ratio* is defined as the minimum distance between a pair of beacons divided by the maximum distance between a beacon and the mobile device:

$$\gamma = \frac{\min_{B_i,B_j} dst(B_i,B_j)}{\max_{B_i} dst(B_i,M)}.$$

Definition 3: Consider the lines going through pairs of beacons. Denote by $ang(B_iB_j, B_{i'}B_{j'})$ the angle between lines B_iB_j and $B_{i'}B_{j'}$—to avoid ambiguity, we require that $0^\circ \le ang(B_iB_j, B_{i'}B_{j'}) \le 90^\circ$. The *minimum beacon angle* is defined as the minimum of such angles:

$$\alpha = \min_{B_i,B_j,B_{i'},B_{j'}} ang(B_iB_j, B_{i'}B_{j'}).$$

The following theorem bounds the maximum localization error possible in our robust localization framework.

Theorem 3: For $n \ge 2k+3$, if $\epsilon \ll \min_{B_i} dst(B_i,M)$ and there are no three beacons in the same line, then the error of our localization algorithm's output is

$$e < \frac{2\epsilon}{\min\left\{\sin\frac{\arcsin(\gamma\sin(\alpha/2))}{2}, \cos\frac{\arcsin(\gamma\sin(\alpha/2))}{2}\right\}}.$$

Proof: Consider the continuous region r. It is in the intersection of at least $k+3$ rings. Since there are at most k dishonest beacons, at least 3 of these rings belong to honest beacons. Suppose that R_{i_1}, R_{i_2}, and R_{i_3} are three rings belonging to honest beacons among the at least $k+3$ rings. Let r' be the continuous region in the intersection of R_{i_1}, R_{i_2}, and R_{i_3} that contains r. Since O is in r, clearly O is also in r'. Next, we show that M is also in r'. Since M is also in the intersection of R_{i_1}, R_{i_2}, and R_{i_3}, we only need to prove the following lemma.

Lemma 1: If $\epsilon \ll \min_{B_i} dst(B_i,M)$ and there are no three beacons in the same line, then the intersection of R_{i_1}, R_{i_2}, and R_{i_3} has only one continuous region.

Proof: We prove by contradiction, as illustrated in Figure 3. Suppose that the intersection of R_{i_1}, R_{i_2}, and R_{i_3} has two continuous regions r_1 and r_2. Choose arbitrary points X_1 from r_1 and X_2 from r_2.

Denote by X_1' (resp., X_2') the intersection of the line segment $B_{i_1}X_1$ (resp., $B_{i_1}X_2$) and the circle

$$dst(X,B_{i_1}) = \tilde{d}_{i_1} - \epsilon.$$

Similarly, denote by X_1'' (resp., X_2'') the intersection of the line segment $B_{i_3}X_1$ (resp., $B_{i_3}X_2$) and the circle

$$dst(X,B_{i_3}) = \tilde{d}_{i_3} - \epsilon.$$

Then clearly,

$$0 \le dst(X_1,X_1'), dst(X_1,X_1''), dst(X_2,X_2'), dst(X_2,X_2'') \le 2\epsilon. \quad (1)$$

We can see that

$$\begin{aligned}
ang(B_{i_1}B_{i_3}, B_{i_1}X_1) &= \arccos(dst(B_{i_1},X_1)^2 + dst(B_{i_1},B_{i_3})^2 \\
&\quad -dst(X_1,B_{i_3})^2) \\
&= \arccos((dst(B_{i_1},X_1') + dst(X_1,X_1'))^2 \\
&\quad +dst(B_{i_1},B_{i_3})^2 - (dst(X_1'',B_{i_3}) \\
&\quad +dst(X_1,X_1''))^2) \\
&= \arccos((\tilde{d}_{i_1} - \epsilon + dst(X_1,X_1'))^2 \\
&\quad +dst(B_{i_1},B_{i_3})^2 \\
&\quad -(\tilde{d}_{i_3} - \epsilon + dst(X_1,X_1''))^2).
\end{aligned}$$

We note that $\tilde{d}_{i_1} > dst(B_{i_1},M) - \epsilon \gg \epsilon$. Similarly, $\tilde{d}_{i_3} \gg \epsilon$. Combining these facts with (1), we have

