Error Minimizing Jammer Localization Through Smart Estimation of Ambient Noise

Zhenhua liu*, Hongbo Liu†, Wenyuan Xu* and Yingying Chen†
*Dept. of Computer Science and Engineering, University of South Carolina, Columbia, SC
Email: {liuz,wyxu}@cse.sc.edu
†Dept. of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ
Email: {hliu3,yingying.chen}@stevens.edu

***Abstract*—Jammer can jeopardize the dependability of wireless networks, and jammer's position information allows the network to cope with jamming leveraging varieties of defense strategies. Thus, in this paper, we address the problem of localizing jammer. Prior work relies on indirect measurements derived from jamming effects, which makes it difficult to accurately localize jammer. We localize jammer by directly using the strength of jamming signals (JSS). Estimating JSS is challenging as they may be embedded in other signals. As such, we devise an estimation scheme based on ambient noise floor and validate it with real world experiments. To improve localization accuracy, we define an evaluation feedback metric to quantify the estimation errors and formulate jammer localization as a non-linear optimization problem, whose optimal solution approaches jammer's true position. We exploit a heuristic search based algorithm for approximating the global optimal solution, and our extensive simulation shows that our error-minimizing-based algorithm outperforms existing algorithms.**

***Keywords*-Jamming, Radio interference, Localization**

I. Introduction

The increasing pervasiveness of wireless technologies, combined with the limited number of unlicensed bands, will continue to make the radio environment crowded, leading to unintentional radio interference across devices with different communication technologies that share the same spectrum, e.g., cordless phones, Wi-Fi network adapters, Bluetooth headsets, microwave ovens, and ZigBee-enabled appliances. Meanwhile, the emerging of software defined radios has enabled adversaries to build intentional jammers to disrupt network communication with little effort. Regardless of unintentional interference or malicious jamming, jammers/interferers may have detrimental impact on network performance – both can be referred as jamming. To ensure the successful deployment of pervasive wireless networks, it is crucial to localize jammers, since the locations of jammers allow a better physical arrangement of wireless devices that cause unintentional radio interference, and enable a wide range of defense strategies for combatting malicious jamming attackers.

In this work, we focus on identifying the location of a jammer. Current jammer-localization approaches mostly rely on parameters derived from the affected network topology, such as packet delivery ratios [11], neighbor lists [6], and nodes' hearing ranges [7]. The use of these indirect measurements derived from jamming effects makes it difficult to accurately localize jammer's position. We seek to localize jammers by directly using the strength of jamming signals (JSS).

Localizing a jammer utilizing jamming signal strength (JSS) is appealing but also challenging. First, the jamming signals are embedded in the regular network traffic. The commonly used received signal strength (RSS) measurement associated with a packet does not correspond to JSS. To overcome the challenges, we devised a scheme that can effectively estimate JSS utilizing the measurement of the ambient noise floor, which is readily available from many commodity devices (e.g., MicaZ motes). Our experiments using MicaZ motes with multiple sender-receiver pairs confirmed the feasibility of estimating the strength of jamming signals under various network traffic conditions.

Second, to improve the accuracy of wireless device localization, the existing RSS-based localization algorithms [1], [2] often rely on obtaining a site survey of radio RSS fingerprints during the training phase. Obtaining a JSS site survey is infeasible for jamming, since a jammer does not cooperate with localization algorithms and the jammer's transmission power is unknown. To improve the accuracy of jamming localization based on JSS, we exploited the random shadowing phenomena in radio propagation and defined an evaluation metric that can quantify the accuracy of the estimated locations. Utilizing such an evaluation metric, we formulated the jammer localization problem as an error minimizing problem and used a simulated annealing algorithm for finding the best solution. Our experiments showed that our error-minimizing-based algorithm outperforms existing jammer localization approaches by 71.82%.

We organize the remainder of the paper as follows. We discuss the related work in Section II and introduce our threat model in Section III. In Section IV, we formulate the jammer localization problem as a non-linear optimization problem. Then, we address the challenge of estimating JSS of a jammer and present our real world experiment validation in Section V. In Section VI, we introduce the simulated annealing algorithm for solving the optimization problem. Finally, we present the simulation validation of our error-minimizing-based localization approaches in Section VII and conclude in Section VIII.

II. Related Work

Countermeasures for coping with jammed regions in wireless networks have been widely investigated. The use of error correcting codes [10] is used to increase the likelihood of decoding corrupted packets. Several other passive approaches are proposed to resume network communication without actively eliminating jammers, which include channel surfing/hopping [4], [9], [17], whereby wireless devices change their working channel to escape from jamming, spatial retreats [8], whereby wireless devices move out of a jammed region geographically, and an anti-jamming timing channel [18], whereby data are communicated via a covert timing channel that is built on a failed-packet-delivery event. Instead of trying to survive in the presence of jamming, we aim at obtaining the locations of jammers to facilitate active defense strategies.

Wireless localization has been an active area, attracting much attention. Infrared [14] and ultrasound [12], [15] are employed for localization, both of which need to deploy a specialized infrastructure for localization. Received signal strength (RSS) [1], [2] is an attractive approach because it can reuse the existing wireless infrastructure. However, aforementioned RSS-based work [1], [2] focused on localizing regular wireless devices and are inapplicable to localize jammers. This is because existing RSS-based methods are built upon the premise that the RSS of various wireless transmitters can be easily measured. Obtaining the strength of jamming signals is a challenging task mainly because jamming signals are embedded in signals transmitted by regular wireless devices.

Identifying jammers' locations becomes an important strategy to cope with jamming. Pelechrinis et al. [11] proposed to localize a jammer by measuring packet delivery ratio (PDR) and performing gradient descent search. Liu *et al.* [6] utilized the network topology changes caused by jamming attacks and estimated the jammer's position by introducing the concept of virtual forces. Sun *et al.* [13] utilized the minimum circle covering technique to form an approximate jammed region for estimating the position of the jammer. However, this work is based on a region-based jamming model, which may not be available in real world jamming scenarios. Liu *et al.* [7] exploited the changes of a node's neighbors (i.e., hearing range changes) caused by jamming attacks to localize a jammer using least square calculations. All of these studies utilized indirect measurements derived from jamming effects to estimate the location of jammers, making it difficult to accurately localize jammers.

