Exploiting Jamming-Caused Neighbor Changes for Jammer Localization

Zhenhua Liu, Hongbo Liu, Wenyuan Xu, and Yingying Chen

Abstract—Jamming attacks are especially harmful when ensuring the dependability of wireless communication. Finding the position of a jammer will enable the network to actively exploit a wide range of defense strategies. In this paper, we focus on developing mechanisms to localize a jammer by exploiting neighbor changes. We first conduct jamming effect analysis to examine how the communication range alters with the jammer's location and transmission power using free space model. Then, we show that a node's affected communication range can be estimated purely by examining its neighbor changes caused by jamming attacks and thus, we can perform the jammer location estimation by solving a least-squares (LSQ) problem that exploits the changes of communication range. Compared with our previous iterative-search-based virtual force algorithm, our LSQ-based algorithm exhibits lower computational cost (i.e., one-step instead of iterative searches) and higher localization accuracy. Furthermore, we analyze the localization challenges in real systems by building the log-normal shadowing model empirically and devising an adaptive LSQ-based algorithm to address those challenges. The extensive evaluation shows that the adaptive LSQ-based algorithm can effectively estimate the location of the jammer even in a highly complex propagation environment.

Index Terms—Jamming, Radio interference, Least squares, Localization.

1 Introduction

The rapid advancement of wireless technologies has enabled a broad class of new applications utilizing wireless networks, such as patient tracking and monitoring via sensors, traffic monitoring through vehicular ad hoc networks, and emergency rescue and recovery based on the availability of wireless signals. To ensure the successful deployment of these pervasive applications, the dependability of the underneath wireless communication becomes utmost important. Among various threats that can undermine the normal wireless communication, jamming attacks are especially harmful towards achieving reliable wireless communication. As the wireless communication medium is shared by nature, an adversary may just inject false messages or emit radio signals to block the wireless medium and prevent other wireless devices from even communicating. Furthermore, the increasingly flexible programming interface of commodity devices makes launching jamming attacks with little efforts. For instance, an adversary can easily purchase a commodity device and reprogram it to introduce packet collisions that force repeated backoff of other legitimate users and thus, disrupt network communications.

To ensure the dependability of wireless communication, much work has been done to detect and defend against jamming attacks. The existing countermeasures for coping with jamming include two types: the proactive conventional physical-layer techniques that provide resilience to interference by employing advanced transceivers [1], e.g., frequency hopping, and the reactive non-physical-layer strategies that defend against jamming leveraging MAC or network layer mechanisms, e.g., adaptive error correcting codes [2], channel adaption [3], spatial relocation [4], or constructing wormholes [5]

Few studies have been done in identifying the physical location of a jammer. However, localizing a jammer is an important task, which not only allows the network to actively exploit a wide range of defense strategies but also provides important information for network operations in various layers. For instance, a routing protocol can choose a route that does not traverse the jammed region to avoid wasting resources caused by failed packet deliveries. Alternatively, once a jammer's location is identified, one can eliminate the jammer from the network by neutralizing it. In light of the benefits, in this paper, we address the problem of localizing a jammer.

Although there has been active research in the area of localizing wireless devices [6]–[8], most of those localization schemes are inapplicable to jamming scenarios. For instance, many localization schemes require the wireless device to be equipped with specialized hardware [6], [9], e.g., ultrasound or infrared, or utilize signals transmitted from wireless devices to perform localization. Unfortunately, the jammer will not cooperate and the jamming signal is usually embedded in the legal signal and thus, is hard to extract, making the signal-based and special-hardware-based

- *Z. Liu and W. Xu are with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29205. E-mail: {liuz,wyxu}@cse.sc.edu*
- *H. Liu and Y. Chen are with the Department of Electrical and Computer Science, Stevens Institute of Technology, Castle Point on Hudson, Hoboken, NJ 07030. E-mail: {hliu3,yingying.chen}@stevens.edu*

approaches inapplicable.

Recent work [10], [11] on jamming localization algorithms relies on metrics other than signals. Without presenting performance evaluation, gradient descent search method based on packet delivery rate (PDR) [11] has been proposed to localize the jammer. In our prior work, we introduce the concept of virtual forces, which are calculated by examining the node state. Guided by virtual forces, the algorithm pushes or pulls the estimated location of the jammer towards its true position iteratively [10].

In this paper, we propose a non-iterative algorithm to localize a jammer, which exploits a node's neighbor list changes caused by jamming attacks. We have discovered that a jammer may reduce the size of a node's hearing range, an area from which a node can successfully receive and decode the packet, and the level of changes is determined by the relative location of the jammer and its jamming intensity. Therefore, instead of searching for the jammer's position iteratively, we can estimate the hearing range by identifying neighbor changes and localize the jammer in one round, which significantly reduces the computational cost yet achieves better localization performance than prior work [10].

We organize the remainder of the paper as follows: we specify our jamming attack model and provide an analysis on jamming effects in Section 3. Then, we discuss our basic LSQ-based algorithm in Section 4. In Section 5, we present our effort in building a realistic propagation model through empirical study, and introduce the adaptive LSQ-based algorithm that can address radio irregularity. In Section 6, we conduct simulation evaluation and present the performance results. Finally, we conclude in Section 7.

2 Related Work

Most of the works dealing with jamming interference is addressed through conventional PHY-layer communication techniques. In these systems, spreading techniques (e.g. frequency hopping) are commonly used to provide resilience to interference [1], [12]. Although such PHY-layer techniques can address the challenges of an RF interferer, they require advanced transceivers.

The issue of detecting jammers was briefly studied by Wood et al. [13], and was further studied by Xu et al. [14], where the authors presented several jamming models and explored the need for more advanced detection algorithms to identify jamming. Jamming detection was also studied in the context of sensor networks [15], [16] and in networks involving frequency hopping [17]. Our work focuses on localizing jammers after jamming attacks have been identified using those jamming detection strategies.

Without localizing jammers, Wood et al. [13] has studied how to map the jammed region. The basic idea is to have the jammed nodes bypass their MAC-layer temporarily and announce the fact that they are jammed. With slight modification, our algorithm identifying neighbor list changes can also map the jammed region.

Moreover, countermeasures for coping with jammed regions in wireless networks have been investigated. The use of error correcting codes [2] is proposed to increase the likelihood of decoding corrupted packets. Channel surfing/ hopping [3], [18], [19], whereby wireless devices change their working channel to escape from jamming, spatial retreats [4], whereby wireless devices move out of jammed region geographically, and anti-jamming timing channel [20], whereby data are communicated via a covert timing channel that is built on failed-packet-delivery event, are proposed to cope with jamming. Additionally, wormhole-based anti-jamming techniques have been proposed as a means to allow the delivery of important alarm messages [5]. The combinations of mask framing, frequency hopping, packet fragmentation, and redundant encoding techniques is proposed to cope with multiple types of jammers [21].

Wireless localization has been an active area, attracting many attentions. Based on localization infrastructure, infrared [6] and ultrasound [9], [22] are employed to perform localization, both of which need to deploy specialized infrastructure for localization. Further, using received signal strength (RSS) [7], [8], [23], [24] is an attractive approach because it can reuse the existing wireless infrastructure. Based on the localization methodology, the localization algorithms can be categorized into range-based and range-free. Range-based algorithms involve estimating distance to anchor points with known locations by utiliziing the measurement of various physical properties, such as RSS [7], [8], [23], [25], Time Of Arrival [26], and Time Difference of Arrival [9]. Range-free algorithms [27]–[30] use coarser metrics to place bounds on candidate positions.

However, little work has been done in localizing jammers. Because of the disturbed network communication under jamming attacks, most of the existing localization methods can not be applied to localize jammers. Recently Pelechrinis et al. [11] proposed to localize the jamming by measuring packet delivery rate (PDR) and performing gradient decent search. However, they did not present performance evaluation. Liu *et al.* [10] utilized the network topology changes caused by jamming attacks and estimated the jammer's position by introducing the concept of virtual forces. The virtual forces are derived from the node states and can guide the estimated location of the jammer towards its true position iteratively. Both jamming localization algorithms [10], [11] are iterative-based, while our algorithm leverages the neighbor changes caused by jamming attacks to lo-

calize jammers in one round.

3 Analysis of Jamming Effects

In this section, we start by outlining basic wireless networks and jammers that we use throughout this paper and briefly reviewing the theoretical underpinning for analyzing the jamming effects. Then, we study the impact of one jammer with an omnidirectional antenna on the wireless communication at two levels: the individual communication range level and the network topology level.

3.1 Network Model and Assumptions

We target to design our solutions for a category of wireless networks with the following characteristics.

Multi-hop. We consider a large-scale network, which is densely deployed. We assume that each node has one transmission rate and communicates in a multihop fashion. One example of such a network could be a sensor network.