$$\begin{aligned}
ang(B_{i_1}B_{i_3}, B_{i_1}X_1) &= \arccos((\tilde{d}_{i_1} - \epsilon + dst(X_1,X_1'))^2 \\
&\quad +dst(B_{i_1},B_{i_3})^2 \\
&\quad -(\tilde{d}_{i_3} - \epsilon + dst(X_1,X_1''))^2) \\
&\approx \arccos((\tilde{d}_{i_1})^2 + dst(B_{i_1},B_{i_3})^2 - (\tilde{d}_{i_3})^2) \\
&\approx \arccos((\tilde{d}_{i_1} - \epsilon + dst(X_2,X_2'))^2 \\
&\quad +dst(B_{i_1},B_{i_3})^2 \\
&\quad -(\tilde{d}_{i_3} - \epsilon + dst(X_2,X_2''))^2) \\
&= \arccos((dst(B_{i_1},X_2') + dst(X_2,X_2'))^2 \\
&\quad +dst(B_{i_1},B_{i_3})^2 - (dst(X_2'',B_{i_3}) \\
&\quad +dst(X_2,X_2''))^2) \\
&= \arccos(dst(B_{i_1},X_2)^2 + dst(B_{i_1},B_{i_3})^2 \\
&\quad -dst(X_2,B_{i_3})^2) \\
&= ang(B_{i_1}B_{i_3}, B_{i_1}X_2).
\end{aligned}$$

Similarly, we can show that

$$ang(B_{i_1}B_{i_2}, B_{i_1}X_1) \approx ang(B_{i_1}B_{i_2}, B_{i_1}X_2).$$

However, when we put the above two equations together, we can get a contradiction. Without loss of generality, we assume that

$$ang(B_{i_1}B_{i_2}, B_{i_1}X_1) < ang(B_{i_1}B_{i_3}, B_{i_1}X_1),$$

since otherwise we can switch the indices i_2 and i_3. It is easy to see

$$\begin{aligned}
ang(B_{i_1}B_{i_2}, B_{i_1}X_1) &= ang(B_{i_1}B_{i_3}, B_{i_1}X_1) \\
&\quad -ang(B_{i_1}B_{i_2}, B_{i_1}B_{i_3}) \\
&\le ang(B_{i_1}B_{i_3}, B_{i_1}X_1) - \alpha \\
&\approx ang(B_{i_1}B_{i_3}, B_{i_1}X_2) - \alpha \\
&= ang(B_{i_1}B_{i_2}, B_{i_1}X_2) \\
&\quad -ang(B_{i_1}B_{i_2}, B_{i_1}B_{i_3}) - \alpha \\
&\le ang(B_{i_1}B_{i_2}, B_{i_1}X_2) - 2\alpha \\
&\approx ang(B_{i_1}B_{i_2}, B_{i_1}X_1) - 2\alpha,
\end{aligned}$$

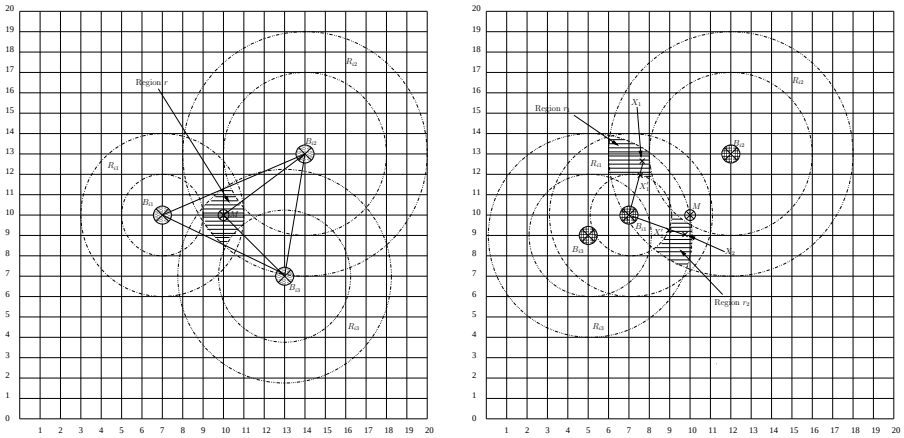

Fig. 3. Intersection of Rings

which is a contradiction. ■

Now we know that both M and O are in r'. We will use this fact to show that

$$e < \frac{2\epsilon}{\min\left\{\sin\frac{\arcsin(\gamma\sin(\alpha/2))}{2}, \cos\frac{\arcsin(\gamma\sin(\alpha/2))}{2}\right\}}.$$

But before we can prove this result, we need another lemma:

