Wireless Netw (2011) 17:531–547
DOI 10.1007/s11276-010-0295-6

Determining the position of a jammer using a virtual-force iterative approach

Hongbo Liu · Zhenhua Liu · Yingying Chen · Wenyuan Xu

Published online: 23 October 2010

Abstract Wireless communication is susceptible to radio interference and jamming attacks, which prevent the reception of communications. Most existing anti-jamming work does not consider the location information of radio interferers and jammers. However, this information can provide important insights for networks to manage its resource in different layers and to defend against radio interference. In this paper, we investigate issues associated with localizing jammers in wireless networks. In particular, we formulate the jamming effects using two jamming models: region-based and signal-to-noise-ratio(SNR)-based; and we categorize network nodes into three states based on the level of disturbance caused by the jammer. By exploiting the states of nodes, we propose to localize jammers in wireless networks using a virtual-force iterative approach. The virtual-force iterative localization scheme is a range-free position estimation method that estimates the position of a jammer iteratively by utilizing the network topology. We have conducted experiments to validate our SNR-based jamming model and performed extensive simulation to evaluate our approach. Our simulation results have showed that the virtual-force iterative approach is highly effective in localizing a jammer in various network conditions when comparing to existing centroid-based localization approaches.

Keywords Jamming · Radio interference · Localization · Virtual force

1 Introduction

As wireless networks become increasingly pervasive, ensuring the dependability of wireless network deployments will become an issue of critical importance. One serious class of threats that will affect the availability of wireless networks are radio interference, or jamming attacks. Jamming attacks can be launched with little effort with two reasons. First, 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. Another reason stems from the fact that most wireless networks consist of commodity devices that can be easily purchased and reprogrammed to interfere with communications. For instance, a device can be programmed to either prevent users from being able to get hold of communication channel to send messages, or introduce packet collisions that force repeated backoff, and thus disrupts network communications.

To ensure the availability of wireless networks, mechanisms are needed for the wireless networks to cope with jamming attacks. In this paper, we explore the task of diagnosing jamming attacks. In particular, how to localize a jammer. Learning the physical locations of the jammers allows the network to further exploit a wide range of

H. Liu (✉) · Y. Chen
Department of ECE, Stevens Institute of Technology,
Castle Point on Hudson, Hoboken, NJ 07030, USA
e-mail: hliu3@stevens.edu

Y. Chen
e-mail: yingying.chen@stevens.edu

Z. Liu · W. Xu
Department of CSE, University of South Carolina,
Columbia, SC 29208, USA
e-mail: liuz@cse.sc.edu

W. Xu
e-mail: wyxu@cse.sc.edu

defense strategies. One can cope with a jammer or an interference source by localizing it and neutralize it through human intervention. Additionally, the location of jammers 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 due to failed packet delivery.

Much work has been done in the area of localizing a wireless device [1–5], but these approaches are not applicable to determine the location of jammers due to three challenges. First, jammers will not comply with localization protocols. Most existing localization schemes either require special hardware, e.g., ultrasound transmitter to measure the time difference of arrival, or require nodes to be localized to participate in localization algorithms, making them inapplicable to localize jammers. Second, the jamming signal is usually embedded in the legal signal and is very hard, if possible, to extract. Finally, as jamming has disturbed network communication, the localization schemes cannot require extensive communication among network nodes. So far, very little work has been done in localizing jammers.

To address the challenges of finding the position of a jammer, we first studied the impact of jamming on network nodes at various locations. Based on the level of disturbance, we divided the network nodes into three main categories: *unaffected* nodes, *jammed* nodes, and *boundary* nodes. Further, we examined two jamming models, *region-based* and *signal-to-noise-ratio(SNR)-based*, to illustrate the underlying principles that govern the state of a node. The region-based model is widely adopted in many literatures, and it determines the impact of jamming purely by examining the received jamming signal power, whereas the SNR-based model exploits the SNR at the receiver, which captures the jamming effects more accurately. We investigated both models to provide a guideline of model selection for studying jamming-related problems.

Second, we propose a virtual-force iterative localization (VFIL) approach that can utilize both jamming models to estimate the position of a jammer. The VFIL method leverages the network topology and exploits the knowledge of the node state to perform position estimation. The basic scheme of the virtual-force iterative approach, VFIL-Tr, assumes the transmission range of the jammer is known. Since in practice the jammer's transmission range is mostly unknown, we further derived a variant of VFIL called VFIL-NoTr, which estimates the jammer's transmission range in each iterative step. Additionally, when utilizing SNR-based jamming model, we observed an oscillation problem, whereby the estimation of the jammer oscillates between two locations other than the true location of the jammer. To address oscillation, we developed an improved VFIL mechanism for both VFIL-Tr and VFIL-NoTr.

Finally, we conducted experiments using MicaZ motes and performed extensive simulations under different network configurations, such as various network node densities and jammer's transmission ranges. Our experimental results validate the jamming models, and we found that the VFIL approach is more effective under the widely-adopted region-based model than the more realistic SNR-based model. Further, simulation results show that our virtual-force iterative approach is less sensitive to node densities and can achieve higher localization accuracy compared with centroid-based approaches.

The rest of the paper is organized as follows. We begin the paper in Sect. 2 by discussing the related work. In Sect. 3, we specify the network models and adversary models being used in this paper. We next formulate the jamming effects using the region-based model and present the virtual-force iterative localization scheme leveraging it in Sect. 4. To better capture the jamming effects, in Sect. 5, we introduce the SNR-based model and validate it via experiments. We describe the improved VFIL scheme in Sect. 5, which can address the oscillation problem occurred in the SNR-based model. Finally, we present the comprehensive simulations that evaluate our VFIL algorithms in Sect. 6 and conclude in Sect. 7.

2 Related work

Coping with jamming and interference is usually a topic that 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 [6, 7]. Although such PHY-layer techniques can address the challenges of an RF interferer, they require advanced transceivers.

Further, the issue of detecting jammers was briefly studied by Wood et al. [8], and was further studied by Xu et al. [9], 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 [10, 11] and in networks involving frequency hopping [12]. Our work focuses on localizing jammers after jamming attacks have been identified using the proposed jamming detection strategies.

Without localizing jammers, Wood et al. [8] 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 slightly modification, our algorithm can not only localize the jammer but also map the jammed region.

Moreover, countermeasures for coping with jammed regions in wireless networks have been investigated. The use of error correcting codes [13] is proposed to increase the likelihood of decoding corrupted packets. Channel surfing/hopping [14–16], whereby wireless devices change their working channel to escape from jamming, spatial retreats [17], whereby wireless devices move out of jammed region geographically, and anti-jamming timing channel [18], 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 [19]. The combinations of mask framing, frequency hopping, packet fragmentation, and redundant encoding techniques is proposed to cope with multiple types of jammers [20].

On the other hand, there has been active work in the area of wireless localization. Based on localization infrastructure, infrared [1] and ultrasound [21, 22] are employed to perform localization, both of which need to deploy specialized infrastructure for localization. Further, using received signal strength (RSS) [2, 4, 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 [2, 4, 23, 25], Time Of Arrival [26], and Time Difference of Arrival [21]. Range-free algorithms [27–30] use coarser metrics to place bounds on candidate positions.

However, little work has been done in localizing jammers. Most of the existing localization methods can not be applied to localize jammers due to the disturbed network communication under jamming attacks. Recently Pelechrinis et al. [31] proposed to localize the jamming by measuring packet delivery rate (PDR) and performing gradient decent search. However, they did not present results of performation evaluation. Our work is novel in that rather than relying on the traditional network communication approaches, we use network topology to achieve better accuracy of localizing jammers comparing to existing range-free algorithms. We further conducted extensive simulation to validate the effectiveness of our approach under two different jamming models.

3 Overview of network model and jamming model

In this section we outline the basic wireless network and jamming models that we use throughout this paper.

3.1 Network model

A wide variety of wireless networks have emerged, ranging from wireless sensor networks, mobile ad hoc network, to mesh networks. The broad range of choice implies that there are many different directions that one can take to tackle the problem of localizing jammers. Devising a generic approach that works across all varieties of wireless networks is impractical. Therefore, as a starting point, we target to tailor our solutions to a category of wireless networks with the following characteristics.

3.1.1 Stationary

We assume that once deployed, the location of each wireless device remains unchanged. We will consider mobility in our future works.

3.1.2 Neighbor-aware

Each node in the network has a number of neighbors, and it maintains a table that records its neighbors' information, such as their locations or activeness. Such a neighbor table is typically maintained by routing protocols. In cases where it is not available from routing protocols, it can be easily achieved by letting each node periodically broadcast hello messages.

3.1.3 Location-aware

Each node knows its location coordinates and its neighbors' locations. This is a reasonable assumption as many applications already require localization services [2, 28].

3.1.4 Omnidirectional

Each node is equipped with an omnidirectional antenna, and each node has the same radio range in all directions.

3.1.5 Able to detect jamming

In this work, we focus on locating a jammer after it is detected. We assume the network is able to identify a jamming attack. The network can utilize one of the existing jamming detection approaches, ranging from measuring simple properties [8, 9] to leveraging more complicated consistency checks.

3.2 Jamming model

In this work, we deal with jammers each equipped with an omnidirectional antenna. Once deployed, the jammer is stationary in the network. There can be multiple jammers in the network. However, multiple jammers do not have overlapped jamming regions.

3.3 Formulation of jamming effects

There are many different attack strategies that a jammer can perform in order to disrupt wireless communications [9]. For example, a constant jammer continually emits a radio signal. Alternatively, the reactive jammer stays quiet when the channel is idle, but starts transmitting a radio signal as soon as it senses activity on the channel, causing a message to be corrupted when it is received.