Stationary. Once deployed, the location of each node remains unchanged. Mobility will be considered in our future works.

Neighbor-Aware. Each node in the network maintains a table that stores the information of its neighbors, such as their locations or activeness. Such a neighbor table is supported by most routing protocols and can be easily implemented by periodically broadcasting beacons. Moreover, each node is able to track the change on its neighbor table.

Location-Aware. Each node is aware of its own location and its neighbors' locations. This can be achieved relatively easy as many applications already require localization services [7].

Homogeneous. Each node is equipped with an omnidirectional antenna and transmits at the same transmission power level.

Adaptive-CCA. Clear channel assessment (CCA) is an essential component of Carrier Sense Multiple Access (CSMA), the de-facto medium access control (MAC) protocols in many wireless networks. In particular, each network node is only allowed to transmit packets when the channel is idle by using CCA as channel detection. Typically, CCA works as follows: before transmitting, a wireless device samples the ambient noise floor for a short period and it will transmit only if the sampled value is larger than a threshold Υ. Studies [31] have shown that adaptive-CCA, which adjusts the threshold Υ based on the ambient noise floor, can achieve better throughput and latency than using a pre-determined threshold Υ. Therefore, we assume that each node employs an adaptive-CCA mechanism in our study.

In this work, we focus on locating a jammer after it is detected. Thus, we assume the network is able to identify a jamming attack, leveraging the existing jamming detection approaches [13], [14].

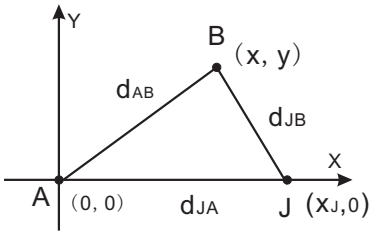


Fig. 1. The coordinate system for the hearing range and the sending range of Node A, wherein A and B are network nodes, and J is the jammer.

3.2 Jamming Model

There are many different attack strategies that a jammer can perform in order to disrupt wireless communications. In this work, we focus on a representative jammer with the following characteristics.

Constant jammer. We use a constant jammer that continually emits a radio signal, regardless whether the channel is idle or not.

Omnidirectional. Each jammer is equipped with an omnidirectional antenna and transmits at the same power level. Thus, every jammer has the same jamming range in all directions.

Non-overlapping. We assume there are one or more jammers in the network, but none of their jamming regions overlap.

3.3 Communication in Non-Jamming Scenarios

Before analyzing the impact of jamming on the communication range, we briefly review the key factors that affect packet deliveries. Essentially, the MAC layer concept, packet delivery ratio (PDR), is determined by the physical metric, signal-to-noise ratio (SNR). At the bit level, the bit error rate (BER) depends on the probability that a receiver can detect and process the signal correctly. To process a signal and derive the associated bit information with high probability, the signal has to exceed the noise by certain amount. Given the same hardware design of wireless devices, the minimum required surplus of signals over ambient noise is roughly the same. We use γ_o to denote the *minimum SNR*, the threshold required to decode a signal successfully. We consider that Node A is unable to receive messages from Node B when $(SNR)_{B\to A} < \gamma_o$, where $(SNR)_{B\to A}$ denotes the SNR of messages sent by B measured at A.

The communication range defines a node's ability to communicate with others, and it can be divided into two components: the ***hearing range*** and the ***sending range***.

- ***The hearing range.*** Consider Node A as a receiver, the hearing range of A specifies the area within which the potential transmitters can deliver their message to A, e.g. for any Transmitter S in A's hearing range, $(SNR)_{S\to A} > \gamma_o$.
- ***The sending range.*** Similarly, consider A as a transmitter, the sending range of A defines the

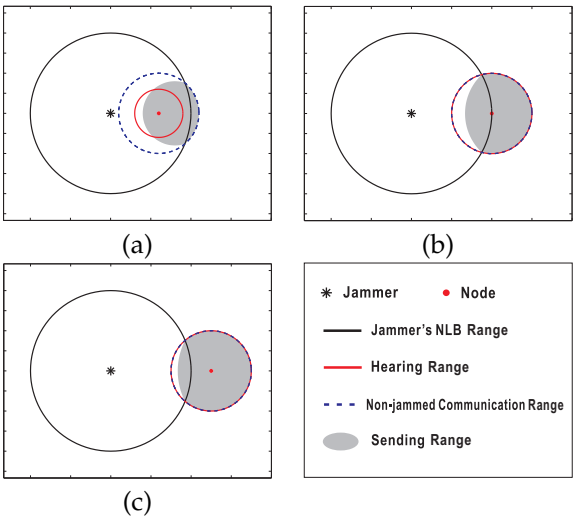


Fig. 2. The hearing range, the sending range, and the non-jammed communication range when the location of a jammer is fixed and a node is placed at different spots: (a) inside the jammer's NLB; (b) at the edge of the jammer's NLB; (c) outside the jammer's NLB.

region within which the potential receivers have to be located to assure receiving A's messages, e.g., for any Receiver R in A's sending range, $(SNR)_{A\to R} > \gamma_o$.

Consider the standard free-space propagation model, the received power is

$$P_R = \frac{P_T G}{4\pi d^2}, \tag{1}$$

where P_T is the transmission power, G is the product of the sending and receiving antenna gain in the LOS (line-of-sight) between the receiver and the transmitter, and d is the distance between them.

Given that in a non-jamming scenario the average ambient noise floor P_N are the same, both the hearing range and the sending range of Node A will be the same, a circle centered at A with a radius of $r_c = \sqrt{\frac{P_T G}{4\pi\gamma_o P_N}}$. This observation coincides with the common knowledge, that is, the communication between a pair of nodes is bidirectional when there are no interference sources.

We note that the hearing range and the sending range characterize a node's ability to receive and to deliver messages that is influenced by environmental factors (e.g., ambience noise or jammer signals) but not by in-network factors (e.g., interference from network nodes).

3.4 The Effect of Jamming on the Communication Range

Applying the free-space model to a jammer, the jamming signals also attenuate with distance, and they reduce to the normal ambient noise level at a circle centered at the jammer. We call this circle the *Noise Level Boundary (NLB)* of the jammer. Since jamming signals are nothing but interference signals that contribute to the noise, a node located within the NLB circle will have bigger ambient noise floor than the one prior to jamming.

For simplicity, much work assumes that when a node is located inside the jammer's NLB circle it loses its communication ability completely, e.g., both its sending range and hearing range become zero. Such assumptions may be valid for nodes that perform CCA by comparing the channel energy with a fixed threshold, as all nodes within the NLB will consider the channel busy throughout the duration that the jammer is active. However, in a network where adaptive-CCA is used, the nodes inside the jamming's NLB circle will still maintain partial communication ability yet weaker than the nodes outside the NLB circle.

To facilitate analyzing the hearing range and the sending range of Node A, we consider a simple network consisting of three players: Jammer J interferes with the communication between Transmitter B and Receiver A, as depicted in Figure 1.

The Hearing Range under Jamming. Consider Node A as the receiver and Node B as the transmitter, the signal-to-noise ratio at A in the presence of Jammer J is

$$(SNR)_{B\to A} = \frac{P_{BA}}{P_N + P_{JA}},$$

where P_{BA} and P_{JA} are the received power of B's signals and the jamming signals at Node A, respectively.

Let's first examine the cases when Node A observes a jamming signal much larger than the normal ambient noise P_N. Assume that the jammer uses the same type of devices as the network nodes, e.g., both use omnidirectional antennas, then the antenna gain product between J and A, and the one between B and A are the same. The SNR can be simplified to,

$$(SNR)_{B\to A} \approx \frac{P_T d_{JA}^2}{P_J d_{AB}^2}. \tag{2}$$

To find the new hearing range under jamming attacks, we search for locations (x, y) that satisfy the equations: $(SNR)_{B\to A} = \gamma_o$. Substituting $d_{AB}^2 = x^2 + y^2$ and $d_{JA}^2 = x_j^2$ to Equation (2), Node A's hearing range when the jamming signal is dominant can be expressed as

$$x^2 + y^2 = \frac{x_j^2}{\beta}, \tag{3}$$

where $\beta = \frac{\gamma_o}{P_T/P_J}$. Thus, the hearing range of Node A is a circle centered at itself with a radius of $r_h = \frac{|x_j|}{\sqrt{\beta}}$. This formula coincides with the intuition: for the same x_j, a louder jamming signal affects network nodes more; given P_T and P_J, the closer a network node is located to the jammer, the smaller its hearing range becomes, as illustrated in Figure 2.