Lemma 2: If there are no three beacons in the same line, then either

$$ang(B_{i_1}M, B_{i_2}M) \geq \arcsin(\gamma\sin(\alpha/2)),$$

or

$$ang(B_{i_1}M, B_{i_3}M) \geq \arcsin(\gamma\sin(\alpha/2)).$$

Proof: Since $ang(B_{i_1}B_{i_2}, B_{i_1}B_{i_3}) \geq \alpha$, we have either $ang(B_{i_1}B_{i_2}, B_{i_1}M) \geq \alpha/2$ or $ang(B_{i_1}B_{i_3}, B_{i_1}M) \geq \alpha/2$. Below we show that, if $ang(B_{i_1}B_{i_2}, B_{i_1}M) \geq \alpha/2$, then

$$ang(B_{i_1}M, B_{i_2}M) \leq \frac{\arcsin(\gamma\sin(\alpha/2))}{2}.$$

Similarly, we can show that, if $ang(B_{i_1}B_{i_3}, B_{i_1}M) \geq \alpha/2$, then

$$ang(B_{i_1}M, B_{i_3}M) \leq \frac{\arcsin(\gamma\sin(\alpha/2))}{2}.$$

Denote by D the distance from B_{i_2} to the line $B_{i_1}M$. Then

$$\begin{aligned} ang(B_{i_1}M, B_{i_2}M) &= \arcsin\left(\frac{D}{dst(B_{i_2}, M)}\right) \\ &= \arcsin\left(\frac{dst(B_{i_1}, B_{i_2})\sin(ang(B_{i_1}B_{i_2}, B_{i_1}M))}{dst(B_{i_2}, M)}\right) \\ &\geq \arcsin\left(\frac{dst(B_{i_1}, B_{i_2})\sin(\alpha/2)}{dst(B_{i_2}, M)}\right) \\ &\geq \arcsin(\gamma\sin(\alpha/2)). \end{aligned}$$

■

Using the above lemma, we know that, without loss of generality, we can assume that

$$ang(B_{i_1}M, B_{i_2}M) \geq \arcsin(\gamma\sin(\alpha/2)).$$

Denote by r'' the continuous region in the intersection of R_{i_1} and R_{i_2} that contains r'. Since both M and O are in r', they should also be in r''.

Each of the two rings involved has a pair of circles. Consider the four intersection points of these two pairs of circles. Without loss of generality, we suppose that the four intersection points are V_1, V_2, V_3, and V_4, ordered in the clockwise direction, and that $\angle V_2V_1V_4$ is acute. Since $\epsilon \ll \min_{B_i} dst(B_i, M)$, we can approximate r'' using the quadrangle $V_1V_2V_3V_4$. It is easy to show that

$$ang(V_1V_2, B_{i_1}M) \approx 90^\circ \approx ang(V_3V_4, B_{i_1}M);$$

thus we know that the line V_1V_2 is parallel to the line V_3V_4. Similarly, we can get that the line V_1V_4 is parallel to the line V_2V_3. Therefore, $V_1V_2V_3V_4$ is a parallelogram. Furthermore, we observe that

$$\begin{aligned} \angle V_2V_1V_3 &= \arcsin\left(\frac{2\epsilon}{dst(V_1, V_3)}\right) \\ &= \angle V_3V_1V_4. \end{aligned}$$

Therefore, $V_1V_2V_3V_4$ is actually a rhombus. In a rhombus, the farthest distance between two points is the length of its longer diagonal line. Therefore,

$$\begin{aligned} e = dst(M, O) &\leq \frac{2\epsilon}{\sin(\angle V_2V_1V_3)} \\ &= \frac{2\epsilon}{\sin\left(\frac{\angle V_2V_1V_4}{2}\right)} \\ &\approx \frac{2\epsilon}{\min\left\{\sin\left(\frac{ang(B_{i_1}M, B_{i_2}M)}{2}\right), \sin\left(90^\circ - \frac{ang(B_{i_1}M, B_{i_2}M)}{2}\right)\right\}} \end{aligned}$$

$$\leq \frac{2\epsilon}{\min\left\{\sin\left(\frac{\arcsin(\gamma\sin(\alpha/2))}{2}\right), \cos\left(\frac{\arcsin(\gamma\sin(\alpha/2))}{2}\right)\right\}}$$

■

VI. Finding Continuous Region 'r'

In this section, we focus on the Step 2 of our LOCOMO framework, which is to efficiently determine the continuous region r. We do not necessarily need to compute the entire continuous region r. Just determining a random point inside of this region would suffice. We present two algorithms for doing this. The first algorithm, called *polynomial-time algorithm*, determines the continuous region r by computing all the boundary arcs of r and then based on the position of a particular boundary arc determines a point inside the continuous region. The second algorithm, called the *fast heuristic algorithm*, employs an interesting heuristic to determine a point and then checks to see if this point lies inside the continuous region r. The advantage of the polynomial-time algorithm is that it is guaranteed to finish in polynomial time (more precisely, $O(n^3 \log n)$ time) even in the worst case. However, in practice, since the worst-case scenario rarely occurs, the heuristic algorithm is much faster than the polynomial-time algorithm. Our experiments show that the heuristic algorithm finishes computing a location in about $2 - 50$ milliseconds (see Section VIII for details). Therefore, the polynomial-time algorithm is mainly of theoretical interests, while the heuristic algorithm is suitable for use in practice.