Despite the diversity of many attack philosophies, the consequences of different jammers are the same. For those nodes which are located near a jammer, their communication are severely disrupted, whereas a node which is far away from a jammer may not be affected by the jammer at all. Based on the degree of disturbance to a node, we can divide network nodes, N, into three non-overlapping categories under jamming attacks: jammed nodes N_J, boundary nodes N_B, and unaffected nodes N_U. Let l_{ij} be the predicate function of the link state *from* node n_i to node n_j. We define l_{ij} as,

$$l_{ij} = \begin{cases} 1 & n_j \text{ can receive packets from } n_i \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

we emphasize that the link state defines the *receiving* capability within one hop, and it is possible that $l_{ij} \neq l_{ji}$. Further, we define L_{mn} as the end-to-end connectivity predicate function from node n_m to n_n, which are two nodes more than one hop away. We denote $L_{mn} = 1$, if there exists a path from node n_m to n_n and the link state of each hop equals 1. Otherwise, $L_{mn} = 0$. Let $Nbr\{n_i\}$ be the set of one-hop neighbors of node n_i before any jammer becomes active. Given L_{mn} and l_{ij}, we define N_J, N_B, and N_U as below,

- **Unaffected node**. $N_U = \{n_u | \forall n_i \in Nbr\{n_u\},\ l_{iu} = 1\}$. The receiving ability of unaffected nodes have not been changed by the jammer at all. Their neighbor set remains the same as the one before a jammer starts to interfere.
- **Jammed node**. $N_J = \{n_j | \forall n_i \in N_U,\ L_{ij} = 0\}$. Due to jamming, the communication of jammed nodes has been severely disturbed. Considering the fact that in a normal scenario a node is able to communicate with any nodes in the network either directly or indirectly through multiple intermediate nodes, we define a jammed node as the one that cannot *receive* packets from any unaffected nodes.
- **Boundary node**. $N_B = \{n_b | (\exists n_i \in N_U,\ L_{ib} = 1)\ \mathit{and}\ (Nbr\{n_b\} \cap N_J \neq \emptyset)\}$. A boundary node is neither a jammed node nor an unaffected node. Its communication ability is partially affected. Part of its neighbors are jammed, but it can still reach at least one unaffected node, possibly, in multiple hops.

In this paper, we consider that a stationary jammer interferes with the network communication at fixed transmission power setup. To make the problem more attackable, we do not consider network topology changes caused by factors other than jamming, e.g., mobility, or hardware failures. As such, the state of each network node will remain unchanged throughout the course that the jammer is active. We denote S_i as the state of the i-th node, and it can be any value in the set of {JAMMED, BOUNDARY, UNAFFECTED}.

Our jamming localization algorithms start to determine the location of the jammer after jamming is detected and the state of each node is identified. One candidate detection algorithm is consistency-based jamming detection algorithm [9], which not only identifies the presence of jamming but also returns the link state of each pair of nodes. Thus, we are able to derive S_i for all nodes in the network.

In addition to determining S_i relying on detection algorithms, it is important to understand the underlying principles that govern the state of a node n_i. In this paper, we will examine two models. The first one is a widely adopted model, whereby a node is considered jammed if it is located within the jammed region. The second model involves checking the signal-to-noise ratio (SNR) at each node. We will study those two models in Sects. 4 and 5, respectively.

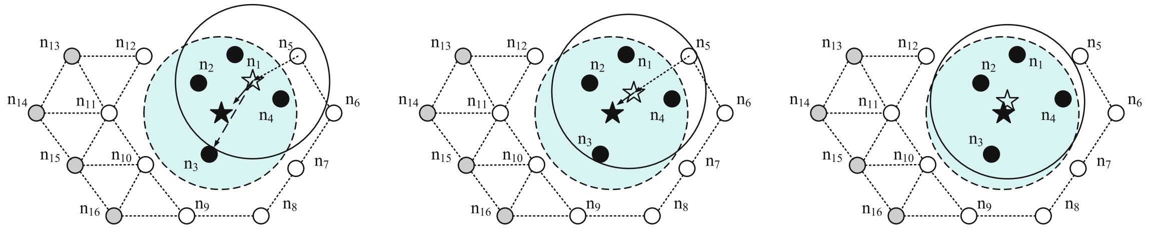
4 Jamming localization in region-based model

4.1 Model formulation of jamming effects

Typically, most recent papers on jamming and wireless networks have modeled the effect of the jammer as a region-based effect [8, 32], i.e., for a given jammer, there exists a jammed region within which a node is unable to communicate with its neighbors and thus is considered jammed. A jammer with a directional antenna may have a sector-shaped jammed region; a jammer with an omnidirectional antenna may have a circular jammed region; and multiple jammers can create a jammed region which is the union of the individual jammed regions. In this section, we focus on the region-based jamming model. To simplify the illustration, we adopt the standard free space propagation model. The power of the received signal at d meters away from the transmitter is

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

where P_T is the transmission power; G is the product of the transmit and receive antenna field radiation patterns in the LOS (line-of-sight) direction (Fig. 1).



(a) Iteration 1 **(b)** Iteration 2 **(c)** Iteration 3

Fig. 1 Iteration of localization steps in Virtual Force Iterative Localization (VFIL) method

Apply the free space propagation model to the jammer, the received jamming signal power decreases inversely proportional to the square of the distance d. Thus, the effect of a jammer is a circle centered at the jammer's location, as shown in Fig. 2. On the edge of the circle, the received jamming signals equal ambient noise level and we call this circle the **Noise Level Boundary (NLB)** of the jammer with radius R_J. A node located inside the NLB circle is considered jammed, as its received jamming signal is larger than the ambient noise level. In contrast, a node outside the NLB circle is considered non-jammed. Therefore, NLB defines the jammed region, and we define the link state of a pair of nodes i and j below,

$$l_{ij} = l_{ji} = \left\{ \begin{array}{lll} 0||Z_i - Z_J||_2 \leq R_J & \text{or} & ||Z_j - Z_J||_2 \leq R_J \\ 1|Z_i - Z_J||_2 > R_J & \text{and} & ||Z_j - Z_J||_2 > R_J \end{array} \right\}. \tag{3}$$

where Z_i is the position of i-th node, Z_J is the position of the jammer and $||*||_2$ is the 2-norm distance. In the region based jamming model, the definition of the three categories N_U, N_J and N_B can be simplified as,

- **Unaffected node**. $N_U = \{n_u|(\forall n_i \in Nbr\{n_u\}, \; ||Z_i - Z_J||_2 > R_J) \text{ and } (||Z_u - Z_J||_2 > R_J)\}$. An unaffected node and all of its neighbors are located outside of NLB of the jammer.
- **Jammed node**. $N_J = \{n_j| \; ||Z_j - Z_J||_2 \leq R_J\}$. A jammed node is located within the NLB of the jammer.

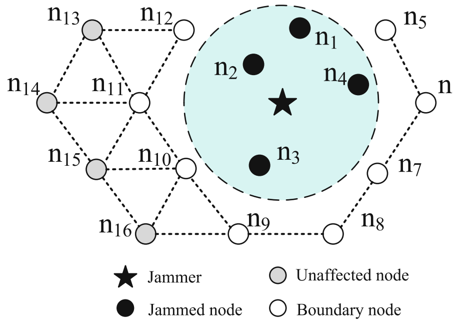


Fig. 2 Illustration of a jamming scenario in a wireless network using the region-based jamming model

- **Boundary node**. $N_B = \{n_b|(||Z_b - Z_J||_2 > R_J) \text{ and } (\exists n_i \in Nbr\{n_b\}, \; ||Z_i - Z_J||_2 \leq R_J)\}$. A boundary node is not jammed itself but it cannot hear from some of its neighbors.

Figure 2 illustrates the classification of different network nodes under a jamming situation. The jammed region is the light blue circle centered at the jammer. Nodes $\{n_1, n_2, n_3, n_4\}$ are jammed nodes; nodes $\{n_5, n_6, n_7, n_8, n_9, n_{10}, n_{11}, n_{12}\}$ are boundary nodes; nodes $\{n_{13}, n_{14}, n_{15}, n_{16}\}$ are unaffected nodes.

4.2 Jammer localization algorithms

In this section, we first overview an existing range-free localization algorithm that can be applied to determine the position of a jammer, Centroid Localization (CL). We then present our approach of Virtual Force Iterative Localization (VFIL).

4.2.1 Centroid localization (CL)

Centroid Localization can be used to localize a jammer, as it performs the estimation without the cooperation of the target nodes. In particular, CL utilizes position information of all neighboring nodes of the node to be localized. In the case of localizing a jammer, the neighboring nodes of the jammer are jammed nodes. Therefore, to infer the position of a jammer, CL collects all coordinates of jammed nodes, and averages over their coordinates as the estimated position of the jammer. Assume that there are m jammed nodes $\{(x_1, y_1), (x_2, y_2),\ldots, (x_m, y_m))\}$. The position of the jammer can be estimated by:

$$\hat{Z}_J = (\hat{x}_J, \hat{y}_J) = \left(\frac{\sum_{k=1}^{m} x_k}{m}, \frac{\sum_{k=1}^{m} y_k}{m} \right). \tag{4}$$

CL only utilizes the coordinates of network nodes, and therefore it is robust against the radio propagation uncertainties in the environment. However, it is extremely sensitive to the distribution of jammed nodes. For example, if the distribution of the jammed nodes is biased towards one side of the jammer, the estimation will be biased as well.

Additionally, it is sensitive to node density. In a uniformly distributed network, increasing the network density will increase the chances that jammed nodes are evenly distributed around the jammer, and thus produce better estimation.

4.2.2 Virtual force iterative localization (VFIL)

Centroid localization method is sensitive to the distribution of the jammed nodes and the network density. To achieve a better localization accuracy, we propose the Virtual Force Iterative Localization method.

VFIL starts with a coarse estimation of the jammer's position derived by CL, and then re-estimate the jammer's position iteratively until the estimated jammer's position is close to the true location. There are several challenges associated with this algorithm: (1) How do we know when the estimated position is close enough? (2) How can we adjust the estimation in each iteration?

4.2.2.1 Termination When the estimated jammer's location is its true position, the estimated jammed region will overlap with the real jammed region. One main characteristic of the real jammed region is that it contains all jammed nodes but none of the boundary nodes. Thus, VFIL should stop when the estimated jammed region covers all the jammed nodes while all boundary nodes fall outside of the region.