Our work is different from prior work in that we formulate the jammer localization problem under an error minimizing framework, aiming to achieve high localization accuracy. Our work localizes a jammer by utilizing the strength of jamming signals directly through measuring the readily available ambient noise floor using commodity wireless devices.

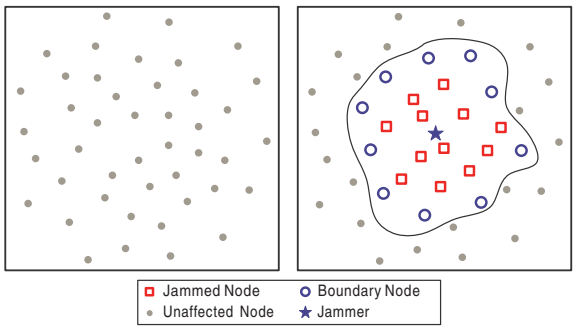


Fig. 1. Illustration of the network nodes classification due to jamming: [Left] prior to jamming and [right] after jamming.

III. Threat Model and Jamming Effect

We focus on one *constant jammer* that continually emits radio signals, regardless of whether the channel is idle or not. Such a jammer can be unintentional radio interferer that is always active or malicious jammer that keeps disturbing network communication. We assume such a jammer is equipped with an omnidirectional antenna and transmits at the same power level, so it has a similar jamming range in all directions. Identifying jammer's position will be performed after the jamming attack is detected, and we assume the network is able to identify jamming attacks, leveraging the existing jamming detection approaches [16], [19].

We first discuss which nodes can participate in jammer localization: the ones that can measure and report JSS. To identify those nodes, we 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. Thus, the network nodes could be classified based on the changes of neighbors caused by jamming. We define that node B is a neighbor of node A if A can receive messages from B prior to jamming. The network nodes can be classified into three categories according to the impact of jamming: *unaffected node*, *jammed node*, and *boundary node*:

- **Unaffected node.** A node is unaffected if it can receive packets from all of its neighbors after jamming is present. This type of node is barely affected by jamming and may not yield accurate JSS measurements.
- **Jammed node.** A node is jammed if it cannot receive messages from any of the unaffected nodes. We note that this type of node can measure JSS, but cannot report their measurements.
- **Boundary node.** A boundary node can receive packets from part of its neighbors but not from all of its neighbors. Boundary nodes can not only measure JSS, but also report their measurements to a designated node for jamming localization.

Figure 1 illustrates an example of network topology changes caused by a jammer. Prior to jamming, all the nodes could receive packets from their neighbors, shown as grey dots. Once the jammer became active (shown as a star), affected nodes lost their neighbors partially or completely. The nodes marked

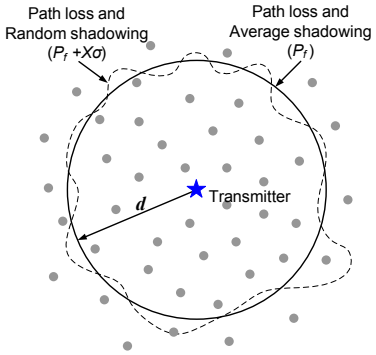


Fig. 2. The contour of RSS subject to path loss is a circle centered at the transmitter, and the contour of RSS attenuated by both path loss and shadowing is an irregular loop fluctuating around the path-loss circle.

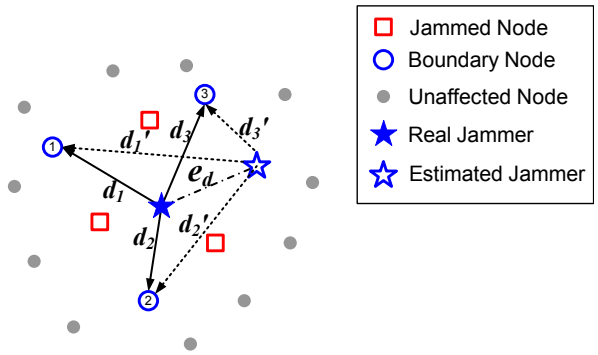


Fig. 3. Illustration of jammer localization basis. When the estimated jammer location is e_d distance away from the true location, the estimated random attenuation is biased and the corresponding standard deviation is larger than the real one.

as red squares lost all of their neighbors and became jammed nodes. The nodes depicted in blue circles are boundary nodes. They lost part of their neighbors but still maintained communication capability to a few neighbors. Finally, the rest of the nodes that remained in grey dots are unaffected nodes, and they can still receive packets from all their neighbors. Note that jammed nodes are usually those nodes located closest to the jammer, whereas the boundary nodes reside in between jammed nodes and unaffected nodes.

In summary, our jammer localization algorithms rely on boundary nodes for sampling and collecting JSS for jammer localization.

IV. Localization Formulation

Essentially, our jammer localization approach works as follows. Given a set of JSS measurements, for every estimated location, we are able to provide a quantitative evaluation feedback indicating how close it is to the true jammer's location. Although unable to adjust the estimation directly, it is possible, from a few candidate locations, to select one that is the closest to the true jammer's location with high probability, making searching for the best estimate feasible. Leveraging this idea, our jammer localization approach comprises two steps: (a) *Signal-Strength Collection.* All boundary nodes obtain JSS measurements locally. (b) *Best-Estimate Searching.* Based on the measured JSS, a designated node will first obtain a rough estimation of the jammer's position using prior work [7]. Then, it refines the estimation by searching for the jammer's position that minimizes the evaluation feedback metric.

There are several challenges associated with this search-based jammer localization approach leveraging JSS:

1) What metric is appropriate to provide a feedback that quantifies the accuracy of jammer's location estimations?
2) How do we obtain strength of jamming signals, which may be embedded in regular transmission?
3) How do we efficiently search for the best estimate?

We address all aforementioned challenges in this work. In this section, we formulate the evaluation feedback metric and model the jammer localization as an optimization problem, and discuss JSS measurement schemes and search algorithms in Section V and Section VI.