Now let's turn to the cases where the jamming signal no longer dominates the ambient noise, e.g., when nodes are located close to the edge of jammer's NLB, as illustrated in Figure 2 (b). The hearing range becomes:

$$x^2 + y^2 = \frac{x_j^2 P_T}{\gamma_o(x_j^2 \mu + P_J)}, \tag{4}$$

where $\mu = 4\pi P_N / G$. Thus, the hearing range of Node A is still a circle centered at itself with a radius of $\sqrt{\frac{P_T G}{4\pi\gamma_o P_N}} \times \sqrt{\frac{1}{1+P_J G/4\pi P_N x_j^2}}$. To avoid the complexities of deriving the antenna gain G, we approximate the node's hearing range with its normal hearing range, e.g., the hearing range without jammers.

In summary, the hearing range of Node A is a circle centered at A with a radius of

$$r_h = \min\left(\frac{|x_j|}{\sqrt{\beta}}, \sqrt{\frac{P_T G}{4\pi\gamma_o P_N}}\right),$$

as illustrated in Figure 2.

The Sending Range under Jamming. To derive the sending range of Node A, we consider A as the transmitter and B as the receiver. Applying the same assumption and approximation, consider the case where the jamming signals are much larger than the white noise, the sending range of Node A is

$$(x - \frac{x_j}{1-\beta})^2 + y^2 = \frac{\beta x_j^2}{(1-\beta)^2}, \tag{5}$$

a circle centered at $(\frac{x_j}{1-\beta}, 0)$ with a radius of $\frac{\sqrt{\beta}|x_j|}{|1-\beta|}$.

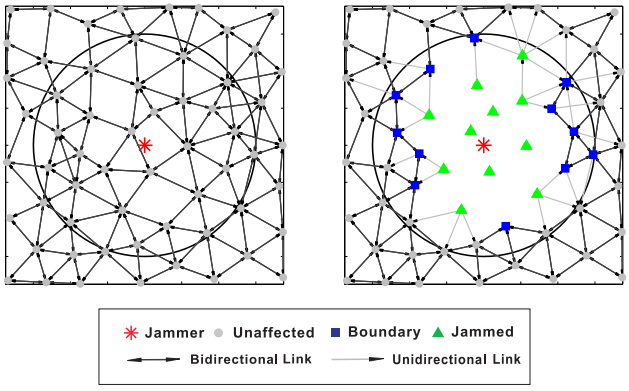
When the jamming signals have attenuated to a value comparable with the white noise, the sending range becomes a non-circular shape. For simplicity, one can approximate the sending range as the intersection of two circles: a normal sending range when no jammer is active and the circle denoted by Equation (5). We depicted sending ranges in Figure 2 when a node is located at various locations.

From Figure 2, we observed that for node A, because of jamming, the hearing range is no longer the same as the sending range, which can cause non-bidirectional links with its neighbors. In fact, interference can explain the commonly observed non-bidirectional communications in wireless networks.

3.5 The Effect of Jamming on Network Topology

In this section, we extend our analysis of jamming impact from the individual node level to the network level, and classify the network nodes based on the level of disturbance caused by the jammer.

Essentially, the communication range changes caused by jamming are reflected by the changes of neighbors at the network topology level. We note that both the hearing range and the sending range shrink due to jamming. We choose to utilize the change of the hearing range and its effect on lost neighbors under jamming, since it can be easily estimated by examining receiving ability at each node.



(a) Jammer is off (b) Jammer is on

Fig. 3. An example of the topology change of a wireless network due to jamming, where the black solid circle represents the jammer's NLB.

We define that node B is a neighbor of node A if A can *receive* messages from B, which is determined by the $(SNR)_{B\to A}$, i.e., the signal-to-noise ratio measured at node B while node A is transmitting. Let $Nbr\{n_i\}$ be the set of neighbors of node n_i before any jammer becomes active, when jammers are present in the network, the network nodes can be classified into three categories according to the impact of jamming: *unaffected node* N_U, *jammed node* N_J, and *boundary node* N_B. Thus, we have

- **Unaffected node.** $N_U = \{n_u | \forall n_i \in Nbr\{n_u\}, (SNR)_{i\to u} > \gamma_0\}$. A node is unaffected, if it can *receive* packets from all of its neighbors.
- **Jammed node.** $N_J = \{n_j | \forall n_i \in N_U, (SNR)_{i\to j} \leq \gamma_0\}$. Essentially, a node n_j is jammed if it cannot *receive* messages from any of the unaffected nodes. We note that two jammed nodes may still be able to communicate with each other.
- **Boundary node.** $N_B = \{n_b | (\exists n_i \in N_U, (SNR)_{i\to b} > \gamma_0)$ *and* $(\forall n_i \in Nbr\{n_b\} \cap N_J, SNR_{i\to b} \leq \gamma_0)\}$. A boundary node can receive packets from part of its neighbors but not from all its neighbors.

Figure 3 illustrates an example of network topology changes caused by a jammer. Prior to jamming, neighboring nodes were connected through bidirectional links. Once the jammer became active, nodes lost their bidirectional links either partially or completely. In the example depicted in Figure 3, the nodes marked as triangles lost all their inbound links (receiving links) from their neighbors and became jammed nodes. Interestingly, some jammed nodes can still send messages to their neighbors, and they may participate in the jamming localization by delivering information to unaffected nodes as described in Section 4. The nodes depicted in rectangles are boundary nodes. They lost part of its neighbors but still maintained partial receiving links, e.g., at least connected to one unaffected

nodes either directly or indirectly. Finally, the rest of nodes depicted in circles are unaffected nodes, and they can still receive from all their neighbors.

4 LSQ-Based Jammer Localization

4.1 Algorithm Description

In the previous sections, we have shown that the hearing range of a node may shrink and its neighbor list may change when a jammer becomes active. The levels of changes are determined by the distance to the jammer and the strength of the jamming signals. The basic idea of our LSQ-based algorithm is to localize the jammer according to the changes of a node's hearing range. To simplify the algorithm description, we start by assuming the node hearing range is known, and we delay the discussion of its estimation to Section 4.2.1.

Consider the example illustrated in Figure 1, if B happens to be located at the edge of A's hearing range, then we have $(SNR)_{B\to A} \approx \gamma_o$ and $d_{AB} = r_{h_A}$. Therefore, we can convert Equation (2) into a general form,

$$(x_A - x_J)^2 + (y_A - y_J)^2 = \beta r_{h_A}^2, \tag{6}$$

where r_{h_A} is the new hearing range of Node A, $\beta = \frac{\gamma_o}{P_T/P_J}$, and (x_A,y_A) and (x_J,y_J) are the coordinates of A and Jammer J, respectively. *In the above equation, the unknown variables includes x_J, y_J, and β. To obtain those three variables, one equation is not enough.*

Suppose that the hearing ranges of m nodes have shrunk to r_{h_i}, $i = \{1,\ldots,m\}$ due to jamming. Then, we have m equations:

$$\begin{aligned}(x_1 - x_J)^2 + (y_1 - y_J)^2 &= \beta r_{h_1}^2\\ (x_2 - x_J)^2 + (y_2 - y_J)^2 &= \beta r_{h_2}^2\\ &\vdots\\ (x_m - x_J)^2 + (y_m - y_J)^2 &= \beta r_{h_m}^2\end{aligned} \tag{7}$$

Assume that we can obtain r_{h_i} for each of m nodes, then we can localize the jammer by solving the above equations. To avoid solving a complicated nonlinear equations, we first linearize the problem by subtracting the m^{th} equation from both sides of the first $m-1$ equations and obtain linear equations:

$$\begin{aligned}&(x_1^2 - x_m^2) - 2(x_1 - x_m)x_J + (y_1^2 - y_m^2) - 2(y_1 - y_m)y_J\\ &\quad = \beta(r_{h_1}^2 - r_{h_m}^2)\\ &(x_2^2 - x_m^2) - 2(x_2 - x_m)x_J + (y_2^2 - y_m^2) - 2(y_2 - y_m)y_J\\ &\quad = \beta(r_{h_2}^2 - r_{h_m}^2)\\ &\quad\vdots\\ &(x_{m-1}^2 - x_m^2) - 2(x_{m-1} - x_m)x_J + (y_{m-1}^2 - y_m^2) - 2(y_{m-1} - y_m)y_J\\ &\quad = \beta(r_{h_{m-1}}^2 - r_{h_m}^2)\end{aligned} \tag{8}$$

Then it can be written in the form of $\mathbf{Az} = \mathbf{b}$ with

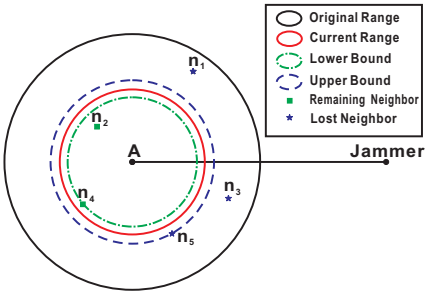


Fig. 4. An illustration of estimating the hearing range of Node A leveraging the change of its neighbor list.

$$\mathbf{A} = \begin{pmatrix} x_1 - x_m & y_1 - y_m & \frac{1}{2}(r_{h_1}^2 - r_{h_m}^2)\\ \vdots & \vdots & \vdots\\ x_{m-1} - x_m & y_{m-1} - y_m & \frac{1}{2}(r_{h_{m-1}}^2 - r_{h_m}^2)\end{pmatrix}$$

and

$$\mathbf{b} = \begin{pmatrix} (x_1^2 - x_m^2) + (y_1^2 - y_m^2)\\ \vdots\\ (x_{m-1}^2 - x_m^2) + (y_{m-1}^2 - y_m^2)\end{pmatrix}.$$

We can estimate the location of the jammer and β by using the least squares (LSQ) method,

$$\mathbf{z} = [x_J, y_J, \beta]^T = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}. \tag{9}$$

4.2 Algorithm Challenges

To localize a jammer using LSQ-based method, two questions have to be answered: (1) how to estimate the radius of a node's hearing range (aka. the hearing radius), and (2) what is the criteria of selecting nodes as candidates to form equation groups?