4.2.2.2 Iteration At each round of location estimation in VFIL, some of the jammed nodes will be located inside the estimated jammed region, while others may be outside. Similarly, some boundary nodes may be included in the estimated jammed region mistakenly caused by the estimation errors. Since the objective of VFIL is to search for an estimation of the jammed region that can cover all the jammed nodes whereas does not contain any boundary nodes, at each iterative step, the jammed nodes that are outside of the estimated jammed region should pull the jammed region toward themselves, while the boundary nodes that are within the estimated jammed region should push the jammed region away from themselves.

To model this push and pull trend, we define two virtual forces, namely *Pull Force* $\mathbf{F}^{\mathbf{i}}_{\mathbf{pull}}$ generated by a jammed node i that is outside of the jammed region, and *Push Force* $\mathbf{F}^{\mathbf{j}}_{\mathbf{push}}$ generated by a boundary node j that is located inside the jammed region. Let $(\hat{X}_J, \hat{Y}_J)$ be the estimated position of the jammer, (x_j, y_j) be the location of a jammed node and (x_b, y_b) be the location of a boundary node. We define $\mathbf{F}^{\mathbf{n_j}}_{\mathbf{pull}}$ and $\mathbf{F}^{\mathbf{n_b}}_{\mathbf{push}}$ as normalized vectors that point to/from the estimated jammer's position as

$$\mathbf{F}^{\mathbf{n_j}}_{\mathbf{pull}} = \left[\frac{x_j - \hat{X}_J}{\sqrt{(x_j - \hat{X}_J)^2 + (y_j - \hat{Y}_J)^2}}, \frac{y_j - \hat{Y}_J}{\sqrt{(x_j - \hat{X}_J)^2 + (y_j - \hat{Y}_J)^2}} \right], \tag{5}$$

and

$$\mathbf{F}^{\mathbf{n_b}}_{\mathbf{push}} = \times \left[\frac{\hat{X}_J - x_b}{\sqrt{(X_J - x_b)^2 + (Y_J - y_b)^2}}, \frac{Y_J - y_b}{\sqrt{(X_J - x_b)^2 + (Y_J - y_b)^2}} \right]. \tag{6}$$

We further define a joint force $\mathbf{F_{joint}}$ as the combination of all $\mathbf{F}^{\mathbf{n_j}}_{\mathbf{pull}}$ and $\mathbf{F}^{\mathbf{n_b}}_{\mathbf{push}}$ based on the formula of force synthesization [33]:

$$\mathbf{F_{joint}} = \frac{\sum_{n_j \in N_{pull}} \mathbf{F}^{\mathbf{n_j}}_{\mathbf{pull}} + \sum_{n_b \in N_{push}} \mathbf{F}^{\mathbf{n_b}}_{\mathbf{push}}}{\left| \sum_{n_j \in N_{pull}} \mathbf{F}^{\mathbf{n_j}}_{\mathbf{pull}} + \sum_{n_b \in N_{push}} \mathbf{F}^{\mathbf{n_b}}_{\mathbf{push}} \right|}, \tag{7}$$

where N_{pull} is the set of jammed nodes that are located outside of the estimated jammed region, and N_{push} is the set of boundary nodes that are located within the estimated jammed region. Following the direction of $\mathbf{F_{joint}}$, VFIL moves the estimated position of the jammer towards the jammer's true position at each iteration.

4.2.2.3 Algorithm walk-through In order to illustrate the VFIL, let us walk through each step of the algorithm. We start with the algorithm that assumes a known jammed range, i.e., the NLB of the jammer is known, and then present the approach that can determine the jammer's location without this assumption.

- **Step 0**. Detect the jamming attack and determine S_i for all network nodes based on the link state returned by the jamming detection algorithm.
- **Step 1**. Estimate the position of the jammer, $\hat{Z}_J$. The intial estimation is obtained by calculating the centroid of all jammed nodes.
- **Step 2**. Derive the estimated jammed region, which is a circle centered at $\hat{Z}_J$ with the radius the same as the jammed range, R_J.
- **Step 3**. Infer the state $\hat{S}_i$ for each node using the estimated jammed region based on the definition of N_J, N_B, and N_U. Derive N_{pull} and N_{push}, and form the joint force $\mathbf{F_{joint}}$.
- **Step 4**. Set an adjustable moving step, and Move the estimated jammer's position along the direction of $\mathbf{F_{joint}}$ to a new estimate position, e.g., $\hat{Z}_J = \hat{Z}_J + \mathbf{F_{joint}} \times \Delta$, where Δ is the step size. The objective is

to reduce the resulting $\mathbf{F_{joint}}$ and make it approach zero in the following iterations.

- **Step 5**. Repeat Step 1 to 4 until all the jammed nodes are included in the estimated jammed region and all the boundary nodes are excluded in the estimated jammed region.

Figure 1(a–c) illustrate the iterative steps of VFIL algorithm. In the first step depicted in Fig. 1(a), $N_{pull} = \{n_3\}$ and $N_{push} = \{n_5\}$. The combined $\mathbf{F_{joint}}$ applied by nodes n_3 and n_5 moves the estimated position of the jammer to a new location whereby the jammed region becomes $\{n_1, n_2, n_3, n_4, n_5\}$, which contains all jammed nodes and a boundary node n_5, as depicted in Fig. 1(b). Thus, in the second iteration, $N_{pull} = \emptyset$ and $N_{push} = \{n_5\}$, and $\mathbf{F_{joint}} = \mathbf{F^{n_5}_{push}}$. Pushed by $\mathbf{F^{n_5}_{push}}$, the jammer is moved to a new estimated location, as depicted in Fig. 1(c), which results in a new jammed region that contains all jammed nodes and excludes all non-jammed nodes. As a result, the algorithm terminates.

4.2.2.4 Estimation of NLB of the jammer In the aforementioned algorithm, we assume the jammed radius is known. Now, we consider the case that the transmission range of the jammer R_J is unknown. We propose to estimate $\hat{R}_J$ in each iteration step. In particular, right after estimating the jammer position, $\hat{Z}_J$, we calculate the jammed range as the one that minimizes the number of *unmatched* nodes, that is, the jammed nodes that are located outside of the estimated jammed range and the boundary nodes that are located within the estimated jammed range. Formally, we define the *unmatched* nodes in terms of N_{pull} and N_{push} as $N_{pull} = \{n_i | S_i = JAMMED \cap \hat{S}_i \neq JAMMED\}$ and $N_{push} = \{n_i | S_i \neq JAMMED \cap \hat{S}_i = JAMMED\}$. Thus, the estimated radius of the NLB of the jammer equals,

$$\hat{R}_J = \arg\min_{R_J}(||N_{push}|| + ||N_{pull}||). \tag{8}$$

After finding the best match of the jammed range in Step 2, VFIL continues to determine N_{pull} and N_{push} and forms the joint force $\mathbf{F_{joint}}$ in Step 4.

The detailed flow chart of VFIL is depicted in Fig. 3. In the rest of the paper, to distinguish these two variants of VFIL, we call the virtual force algorithm that assumes the awareness of R_J as VFIL-Tr, and the one without the knowledge of R_J as VFIL-NoTr.

4.2.2.5 Convergence Based on the observation of our simulation study, in most cases, VFIL converges within 100 iterations towards the true position of the jammer. In rare cases, the algorithm will fluctuate around the true position instead of converging towards the true position quickly. After careful examination, we found that the fluctuated estimations in such cases are already very close to the true position. Therefore, after a threshold of iterations, we stop the iterations and use the current estimation value as the final localization estimate.

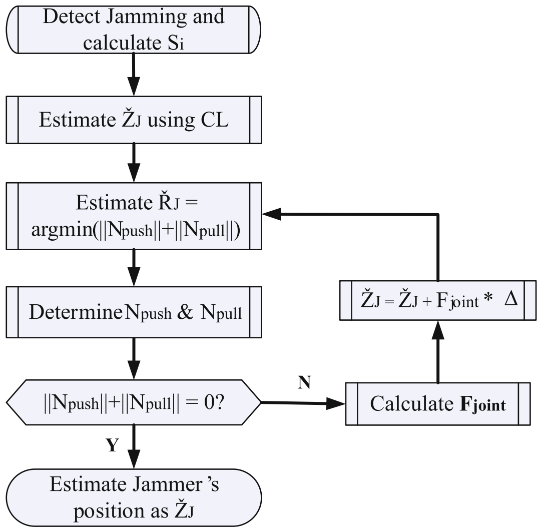

Detect Jamming and calculate Si
Estimate ŽJ using CL
Estimate ŘJ = argmin(||Npush||+||Npull||)
Determine Npush & Npull
||Npush||+||Npull|| = 0?
N
Y
Calculate Fjoint
ŽJ = ŽJ + Fjoint * Δ
Estimate Jammer 's position as ŽJ

Fig. 3 Flow chart of Virtual Force Iterative Localization algorithm without the knowledge of jammed range (VFIL-NoTr)

5 Jamming localization in SNR-based model

5.1 Model formulation of jamming effects

5.1.1 The limitation of the region based model

The region-based jamming model essentially only considers the path loss of the jamming signal while ignores the signal from the sender S, and thus it does not provide a complete depiction of the complex relationships between the transmission power of senders and the jammer, as well as the geometry of the deployment. Essentially, the probability of success reception of a packet is primarily a function of the signal-to-noise ratio (SNR) at the receiver R. In the jamming scenario, the “noise’ includes ambient noise P_N and jamming signals P_J,

$$\mathrm{SNR} = \frac{P_{SR}}{P_N + P_{JR}} \approx \frac{P_S G_{SR}}{P_J G_{JR}} \times \frac{d^2_{JR}}{d^2_{SR}}, \tag{9}$$

where P_{SR} is the received power of desired signal, P_N is the noise, and P_{JR} is the received jamming power; P_S and P_J are the transmission power of S and J respectively; G_{SR} and G_{JR} are the antenna field patterns in the line-of-sight(LOS) direction between S and R, and between J and R. The approximation holds when the ambient noise level is neglectable compared to the signal and jamming power.