A. Radio Propagation Basics

The intuition of the evaluation feedback metric of estimated localization comes from the basics of radio propagation over a medium with impedances. To illustrate our jammer localization approach, we use the log-normal shadowing model [3], which captures both path loss and shadowing. Let P_f be the received signal strength subject to path loss attenuation only, and let the power of a transmitted signal be P_t. The received signal power in `dBm` at a distance of d can be modeled as the sum of P_f and a variance caused by shadowing and other random attenuation,

$$P_r = P_f + X_\sigma \tag{1}$$

$$P_f = P_t + K - 10\eta \log_{10}(d), \tag{2}$$

where X_σ, caused by shadowing, is a Gaussian zero-mean random variable with standard deviation σ, K is a unitless constant which depends on the antenna characteristics and the average channel attenuation, η is the Path Loss Exponent (PLE), and d_0 is a reference distance. In a free space, η is 2 and X_σ is always 0.

Figure 2 illustrates contours of a constant received strength and the relationship between shadowing and path loss. The attenuation caused by shadowing at any single location, distance d away from the transmitter, may exhibit variation (denoted by dash curves in Figure 2); the average attenuation and average signal strength (denoted by the solid-line circle centered at the transmitter) are roughly the same [3].

B. Localization Evaluation Metric

We quantify the evaluation feedback metric e_z as the estimated standard deviation of the random attenuation X_σ, as if the jammer was indeed located at the estimated location.

Calculating e_z. Assume a jammer J located at (x_J, y_J) starts to transmit at the power level of P_J, and m nodes located at $\{(x_i, y_i)\}_{i\in[1,m]}$ become boundary nodes. To calculate e_z, each boundary node will first measure JSS locally (the details will be discussed in Section V), and we denote the JSS

Algorithm 1 Evaluation feedback metric calculation.

procedure EVALUATEMETRIC($\hat{\mathbf{z}}, \mathbf{p}$)
  for all $i \in [1, m]$ **do**
    $\hat{X}_{\sigma_i} = P_{r_i} - P_{f_i}(\hat{\mathbf{z}})$
  end for
  $e_z = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\hat{X}_{\sigma_i} - \hat{\bar{X}}_{\sigma})^2}$
end procedure

measured at boundary node i as P_{r_i}. Let the current estimation of the jammer J's location and the transmission power be

$$\hat{\mathbf{z}} = [\hat{x_J}, \hat{y_J}, \hat{P_J} + \hat{K}].$$

Then, we can estimate P_{f_i}, the JSS subject to path loss only at boundary node i as

$$\begin{aligned} P_{f_i}(\hat{d}_i) &= \hat{P}_J + \hat{K} - 10\eta \log_{10}(\hat{d}_i) \\ \hat{d}_i(\hat{\mathbf{z}}) &= \sqrt{(\hat{x}_J - x_i)^2 + (\hat{y}_J - y_i)^2} \end{aligned} \tag{3}$$

The random attenuation (shadowing) between the jammer J and boundary node i can be estimated as

$$\hat{X}_{\sigma_i} = P_{r_i} - P_{f_i}(\hat{d}_i). \tag{4}$$

The evaluation feedback metric for the estimation $\hat{\mathbf{z}}$ is the standard deviation of estimated $\{\hat{X}_{\sigma_i}\}_{i\in[1,m]}$,

$$e_z(\hat{\mathbf{z}}, \mathbf{p}) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\hat{X}_{\sigma_i} - \hat{\bar{X}}_{\sigma})^2}, \tag{5}$$

where $\bar{X}_{\sigma}$ is the mean of X_{σ_i}. One of the biggest advantages of this definition is that by subtracting $\bar{X}_{\sigma}$, e_z is only affected by $(\hat{x}_J, \hat{y}_J)$ and is independent of the estimated jamming power $\hat{P}_J + \hat{K}$.

The property of e_z. The definition of e_z has the following property. When the estimated jammer's location equals the true value, e_z is the real standard deviation of X_{σ}. When there is an estimation error (the estimated location is e_d distance away from the true location), e_z will be biased and will be larger than the real standard deviation of X_{σ}. The level of bias is affected by e_d: the larger e_d is, the bigger the estimated standard deviation of X_{σ} will likely be. The detailed relationship between e_z and e_d will be discussed in Section VI-A.

Here, we illustrate the property of e_z using the example depicted in Figure 3, where 3 boundary nodes are $\{d_1, d_2, d_3\}$ distance away from the jammer J. Let $\{X_{\sigma_1}, X_{\sigma_2}, X_{\sigma_3}\}$ be the true shadowing attenuation between the boundary nodes and J. Applying the true location of J to Eq. (5), $e_z(\mathbf{z}, \mathbf{p})$ equals to the true standard deviation of X_{σ_i}. If the estimated location of J is at (x'_J, y'_J), the estimated distances between the three boundary nodes to J are $\{d'_1, d'_2, d'_3\}$. In this example, $d'_1 > d_1$, $d'_2 > d_2$, and $d'_3 < d_3$; and $X'_{\sigma_1} > X_{\sigma_1}$, $X'_{\sigma_2} > X_{\sigma_2}$, and $X'_{\sigma_3} < X_{\sigma_3}$. Thus, the e_z corresponding to (x'_J, y'_J) is larger than the one calculated using the true location of J.

We note that the relationship between e_z and e_d is independent to the distribution of X_{σ}. Thus, in cases where the log-normal shadowing model does not match with the real radio propagation, e_z can still provide quantitative feedback of e_d.

Problem formulation. We can thus formulate the jammer localization problem as an optimization problem,

Problem 1:

$$\begin{aligned} \underset{\mathbf{z}}{\text{minimize}} \quad & e_z(\mathbf{z}, \mathbf{p}) \\ \text{subject to} \quad & \mathbf{p} = \{P_{r_1}, \ldots, P_{r_m}\}; \end{aligned}$$

where $\mathbf{z}$ are the unknown variable vector of the jammer, e.g., $\mathbf{z} = [x_J, y_J, P_J + K]$, and $\{P_{r_i}\}_{i\in[1,m]}$ are the JSS measured at the boundary nodes $\{1, \ldots, m\}$. As we will show in Section VI-A, the estimated location of the jammer at which e_z is minimized, matches the jammer's true location with small estimation errors.

