Security and Safety, Vol. 3, 2023025 (2024)

Security and Safety in Physical Layer Systems

Research Article

Open Access

Information Network

Topology-based multi-jammer localization in wireless networks

Hongbo Liu[1,*], Yingying Chen[2], Wenyuan Xu[3], Zhenhua Liu[4], and Yuchen Su[1]

[1] *University of Electronic Science and Technology of China, Chengdu 611731, China*
[2] *Rutgers University, New Brunswick, NJ 08902, USA*
[3] *Zhejiang University, Zhejiang 310058, China*
[4] *Wells Fargo, Charlotte, NC 28262, USA*

Received: 27 April 2023 / Revised: 1 August 2023 / Accepted: 17 August 2023 / Published online: 4 October 2023

Abstract Jamming attacks and unintentional radio interference are some of the most urgent threats harming the dependability of wireless communication and endangering the successful deployment of pervasive applications built on top of wireless networks. Unlike the traditional approaches focusing on developing jamming defense techniques without considering the location of jammers, we take a different viewpoint that the jammers' position should be identified and exploited for building a wide range of defense strategies to alleviate jamming. In this paper, we address the problem of localizing multiple jamming attackers coexisting in wireless networks by leveraging the network topology changes caused by jamming. We systematically analyze the jamming effects and develop a framework that can partition network topology into clusters and can successfully estimate the positions of multiple jammers even when their jamming areas are overlapping. Our experiments on a multi-hop network setup using MicaZ sensor nodes validate the feasibility of real-time collection of network topology changes under jamming and our extensive simulation results demonstrate that our approach is highly effective in localizing multiple attackers with or without the prior knowledge of the order that the jammers are turned on.

Keywords Jammer localization, wireless networks, topology

Citation Liu H, Chen Y and Xu W et al. Topology-based multi-jammer localization in wireless networks. Security and Safety 2024; **3**: 2023025. https://doi.org/10.1051/sands/2023025

1 Introduction

The broadcast-based communication has made wireless networks vulnerable to jamming attacks and to radio interference. The increasingly flexible programming interfaces of commodity devices (*e.g.*, software-defined radios) have enabled adversaries to build jammers with little effort to disturb network communication. Even without malicious jammers, the tension between the proliferation of wireless technologies and the limited number of unlicensed bands has made and will continue to make the radio environment crowded, causing unintentional radio interference across devices that leverage different wireless technologies but share the same spectrum, *e.g.*, WiFi and Bluetooth. Multiple instances of jamming attacks and unintentional radio interference may co-exist in the network, and they will continue to be one of the most urgent threats harming the dependability of wireless communication and endangering the successful deployment of pervasive applications built on top of wireless networks. Since both jamming attacks and radio interference can prevent networks from delivering information, we use the term *jamming* to refer to both threats in this paper.

To ensure the dependability of wireless communication, much work has been done to detect and defend against jamming attacks. In terms of detection, single-statistics-based and consistent-check-based algorithms [1] have been proposed. The existing countermeasures for coping with jamming include two types: the proactive conventional physical-layer techniques that provide resilience to interference by employing advanced transceivers [2], *e.g.*, frequency hopping, and the reactive non-physical-layer strategies that defend against jamming leveraging MAC or network layer mechanisms [3, 4], *e.g.*, adaptive error correcting codes, channel adaption. Those defense technologies provide useful methods to alleviate jamming. However, they primarily rely on the network to passively adjust themselves without leveraging the information of the jammer. We take a different viewpoint, that is, networks should identify the physical location of a jammer and use such information to actively exploit a wide range of defense strategies in various layers. For instance, a routing protocol can choose a route that does not traverse the jammed region to avoid wasting resources caused by failed packet deliveries. Furthermore, once a jammer's location is identified, one can eliminate the jammer from the network by neutralizing it. This approach is especially useful for coping with unintentional radio interference that is turned on accidentally. In this paper, we aim to address the problem of localizing multiple jamming attackers coexist in a wireless network.