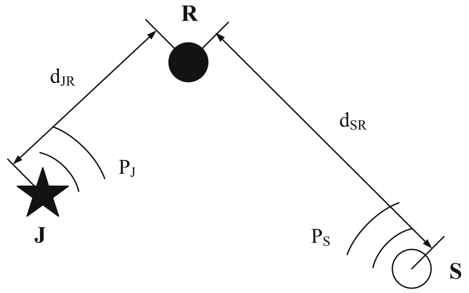


Fig. 4 Illustration of received signals at the receiver *R*

Figure 4 illustrates the relationship among P_S, P_J, d_{SR} and d_{JR}. From Eq. 9, for a given set of devices and transmission power setup, the SNR at *R* is a function of its distances to *J* and *S*, e.g., d_{JR}, d_{SR}.

5.1.2 PDR versus SNR

To understand the relationship between SNR and the probability of the successful reception of packets, we carried out several experiment studies using three Crossbow MicaZ motes. We use the metric packet delivery ratio (PDR), the percentage of successfully delivered packets out of the total delivery attempts, to qualify the packet reception quality. In our experiments, two MicaZ motes act as the sender *S* and the receiver *R* respectively, and a third mote *J* continuously sends out random bits to interfere with the communication between *S* and *R*. Throughout the experiments we used the same motes as *R*, *S*, and *J* to eliminate the impact that might be caused by hardware variance.

There are many ways to create different SNR at *R*. In our experiment, we fixed the locations of *S* and *J* with 40 inches separation, and moved *R* in a 50 × 100 inches rectangular with the step size of 5 inches. At each location, *R* has a unique pair of (d_{JR}, d_{SR}), and it can lead to different SNR values at *R*. At each spot, *R* measures two statistics, the number of received packets (the sender *S* will transmit 1,000 packets in total) and RSSI (Received Signal Strength Indication) values in two scenarios. The first case involves measuring RSSI when both *S* and *J* are active, and the second involves keeping *J* active while *S* turned off. Those two measurements are used to calculate SNR at each location. We plot the measured (SNR, PDR) in Fig. 5.

Figure 5 demonstrates a sharp cliff phenomenon: when SNR is smaller than 0 dB, the PDR is 0; as SNR value increases beyond some threshold value, the PDR jump to 100%. This cliff phenomenon coincides with the theoretical result, i.e., the success reception of a packet is primarily determined by SNR at the receiver. Thus, we can define the link state l_{ij} based on a threshold model. Formally, the link state from node n_i to n_j is

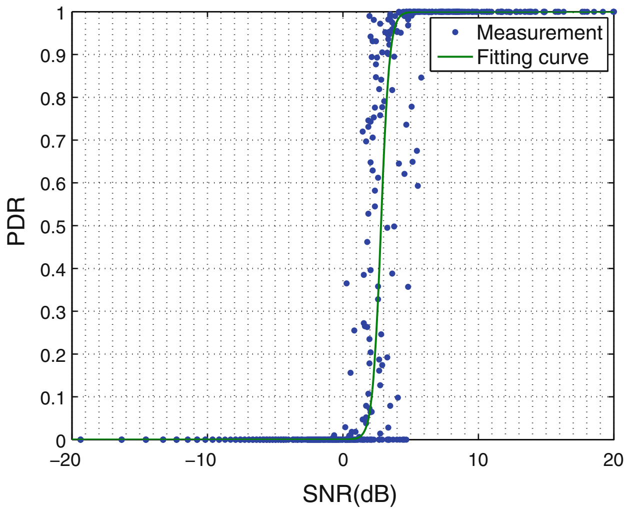


Fig. 5 Experiment results of the relationship between Packet Delivery Ratio (PDR) to Signal-to-Noise Ratio (SNR), using MicaZ

$$l_{ij} = \begin{cases} 0 & \mathrm{SNR}_{ij} \le \gamma_0 \\ 1 & \mathrm{SNR}_{ij} > \gamma_0 \end{cases} \qquad (10)$$

where SNR_{ij} is the SNR measured at node n_j when node n_i is transmitting and all other network nodes remain silent. γ_o is the threshold SNR, above which packets can be received successfully, and we call it **Decodable SNR threshold**. To determine the value of γ_0 empirically, we performed a logistic regression over the measured data pairs (SNR, PDR), and get γ_o = 3.42 dB when PDR = 95%.

- **Jamming Effects in SNR-based Model**. Applying Eq. 10 to the definition of network nodes, N_J, N_B, and N_U, in the SNR-based jamming model we have
- **Unaffected node**. $N_U = \{n_u | \forall n_i \in Nbr\{n_u\}, SNR_{iu} > \gamma_o\}$. A node is unaffected, if it can receive packets from all of its neighbors.
- **Jammed node**. $N_J = \{n_j | \forall n_i \in N_U,\ L_{ij} = 0\}$. Essentially, a node n_j is jammed if it cannot communicate with any of the unaffected nodes. We note that two jammed nodes may still be able to communicate with each other. However, they cannot communicate with any of the unaffected nodes.
- **Boundary node**. $N_B = \{n_b | (\exists n_i \in N_U,\ L_{ib} = 1)\ \mathit{and}\ (\forall n_i \in Nbr\{n_b\} \cap N_J, SNR_{ib} \le \gamma_o)\}$. A boundary node can receive packets from part of its neighbors but not from all its neighbors.

To illustrate the difference between those two jamming models, we consider a network scenario shown in Fig. 6, where the light blue circle depicts the NLB circle of the jammer. In the region-based model, nodes $\{n_1, n_2, n_3, n_7, n_8, n_{11}, n_{12}\}$ are considered jammed, since they are located inside the NLB. In the SNR-based model, however, only nodes $\{n_1, n_2, n_{11}\}$ are jammed because none of them

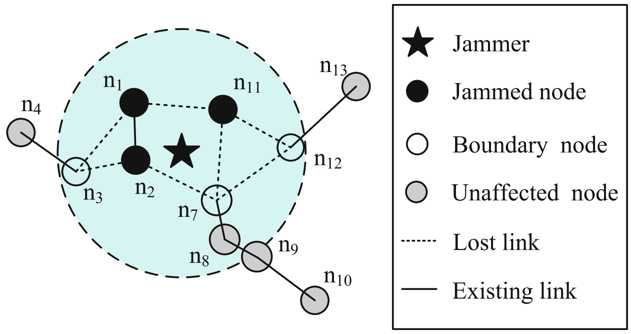


Fig. 6 Illustration of a jamming scenario under the SNR-based jamming model

is able to communicate with the majority of the network. Interestingly, although nodes n_1 and n_2 are jammed, they can still communicate with each other. This is because n_1 and n_2 are so close to each other that their received SNR_{12} and SNR_{21} are both higher than the decodable threshold γ_o. In fact, we have observed such a phenomenon during our PDR versus SNR experiment using MicaZ motes [34]. Additionally, despite the fact that node n_8 is located within NLB, it is unaffected, because it does not lose the communication ability with all its neighbors. Similarly, $\{n_4, n_8, n_9, n_{10}, n_{13}\}$ are unaffected. Finally, nodes $\{n_3, n_7, n_{12}\}$ are not jammed but boundary nodes, because they can reach at least one of the unaffected nodes.

5.2 Enhanced virtual force iterative localization algorithm

Once the presence of the jammer is detected, the jammer can be localized using the VFIL scheme described in Sect. 4.2. Different from the region-based model, we applied the SNR-based model to determine the estimated status of the nodes in every iteration. To make the calculation efficient, we adopted the concept of *nearest* and *furthest* un-jammed neighbors. Essentially, a node n_i is unaffected, if it can receive packets from the neighbor located furthest away from it, since the furthest neighbor will create the weakest signal at n_i among all its neighbors and the ability to receive from furthest neighbor indicates the ability to receive from all neighbors. Similarly, a node n_i is a boundary node, if it cannot receive packets from its furthest neighbor yet can hear from its nearest un-jammed neighbor. A node is jammed if it cannot hear from its nearest un-jammed neighbor.

Applying the SNR-based model to VFIL algorithm, we found that in some special scenarios, the estimated jammer position will not converge to the true location of the jammer, instead it oscillates between locations other than the true location of the jammer. Even if different step sizes of the virtual force are adopted, the oscillation phenomenon persists. To address the extra oscillation problem, an improved VFIL scheme is needed. We next describe the oscillation problem that we encountered.

5.2.1 *Oscillation problem*

As defined in Sect. 3, each node n_i has a state, S_i, that is determined by the true location of the jammer. In each iteration, we infer the estimated node state, $\hat{S}_i$, based on the estimated position of the jammer, and the estimated state $\hat{S}_i$ may not match its actual state S_i. Under such a situation, the node n_i becomes an unmatched node. The objective of the VFIL algorithm is to find an estimated location of the jammer when all the nodes are in their actual states. Thus, the VFIL algorithm will use the joint virtual force to pull or push the estimated jammer's position to the next estimated location and seek to make these unmatched nodes to be in their actual states. However, it is possible that the next estimated location results in another group of unmatched nodes and the resulting joint virtual force moves the estimated jammer's position back to the previous location. We define this situation as the *oscillation* problem. When oscillation occurs between two estimated locations of jammer's position, the localization algorithm will not converge.

5.2.2 *Example*

In Fig. 7, we provide a detailed example to illustrate the oscillation problem when localizing a jammer. In this example, the estimated positions of the jammer oscillate between E_1 and E_2. Under jamming, $\{n_3, n_7, n_8, n_9\}$ are non-jammed nodes. In particular, nodes $\{n_3, n_7\}$ are boundary nodes, and nodes $\{n_8, n_9\}$ are unaffected. However, when the estimated jammer's location is at E_1 as shown in Fig. 7(a), n_3 becomes an unmatched node with its state as $\hat{S}_3 = JAMMED$, and it generates Push Force $\mathbf{F}_{\mathbf{push}^{\mathbf{n_3}}}$. Thus, the virtual force, $\mathbf{F_1} = \mathbf{F}^{\mathbf{n_3}}_{\mathbf{push}}$, will push the next position estimation to E_2 as shown in Fig. 7(b).