Algorithm 2 Acquiring the Ambient Noise Floor (ANF). ANF approximates the strength of jamming signals.

procedure MEASUREJAMMINGRSS
  $\mathbf{s} = \{s_1, s_2, ..., s_n\}$ = MeasureRSS()
  if var($\mathbf{s}$) < varianceThresh **then**
    $\mathbf{s}_a = \mathbf{s}$
  else
    $JssThresh = \min(\mathbf{s}) + \alpha[\max(\mathbf{s}) - \min(\mathbf{s})]$ ▷ $\alpha \in [0, 1]$
    $\mathbf{s}_a = \{s_i | s_i < JssThresh,\ s_i \in \mathbf{s}\ \}$
  end if
  return mean($\mathbf{s}_a$)
end procedure

V. MEASURING JAMMING SIGNALS

Received signal strength (RSS) is one of the most widely used measurements in localization. For instance, a WiFi device can estimate its most likely location by matching the measured RSS vector of a set of WiFi APs with pre-trained RF fingerprinting maps [1] or with predicted RSS maps constructed based on RF propagation models [20]. However, obtaining the signal strength of jammer (JSS) is a challenging task mainly because jamming signals are embedded in signals transmitted by regular wireless devices. The situation is complicated because multiple wireless devices are likely to send packets at the same time, as jamming disturbs the regular operation of carrier sensing multiple access (CSMA). For the rest of this paper, we refer the regular nodes' concurrent packet transmissions that could not be decoded as a collision. While it is difficult, if ever possible, to extract signal components contributed by jammer or collision source, we discover that it is feasible to derive the JSS based on periodic ambient noise measurement. In the following subsections, we first present basics of ambient noise with regard to jamming signals, and then introduce our scheme to estimate the JSS. Finally, we validate our estimation schemes via real world experiments.

A. Basics of Ambient Noise Floor

In theory, *ambient noise* is the sum of all unwanted signals that are always *present*, and the ambient noise floor (ANF) is the measurement of the ambient noise. In the presence of

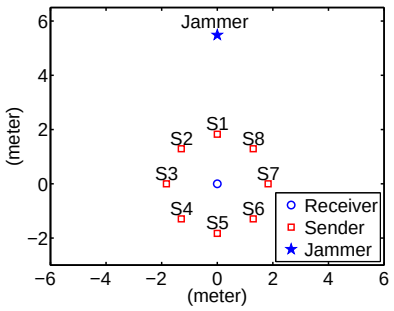


Fig. 4. An Illustration of Network Deployment.

constant jammer, the ambient noise includes thermal noise, atmospheric noise, and jamming signals. Thus, it is

$$P_N = P_J + P_W, \tag{6}$$

where P_J is the JSS, and P_W is the white noise comprising thermal noise, atmospheric noise, etc. Realizing that at each boundary node P_W is relatively small compared to P_J, the ambient noise floor can be roughly considered as JSS. Thus, estimating JSS is equivalent to deriving the ambient noise floor (ANF) at each boundary node. In this work, we consider the type of wireless devices that are able to sample ambient noise regardless of whether the communication channel is idle or busy, e.g., MicaZ sensor platforms; and derive the ANF based on ambient noise measurements.

A naive approach of estimating the ANF could be sampling ambient noise when the wireless radio is idle (e.g., neither receiving nor transmitting packets). Such a method may not work in all network scenarios, since it may result in an over-estimated ANF. For example, in a highly congested network, collision is likely to occur, and the collided signals may be treated as part of the ANF at the receiver, resulting in an inflated ANF. This is exactly the situation we want to avoid.

B. Estimating Strength of Jamming Signals

To derive JSS, our scheme involves sampling ambient noise values regardless of whether the channel is idle or busy. In particular, each node will sample n measurements of ambient noise at a constant rate, and denote them as $\mathbf{s} = [s_1, s_2, \ldots, s_n]$. The measurement set $\mathbf{s}$ can be divided into two subsets ($\mathbf{s} = \mathbf{s}_a \cup \mathbf{s}_c$).

1) $\mathbf{s}_a = \{s_i | s_i = P_J\}$: the ambient noise floor set of ambient noise measurements when only jammer is active;
2) $\mathbf{s}_c = \{s_i | s_i = P_J + P_C\}$: the combined ambient noise set of ambient noise measurements that include both jamming signals (P_J) and signals from one or more senders (P_C).

The JSS is approximately the average of ANFs, i.e., mean($\mathbf{s}_a$). In a special case where no sender has ever transmitted packets throughout the process of obtaining n measurements, $\mathbf{s}_c = \emptyset$ and $\mathbf{s}_a = \mathbf{s}$.

Motivated by these observations, we designed an algorithm (referred as Algorithm 2) to calculate the average of ANFs, which is considered it as the JSS: A regular node will take n measurements of the ambient noise measurements. It will consider the ANF as the average of all measurements if no sender has transmitted during the period of measuring; otherwise, the ANF is the average of $\mathbf{s}_a$ by filtering out $\mathbf{s}_c$ from $\mathbf{s}$. The intuition of differentiating those two cases is that if only jamming signals are present, then the variance of n measurements will be small; otherwise, the ambient noise measurements will vary as different senders happen to transmit.

The correctness of the algorithm is supported by the fact that $\mathbf{s}_a$ is not likely to be empty due to carrier sensing, and the JSS approximately equals to the average of $\mathbf{s}_a$. The key question is how to obtain $\mathbf{s}_a$. To do so, we set the upper bound (i.e., `JssThresh`) of $\mathbf{s}_c$ in Algorithm 2 as α percentage of the amplitude span of ambient noise measurements. We validate the feasibility of obtaining $\mathbf{s}_a$ using a filtering bound in next experimental subsection.