Although there has been active research in the area of localizing a wireless device [5–7], most of those localization schemes are inapplicable to jamming scenarios. For instance, many localization schemes require the wireless device to be equipped with specialized hardware [5, 8], *e.g.*, ultrasound or infrared, or utilize signals transmitted from wireless devices to perform localization. Unfortunately, the jammer will not cooperate and the jamming signal is usually embedded in the legal signal and thus, is hard to extract, making the signal-based and special-hardware-based approaches inapplicable. Regarding jammer localization, only a few algorithms [9, 10] have been proposed to localize one jammer. Without presenting a performance evaluation, Pelechrinis *et al.* [10] proposed to localize the jammer by performing a gradient descent search based on packet delivery rate (PDR). Liu *et al.* [9, 11] have designed two algorithms that utilize the network topology changes caused by jamming attacks to estimate the jammer's position: one is a virtual-force-based jammer localization algorithm [9] and the other is a least-squares-based localization scheme [11]. Prior work can localize *one* jammer, but will not perform well in the presence of multiple jammers, which can cause severe network communication disturbance on a large scale. Furthermore, multiple jammers may have overlapping jamming regions and form only one connected jammed area. Therefore, the problem of multiple-jammer localization still persists and needs appropriate solutions.

In this paper, we systematically studied the effects of multiple jammers and developed a framework utilizing network topology changes under jamming to locate multiple jamming attackers. The two main components in our framework, namely the *automatic network topology partitioner* and *intelligent multi-jammer localizer*, work together to derive different categories of node clusters and achieve high localization accuracy even under overlapping jammed areas. We conducted real experiments using MicaZ sensor nodes in a multi-hop network setup with two jammers. Our experimental results confirmed that we are able to collect the network topology changes in spite of the disturbed network communication under jamming. We further performed simulations under various large-scale network setups. Our extensive simulation results demonstrated that our framework can effectively partition the network topology under the presence of multiple jamming attackers and further localize these jammers with high accuracy with or without prior knowledge of the order in the jammers are turned on. The main contribution of this paper is summarized as follows:

- We propose a new multi-jammer localization framework consisting of an automatic network topology partitioner and intelligent multi-jammer localizer, which could deal with both cases of sequentially and simultaneously turning on jammers.
- To localize multiple jammers with appropriate localization methods, we develop a new localization strategy in our intelligent multi-jammer localizer based on the jammed and boundary cluster classification results.
- We conducted real experiments using MicaZ sensor nodes in a multi-hop network setup, and the experimental results confirmed the capability of our proposed automatic network topology partitioner to collect the network topology changes under jamming.

- To evaluate the localization performance of the proposed framework, we conduct large-scale simulations by taking the example with 3 jammers both sequentially and simultaneously turning on for illustration. The localization results confirm the feasibility of our proposed multiple jammer localization strategy.

The remainder of the paper is organized as follows: We first discuss our work in the context of existing studies in Section 2. We then specify our jamming attack model and analyze the jamming effects in Section 3. The feasibility of our approach utilizing the network topology changes is validated through real experiments in Section 4. We present our framework and algorithms that are developed to partition the network topology and localize multiple jamming attackers in Section 5. We next conduct extensive simulations to validate our approach in Section 8. Finally, we conclude our work in Section 9.

2 Related work

Jamming and radio interference are known threats and have attracted much attention [12]. Traditionally, jamming is addressed through conventional PHY-layer communication techniques, *e.g.*, spreading techniques. While these techniques provide resilience to interference [2], they require advanced transceivers. Jamming detection has been studied by Xu *et al.* [1] in the context of commodity wireless devices and in the context of sensor networks [13]. Our work focuses on localizing jammers after identifying jamming attacks using these detection strategies.

Numerous countermeasures have been proposed for coping with jamming in commodity wireless networks. Defense strategies include the use of error correcting codes [3] to increase the likelihood of decoding corrupted packets, channel hopping [4, 14, 15] to adapt the working channel to escape from jamming, spatial retreats [16] to move out of jammed region geographically, anti-jamming timing channel [17], and wormhole-based anti-jamming techniques [18].