4.2.1 *Estimating the Hearing Radius.*

To estimate the hearing radius of Node A after a jammer becomes active, Node A should examine its neighbor list and identify two specially-located nodes: its furthest neighbor that Node A *can* still hear from and its closest node that Node A *cannot* hear. Since the distances to those two special nodes provide the lower bound and the upper bound of A's hearing radius, A's hearing radius can be estimated as the mean value of those bounds.

Consider the example illustrated in Figure 4, before the jammer started to disturb the network communication, Node A had a neighbor list of $\{n_1, n_2, n_3, n_4, n_5\}$. Once the jammer became active, A's neighbors reduced to $\{n_2, n_4\}$ and we call this set the *Remaining Neighbor Set*. At the same time, A can no longer hear from $\{n_1, n_3, n_5\}$, the *Lost Neighbor Set*. The estimated upper bound of A's hearing radius r_u equals the distance to n_5, the nearest node in the lost neighbor set; the estimated lower bound r_l equals the distance to n_4, the furthest node in the remaining neighbor set. As a result, the true hearing radius r_{h_A} is sandwiched between $[r_l, r_u]$ and can be estimated as $\hat{r}_{h_A} = (r_u + r_l)/2$.

The estimation error of the hearing radius e_h depends on $(r_u - r_l)$ and can be any value in $[0, (r_u -$

$r_l)/2]$. When the distances between any two nodes are uniformly distributed, the estimation error e_h follows uniform distribution with the expected value as $\frac{r_u - r_l}{4}$.

4.2.2 Selecting m Nodes.

The nodes that can contribute to the jamming localization have to satisfy the following requirements: (1) they have a reduced hearing range and their neighbor list has changed; (2) the new hearing range under jamming attacks can be estimated; and (3) they are able to transmit their new hearing radius out of the jammed area.

Although an unaffected node may have a slightly reduced hearing range, its neighbor list remains unchanged. Therefore, its hearing radius cannot be estimated and neither can it contribute an equation to localize the jammer. Likewise, although a jammed node's hearing range is decreased severely, its remaining neighbor set may be empty, preventing it from estimating the up-to-date hearing radius accurately. Even in cases when they may estimate their hearing ranges with the help of "Jammed Cluster", they may not be able to transmit their estimations out of the jammed area due to communication isolation. In short, most of the jammed nodes are not suitable for jamming localization. Only those that can estimate their reduced hearing ranges and are able to send out messages to unaffected nodes can be used.

Finally, with regard to boundary nodes, the hearing range of a boundary node is reduced. Leveraging their reduced neighbor lists, their hearing radii can be estimated. More importantly, they can still communicate with unaffected nodes within finite steps. Therefore, all boundary nodes shall be used to participate the jamming localization.

In summary, we use the following nodes to form the equation group for jamming localization: all the boundary nodes and the jammed nodes that can estimate their reduced hearing ranges and are able to send out messages to unaffected nodes.

5 Localizing a Jammer in Reality

The previous analysis that exploits the free-space model provides insights in understanding the jamming effect and underlying theoretical basis for our localization algorithms. However, real wireless communication operates in complex propagation environments full of absorption, reflection, scattering, and diffraction, and it cannot be accurately modeled by the free-space model. Because of those characteristics associated with realistic radio propagation, several challenges arise when implementing our localization algorithm in practice. Thus, in this section, we first performed experimental measurements in a real environment to understand radio propagation in practice and then, selected a model that can represent realistic radio propagation. Finally, we modified the LSQ-based algorithm to address challenges induced by the complex radio propagation.

In practice, we envision that each node periodically samples the PDRs from all its neighbors and deliver them to a designed node that is unaffected. Once the jammer is detected, the designated node runs the LSQ-based algorithm. In other words, the data is acquired distributively but the localization is performed at one node.

5.1 Empirical Study

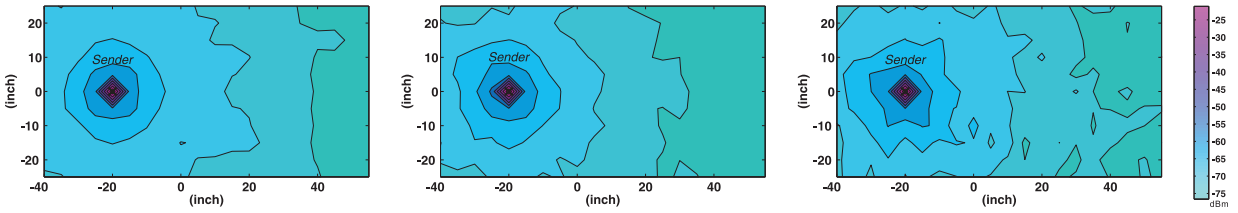
We conducted experiments in a $50ft \times 50ft$ basement with 9-foot ceiling and several columns supporting the ceilings. We chose to use micaZ sensor nodes [32], which have a 2.4-2.48 GHz Chipcon CC2420 Radio, and set the operation frequency to 2.4 GHz. Due to space limitation, we tuned the transmission power to -15 dBm or -25 dBm to reduce the communication range of sensors. We have performed two sets of experiments, one for measuring the Received Signal Strength (RSS) and the other for Packet Delivery Ratio (PDR).

RSS Contours. In the first set of experiments, we placed the transmitter at $(-20, 0)$ and measured RSS values in the rectangular grid located at $[-40, 50] \times [-25, 25]$ inches with a grid size of 5 inches. To eliminate the potential errors caused by variances among multiple radio chips, we only used one micaZ sensor as the receiver and move it to each grid vertex. The resulted RSS contours, which is shown in Figure 5 (a), appear to be approximate concentric circles yet with irregular edges.

PDR Contours. Similarly, in the second set of experiments, we studied the PDR [1] from a sender to a receiver at each grid vertex in the presence of a jammer. Three micaZ sensors were used: We placed the sender at $(20, 0)$, the jammer at $(-20, 0)$, and moved the receiver to one of the grid vertices in the same rectangular grid. We chose to implement a constant jammer [14] on a micaZ sensor to avoid additional RSS empirical study of the jammer's radio, and we set the jammer's transmission power to -15dBm, larger than the sender's transmission power, -25dBm.

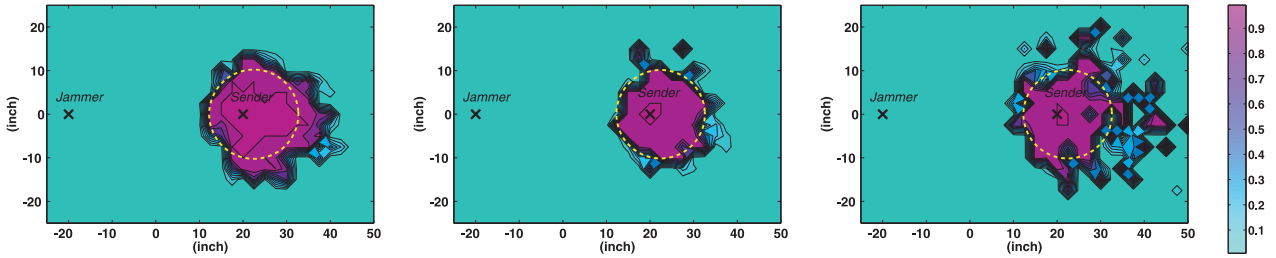
We depicted the measured PDR contours over the rectangular grid of $[-25, 50] \times [-25, 25]$ inches in Figure 6 (a). The area where $PDR > 0$ actually maps to the sending range, and exhibits irregularity that coincides with common observations in wireless communication. As a comparison, we depicted the theoretical sending range of the sender using the free space model as the dashed circle in Figure 6 (a). Although the theoretical sending range overlaps with the real sending range, the real one is highly irregular and one can predict that the performance of localization algorithm will suffer in a real system.