C. Experiment Validation

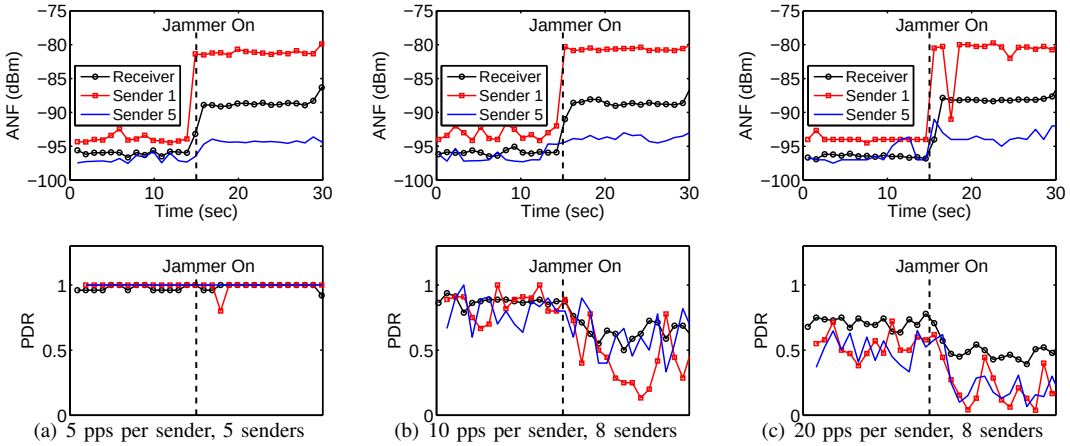
1) Methodology: To verify our algorithm that derives JSS, we conducted experiments involving one receiver and eight senders, implemented on MicaZ nodes. We deployed them on an outdoor playground, and conducted a set of experiments to evaluate the performance of Algorithm 2 at different conditions: various numbers of senders, distance to the jammer, ambient noise sample rates and network traffic.

The deployment layout is illustrated in Figure 4: a receiver at the origin $(0, 0)$, 8 senders evenly spread out at the border of circle with the radius of 1.8 meters, and the jammer at 5.5 meters to the receiver. The receiver remained silent, and each sender broadcast packets of size 120 bytes at constant rates: $\{5, 10, 20\}$ packets per second (pps). Throughout the experiments, both the receiver and senders sample ambient noise values at constant rates: the receiver samples ambient noise at a higher rate (90 samples per second) and the sender at lower rates ($\{18, 9, 5\}$ samples per second for $\{5, 10, 20\}$ pps respectively). Each ambient noise sample is an average over a period of 8 symbols (1.024 ms) as supported by MicaZ hardware, and the ANFs are estimated offline using Algorithm 2 in Matlab. Throughout our experiments, we set $\alpha = 0.4$ for filtering the ambient noise floor set $\mathbf{s}_a$.

2) Jammer and Traffic Rates: To study how well Algorithm 2 estimates the strength of jamming signals at various traffic rates, we chose three cases corresponding to different congestion levels, and used packet delivery ratios (PDRs)[1] to indicate the congestion level. The three cases are a low traffic case with 100% PDR (5 senders, each transmitting at 5 pps), a slightly-congested case with about 90% PDR (8 senders, each transmitting at 10 pps), and a highly-congested case with about 60% PDR (8 senders, each transmitting at 20 pps).

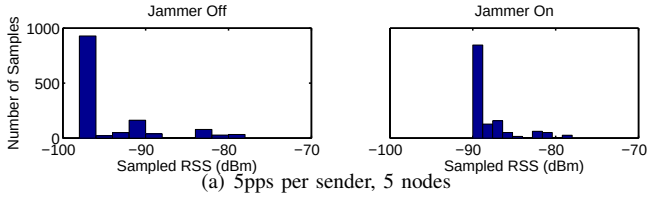
Figure 5 illustrates the time series of the estimated ANF using Algorithm 2 and the measured PDR. To make the plot recognizable, we depicted two senders (1 and 5) that are located at different distances to the jammer, and omitted others

[1]PDR is the percentage of successfully delivered packets.

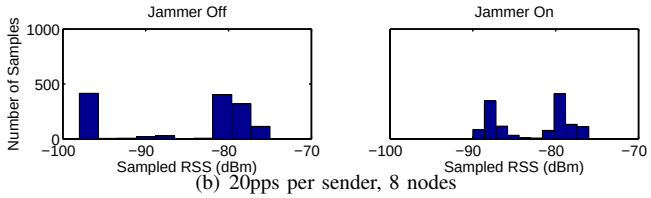


(a) 5 pps per sender, 5 senders (b) 10 pps per sender, 8 senders (c) 20 pps per sender, 8 senders

Fig. 5. Time series plots of estimated ANFs when the jammer was turned on in three scenarios: a low traffic case (5 senders, 5 packets per second (pps) each), a slightly-congested case (8 senders, 10 pps each), and a highly-congested case (8 senders, 20pps each). The estimated average ANFs were stable and jumped as soon as the jammer was turned on, indicating that Algorithm 2 is effective in estimating JSS.

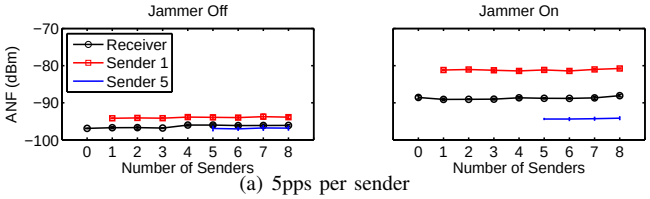


(a) 5pps per sender, 5 nodes

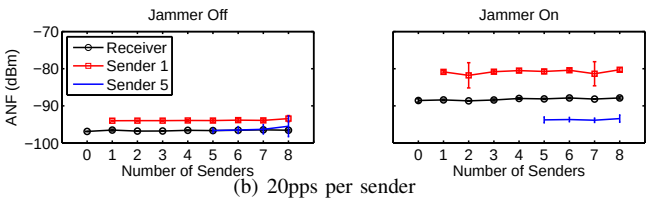


(b) 20pps per sender, 8 nodes

Fig. 6. Histogram of periodic RSS measurements in two cases: a low and a high traffic load.