When the position estimation of the jammer becomes E_2, node n_3 can receive from n_4, and its state becomes *BOUNDARY*. However, nodes n_7, n_8, and n_9 become jammed nodes, which are different from their actual states. The joint virtual force $\mathbf{F_2}$ formed by n_7, n_8, and n_9 is opposite to $\mathbf{F_1}$ and will push the jammer's estimated location back to E_1. Thus, the estimated jammer's positions will be oscillated between E_1 and E_2.

Nodes $\{n_3\}$ and $\{n_7, n_8, n_9\}$ are two groups of unmatched nodes generating virtual force alternatively to move the estimated jammer's position back and forth between E_1 and E_2. In the first group, n_3 is the only node in this group, while in the second group, n_9 will determine whether the

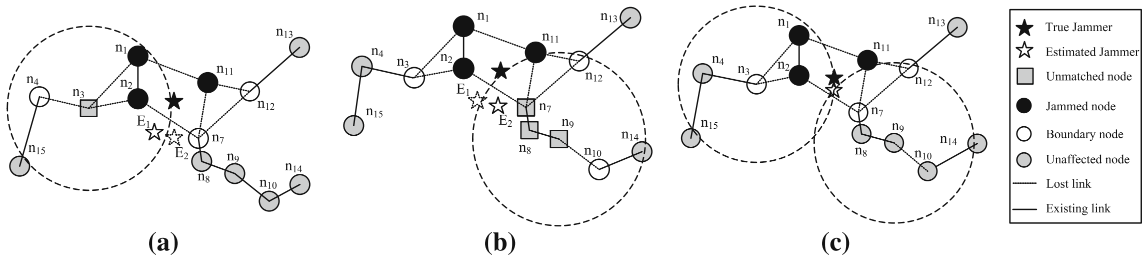


Fig. 7 Illustration of oscillation: **a** when the estimated jammer's position is at E_1, the boundary node n_3 becomes a jammed node, and it forms the virtual force to push next estimated jammer's position to E_2; **b** when the estimated jammer's position is at E_2, nodes n_7, n_8 and n_9 become jammed nodes, which do not match their actual states, and they form the virtual force to push the estimated jammer's position back to E_1; **c** the final estimated location of the jammer is considered as the intersection point $\hat{E}$ of the SNR boundaries that satisfies all jamming conditions

whole group is jammed or not as n_9 is the only path for n_7 and n_8 to transfer packets. Therefore, n_3 and n_9 are the key nodes that lead to oscillation. We call this kind of nodes as *bridge* nodes. A bridge node oscillates its state between *JAMMED* and *BOUNDARY* as the estimated locations of the jammer fluctuates among a set of positions.

5.2.3 Resolving oscillation

To solve the oscillation, we introduce the concept of the SNR boundary of the bridge nodes. Essentially, given the location of the network topology, the requirement of making the bridge node's estimated state match its true value confines the location of the jammer. For example, as depicted in Fig. 7(a), n_4 is n_3's nearest non-jammed neighbor node. To ensure that n_3 is able to receive from n_4, the ratio of n_3's receiving power from n_4 to its receiving power to the jammer, $\mathrm{SNR}_{4,3}$, should be greater than the threshold γ_0: $\frac{P_{n_4} d^2_{n_3,E_1}}{P_J d^2_{n_3,n_4}} > \gamma_0$. Thus, the jammer has to be located outside the circle that is centered at n_3 with a radius of $r_{n_3} = \sqrt{\frac{P_J d^2_{n_3,n_4} \gamma_0}{P_{n_4}}}$, depicted as the dashed circle in Fig. 7(a). We call this circle as **SNR boundary**.

Similarly, to ensure that n_9 is an unaffected node, n_9 should be able to receive packets from its furthest neighbor n_{10}. The radius of the SNR boundary of n_9 is confined by $r_{n_9} = \sqrt{\frac{P_J d^2_{n_9,n_{10}} \gamma_0}{P_{n_{10}}}}$ and the jammer has to be located outside the SNR boundary of n_9, shown in Fig. 7(b). Since both n_3 and n_9 are non-jammed nodes, the estimated location of the jammer should be outside of their SNR boundaries. Neither E_1 nor E_2 satisfy this constrain of SNR boundary. To leverage SNR boundary, we estimate the location of the jammer as one of the intersection points of SNR boundaries. In particular, we extend the Step 4 in the Virtual Force Iterative Localization algorithm as following:

– **Step 4.1**. Once an oscillation is detected, record all the unmatched nodes found in oscillation;
– **Step 4.2**. Identify bridge nodes based on the definition and append their SNR boundaries into list L;
– **Step 4.3**. Calculate the intersection points of all the SNR boundaries in list L.
– **Step 4.4**. Check the intersection points one by one against the jamming condition, i.e., state of each network node. The first point that satisfies the jamming condition is returned as the estimated jammer's position. If no intersection point satisfies the jamming condition or no intersection points are found, recalculate the virtual force using the nodes in L, and use the bridge nodes in L to adjust the step size, then go back to Step 2 (in the original algorithm). The motivation of adjusting the step size is to get a different set of bridge nodes and consequently a different set of intersection points, increasing the chances of finding one intersection point that satisfies the jamming condition.

Based on the improved VFIL, continuing the example illustrated in Fig. 7, we estimate the location of the jammer as the intersection point $\hat{E}$, whereby the derived states of all nodes match their true states, as depicted in Fig. 7(c). We will evaluate the effectiveness of our improved algorithm in the next Section.

6 Simulation evaluation

In this section, we evaluate the effectiveness of our virtual force iterative localization approach under both the region-based model (RBM) as well as the SNR-based model (SBM) through simulation.

6.1 Methodology

We implemented our own simulator using Matlab. We simulated a wireless network environment in a 300-by-300 feet field, within which network nodes were uniformly distributed. The transmission range of each node was set to 30 feet. We evaluated the performance of three algorithms, VFIL-Tr, VFIL-NoTr, and CL, in various network conditions, including different network node densities and jammer's NLB radius. To study the impact of those network parameters on the algorithms, we placed the jammer at the center of the simulation area so that the jammer was surrounded by multiple network nodes. Later, we investigated the effect of the jammer's position on the algorithm performance by randomly placing the jammer anywhere within the simulation area, including the edge of the network. To capture the average trend, we run each algorithm in each network setup 5,000 times.

6.2 Metrics

We used the following metrics to evaluate the performance of our approaches.

6.2.1 Estimation error of the jammer's NLB radius

One intermediate step of the VFIL-NoTr is to estimate the jammer's Noise-Level-Boundary (NLB) radius, which might affect the localization performance. Thus, we studied jammer's NLB estimation errors. In particular, we define such errors as the difference between the true radius of the jammer's NLB and the estimated one using VFIL-NoTr. To provide a statistical view of the estimation accuracy, we present the Cumulative Distribution Functions (CDFs) of the range estimation errors.

6.2.2 Localization error of the jammer's position

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. Similarly, to capture the statistical characteristics, we studied the average errors under multiple experimental runs. We present the CDFs of the localization errors obtained from all experimental runs.

6.3 Results

6.3.1 Sensitivity of node density

We first studied the effects of various network node densities on the localization performance. To adjust the network node densities, we varied the total number of nodes, *N*, deployed in the simulation. In particular, we chose *N* to be 200, 300, and 400, respectively. Figure 8 presents the localization results when VFIL-Tr, VFIL-noTr, or CL was implemented using either the region-based jamming model (RBM) or SNR-based jamming model (SBM). The radius of the jammer's NLB was fixed at 65.6 feet in both jamming models. Further, we set the SNR threshold, γ_0, as 1.1 in the SNR-based model.

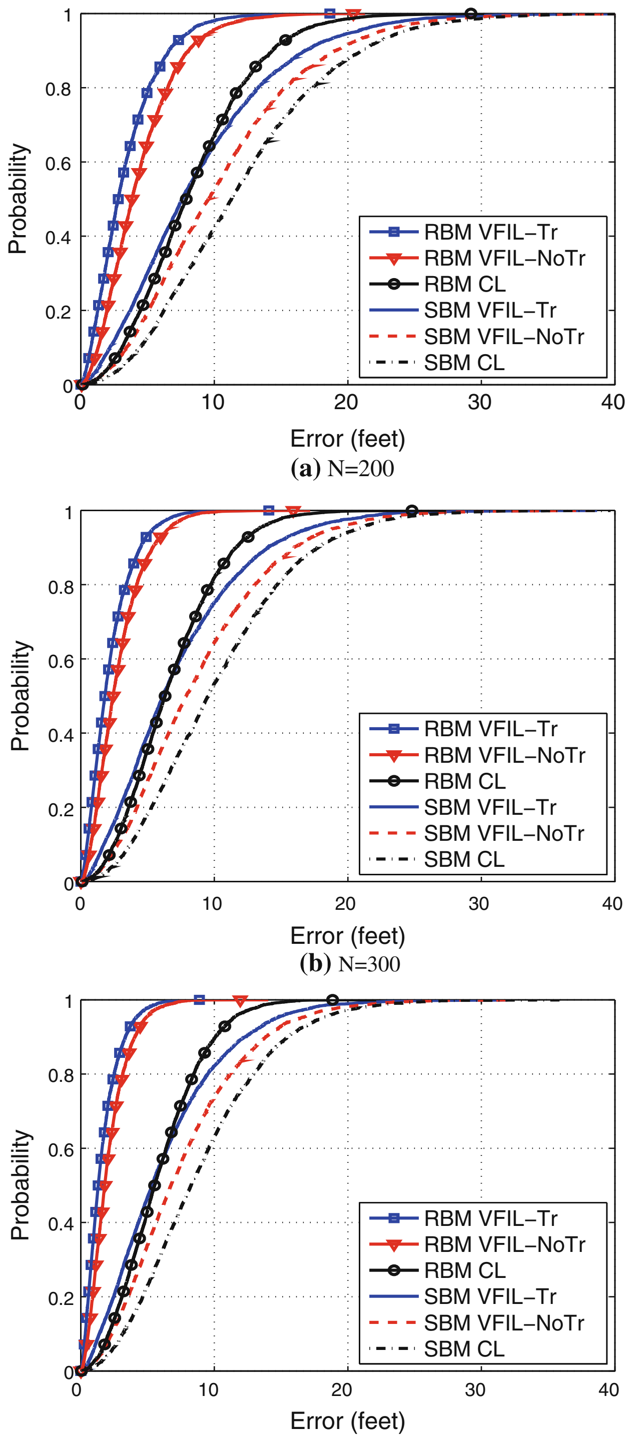
Overall, we observed that the higher the node density, the better the localization accuracy. Among all algorithms, VFIL-Tr achieves the best performance consistently with all node density setup and in both jamming models, whereas CL method performs the worst. Additionally, we found that all the localization algorithms adopting the region-based model outperform those using the SNR-based model. This is because the region based model simplifies the jamming situation, and considers a relatively regular jammed area. Such regularity in the region-based model makes the estimation of the jammer's position more accurate. Since the SNR-based model provides a better approximation of real wireless communications, the observation that all the algorithms perform better in the region-based model emphasizes the necessity of adopting the more realistic SNR-based model when studying jamming-related problems.