A. Polynomial-time Algorithm

Before we present the polynomial-time algorithm for finding the continuous regin r, we require a lemma that gives the relationship between the continuous region and the continuous arcs on the boundary of this region.

Definition 4: A ring is *related* to a continuous arc if the continuous arc is inside the ring but not on the boundary of this ring.

Lemma 3: Suppose that r is a continuous region and c is a continuous arc on the bounday of r. Then r is in the intersection of at least $k+3$ rings if and only if at least $k+2$ rings are related to c.

We skip the proof of Lemma 3 since it is straightforward. Now, to determine a continuous region in the intersection of at least $k+3$ rings, the algorithm needs to count the number of rings related to each continuous arc and find a continuous arc that at least $k+2$ rings are related to. (It is easy to check if a ring is related to a *continuous* arc by comparing the distance between the arcs end points and the center of the ring to the inner and outer radii of the ring.) Once such an arc is found, depending on whether the arc is on an outer boundary circle or an inner boundary circle, a random point can be picked from either the inner region or the outer region of the arc respectively. The details of the algorithm are as shown in Algorithm 2.

```
1: Let S be a set initially containing the two boundary circles of ring R_1.
2: for i = 2, ..., n do
3:    Let S_i be a set initially containing the two boundary circles of ring R_i.
4:    for each arc in S and each arc in S_i do
5:       if the above two arcs intersect then
6:          Split each of these two arcs using the intersection(s), and replace them in the corresponding arc sets (S or S_i) with the new splitted arcs (result of the splitting operation).
7:       end if
8:    end for
9:    Let S = S ∪ S_i.
10: end for
11: for each arc c_j in S do
12:    Set the corresponding counter λ_j to 0.
13:    for i = 1, ..., n do
14:       if R_i is related to c_j then
15:          λ_j = λ_j + 1.
16:       end if
17:    end for
18:    if λ_j ≥ k + 2 then
19:       if c_j is on an inner boundary circle then
20:          Output is defined on the side out of this circle.
21:       else if c_j is on an outer boundary circle then
22:          Output is defined on the side inside this circle
23:       end if
24:       Stop the algorithm.
25:    end if
26: end for
```

Algorithm 2: Polynomial-time Algorithm for Finding the Continuous Region

Lemma 4: The worst-case time complexity of the above algorithm is $O(n^3 \log n)$.

B. Fast Heuristic Algorithm

Though the worst case time complexity of the polynomial-time algorithm is polynomial ($O(n^3 \log n)$) in terms of the total number of beacon nodes, in practice its efficiency needs improvement. Trial experimental runs for the polynomial-algorithm have shown that the execution time is of the order of seconds which may not be efficient for most applications. (For sake of brevity, we do not include the results of those experiments in this paper.) To overcome this problem, we propose an efficient heuristic-based technique.

The heuristic we use is as follows. Note that $k+3$ is already a large number of rings. Since the region r is contained in at least $k+3$ rings, the rings containing r are intersecting with large numbers of other rings. Therefore, if a ring R_i is intersecting with a large number of rings, it is very likely that R_i contains r. So, we should first consider the rings intersecting with the maximum

numbers of other rings. The details of our heuristic algorithm is shown in Algorithm 3.

```
Count the number of rings intersecting with each ring.
for each ring R_i, in the order of decreasing number of rings intersecting with it do
    for each ring R_j, R_j ≠ R_i, in the order of decreasing number of rings intersecting with it do
        Compute the intersection points of the boundary circles of R_i and R_j.
        for k = 1, ..., κ do
            Choose a random intersection point computed above.
            Choose a random point O near this intersection point (such that the distance between them is less than ε).
            Count the number of rings containing O.
            if there are at least k + 3 rings containing O then
                Output O.
                Stop the Algorithm.
            end if
        end for
    end for
end for
```

Algorithm 3: Fast Heuristic Algorithm for Finding the Continuous Region

VII. Extension to 3-Dimensional Localization

So far we have only considered localization in a 2-dimensional space. In certain environments (like mountains and valleys), 3-dimensional localization is needed. In this section, we extend our results to the 3-dimensional space.

The first result we obtain is the lower bound theorem. It turns out that we need one more honest beacon node than in the 2-dimensional case.

Theorem 4: Suppose that $n \leq 2k+3$. Then, for any distance-based 3-dimensional localization algorithm, for any locations of the beacons, there exists a scenario in which $e \rightarrow +\infty$.

With $2k+4$ beacon nodes, we can also establish a bounded error for 3-dimensional localization. But to obtain this result, we need to first introduce a few new definitions.