1. PDR was measured at the receiver by calculating the ratio of the number of correctly decoded packets with respect to the expected number of transmitted packets.



(a) Real measurements (b) $\sigma = 1.0$ (c) $\sigma = 2.0$

Fig. 5. RSS contours with a transmitter located at $(-20, 0)$. (a) was obtained through experiments. (b) and (c) are simulated contours using the signal propagation model with shadow fading.



(a) Real measurements (b) $\sigma = 1.0$ (c) $\sigma = 2.0$

Fig. 6. PDR contours with a stationary sender at $(20, 0)$ and a jammer at $(-20, 0)$, where the dash circle is the sending contour derived using free space model.(a) was obtained through experiments. (b) and (c) are simulated contours using the signal propagation model with shadow fading.

thus, an extensive performance study of the LSQ-based algorithm is in need.

5.2 Log-Normal Shadowing Model

To prepare for the extensive performance study, we targeted at discovering a realistic propagation model that can help to build our simulation tools. To balance the trade-off of modeling, we chose a simple model that captures the essential of signal propagation without using computer-aided modeling tools: log-normal shadowing model. The log-normal shadowing model captures both path loss versus distance along with the random attenuation due to blockage from objects in the signal path [33], and it has the following form,

$$PL(d) = PL(d_0) - 10 \cdot \eta \cdot \log(\frac{d}{d_0}) + X_\sigma, \qquad (10)$$

where $PL(d)$ is the path loss at distance d, $PL(d_0)$ is the known path loss at a reference distance d_0, η is the Path Loss Exponent(PLE), and X_σ is a Gaussian zero-mean random variables with standard deviation σ. When $\eta = 2$ and $\sigma = 0$, Equation (10) regresses to,

$$PL(d) = PL(d_0) - 20 \cdot \log(\frac{d}{d_0}), \qquad (11)$$

which is the log normal form of the standard free-space propagation model listed as equation 1.

There are two unknown parameters in the log-normal shadowing model: PLE η and the standard deviation σ. We determined those parameters using our experimental measurements in two steps: We first focus on the calculation of PLE η, since X_σ is a zero-mean variable and statistically it should have little impact to the average of path loss. Second, we obtain the standard deviation of X_σ based on the fitted η.

To obtain the best estimation of PLE η, we search for the value of η that minimizes the mean-square error (MMSE) [33] between the modeled RSS value and the empirical measurements,

$$MSE(\eta) = \sum_{i=1}^{n} [PL_{measured}(d_i) - PL_{model}(d_i)]^2, \qquad (12)$$

where n is the number of measurements. our empirical study generated a fitted η of 2.11. Further, we estimated σ using several methods and selected $\sigma = 1.0$, as it produces RSS contours (Figure 5 (b)) and PDR contours (Figure 6 (b)) exhibiting closest irregularity to the original empirical measurements.

As a reference, we also generated RSS contours and PDR contours when σ is larger than the empirical measurements, e.g., $\sigma = 2.0$, in Figure 5 (c) and Figure 6 (c), respectively. Although those contours exhibit a higher degree of irregularity and do not emulate our empirical measurements well, evaluating our algorithm in such environments can provide guidance on predicting the algorithm's performance in a highly complicated environment.

5.3 Dealing with Signal Irregularity

The irregularity of the hearing range caused by random attenuation and multi-path propagation in a complex radio environment can create much larger estimation errors of hearing radii than the one obtained assuming the free-space model. The larger estimation errors, in turn, can impair the localization accuracy, especially in the cases when not enough equations are available to 'cancel out' those large estimation errors. Thus, the estimated location of the jammer could be

Algorithm: Adaptive_LSQ_localization
S: the set of boundary nodes
$\hat{J}_L = LSQ\_localization()$
$\hat{J}_C = Centroid\_localization()$
$d_m = \max_{Z_i, Z_j \in S} ||Z_i - Z_j||_2$
if $||\hat{J}_L - \hat{J}_C||_2 < a \times d_m$ **then**
  return $\hat{J}_L$
else
  return $\hat{J}_C$
end

Algorithm 1: The adaptive LSQ-based localization algorithm which incorporates the Centroid method with the original LSQ method. We empirically selected $a = 0.4$

very far away from its true location, even out of the jammed region.

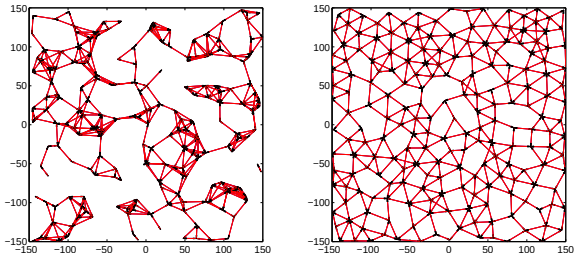
To assure that the estimated location of the jammer is inside the jammed region, we utilize centroid-based localization (CL) algorithm, which estimates the position of the jammer by averaging over coordinates of all boundary nodes. Formally, consider that there are m boundary nodes $\{(x_i, y_i)\}_{i=1...m}$, the position of the jammer can be estimated by:

$$\hat{J} = (\hat{x}_J, \hat{y}_J) = (\frac{\sum_{k=1}^{m} x_k}{m}, \frac{\sum_{k=1}^{m} y_k}{m}). \quad (13)$$

Although CL is extremely sensitive to the distribution of boundary nodes and does not provide accurate estimation consistently [10], it always produce a position surrounded by all boundary nodes and can serve as correction when the LSQ-based algorithm fails to perform.

Thus, we proposed an adaptive LSQ-based localization algorithm that combines the CL method with the LSQ algorithm, as shown in Algorithm 1. We note the name difference between the *adaptive* LSQ-based algorithm and the *pure* LSQ-based algorithm, as the latter does not integrate the CL method. In the `Adaptive_LSQ_localization` algorithm, the jammer's position are first estimated using both CL (`Centroid_localization`) and LSQ (`LSQ_localization`) independently. We note that `LSQ_localization` returns *infinity*, when the number of equations is less than the unknown variables. We estimated the span of the jammed region as the maximum distance between all boundary nodes. When the difference of both estimations are larger than a times the span of the jammed region, indicating abnormally large estimation errors, the estimation using CL is returned. Otherwise, the estimation using LSQ algorithm is preferred. We selected a to be 0.4, as it produces the best performance empirically.

We note that `Adaptive_LSQ_localization` algorithm does not work well when the jammer is indeed located outside the network and all affected nodes. However, such cases do not impose much concern in practice, as a jammer is less likely to place itself outside of the network, afraid of not fulfilling its objective to disrupt the communication ability of as many nodes as possible. Even if such cases do happen, such situations can be detected by examining the positions of affected nodes with regard to the network edges and then, one can choose to localize the jammer using LSQ-based algorithm instead of CL method.



(a) Simple deployment (b) Smart deployment

Fig. 7. Two deployments on a network with $N = 200$.

6 Experiment Validation

In this section, we evaluate the performance of the LSQ-based localization algorithm using both the free-space model and the log-normal shadowing model.

6.1 Simulation Methodology and Performance Metrics

We have built our simulator that utilizes both the free space model and the log-normal shadowing model in MATLAB. For both models we took advantage of our empirical study, for instance, we set $\gamma_o = 2.20$, a parameter measured using MicaZ sensors in our prior work [10]. We chose to evaluate the performance of our localization algorithm using two representative network deployments: simple deployment and smart deployment. The nodes' coordinates in the simple deployment follow a uniform distribution, corresponding to a random deployment, e.g., sensors are randomly disseminated to the battlefield or the volcano vent. Nodes may cluster together at some spots while may not cover other areas, as shown in Figure 7(a). The smart deployment involves carefully placing nodes so that they cover the entire deployment region well and the minimum distance between any pair of nodes is bounded by a threshold, as shown in Figure 7(b). This type of deployment can be achieved using location adjustment strategies [34] after deployment.

For both types of deployment, we simulated a wireless network deployed in a 300m-by-300m region, and evaluated the algorithms in various network conditions, including different network node densities, jammer's NLB radius, and etc. The normal communication range of each node was set to 30m. Unless specified, we placed the jammer at the center of the network $(0, 0)$ and set the jammer's NLB to 60m. Later, we investigated the effect of the jammer's position on the algorithm performance by placing the jammer at the edge of the network.

To evaluate the accuracy of localizing the jammer, we define the localization error as the Euclidean distance between the estimated jammer's location and the true location. To capture the statistical characteristics, we studied the average errors under multiple experimental rounds (1000 times) and we presented both the means and the Cumulative Distribution Functions(CDF) of the localization error.