(a) 5pps per sender



(b) 20pps per sender

Fig. 7. An illustration of the estimated ambient noise floor with an increasing number of senders.

that showed the same trend. The estimated ANFs of all nodes jumped once the jammer became active at the 15^{th} second, which shows that Algorithm 2 is able to adaptively calculate the ANF. Encouragingly, prior to jamming, even in the highly congested case where the average PDRs are less than 50% as shown in Figure 5(c), the receivers were able to derive a stable ANF. Sender 5 did miss a few estimation, because of its extreme low ambient noise sampling rate (5 samples per second). We note that prior to jamming the estimated ANF for the senders and receivers are different because the devices were not calibrated. After the jammer became active, the differences in ANFs for senders 1, 5, and the receiver were caused by their different distances to the jammer.

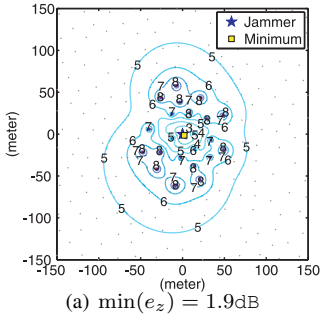
To better understand the distribution of ambient noise measurements, we showed the periodic ambient noise measurement histograms of the low traffic and highly-congested cases in Figure 6. In both cases, as the jammer was being turned on, the ANF increased from around -96 dBm to -85 dBm. Compared between those two cases, the percentage of ambient noise measurements when both the jammer and senders were transmitting (> -80 dBm) was larger in the highly congested cases, suggesting the increasing difficulty of deriving ANFs with the increase of traffic rates.

3) The Number of Senders: In order to study the impact of the number of colliding sources and the network traffic, we increased the number of senders sequentially from 0 to 8, and summarized the estimated ANFs in two scenarios (in Figure 7): jammer off and jammer on. In general, the increase of the senders does not have much influence on the correctiveness of ambient noise floor estimation in all cases. Sender 1 transmitting at 20 packets per second did show a higher variance of estimation. That is caused by its low ambient noise sampling rate.

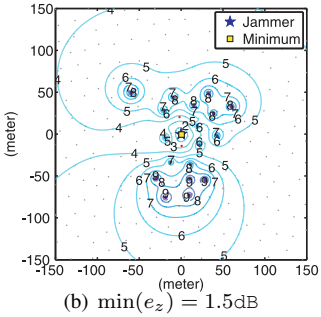
In summary, all of our experiments in real world scenarios have shown encouraging results and suggest the JSS can be estimated using Algorithm 2.

VI. Finding the Best Estimation

The jammer localization problem can be modeled as a non-linear optimization problem (defined in Problem 1), and finding a good estimation of jammer's location is equivalent to seeking the solution that minimizes the evaluation feedback metric e_z. In this section, we illustrate the relationship between

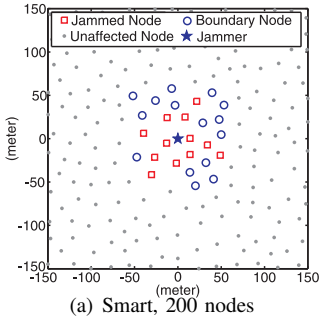


(a) $\min(e_z) = 1.9\text{dB}$

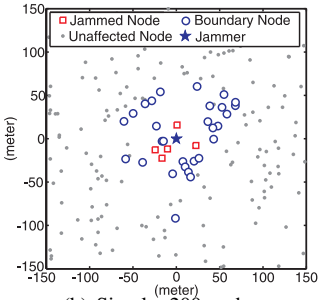


(b) $\min(e_z) = 1.5\text{dB}$

Fig. 8. An illustration of error contours for e_z in a network of 200 nodes. e_z reaches its minimum at a location close to the jammer's true position.



(a) Smart, 200 nodes



(b) Simple, 200 nodes

Fig. 9. Illustration of two deployments: (a) Smart deployment, (b) Simple deployment.

e_z and e_d (the distance between the true jammer's location and the estimated one), which suggests that greedy algorithms that search for successively better solutions are unable to find the global optimal value. Instead, we use heuristic search algorithms that rely on guided random processes to approach the global optimum without converging to a local minimum.

A. Error Analysis

The evaluation feedback metric e_z is a nonlinear function of the estimated location of the jammer and the measured RSS values. To understand e_z, we performed a numerical simulation and derived the numerical values of e_z on a grid of points in a 300-by-300 meter square, within which 200 nodes were randomly deployed with a transmission power of -45 dBm. Additionally, the jammer transmitted at a power level of -38 dBm, and affected about 20 boundary nodes. To examine the impact of an inaccurate estimation of P_J, we set the estimated jamming power $\hat{P}_j$ to -25 dBm, much larger than the true jamming power. To get enough resolution, we set the grid step to 0.5m and in total calculated $360,000$ data points for each network topology. We chose two representative network topologies and depicted their error contours in Figure 8, from which we drew the following observations:

1) Despite the inaccurate estimation of jamming power, the global minimum of e_z is close to the true location of the jammer, suggesting the estimated location that minimizes e_z is an accurate estimation of a jammer's position.
2) At each boundary node (marked by blue circles in Figure 8), e_z reaches its local maximum. This is because that at boundary node i, $\hat{d}_i$ is close to 0, which causes σ_i to be an outlier, raising the variance of e_z.
3) Interestingly, e_z is not strictly proportional to e_d. When the estimated location is in the close vicinity of the true value, the smaller e_d is, the smaller e_z becomes; when the estimated location increases to more than 100m, the larger e_d, the smaller e_z.

The combination of the 2nd and 3rd observations makes greedy algorithms impractical. For instance, the gradient descent, which moves towards the steepest decreasing direction of e_z, will not be able to climb the 'hill' of the global maximum at a boundary node, nor will it be guaranteed to search towards the global minimum solution. Thus, we examine heuristic searching algorithms to find the global minimum.

B. Searching Algorithm Description.

To search for the best estimation, we propose to use a simulated annealing algorithm (SA) [5]. Simulate annealing algorithms search for the optimal solutions by modeling the physical process of heating a material and then carefully lowering the temperature to decrease defects. At each iteration, the simulated annealing algorithm compares the current solution with a randomly-generated new solution. The new solution is generated according to a probability distribution with a scale proportional to the temperature, and it will replace the current solution based on a probability governed by both the new object function value and the temperature. By accepting 'worse' solutions occasionally, the algorithm avoids being trapped in local minima, and is able to explore solutions globally. As the temperature decreases, the annealing algorithm reduces its search scale so that it converges to a global minimum with high probability.