For each beacon B_i, we define a global shell just as we defined the ring for the 2-dimensional case:

$$\tilde{d}_i - \epsilon < dst(B_i, X) < \tilde{d}_i + \epsilon.$$

For simplicity, we still use R_i to denote the above global shell. The globes on the boundary of these shells are called the *boundary globes*; the inner globe of a shell is called an inner boundary globe, while the outer globe of a shell is called an outer boundary circle. A *continous 3-dimensional region* is part of the space such that its boundary consists of parts of boundary globes, and that no boundary globe goes through its internal. Our 3-dimensional localization algorithm finds a continuous 3-dimensional region r such that r is in the intersection of at least $k+4$ global shells. The output O of the algorithm is defined as a random point in the region r.

Definition 5: Consider the planes going through triples of beacons. Denote by $ang(B_{i_1}B_{i_2}B_{i_3}, B_{i'_1}B_{i'_2}B_{i'_3})$ the angle between the two planes $B_{i_1}B_{i_2}B_{i_3}$ and $B_{i'_1}B_{i'_2}B_{i'_3}$—to avoid ambiguity, we require that $0^\circ \leq ang(B_{i_1}B_{i_2}B_{i_3}, B_{i'_1}B_{i'_2}B_{i'_3}) \leq 90^\circ$. The *minimum beacon plane angle* is defined as the minimum of such angles:

$$\alpha^\star = \min_{B_{i_1},B_{i_2},B_{i_3},B_{i'_1},B_{i'_2},B_{i'_3}} ang(B_{i_1}B_{i_2}B_{i_3}, B_{i'_1}B_{i'_2}B_{i'_3}).$$

Given the above definitions, we can now state our main (positive) result on 3-dimensional localization.

Theorem 5: For $n \geq 2k+4$, if $\epsilon \ll \min_{B_i} dst(B_i, M)$ and there are neither three beacons in the same line nor four beacons in the same plane, then the error of our localization algorithm's output is

$$e < 2\epsilon\sqrt{\frac{1}{\beta^2} + (\frac{1}{\sin\alpha^\star} + \frac{1}{\beta \cdot \tan\alpha^\star})^2},$$

where $\beta = \min\left\{\sin\left(\frac{\arcsin(\gamma\sin(\alpha/2))}{2}\right), \cos\left(\frac{\arcsin(\gamma\sin(\alpha/2))}{2}\right)\right\}$.

VIII. Evaluation

To evaluate the performance of LOCOMO (with the heuristic algorithm for computing r), we carry out extensive experiments using the ns2 network simulator tool. The setup of our experiments is as follows: The simulation area is $500\text{m} \times 500\text{m}$. The radio transmission range is 500m. We use 802.11 as the MAC protocol. The distance between each beacon node and the mobile device is computed using the Received Signal Strength Indicator (RSSI) technique.

Our experiments consider two different distributions of measurements errors: uniform distribution and normal distribution. For each of these two distributions, we study how the number of malicious beacons (k) and the magin of measurement error (ϵ) influence the localization error and the computational time. We do not evaluate the communication overhead because it depends on the method used to measure distance. Since LOCOMO is a general framework independent of the method to measure distance, the communication overhead is not directly related with LOCOMO.

A. Experiments with Uniform Measurement Error

Now we study the scenario in which the random measurement error is uniformly distributed over $[-\epsilon, \epsilon]$. First, we observe the performance of LOCOMO for each values of ϵ, when the number of malicious nodes (k) in the network increases but the total number of nodes in

the network in each simulation is fixed ($2k_{max}+3$). Here k_{max} is some maximum value for k. Such a setup ensures that for smaller values of $k, k < k_{max}$, there are more than $k+3$ honest beacon nodes available for localization and when $k = k_{max}$, there will be enough number of honest beacon nodes (exactly, $k+3$) for the target node to localize itself correctly. We run the simulation of LOCOMO for each value of ϵ from 0 to 5.0 and each value of k from 0 to 20. Thus, $k_{max} = 20$. In each simulation, the total number of beacon nodes available for localization is $2k_{max}+3 = 43$.

We run the simulation for each pair of k and ϵ for 100 times and average the localization error. From Figure 4, we can see that the average localization error (e) is increasing when ϵ increases, which is very natural. However, e is decreasing when k increases—this looks counter-intuitive. The reason lies in our heuristic algorithm for computing r: It takes k as input and computes the intersection of $k+3$ rings. Hence, the larger k is, the intersection of more rings the algorithm needs to find, and the better precision we have in the localization result.