6.2 Performance under the Free-Space Model

To examine the effectiveness of the pure LSQ-based algorithm, we firstly compared its performance with the virtual force iterative localization algorithm (VFIL) from our prior work [10] using the free-space model. To make a fair comparison, we adopted a version of the VFIL algorithm that also does not rely on the information of the jammer's NLB, just as the LSQ-based algorithm, and we executed both pure LSQ-based and VFIL algorithms on the same set of network topologies.

Impact of the Node Density. We first investigated the impact of the node density on the performance of both the LSQ and VFIL methods. To adjust the network node density, we varied the total number of nodes deployed in the 300-by-300 meter region in the simulation. In particular, we chose to run the experiments on the networks of N nodes, and $N = \{200, 300, 400\}$.

We depicted the mean errors for both LSQ and VFIL in Figure 8 (a). Firstly, we observed that LSQ outperformed VFIL consistently in all node densities and node deployment setups. The LSQ's mean errors fall between 1m and 3m, much smaller than the errors of VFIL, which ranges from 9m to 25m.

The performance difference can be explained as the following: The VFIL algorithm iteratively searches for the estimated jammer's location until it finds one that under the assumption of a jammer resided there the derived nodes' categories match with their true categories, e.g., unaffected, jammed, or boundary. Thus, such an estimation is only good-enough but not optimal. In comparison, the LSQ algorithm calculates the location that minimizes all hearing range estimation errors at one step.

Secondly, with the increasing network node densities, the performance of both algorithms improves. Since both algorithms rely on the number of affected nodes to improve the estimation accuracy, the higher the densities, the smaller the mean estimation errors.

Finally, both algorithms performed better in a smart deployment than a simple deployment. In a simple deployment, nodes were not evenly distributed. Thus, when a jammer was placed within an area sparsely covered, without enough affected nodes to provide constraints, the accuracy of the jammer's location estimation suffered. In contrast, the nodes in a smart deployment covered the entire network region evenly, and they supplied reasonable amount of information for the algorithms to localize the jammer. Therefore, both algorithms achieved better localization accuracy in a smart deployment.

We also provided a view of Cumulative Distribution Function (CDF) curves for both algorithms in Figure 9. To make the plot readable, we showed the results of 200 and 400 node cases, omitting the almost overlapped 300-node result. Again, we observed that the LSQ outperformed VFIL constantly. Particularly, under the smart deployment, 90% of the time LSQ can estimate the jammer's location with an error less than 4.2m, while VFIL can only achieve 18.8m 90% of the time, resulting in an improvement of 80%. While under a simple deployment, LSQ improved the localization accuracy by 95%, as its estimation errors were less than 5.9m 90% of the time versus 47.5m for VFIL.

Impact of the Jammer's NLB Range. To study the effects of various jammer's NLB ranges to the localization performance, we examined networks with 300 nodes and set the jammer's NLB radius to 40m, 60m, 80m and 100m, respectively. The results were plotted in Figure 8(b) showing that the LSQ method still largely outperformed the VFIL method by over 60%. Additionally, we noticed that the localization errors of VFIL decreased linearly when the jammer's NLB range increased. However, the errors of LSQ only lessened when the NLB range increased from 40m to 60m and became steady afterwards. This is because the number of affected nodes in the 40m NLB range scenario was not enough for LSQ to localize the jammer accurately, i.e., the number of equations that can be created for the LSQ algorithm was not enough. When the NLB range became large enough (e.g., larger than 60m), the LSQ algorithm had enough equations to produce estimation with similar average errors.

Impact of the Jammer's Position. We investigated the impact of the jammer's position by placing it at the center $(0, 0)$ and at the corner $(130, -130)$, respectively. In both cases, we set the jammer' NLB range to 60m and used 300-node networks.

Figure 8(c) shows that the performance of both LSQ and VFIL degraded when the jammer is at the corner of the network. Because the affected nodes were located on one side of the jammer, causing the estimated location biased towards one side. However, in both simple and smart deployments, LSQ still maintained a localization error less than 10m, which is 1/3 of a node's transmission range. VFIL produced errors of more than 40m when the jammer was at the corner, making the results of jammer localization unreliable. Thus, LSQ is less sensitive to the location of the jammer. The observations of the CDF results in Figure 11 provide a consistent view with the mean errors.

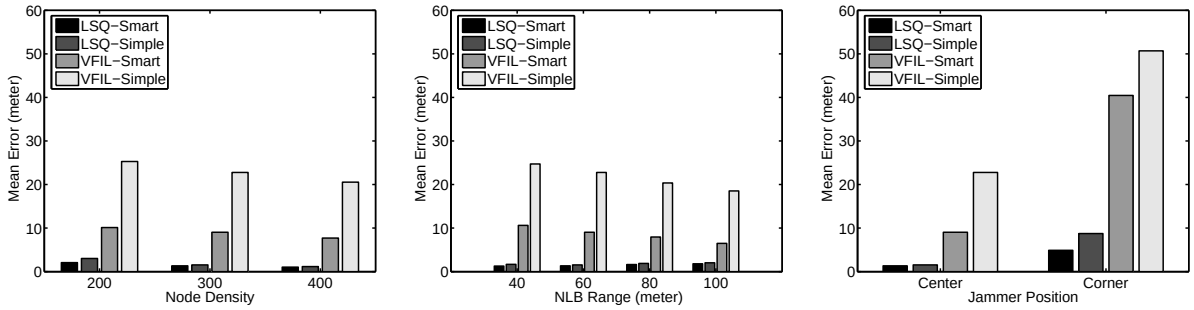


Fig. 8. The impact of various factors on the performance of LSQ and VFIL algorithms under the free-space model: (a) node density; (b) jammer's NLB range; (c) jammer's position in the network.

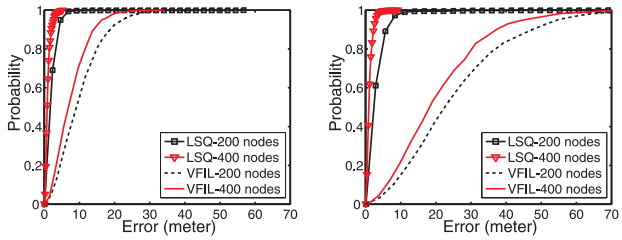


Fig. 9. Cumulative Distribution Function (CDF) of the localization errors with regard to different node densities under the free-space model.

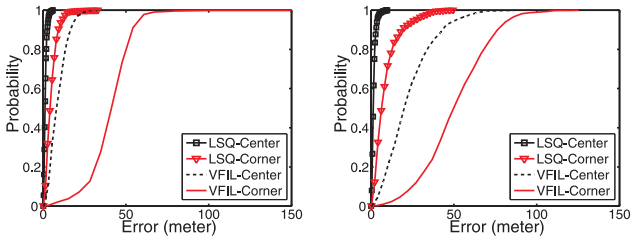


Fig. 11. Cumulative Distribution Function (CDF) of the localization errors when the jammer is placed in the center or at the corner, under the free-space model.

6.3 Performance under the Shadowing Model

We evaluated the performance of the adaptive LSQ-based localization algorithms by emulating a real environment. Particularly, we adopted the log-normal model and tuned the parameters obtained from our empirical study, e.g., $\eta = 2.11$ and $\sigma = 1.0$. As such, we utilized the advantages of simulation methodology, e.g., flexibility, low cost, and no physical space limitation yet captured major characteristics of real-world implementation.

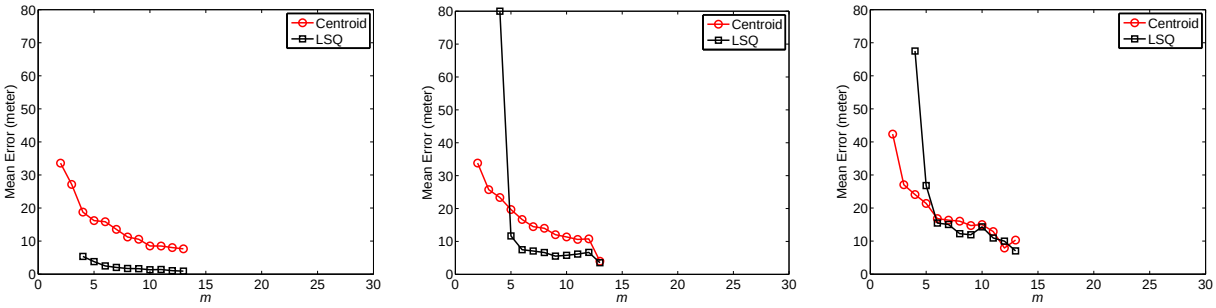
Similar to the experiments assuming free-space model, we studied the localization accuracy of the adaptive LSQ-based algorithm under shadowing model in various network configurations, including node densities, jammer's NLB ranges, and jammer's positions in the network. In addition, we studied the impact of the standard deviation σ to the localization performance. We didn't, however, present the performance evaluation results for VFIL algorithm because VFIL does not always converge under shadowing model. We did compare the performance of centroid-based algorithm and pure LSQ-based algorithm to analyze their roles in the adaptive LSQ-based algorithm.