We compared the median error, e.g., the estimation error at the 50th percentile, for each algorithm in each experiment setup. Figure 8(a) shows the results when the number of nodes is 200, e.g., $N = 200$. As far as the region-based model is concerned, the median estimation error of VFIL-Tr is 2.8 ft, which means 50% of the time VFIL-Tr can estimate the jammer's location with an error less than 2.8 ft. In comparison, CL can only achieve a median estimation error of 7.9 ft. Thus, the VFIL-Tr outperforms CL by 65%. Similarly, VFIL-NoTr has a median error of 3.8 ft, exhibiting a performance improvement of 52% compared with CL.

Under the SNR-based model, VFIL-Tr improves the localization accuracy by 34% with an median error of 7.5 ft versus 11.3 ft for CL, whereas VFIL-NoTr improves the localization accuracy by 15% with a median error of 9.6 ft.

Furthermore, when the number of nodes increases to $N = 300$, as presented in Fig. 8(b), under the region based model the improvement of the localization accuracy in terms of the median error is 71% between CL and VFIL-Tr methods (from 6.3 ft to 1.8 ft) and is 62% between CL and VFIL-NoTr (from 7.9 ft to 2.4 ft). Under the SNR-based model, the accuracy improvement of the median error between CL and VFIL-Tr methods is 35% (from 9.6 ft to 6.2 ft) and 18% between CL and VFIL-NoTr (from 9.6 ft to 7.9 ft).

Finally, in Fig. 8(c) where the node density increases to $N = 400$, under the region-based model, the median error of localization improves by 76% between CL and VFIL-Tr methods (from 5.5 ft to 1.3 ft) and by 67% between CL and VFIL-NoTr (from 5.5 ft to 1.8 ft), respectively.



(a) N=200

(b) N=300

(c) N=400

Fig. 8 Performance impact under different node densities when the jammer's NLB radius is fixed at 65.6 ft

Similarly, under the SNR-based model, the median errors improves by 35% between CL and VFIL-Tr methods (from 8.3 ft to 5.4 ft) and by 24% between CL and VFIL-NoTr (from 8.8 ft to 6.8 ft), respectively.

As a conclusion, compared with the CL method, the Virtual Force Iterative Localization (VFIL) approaches improve the localization accuracy, and the performance improvement increases as the node density increases.

6.3.2 *Estimation of the jammer's NLB radius*

We next studied the accuracy of the jammer's NLB radius estimated by VFIL-NoTr. Figure 9 depicts the CDFs of the jammer's NLB radius estimation errors in various simulation setups including different node densities and two jamming models. In general, we found that our VFIL approaches can estimate the jammer's NLB radius more accurately in the region-based jamming model than the one obtained under the SNR-based jamming model. In particular, in Fig. 9(a, c), we observed that for the region-based jamming model, a higher node density yields smaller estimation errors, and the accuracy of the radius estimation improves by 22% at 50th percentile (from 0.98 ft to 0.77 ft) when $N = 400$ compared with the one obtained when $N = 200$.

In the SNR-based model, however, we observed that the jammer's NLB radius estimation is less sensitive to node density changes. This is because as the node density changes, the change rates of the number of boundary nodes are different in two jamming models. When the region-based model is concerned, the number of boundary nodes will increase proportionally to the node density changes. Since the number of boundary nodes is determined by the relatively regular jammed area and the node distribution during the deployment (e.g., uniform distribution). However, the less regular jammed area in the SNR-based model will make the number of boundary nodes less deterministic. Thus, even when the node density increases, the number of boundary nodes may not increase in the SNR-based model.

Further, Fig. 9(b, d) present the CDFs of the jamming-radius estimation errors using two jamming radii, 65.6 ft and 92.8 ft, in 200-node topologies. We found that a larger jamming NLB circle produces better estimations. This is because a larger jamming NLB circle causes a larger number of nodes to be jammed, which provides additional topology constraints for radius estimating, e.g., to ensure that the estimated node state confirms with the real state. Because of the increased jamming NLB radius, the median errors of the jammer's NLB radius estimation are improved by about 20% in both jamming models.

6.3.3 *Impact of the jammer's NLB radius*

After examining the estimation accuracy of the jammer's NLB radius, we studied the impact of different jammer's NLB radius, R_J, on the location estimation accuracy. In this set of experiments, we fixed the number of nodes to 200

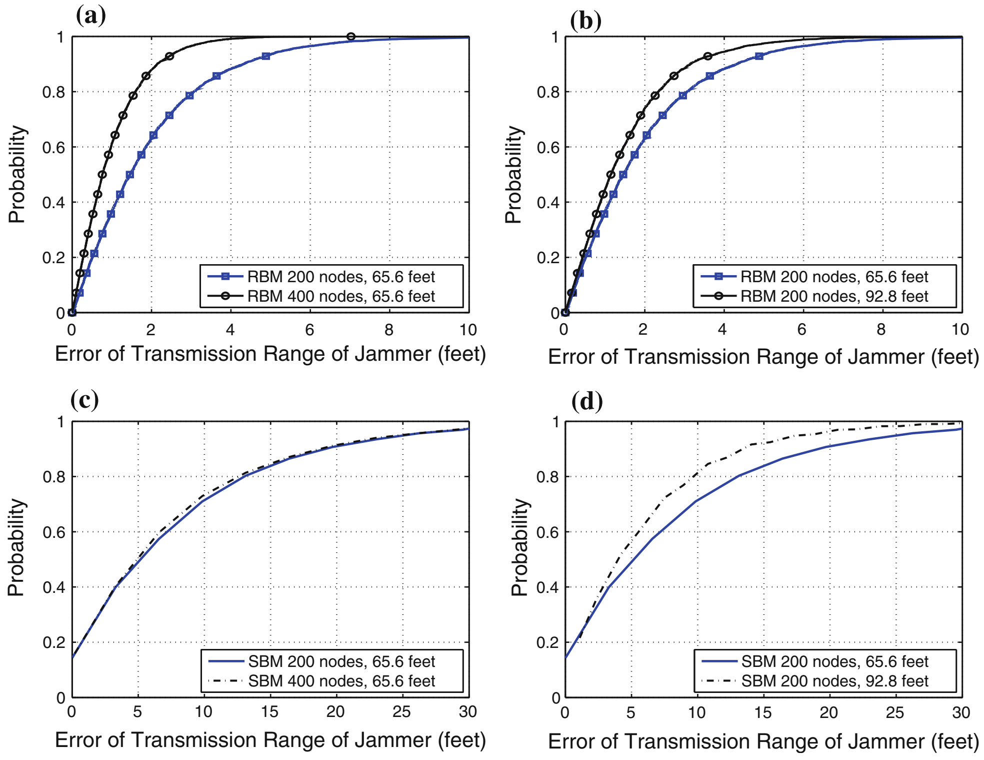


Fig. 9 Error CDFs when estimating the jammer's NLB radius using VFIL-NoTr

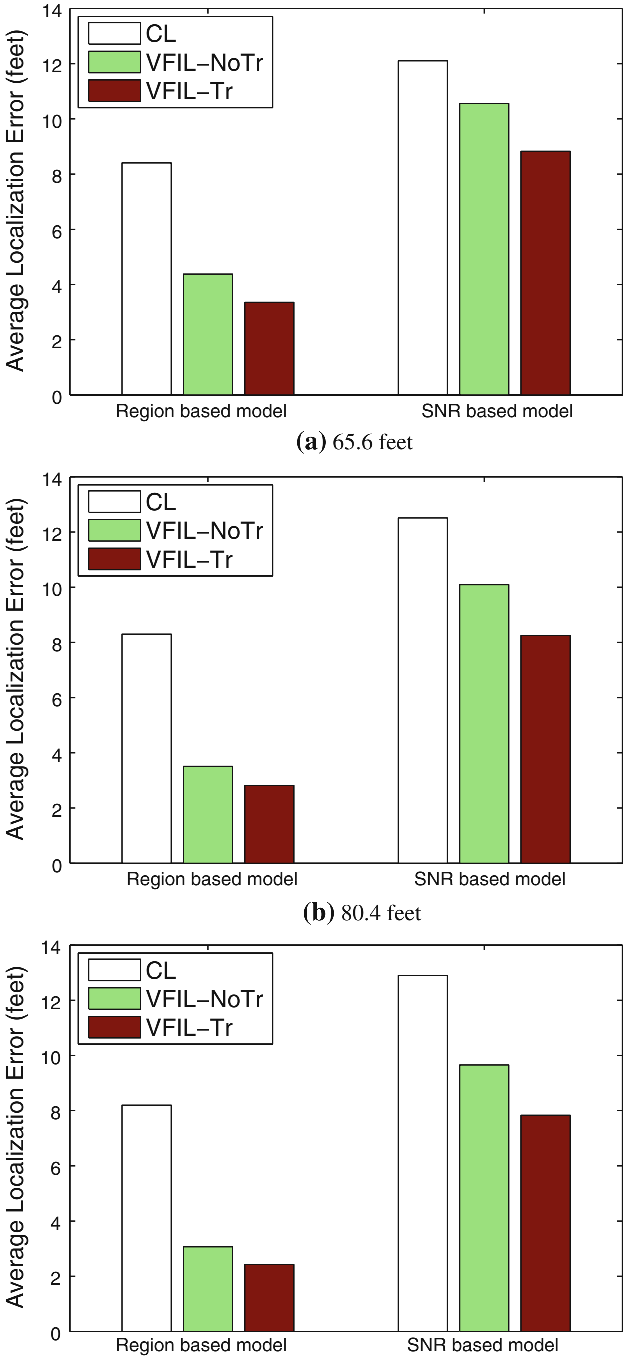
and set the jammer's NLB radius to 65.6, 80.4, and 92.8 ft, respectively. Figure 10 presents the resulted average localization errors of VFIL and CL algorithms using both the region-based and SNR-based models respectively. In both jamming models, we observed the similar trend of localization performance as the one mentioned in Sect. 6.3.1: VFIL-Tr achieves the best performance regardless of the values of jammer's NLB radii, while CL performs the worst; and our VFIL methods perform better using the region-based model than using the SNR-based model. In particular, when the jammer's NLB radius is 65.6 ft, under the region-based model, the mean error of VFIL-Tr has an improvement of 70% over the one of the CL method (from 8.4 ft to 3.4 ft), whereas the VFIL-NoTr has a corresponding improvement of 61% (from 8.4 ft to 4.4 ft). Under the SNR-based model, the mean error of VFIL-Tr improves by 38% compared with the one in the CL method (from 12.1 ft to 8.8 ft), and the mean error of VFIL-NoTr improves by 22%(from 12.1 ft to 10.6 ft).