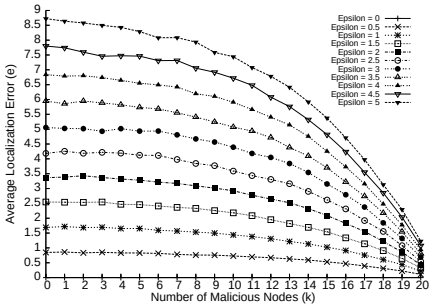

Fig. 4. Localization Error for Uniform Measurement Error

We emphasize that the increased localization precision here (for larger k) is actually *the reward of spending more time on localization*: As shown in Figure 5, the average simulation time is increasing in k.[1] Consequently, it is *not* the increase of k that gives us better precision in localization; it is the increase of computational time that gives us better precision.

To see this more clearly, we have another set of experiments in which our heuristic algorithm is slightly modified: For every value of k, the algorithm always looks for the intersection of $k_{max}+3(=23)$ rings rather than $k+3$ rings. In other words, for different values of k, we always find the intersection of the same number of rings. We keep everything else intact and run the simulation again for each pair of k and ϵ for 100 times.

As shown in Figure 6, in this second set of experiments, the average localization error e now goes

[1]Note that the average simulation time increases pretty slowly, except when k approaches k_{max}, the maximum number of malicious nodes we can tolerate. Moreover, even when $k = k_{max}$, the average simulation time is only about 50 milliseconds. Thus, the time cost of LOCOMO is reasonable, although it increases in k.

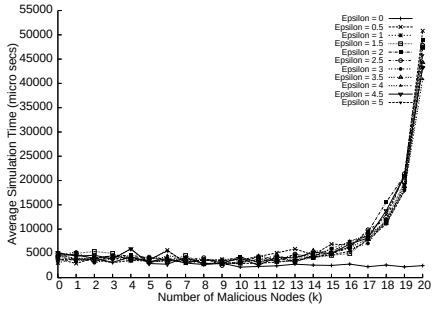

Fig. 5. Simulation Time for Uniform Measurement Error

up with an increasing k. This is consistent with our intuition that more malicious beacon nodes should lead to worse localization precision. Figure 7 shows that, in the second set of experiments, the average simulation time still increases in k, but increases *much more slowly*. Putting these two observations together, we have verified our conjecture on the change of localization precision in the first set of experiments: The localization error actually should increase when k goes up, but it becomes decreasing when we spend much more computational time for larger k.

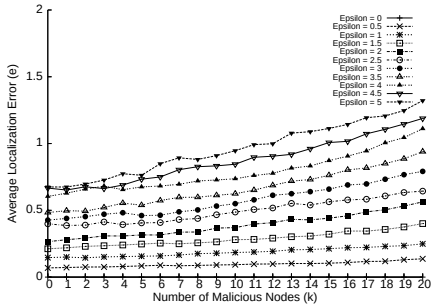

Fig. 6. Localization Error for Uniform Measurement Error with Modified Algorithm

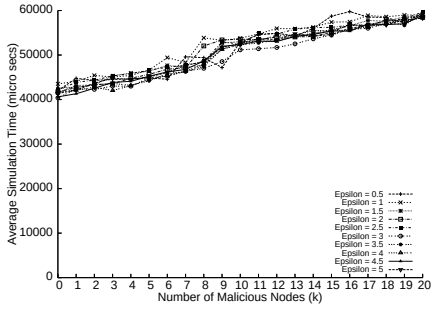

Fig. 7. Simulation time for Uniform Measurement Error with Modified Algorithm

To summarize, there is a tradeoff among the time we spend on localization, the number of malicious beacon

nodes we can tolerate, and the precision of the localization result. To have better precision in localization, we have to either increase the computational time, or decrease the number of malicious beacons we can tolerate. In practice, the precision we can obtain is decided by our time constraint and malicious beacon constraint.

The influence of ϵ on the performance of LOCOMO is relatively simple. A smaller ϵ can make the localization error e smaller, but it does not have any significant influence on the computational time needed.

B. Experiments with Normal Measurement Error

To ensure that our evaluation result is not restricted to a uniformly distributed measurement error, we repeat all our experiments with a normally distributed measurement error. Here we keep all our experiment parameters intact, except that the distance measurement error follows a normal (Gaussian) distribution with mean 0 and variance $\frac{\epsilon}{2}$. However, we need to make sure that the measurement error value is between $[-\epsilon,+\epsilon]$. Therefore, we modify the distribution a little such that the probability density outside $[-\epsilon, +\epsilon]$ becomes 0; the probability density inside the interval $[-\epsilon, +\epsilon]$ is scaled up a little accordingly.

Figure 8 shows the localization error for each pair of (k, ϵ) when the measurement error follows the normal distribution. Figure 9 shows the corresponding simulation time. We can see that the curves are analogous to those in Figures 4 and 5.