Centroid-Based vs. Pure LSQ-Based Algorithm. We first conduct a performance comparison between the pure LSQ-based algorithm and centroid localization (CL). In particular, we examined both algorithms in a 200-node network using both deployments, and set the jammer's NLB range to 60m. We set the PLE, η, to 2.11 while cycled σ through $\{0, 1.0, 2.0\}$. The mean errors for both algorithms with regards of the number of usable affected nodes, m, are plotted in Figure 10.

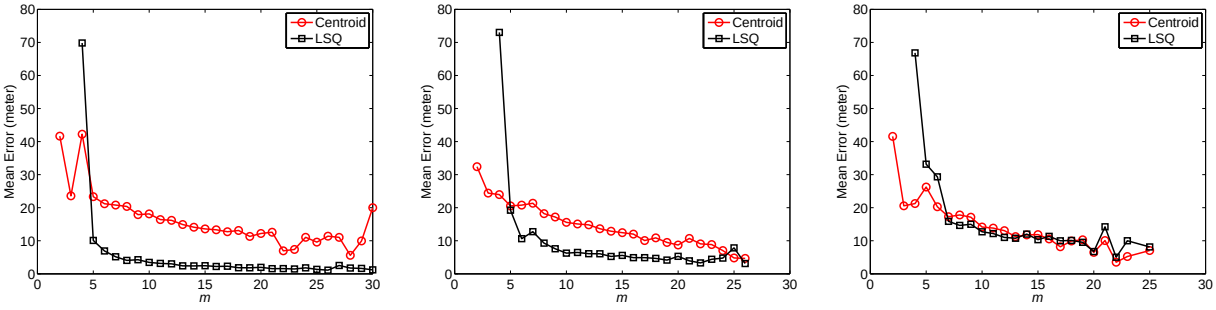
From the results depicted in Figure 10 (a)-(f), we observed that the range of m in the smart deployment is always smaller than 15, while m in the simple deployment can reach up to 30. Regardless whether the smart deployment or the simple deployment is adopted, the performance of both the pure LSQ-based and the CL algorithms improves with the increasing of m. However, the LSQ-based algorithm almost always outperformed the CL algorithm when $m \geq 6$ and underperformed when $m < 6$. This observation suggests that the adaptive LSQ-based algorithm can provide better location estimation of the jammer by combining the better estimation of pure LSQ-based and the CL algorithm.

Additionally, we note that the centroid-based algorithm is not sensitive to the value of σ, while the performance of the pure LSQ-based algorithm decreases, yet still better than or close to the centroid-based algorithms, when σ increases from 0 to 2.0. For instance, in the smart deployment, the average estimation error of pure LSQ-based method is around 4 when $\sigma = 0$, the average error becomes 8 when σ increase to 1.0, and the average error is more than 10 when $\sigma = 2.0$. This suggests that when σ is extremely large, CL might perform better than LSQ-based algorithm.

We have also conducted simulations to examine the performance of both algorithms in 300-node networks

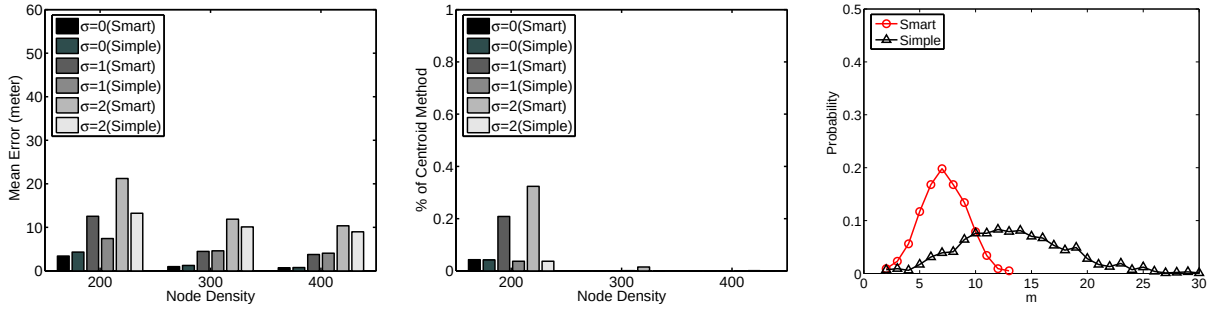


(a) σ=0, Smart (b) σ=1.0, Smart (c) σ=2.0, Smart



(d) σ=0, Simple (e) σ=1.0, Simple (f) σ=2.0, Simple

Fig. 10. Performance comparison between pure LSQ-based and CL algorithm with $N = 200$ and jammer's NLB range as 60m. We plotted the mean errors based on the number of usable affected nodes, m.



(a) Mean error (b) The percentage of CL usage (c) Probability($N = 200$)

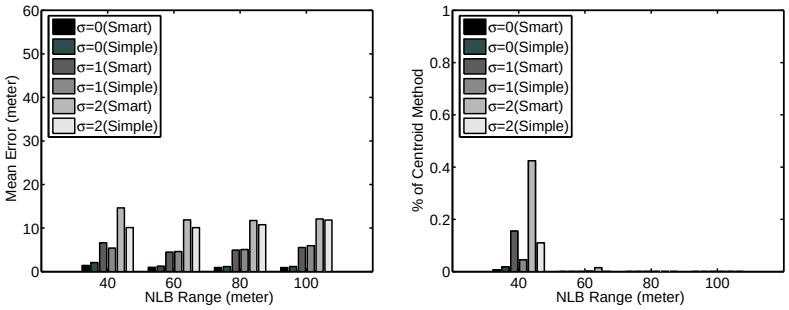
Fig. 12. Adaptive-LSQ: The impact of the node density with a jammer's NLB range of 60m.

and 400-node networks, but we did not present the results as they exhibit similar trends.

Impact of the Node Density. We next investigated the impact of the node density on the adaptive LSQ-based localization algorithm by setting the N to $\{200, 300, 400\}$ while fixing the jammer's NLB range to 60m. We plotted the mean estimation errors with $\sigma = \{0, 1.0, 2, 0\}$ in Figure 12 (a) and the percentage of cases that CL is used in Figure 12 (b) . Similar to the results obtained using the free space model, as N increases, the performance of the adaptive LSQ-based algorithm improves for all σ. Particularly, the mean error of the adaptive LSQ-based algorithm when $\sigma = 0$ is similar to the one obtained using the free-space model, which confirms with the fact that the shadowing model with $\sigma = 0$ regresses to the free space model. We note the slight difference in mean errors of those two cases is caused by the usage of CL algorithm in Adaptive-LSQ.

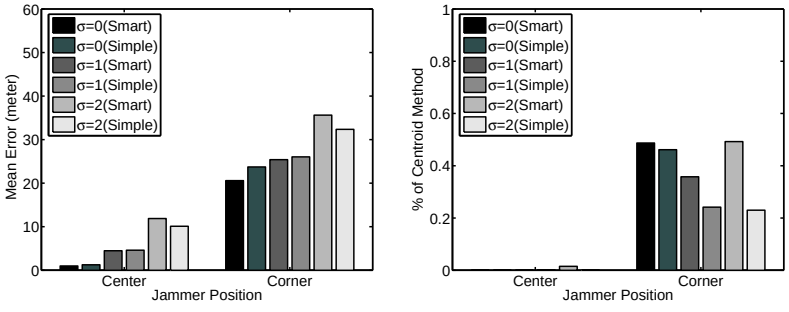
Additionally, for the same N and jammer's NLB range, we observed that as σ increases, the performance of adaptive LSQ-based algorithm decreases. This is caused by the increasing degrees of the hearing range's irregularity. Interestingly, unlike the free space model, when σ is larger than 0, the smart deployment cases no longer perform better than the simple deployment cases. This difference is particularly pronounced in the case of $N = 200$, and the probability distribution of the numbers of usable affected nodes (as shown in Figure 12 (c)) reveals that the larger mean error of the smart deployment is essentially caused by its smaller m, i.e., smaller amount of available equations in the LSQ-problem and cannot average out the errors caused when estimating the hearing range in a complicated radio environment.

Impact of the Jammer's NLB Range. Similar to the experiment in the free space model, we measured the mean localization errors of 300-node networks



(a) Mean errors (b) The percentage of CL usage

Fig. 13. Adaptive-LSQ: the impact of the jammer's NLB range with $N = 300$.