Wireless localization, which has been an active area of research, employs infrared [5] and ultrasound [8, 19] based on localization infrastructure, both of which require specialized infrastructure for localization. Furthermore, received signal strength (RSS) [6, 7, 20, 21] is an attractive approach because it can reuse the existing wireless infrastructure.

Localization algorithms can be categorized into range-based and range-free based on the localization methodology. Range-based algorithms involve estimating the distance to anchor points with known locations by utilizing the measurement of various physical properties, such as RSS [7], Time of Arrival [22], and Time Difference of Arrival [8]. Range-free algorithms [23–26] use coarser metrics to place bounds on candidate positions. However, these approaches are mostly inapplicable to localize jammers, as the jammer will not cooperate, and the regular radio signal is disturbed under jamming.

Recently, several methods have been proposed for localizing individual jammers. Pelechrinis *et al.* [10] suggested measuring PDR and performing a gradient descent search to locate the jammer. Liu *et al.* [9] proposed a method that estimates the jammer's position by using virtual forces derived from network topology changes caused by jamming attacks. Both methods [9, 10] require an iterative search for the jammer's location. Liu *et al.* [11] developed a lease-squares-based algorithm that leverages the change of hearing range caused by jamming to localize the jammer in one round. Kim *et al.* [27] used power adaptation techniques to localize a jammer, assuming an ideal Friis transmission model and not considering radio irregularity. However, these algorithms only localize one jammer and may not be effective in locating multiple jammers, which is the problem addressed in this paper.

3 Model

In this section, we introduce our adversary and network models, followed by an analysis of jamming effects in the presence of multiple jammers.

3.1 Adversary model

We consider the presence of multiple jammers in the network and aim to localize each of them. Regardless of their attack strategies, jammers have similar effects: nodes close to jammers experience severe communication disruption, while nodes farther away may not be affected at all. Therefore, we assume that jammers transmit at the same fixed power level, without investigating diverse jamming attack philosophies. To simplify the problem, we consider the scenario where two jammers transmit at the same power level and become active either sequentially or simultaneously. The simultaneous activation of two jammers presents greater challenges than the sequential case since it does not reveal any information about individual jammers but all of them as a whole. It is worth noting that our strategies can be easily extended to localize more than two jammers.

3.2 Network model

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

- **Stationary.** The nodes in our considered network remain stationary after their deployment. In future works, we will investigate mobile nodes.
- **Neighbor-aware.** Each node is equipped with an omnidirectional antenna and transmits at the same transmission power level. Therefore, each node shares the same communication range and can only receive messages from other nodes within its communication range. We refer to these nodes as "neighbors" of a particular node. To keep track of their neighbors, each node maintains a neighbor table containing the IDs of neighboring nodes and updates this table as needed. This is supported by most routing protocols and can be easily implemented by periodically broadcasting beacons.
- **Location-aware.** Each node is aware of its own location. This is reasonable as most wireless devices are equipped with GPS or some other approximate but less burdensome localization algorithms [6, 7, 12].
- **Able to detect jamming.** In this work, we focus on locating jammers after they are detected. Thus, we assume the network is able to identify a jamming attack and the number of jammers, leveraging the existing jamming detection approaches [1, 12, 28].

3.3 Analysis of jamming effects

3.3.1 General jamming effects

In order to present a comprehensive understanding of the complex relationship between the transmission power of a wireless node and a jammer, we adopt the signal-to-interference-plus-noise-ratio (SINR) model. This model takes into account the ambient noise P_N and jamming signals P_J as part of the "noise" in a jamming scenario. The SINR can be expressed for a sender-receiver pair (S, R):

$$\text{SINR} = \frac{P_{SR}}{P_N + P_{JR}} \tag{1}$$

where P_{SR} is the received power of the desired signal, P_N represents the noise, and P_{JR} is the received power level of the jamming signal. In our work, we define the link state l_{ij} *from* node n_i to n_j using a threshold model. Specifically, we define the link state from node n_i to n_j as:

$$l_{ij} = \begin{cases} 0 & \text{SINR}_{ij} \leq \gamma_0 \\ 1 & \text{SINR}_{ij} > \gamma_0 \end{cases} \tag{2}$$

where SINR_{ij} represents the SINR measured at node n_j when node n_i is transmitting, and all other network nodes remain silent. γ_o is the threshold SINR, above which packets can be received successfully. We refer to this threshold as the **Decodable SINR threshold** [11].