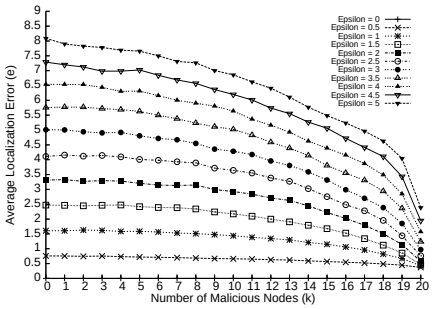

Fig. 8. Localization Error for Normal Measurement Error

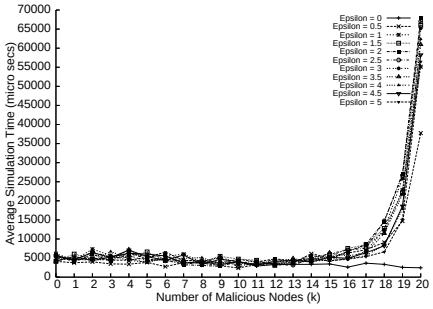

Fig. 9. Simulation Time for Normal Measurement Error

We also experiment with the modified algorithm in which we find the intersection of the same number of rings for different values of k. Figure 10 and 11 show the localization error and simulation time, respectively. Again they are analogous to Figures 6 and 7, respectively. Therefore, our evaluation result is valid for different distributions of measurement errors.

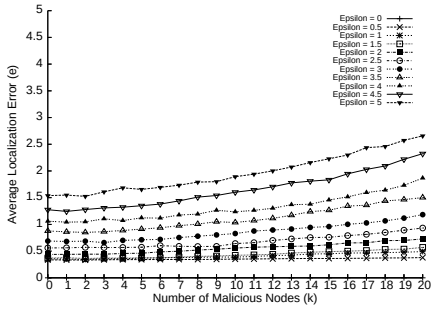

Fig. 10. Localization Error for Normal Measurement Error with Modified Algorithm

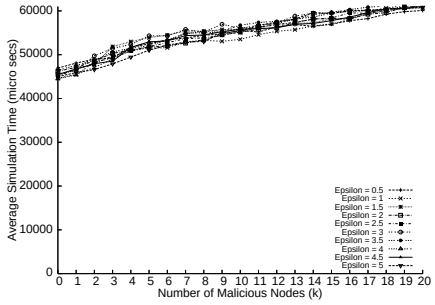

Fig. 11. Simulation time for Normal Measurement Error with Modified Algorithm

IX. Conclusion and Open Question

In this paper, we have presented LOCOMO, a distance-based localization framework for mobile devices against malicious beacon nodes. LOCOMO is optimal in the sense that it provides a guaranteed degree of localization accuracy against the maximum possible number of malicious beacon nodes. LOCOMO can be used with different algorithms for finding the intersection of rings. In particular, there is a polynomial-time algorithm that guarantees LOCOMO to finish in polynomial time even in the worst case. There is also a fast heuristic algorithm that is suitable for use in practice. LOCOMO can be extended for 3-dimensional localization. Our evaluations demonstrate that LOCOMO provides good localization precision with a very small time cost.

An open question is what is the best algorithm to find the intersection of rings, in terms of worst-case complexity and in terms of average computational time? We leave this question for future work.

REFERENCES

[1] P. Bahl and V. N. Padmanabhan. Radar: an in-building RF-based user location and tracking system. In *IEEE INFOCOM Conference Proceedings*, pages 775–784. IEEE Communications Society, March 2000.

[2] J. Bruck, J. Gao, and A. A. Jiang. Localization and routing in sensor networks by local angle information. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 181–192, New York, NY, USA, 2005. ACM Press.

[3] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, pages 28–34, Oct 2000.

[4] L. Doherty, L. E. Ghaoui, and K. S. J. Pister. Convex position estimation in wireless sensor networks. In *IEEE INFOCOM Conference Proceedings*, Anchorage, April 2001. IEEE Communications Society.

[5] T. Eren, D. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. Anderson, and P. Belhumeur. Rigidity, computation and randomization of network localization. In *The IEEE INFOCOM 2004. Proceedings.*, Hong Kong, China, April 2004. IEEE Computer and Communications Society.

[6] L. Fang, W. Du, and P. Ning. A beacon-less location discovery scheme for wireless sensor networks. In *IEEE INFOCOM Conference Proceedings*. IEEE Communications Society, March 2005.

[7] D. Goldenberg, A. Krishnamurthy, W. Maness, Y. Yang, A. Young, A. Morse, A. Savvides, and B. Anderson. Network localization in partially localizable networks. In *The IEEE INFOCOM 2005. Proceedings*, Miami, FL, March 2005.

[8] M. Greis. *Tutorial for the Network Simulator "ns"*. VINT group, 2005. http://www.isi.edu/nsnam/ns/.