Additionally, we observed that the VFIL methods leveraging the SNR-based model are less sensitive to the jamming range than the one using the region-based model. In particular, when the jammed range increases from 65.6 ft to 92.8 ft, as depicted in Fig. 10, the mean errors of both VFIL-Tr and VFIL-NoTr using the region-based model decrease by 30%, dropping from 3.4 ft to 2.4 ft for VFIL-Tr and from 4.4 ft to 3.1 ft for VFIL-NoTr, respectively. In cases of the SNR-based model, the mean error of VFIL-Tr decreases from 8.8 ft to 7.8 ft, resulting in an improvement of 11%, and the one of VFIL-NoTr is reduced from 10.6 ft to 9.7 ft, leading to an improvement of 8%.

6.3.4 *Impact of jammer's position*

In this round of experiments, we investigated the impact of the jammer's position on the localization performance by randomly placing the jammer anywhere in the network, instead of restricting the jammer within the center of the simulation area. We cycled through all three algorithms, VFIL-Tr, VFIL-NoTr, and CL, in three network scenarios, whereby ($N = 200$, $r_j = 65.6$ ft), ($N = 200$, $r_j = 80.4$ ft), and ($N = 300$, $r_j = 65.6$ ft), respectively. The resulted error CDF curves are plotted in Fig. 11, from which we can draw the same conclusion: VFIL-Tr still achieves the best performance no matter which type of jamming model is used.

However, compared Fig. 11(a) with Fig. 8(a) which has the same network density and jammer's NLB radius but different jammer placement constraint (in the center of networks), we observed that the performance of all jammer localization algorithms degrades as a result of random placement. This is because random positioning will include cases where the jammer is placed close to the edge of the network. Under such situations, the jammed nodes and boundary nodes are resided at one side of the jammer, causing the estimated jammer location biased towards the



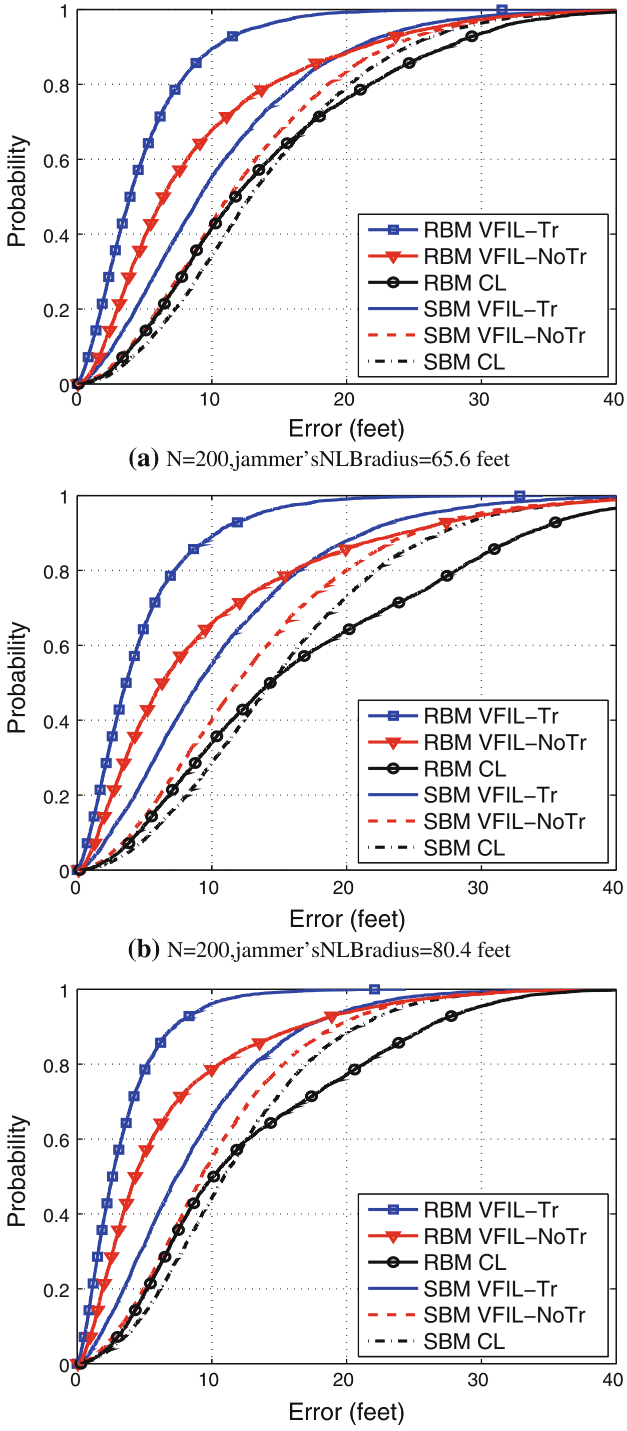
(a) 65.6 feet

(b) 80.4 feet

(c) 92.8 feet

Fig. 10 Impact of different radii of jammer's NLB circle when the number of network nodes is 200

side where the jammed nodes and the boundary nodes locate. Such performance degradation will not impose much concern in practice, because a jammer is less likely to place itself on the edge of the network, afraid of not fulfilling its objective to disrupt the communication ability of as many nodes as possible.



(a) N=200,jammer'sNLBradius=65.6 feet

(b) N=200,jammer'sNLBradius=80.4 feet

(c) N=300,jammer'sNLBradius=65.6 feet

Fig. 11 Impact of jammer's position in the network

Additionally, we noticed the similar performance trend in Fig. 11(b, c), indicating the generality of our observations across various node densities and jammer's NLB radii.

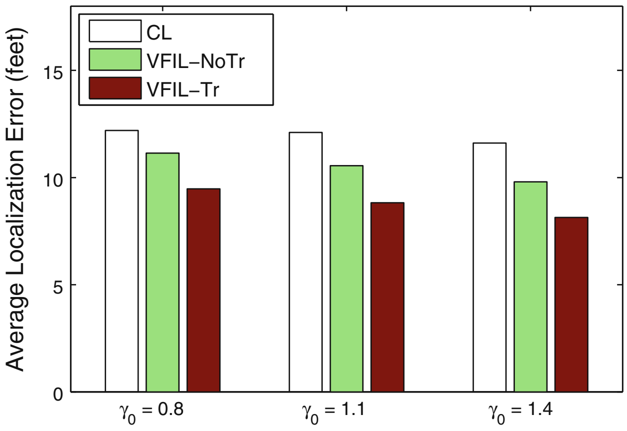


Fig. 12 Impact of different SNR threshold when the number of nodes is 200 and the radius of the jammer's NLB is set to 65.6 ft

6.3.5 *Impact of the decodable SNR threshold*

Finally, we examined the impact of the decodable SNR threshold on the localization algorithms that adopt the SNR-based jamming model. We studied their performance in a network with 200 nodes and a jammer with a NLB radius of 65.6 ft. Figure 12 depicts the average localization error estimated by VFIL and CL methods when the decodable SNR threshold γ_0 is set to 0.8, 1.1, and 1.4, respectively. Again, we discovered that VFIL-Tr performs best for all decodable-SNR threshold values, whereas CL performs the worst. Additionally, as the value of γ_0 increases, we found that the localization accuracy improves for all three algorithms. In particular, when γ_0 is increased from 0.8 to 1.1 and then to 1.4, the average localization error of CL decreases from 12.2 ft to 12.1 ft and then to 11.6 ft, which is an improvement of 5%. Compared with CL method, the VFIL methods exhibit a larger performance improvement: 15% (from 9.5 ft to 8.1 ft) for VFIL-Tr and 12% (from 11.1 ft to 9.8 ft) for VFIL-NoTr. The performance improvement created by increased γ_0 can be explained as the following. Given the same number of network nodes and the jammer's NLB radius, a bigger γ_0 makes a larger number of nodes to be jammed, which increases the number of constraints and consequently improves the localization accuracy.