(a) Mean errors (b) The percentage of CL usage

Fig. 14. Adaptive-LSQ: the impact of the jammer's position with $N = 300$ and a jammer's NLB range as 60m.

when changing the jammer's NLB radius to 40m, 60m, 80m and 100m, respectively, and plotted the results in Figure 13. Again, we observed that as σ increases the performance of the adaptive LSQ-based algorithms decreases. However, as the jammer's NLB range increases, the accuracy of the estimated jammer's location does not change much.

Impact of the Jammer's Position. We placed jammer at the corner $(130, -130)$ and at the center $(0, 0)$, respectively, and depicted the mean errors of the adaptive LSQ-based algorithm when setting the jammer's NLB range to 60m and N to 300 in Figure 14. We observed similar trend of the localization accuracy to the free space model. In particular, when the jammer is located at the center, the adaptive LSQ-based algorithm can localize the jammer with an accuracy better than 12m on average. However, when the jammer is located at the corner, the mean estimation errors become around 30m. The increase in estimation errors is because of the usage of CL. The performance of CL algorithm depends on the distribution of the affected nodes and a jammer located at the network corner produces a biased distribution of the affected nodes. The percentage of the CL usage is depicted in Figure 14(b).

7 Conclusion

In this work, we addressed the problem of localizing a jammer in wireless networks and proposed a least squares (LSQ) based localization algorithm that estimates the jammer's location by utilizing the changes of neighbor nodes caused by jamming. We have analyzed the impact of a jammer on both a node's hearing range and sending range. And we have shown that the levels of a node's hearing range changes is determined quantitatively by the distance between the node to the jammer. The change of a node's hearing range can be estimated by exploiting the changes of its neighbors. Therefore, we can localize the jammer by examining the neighbor list changes of multiple nodes and constructing a least-squares problem. Our approach does not depend on measuring signal strength inside the jammed area, nor does it require to deliver information out of the jammed area. Thus, it works well in the jamming scenarios where network communication is disturbed.

We analyzed and evaluated our LSQ-based jammer localization algorithms in both the free space and the shadowing model that represents the real radio environment. Under the free space model, we compared our LSQ-based jammer localization algorithm with our prior work, i.e., the virtual force iterative localization algorithm (VFIL) that involves searching for the location of the jammer iteratively. Since the LSQ-based approach finishes the location estimation in one step, it significantly reduces the computation cost while achieving better performance. We have shown that our LSQ-based approach outperforms the VFIL regardless of node distributions, network node densities, jammer's transmission ranges, and jammer's positions by simulation.

To address the irregularity that exists in real systems, we studied our LSQ-based algorithm in the

shadowing model. Particularly, we extended the pure LSQ-based localization scheme to an adaptive version by combining it with the centroid method. To evaluate the adaptive LSQ-based algorithm, we built our simulation environment based on the shadowing model and used the parameters obtained empirically from our experiments. Our extensive simulation results have confirmed that the adaptive LSQ-based algorithm is effective in localizing jammers in all experiment configurations, even in a highly complicated radio environment.

REFERENCES

[1] J. G. Proakis, *Digital Communications*, 4th ed. McGraw-Hill, 2000.

[2] G. Noubir and G. Lin, "Low-power DoS attacks in data wireless lans and countermeasures," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, no. 3, pp. 29–30, 2003.

[3] W. Xu, W. Trappe, and Y. Zhang, "Channel surfing: defending wireless sensor networks from interference," in *IPSN '07: Proceedings of the 6th international conference on Information processing in sensor networks*, 2007, pp. 499–508.

[4] K. Ma, Y. Zhang, and W. Trappe, "Mobile network management and robust spatial retreats via network dynamics," in *Proceedings of the 1st International Workshop on Resource Provisioning and Management in Sensor Networks (RPMSN05)*, 2005.

[5] M. Cagalj, S. Capkun, and J. Hubaux, "Wormhole-Based Anti-Jamming Techniques in Sensor Networks," *IEEE Transactions on Mobile Computing*, pp. 100 – 114, January 2007.

[6] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems*, vol. 10, no. 1, pp. 91–102, Jan. 1992.

[7] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, March 2000, pp. 775–784.

[8] Y. Chen, J. Francisco, W. Trappe, and R. P. Martin, "A practical approach to landmark deployment for indoor localization," in *Proceedings of the Third Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, September 2006.

[9] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*, Aug 2000, pp. 32–43.

[10] H. Liu, W. Xu, Y. Chen, and Z. Liu, "Localizing jammers in wireless networks," in *Proceedings of IEEE PerCom International Workshop on Pervasive Wireless Networking (IEEE PWN)*, 2009.

[11] K. Pelechrinis, I. Koutsopoulos, I. Broustis, and S. V. Krishnamurthy, "Lightweight jammer localization in wireless networks: System design and implementation," in *Proceedings of the IEEE GLOBECOM*, December 2009.

[12] C. Schleher, *Electronic Warfare in the Information Age*. MArtech House, 1999.

[13] A. Wood, J. Stankovic, and S. Son, "JAM: A jammed-area mapping service for sensor networks," in *24th IEEE Real-Time Systems Symposium*, 2003, pp. 286 – 297.

[14] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, 2005, pp. 46–57.

[15] M. Çakiroğlu and A. T. Özcerit, "Jamming detection mechanisms for wireless sensor networks," in *InfoScale '08: Proceedings of the 3rd international conference on Scalable information systems*. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 1–8.

[16] R. Mraleedharan and L. A. Osadciw, "Jamming attack detection and countermeasures in wireless sensor network using ant system," in *Proceedings of the SPIE in Wireless Sensing and Processing*, vol. 6248, 2006, p. 62480G.

[17] J. T. Chiang and Y.-C. Hu, "Cross-layer jamming detection and mitigation in wireless broadcast networks," in *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2007, pp. 346–349.

[18] V. Navda, A. Bohra, S. Ganguly, R. Izmailov, and D. Rubenstein, "Using channel hopping to increase 802.11 resilience to jamming attacks," in *IEEE Infocom Minisymposium*, May 2007, pp. 2526 – 2530.

[19] S. Khattab, D. Mosse, and R. Melhem, "Modeling of the channel-hopping anti-jamming defense in multi-radio wireless networks," in *Mobiquitous '08: Proceedings of the 5th Annual International Conference on Mobile and Ubiquitous Systems*. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 1–10.

[20] W. Xu, W. Trappe, and Y. Zhang, "Anti-jamming timing channels for wireless networks," in *WiSec '08: Proceedings of the first ACM conference on Wireless network security*. New York, NY, USA: ACM, 2008, pp. 203–213.

[21] A. D. Wood, J. A. Stankovic, and G. Zhou, "Deejam: Defeating energyefficient jamming in ieee 802.15.4-based wireless networks," in *Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2007.

[22] A. Ward, A. Jones, and A. Hopper, "A new location technique for the active office," *IEEE Personal Communications*, vol. 4(5), pp. 42–47, 1997.

[23] Y. Chen, K. Kleisouris, X. Li, W. Trappe, and R. P. Martin, "The robustness of localization algorithms to signal strength attacks: a comparative study," in *Proceedings of the International Conference on Distributed Computing in Sensor Systems (DCOSS)*, June 2006, pp. 546–563.

[24] G. Chandrasekaran, M. A. Ergin, J. Yang, S. Liu, Y. Chen, M. Gruteser, and R. Martin, "Empirical evaluation of the limits on localization using signal strength," in *Proceedings of the Third Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, June 2009.

[25] J. Hightower, G. Borriello, and R. Want, "Spoton: An indoor 3d location sensing technology based on RF signal strength," University of Washington, Dept. of Computer Science and Engineering, Technical Report 00-02-02, February 2000.

[26] P. Enge and P. Misra, *Global Positioning System: Signals, Measurements and Performance*. Ganga-Jamuna Pr, 2001.

[27] T. He, C. Huang, B. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes in large scale sensor networks," in *Proceedings of the Ninth Annual ACM International Conference on Mobile Computing and Networking (MobiCom'03)*, 2003.

[28] N. Bulusu, J. Heidemann, and D. Estrin, "Gps-less low-cost outdoor localization for very small devices," *IEEE Personal Commun. Mag.*, vol. 7, pp. 28–34, 2000.

[29] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, 2001, pp. 2926–2931.

[30] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," in *Proceedings of the Fourth ACM International Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc)*, Jun 2003, pp. 201–212.

[31] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, 2004, pp. 95–107.

[32] Crossbow Technology, available at http://www.xbow.com/.

[33] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[34] G. Wang, G. Cao, and T. L. Porta, "Movement-assisted sensor deployment," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 640–652, 2006.