[9] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 81–95, New York, NY, USA, 2003. ACM Press.

[10] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, August 2001.

[11] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning System: Theory and Practice*. Springer Verlag, 1997.

[12] X. Ji and H. Zha. Sensor positioning in wireless ad-hoc sensor. networks using multidimensional scaling. In *Proceedings of IEEE INFOCOM 2004*, March 2004.

[13] C. A. R. Jr. Sensors bolster army prowess. *SIGNAL Magazine, AFCEA's International Journal*, 2004. http://www.afcea.org/signal/articles/anmviewer.asp?a=30.

[14] W. R. P. Jr., T. H. de Paula Figueiredo, H. C. Wong, and A. A. Loureiro. Malicious node detection in wireless sensor networks. In *18th International Parallel and Distributed Processing Symposium, 2004. Proceedings.*, page 24. IEEE Computer Society, April 2004.

[15] B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *The Sixth Annual International Conference on Mobile Computing and Networking(MOBICOM)*. ACM SIGMOBILE, August 2000.

[16] Y. B. Ko and N. Vaidya. Location-aided routing(lar) in mobile ad hoc networks. In *The Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking(MOBICOM)*, pages 66–75. ACM SIGMOBILE/IEEE Communications Society, October 1998.

[17] L. Lazos and R. Poovendran. Serloc: secure range-independent localization for wireless sensor networks. In *WiSe '04: Proceedings of the 2004 ACM workshop on Wireless security*, pages 21–30, New York, NY, USA, 2004. ACM Press.

[18] D. Liu, P. Ning, and W. Du. Attack-resistant location estimation in sensor networks. In *The Fourth International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pages 99–106. ACM SIGBED and IEEE Signal Processing Society, April 2005.

[19] D. Liu, P. Ning, and W. Du. Detecting malicious beacon nodes for secure location discovery in wireless sensor networks. In *The 25th International Conference on Distributed Computing Systems (ICDCS '05)*, pages 609–619. IEEE Computer Society, June 2005.

[20] K. Lorincz, D. Malan, T. R. F. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, S. Moulton, and M. Welsh. Sensor networks for emergency response: Challenges and opportunities. *IEEE Pervasive Computing, Special Issue on Pervasive Computing for First Response*, Oct-Dec 2004.

[21] D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 50–61, New York, NY, USA, 2004. ACM Press.

[22] D. Niculescu and B. Nath. DV based positioning in ad hoc networks. *Journal of Telecommunication Systems*, 2003.

[23] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *The Sixth Annual International Conference on Mobile Computing and Networking(MOBICOM)*, pages 32–43. ACM SIGMOBILE, August 2000.

[24] S. Ray, R. Ungrangsi, F. de Pellegrini, A. Trachtenberg, and D. Starobinski. Robust location detection in emergency sensor networks. In *IEEE INFOCOM Conference Proceedings*, pages 1044–1053, San Francisco, March 2003. IEEE Communications Society.

[25] N. Sastry, U. Shankar, and D. Wagner. Secure verification of location claims. In *WiSe '03: Proceedings of the 2003 ACM workshop on Wireless security*, pages 1–10, New York, NY, USA, 2003. ACM Press.

[26] A. Savvides, W. Garber, S. Adlakha, R. Moses, and M. Srivastava. On the error characteristics of multihop node localization in ad-hoc sensor networks. In *Proceedings of the Second International Workshop on Information Processing in Sensor Networks (IPSN'03)*, pages 317–332, Palo Alto, California, USA, April 2003.

[27] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz. Localization from connectivity in sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 15(11):961–974, 2004.

[28] R. Stoleru and J. A. Stankovic. Probability grid: A location estimation scheme for wireless sensor networks. In *IEEE Sensor and Ad Hoc Communications and Networks*, pages 430–438. IEEE Communications Society, October 2004.

[29] Y.-C. Tseng, M.-S. Pan, and Y.-Y. Tsai. Wireless sensor networks for emergency navigation. *Computer*, 39(7):55–62, 2006.

[30] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *ACM Transaction on Information Systems*, pages 91–102, Jan 1992.

[31] K. Yedavalli, B. Krishnamachari, S. Ravula, and B. Srinivasan. Ecolocation: A sequence based technique for rf-only localization in wireless sensor networks. In *Proceedings of the Fourth International Conference on Information Processing in Sensor Networks (IPSN '05)*, Los Angeles, CA, USA, April 2005.

[32] L. Yu, N. Wang, and X. Meng. Real-time forest fire detection with wireless sensor networks. In *Proceedings. 2005 International Conference on Wireless Communications, Networking and Mobile Computing*, volume 2, pages 1214–1217, September 2005.