C. Reducing Searching Space

Simulated annealing algorithm is search-based, and its efficiency depends on the searching space. To improve the search efficiency, we first limited the range of each variable. For example, the coordinate of jammer (x_J, y_J) should reside inside the jammed area, which can be estimated by examining the positions of both jammed nodes and boundary nodes[2]. We also restricted the jammer's transmission power to the range of $[-50, 0]$ dBm. Note that this restriction is less important in terms of minimizing localization accuracy, since our objective function e_z does not depend on it. For the initial estimated position of jammer, we set the initial value to an estimation obtained by Adaptive LSQ methods proposed by Liu *et al.* [7].

VII. Performance Validation

A. Evaluation Methodology

In this section, we evaluated the performance of our jammer localization approach that utilizes the error minimizing frame-

[2]Even if in rare cases that the jammer is outside the deployed area, the layout of jammed nodes and boundary nodes (e.g. at the boundary of the network) will indicate the jamming regions.

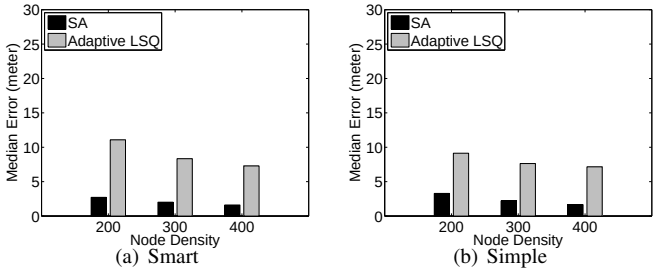


Fig. 10. Impact of node density on median localization errors, where $\{200, 300, 400\}$ nodes are distributed in 300-by-300 meter field following (a) a smart deployment, and (b) a simple deployment.

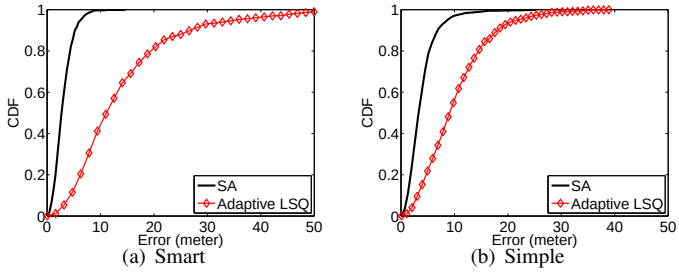


Fig. 11. Cumulative Distribution Function(CDF) of the localization errors in a 200-node network.

work. We studied our heuristic search algorithm–simulated annealing algorithm (SA)–for finding the best estimate of jammer's position; and compared SA to the prior work by Liu *et al.* [7], i.e. the Adaptive LSQ algorithm. We developed a simulator in Matlab. We simulated the underlying radio propagation according to the log-normal shadowing model, and used the Simulated Annealing function provided in the Global Optimization Toolbox in Matlab. To make a fair comparison, we set the parameters of the shadowing model to the same values as the ones used in the prior work by Liu *et al.* [7] (e.g., the path loss component $\eta = 2.11$) except for the standard deviation of the random attenuation σ, where we set $\sigma = 2.0$ rather than $\sigma = 1.0$ in order to examine the robustness of our algorithms in a highly irregular radio environment. Additionally, we also studied two types of network deployment: the smart deployment with nodes following a uniform coverage and the simple deployment with a random coverage, as shown in Figure 9.

We studied our SA algorithm with regard to a variety of factors, including node densities and jammer's transmission power. To capture the statistical performance, for each node density setup, we randomly generated 1000 network topologies for both the smart and simple deployment, respectively. For the rest of this paper, unless specified, the jammer's transmission power was set to -38 dBm and the transmission power of regular nodes was set to -45 dBm[3], resulting in a jamming range approximately twice of the communication range of a regular node.

Performance Metrics. To evaluate the accuracy of localizing the jammer, we defined the localization error e_d as the Euclidean distance between the estimated jammer's location and the true jammer's location, and we presented both the medians and the Cumulative Distribution Functions(CDF) of e_d to show the statistical characteristics.

B. Evaluation Results

Impact of Node Density. We first investigated the impact of node density on the accuracy of localizing one jammer by deploying $\{200, 300, 400\}$ nodes in our 300-by-300 meter network and fixing the jammer at the center $(0, 0)$.

[3]For instance, to set the transmission power of a MicaZ node to -45 dBm, one can select the power level of -25 dBm and attach a -20 dB attenuator.

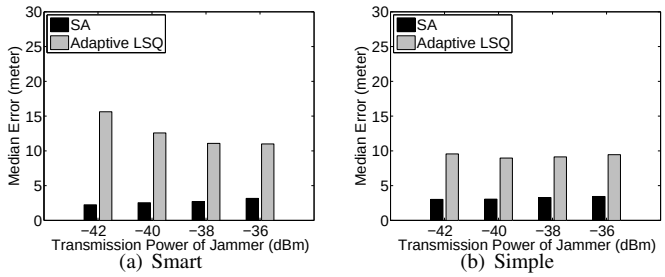
We depicted the median localization errors for the simulated annealing algorithm (SA) and Adaptive LSQ algorithm in Figure 10. Firstly, we observed that SA consistently outperformed Adaptive LSQ algorithm in all the node densities and deployment setups. The median errors for SA that ranges from 1.6 to 3.3 meters, are much smaller than the errors of Adaptive LSQ that range from 7 to 11 meters.

Secondly, as the network node density increases, the accuracy of all algorithms improves: median errors reduced from 2.7 to 1.6 meters for SA in a smart deployment and from 7.2 to 11 meters for Adaptive-LSQ in a smart deployment. This is because, given a jammer, a higher node density results in a larger number of boundary nodes, which in turn improve the accuracy of our algorithm.

Finally, we noticed that SA performed slightly better in a smart deployment than a simple deployment. In a smart deployment, nodes were evenly distributed and thus the estimated strength of jamming signals were uniformly distributed in space, which is likely leading to a better estimation of random attenuation, compared to a simple deployment where the estimated strength of jamming signals may be clustered together in space and result in biased random attenuation estimation.