7 Conclusion

In this paper, we explored the task of diagnosing jamming attacks. In particular, we focused on localizing the jammer after a jamming attack is identified. We formulated the jamming effects in two jamming models, region-based and signal-to-noise-ratio(SNR)-based. The region-based model applies the free space propagation model to the received jamming signal power, whereas the SNR-based model utilizes the signal-to-noise-ratio at the receiver to better capture the effects of the jammer. Further, we categorized the network nodes into three states under jamming: `JAMMED`, `BOUNDARY`, and `UNAFFECTED`. By exploiting the state of each network node, we developed virtual-force iterative localization (VFIL) algorithm that utilizes the network topology to iteratively adjust the estimated location of a jammer until it reaches a close approximate of the true location. VFIL does not depend on the measuring signal strength inside the jammed region, and thus it is not affected by the disturbed network communication caused by jamming. VFIL has two variants: VFIL-Tr assumes the NLB of the jammer is known, whereas VFIL-NoTr needs to estimate the NLB of the jammer when estimating the jammer's location.

Our experiments involving MicaZ motes show that the SNR-based model is a realistic jamming model in practice. Since the region-based model is widely used in many literatures, we evaluated localization algorithms using both models. Further, we conducted extensive simulation to study the impact of various network factors on the performance of our virtual-force iterative approach under both jamming models. Those factors include network node densities, jammer's NLB radius, and jammer's positions in the network. Our simulation results have shown that the virtual-force iterative approach is effective in localizing the jammer with high accuracy and outperforms the existing centroid-based methods. Additionally, we observed that all localization algorithms exhibit better performance using region-based jamming model, emphasizing the importance of adopting a realistic jamming model to better capture the algorithm performance.

Acknowledgments Preliminary results of this paper have been presented in part in IEEE SPEUCS 2007 [35] and IEEE PWN 2009 [36]. The work was supported in part by National Science Foundation Grants CNS-0845671 and CNS-0954020.

References

1. Want, R., Hopper, A., Falcao, V., & Gibbons, J. (1992). The active badge location system. *ACM Transactions on Information Systems, 10*(1), 91–102.
2. Bahl, P., & Padmanabhan, V. N. (2000). 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.
3. He, T., Huang, C., Blum, B. M., Stankovic, J. A., & Abdelzaher, T. (2005). Range-free localization and its impact on large scale sensor networks. *ACM Transactions on Embedded Computing Systems, 4*, 877–906.
4. Chen, Y., Francisco, J., Trappe, W., & Martin, R. P. (2006). 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).

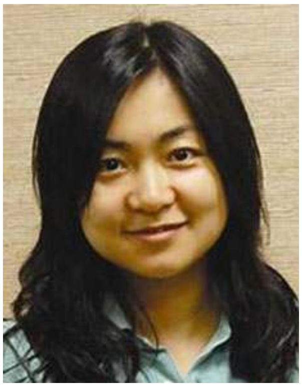
5. Kleisouris, K., Chen, Y., Yang, J., & Martin, R. P. (2008). The impact of using multiple antennas on wireless localization. In *Proceedings of the fifth annual IEEE communications society conference on sensor, mesh and ad hoc communications and networks (SECON)*. June 2008.
6. Proakis, J. G. (2000). *Digital communications* (4th ed.). Singapore: McGraw-Hill.
7. Schleher, C. (1999). *Electronic warfare in the information age*. Norwood: MArtech House.
8. Wood, A., Stankovic, J., & Son, S. (2003). JAM: A jammed-area mapping service for sensor networks. In *24th IEEE real-time systems Symposium*. pp. 286–297.
9. Xu, W., Trappe, W., Zhang, Y., & Wood, T. (2005). 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*. pp. 46–57.
10. Çakiroğlu, M., & Özcerit, A. T. (2008). 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), pp. 1–8.
11. Mraleedharan, R., & Osadciw, L. A. (2006). Jamming attack detection and countermeasures in wireless sensor network using ant system," in *Proceedings of the SPIE in wireless sensing and processing*, (Vol. 6248). p. 62480G.
12. Chiang, J. T., & Hu, Y.-C. (2007). 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, pp. 346–349.
13. Noubir, G. & Lin, G. (2003). Low-power DoS attacks in data wireless lans and countermeasures. *SIGMOBILE Mobile Computing and Communications Review, 7*(3), 29–30.
14. Xu, W., Trappe, W., & Zhang, Y. (2007). Channel surfing: Defending wireless sensor networks from interference. In *IPSN '07: Proceedings of the 6th international conference on information processing in sensor networks*. pp. 499–508.
15. Navda, V., Bohra, A., Ganguly, S., Izmailov, R., & Rubenstein, D. (2007). Using channel hopping to increase 802.11 resilience to jamming attacks. In *IEEE infocom minisymposium*. May 2007, pp. 2526–2530.
16. Khattab, S., Mosse, D., & Melhem, R. (2008). 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), pp. 1–10.
17. Ma, K., Zhang, Y., & Trappe, W. (2005). Mobile network management and robust spatial retreats via network dynamics. In *Proceedings of the the 1st international workshop on resource provisioning and management in sensor networks (RPMSN05)*.
18. Xu, W., Trappe, W., & Zhang, Y. (2008). 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, pp. 203–213.
19. Cagalj, M., Capkun, S., & Hubaux, J. (2007). Wormhole-based anti-jamming techniques in sensor networks. In *IEEE transactions on mobile computing*. January 2007, pp. 100–114.
20. Wood, A. D., Stankovic, J. A., & Zhou, G. (2007). 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)*.
21. Priyantha, N., Chakraborty, A., & Balakrishnan, H. (2000). The cricket location-support system. In *Proceedings of the ACM international conference on mobile computing and networking (MobiCom)*. Aug 2000, pp. 32–43.
22. Ward, A., Jones, A., & Hopper, A. (1997). A new location technique for the active office. *IEEE Personal Communications, 4*(5), 42–47.
23. Chen, Y., Kleisouris, K., Li, X., Trappe, W., & Martin, R. P. (2006). 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. Chandrasekaran, G., Ergin, M. A., Yang, J., Liu, S., Chen, Y., Gruteser, M., & Martin, R. (2009). 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. Hightower, J., Borriello, G., & Want, R. (2000). 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. Enge, P., & Misra, P. (2001). *Global positioning system: Signals, measurements and performance*. Ganga-Jamuna Pr.
27. He, T., Huang, C., Blum, B., Stankovic, J. A., & Abdelzaher, T. (2003). 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)*.
28. Bulusu, N., Heidemann, J., & Estrin, D. (2000). Gps-less low-cost outdoor localization for very small devices. *IEEE Personal Communications Magazine, 7*, 28–34.
29. Niculescu, D., & Nath, B. (2001). Ad hoc positioning system (APS). In *Proceedings of the IEEE global telecommunications conference (GLOBECOM)*. pp. 2926–2931.
30. Shang, Y., Ruml, W., Zhang, Y., & Fromherz, M. P. J. (2003). 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. Pelechrinis, K., Koutsopoulos, I., Broustis, I., & Krishnamurthy, S. (2009). Lightweight jammer localization in wireless networks: System design and implementaion. In *Proceedings of IEEE global telecommunication cofference (GLOBECOM)*. December 2009.
32. Deng, J., Han, R., & Mishra, S. (2004). Intrusion tolerance and anti-traffic analysis strategies for wireless sensor networks. In *DSN '04: Proceedings of the 2004 international conference on dependable systems and networks*. Washington, DC, USA: IEEE Computer Society, p. 637.
33. Kleppner, D., & Kolenkow, R. J. (1973). *An introduction to mechanics*. New York: McGraw-Hill.
34. "Crossbow Technology Inc." white paper available at http://www.xbow.com.
35. Xu, W. (2007). On adjusting power to defend wireless networks from jamming. In *Proceedings of the first workshop on the security and privacy of emerging ubiquitous communication systems (SPEUCS)*.
36. Liu, H., Xu, W., Chen, Y., & Liu, Z. (2009). Localizing jammers in wireless networks. In *Proceedings of IEEE PerCom international workshop on pervasive wireless networking (IEEE PWN)*.

Author Biographies

Hongbo Liu is a Ph.D. candidate of the Electrical and Computer Engineering Department at Stevens Institute of Technology. His research interests include information security & privacy, wireless localization and location based services (LBS), wireless and sensor networks. He is currently working in the Data Analysis and Information SecuritY (DAISY) Lab with Prof. Yingying Chen. He got his Master degree in communication engineering from Department of Communication and Information Engineering of University of Electronic Science and Technology of China in 2008. He received his Bachelor's degree from Department of Communication and Information Engineering of University of Electronic Science and Technology of China, China, in 2005.

Zhenhua Liu is a Ph.D. candidate of Computer Science and Engineering Department at University of South Carolina. His research interests include anti-jamming defense, and location privacy in sensor networks. He is currently working in the Arena for Research on Emerging Networks and Applications (ARENA) lab with Prof. Wenyuan Xu. He received his Bachelor degree of Electronics and Information Engineering from Department of Information Science and Technology in Central South University of China in 2006.

Yingying Chen received her Ph.D. degree in Computer Science from Rutgers University. She is currently an assistant professor in the Department of Electrical and Computer Engineering at Stevens Institute of Technology. Her research interests include wireless and system security and privacy, wireless networking, and distributed systems. She has coauthored the book Securing Emerging Wireless Systems and published extensively in journal and conference papers. Prior to joining Stevens Institute of Technology, she was with Bell Laboratories and the Optical Networking Group, Lucent Technologies. She received the IEEE Outstanding Contribution Award from IEEE New Jersey Coast Section each year 2005–2009. She is the recipient of the NSF CAREER award. She is also the recipient of the Best Technological Innovation Award from the International TinyOS Technology Exchange in 2006, as well as the Best Paper Award from the International Conference on Wireless On-demand Network Systems and Services (WONS) in 2009.



Wenyuan Xu received her Ph.D. degree in Electrical and Computer Engineering from Rutgers University in 2007. She is currently an assistant professor in the Department of Computer Science and Engineering, University of South Carolina. Her research interests include wireless networking, network security and privacy. Dr. Xu is a coauthor of the book Securing Emerging Wireless Systems: Lower-layer Approaches, Springer, 2009. She received NSF Career Award in 2009. She has served on the technical programs for several IEEE/ACM conferences on wireless networking and security.