Figure 11 shows the Cumulative Distribution Function (CDF) curves for all algorithms under both deployments in a 200-node network. Again, we observed that SA consistently outperformed the Adaptive LSQ algorithm. In particular, under a smart deployment, 90% of the time, SA can estimate jammer's location with an error less than 5.3 meters, while Adaptive LSQ can only reach an error less than 26.8 meters 90% of the time, resulting in a 5-fold improvement of the estimation accuracy. Similarly, in a simple deployment, 90% of the time, all of our localization algorithms can gain 2.6 times improvement compared to the Adaptive LSQ: an estimation errors less than 6.9 meters for our algorithms versus 18.1 meters for Adaptive LSQ.

Impact of the Jamming Power. To study the effects of various transmission power of jammer to the localization performance, we examined networks with 200 nodes in a 300-by-300 meter field and set the jammer's transmission power to $\{-42, -40, -38, -36\}$ dBm, respectively. The results are plotted in Figure 12, which shows that SA greatly outperformed the Adaptive LSQ algorithm by over 63% for all



(a) Smart (b) Simple

Fig. 12. Impact of jammer's transmission power on median localization errors, where the jammer's transmission power was set to $\{-42, -40, -38, -36\}$ dBm, respectively.

the jamming power levels. The localization errors of SA was slightly affected by the jamming power: an increment of 0.9 meters for the smart deployment and 0.4 meters for the simple deployment when the jamming power increased from -42dBm to -36dBm. This can be explained as the following. Two main factors that influence the accuracy of our error-minimizing-based approach are the spatial density of the estimated strength of jamming signals and the search space for the heuristic search algorithm. For a given node density, as the jamming power increases, the search space for jammer's position grows. As a result, the probability of finding an accurate location estimation reduces, and the average estimation error increases.

VIII. Concluding Remarks

We studied the problem of minimizing errors when localizing a jammer in wireless networks. The jammer could be some wireless device causing unintentional radio interference or malicious jammer disturbing the network. Most of the existing jammer-localization schemes rely on the indirect measurements of network parameters affected by jammer, such as packet delivery ratios, neighbor lists and nodes' hearing ranges, which makes it difficult to accurately localize jammer. In this work, we localized jammer by exploiting directly the strength of jamming signals (JSS). Estimating the JSS is considered challenging since they are usually embedded in other signals. Our estimation scheme derives ambient noise floors as the JSS leveraging the available signal strength measuring capability in wireless devices. The scheme samples signal strength regardless of whether the channel is busy or idle, and estimates the ambient noise floor by filtering out regular transmission (if any) to obtain the JSS. Our experiments utilizing MicaZ motes show that the derived ambient noise floor maps to the JSS even in scenarios with high traffic and heavy congestion.

To reduce the estimation errors, we further designed an error-minimizing-based framework to localize jammer. In particular, we defined an evaluation feedback metric that quantifies the estimation errors of jammer's position, and treated this metric as the objective function for minimizing. We analyzed the relationship between the evaluation feedback metric and estimation errors, and the results show that the location that minimizes the feedback metric approaches jammer's true locations and greedy algorithms are not guaranteed to find this location. As such, we examined simulated annealing (SA) algorithms to find the best location estimation. We compared our SA-based jammer localization algorithm with existing localization algorithms under various network conditions: node densities and jammer's transmission power. Our extensive simulation results have confirmed that our error-minimizing-based search algorithm outperforms the existing algorithms.

References

[1] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM*, pages 775–784, 2000.

[2] Y. Chen, J. Francisco, W. Trappe, and R. P. Martin. A practical approach to landmark deployment for indoor localization. In *SECON*, 2006.

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

[4] S. Khattab, D. Mosse, and R. Melhem. Modeling of the channel-hopping anti-jamming defense in multi-radio wireless networks. In *Proceedings of Annual International Conference on Mobile and Ubiquitous Systems*, pages 1–10, 2008.

[5] P. V. Laarhoven and E. Aarts. *Simulated Annealing: Theory and Applications*. Springer, 1987.

[6] H. Liu, Z. Liu, Y. Chen, and W. Xu. Determining the position of a jammer using a virtual-force iterative approach. *Wireless Networks*, 17:531–547, 2010.

[7] Z. Liu, H. Liu, W. Xu, and Y. Chen. Exploiting jamming-caused neighbor changes for jammer localization. *IEEE Transactions on Parallel and Distributed Systems*, 23(3), 2012.

[8] K. Ma, Y. Zhang, and W. Trappe. Mobile network management and robust spatial retreats via network dynamics. In *Proceedings of International Workshop on Resource Provisioning and Management in Sensor Networks*, 2005.

[9] 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*, pages 2526 – 2530, 2007.

[10] G. Noubir and G. Lin. Low-power DoS attacks in data wireless LANs and countermeasures. *Mobile Computting Communications Review*, 7(3):29–30, 2003.

[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*, 2009.

[12] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *MobiCom*, pages 32–43, 2000.

[13] Y. Sun and X. Wang. Jammer localization in wireless sensor networks. In *Proceedings of International Conference on Wireless communications, networking and mobile computing*, pages 3113–3116, 2009.

[14] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, 1992.

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

[16] A. Wood, J. Stankovic, and S. Son. JAM: A jammed-area mapping service for sensor networks. In *Proceedings of IEEE Real-Time Systems Symposium*, pages 286–297, 2003.

[17] W. Xu, W. Trappe, and Y. Zhang. Channel surfing: defending wireless sensor networks from interference. In *Proceedings of International conference on Information processing in sensor networks*, pages 499–508, 2007.

[18] W. Xu, W. Trappe, and Y. Zhang. Anti-jamming timing channels for wireless networks. In *Proceedings of ACM conference on Wireless network security*, pages 203–213, 2008.

[19] W. Xu, W. Trappe, Y. Zhang, and T. Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In *MobiHoc*, pages 46–57, 2005.

[20] J. Yang, Y. Chen, and J. Cheng. Improving localization accuracy of rss-based lateration methods in indoor environments. *Ad Hoc & Sensor Wireless Networks*, 11(3-4):307–329, 2011.