In order to describe the impact of jammers on wireless networks, we introduce a classification of network nodes into three categories: *unaffected node* N_U, *jammed node* N_J, and *boundary node* N_B. Let $Nbrn_i$ denote the set of neighbors of node n_i before any jammer becomes active, other notations are explained in Table 1. We then derive the SINR-based jamming model as follows:

- **Unaffected node.** $N_U = \{n_u | \forall n_i \in Nbr\{n_u\}, \text{SINR}_{iu} > \gamma_o\}$. 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 *receive* messages from any of the unaffected nodes. We note that two nodes in the jammed region may still be able to communicate with each other, but we still call them jammed nodes. However, they cannot communicate with any of the unaffected nodes.

Table 1. Notation summary

Variable or Function	Description
N_J^i	Jammed nodes when ith jammer is on;
N_B^i	Boundary nodes when ith jammer is on;
G	Neighborhood adjacency matrix;
JC	The clusters of jammed nodes;
BC	The clusters of boundary nodes;
$\hat{J}_i$	The estimated position of ith jammer;
$MST()$	Minimum spanning tree method;
$Centroid()$	Centroid-based method;
$AdaptiveLSQ()$	Adaptive Least Squares method;
$Mirroring()$	Mirroring method;
$Gauss\text{-}Newton(\ )$	Gauss-Newton searching method;

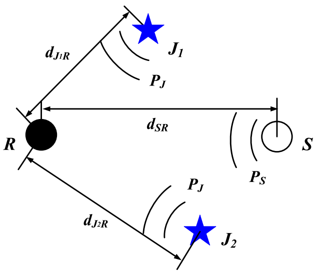


Figure 1. An illustration of a multi-jammer scenario in a wireless network

- **Boundary node.** $N_B = \{n_b | (\exists n_i \in N_U, L_{ib} = 1) \text{ and } (\forall n_i \in Nbr\{n_b\} \cap N_J, \text{SINR}_{ib} \leq \gamma_o)\}$. A boundary node can receive packets from part of its neighbors but not from all its neighbors.

3.3.2 Effects of multiple jammers

In scenarios where multiple jammers are present, the jammers can be turned on either sequentially or simultaneously. We analyze the jamming effects by considering these two typical ways of conducting jamming attacks. Figure 1 depicts a network set up with one sender-receiver pair (S, R) and two jammers (J_1, J_2), where the distance between S and R is d_{SR}, and J_1 and J_2 with the transmission power P_J are d_{J_1R} and d_{J_2R} away from R. The interfering signal from both jammers, J_1 and J_2, arriving at the legitimate receiver S will mix together and create even stronger interference to the reception of the legitimate signal, resulting in more complexity in the jamming scenarios. We next use this setup to illustrate the different jamming effects when two jammers are turned on either sequentially or simultaneously.

Sequentially turning on jammers. When the two jammers J_1 and J_2 are turned on sequentially, the network communication will experience changes and disruptions twice. When the first jammer J_1 is turned on, according to equation (1), the SINR at receiver R in the presence of Jammer J_1 becomes:

$$\text{SINR}^1 = \frac{P_{SR}}{P_N + P_{J_1R}}, \tag{3}$$

and the link state from node n_i to n_j is still defined by equation (2).

After the second jammer J_2 is turned on, the SINR-based jamming model at R is changed to:

$$\text{SINR}^{1,2} = \frac{P_{SR}}{P_N + P_{J_1R} + P_{J_2R}}. \tag{4}$$

The link state from node n_i to n_j may change or remain the same depending on SINR. In total, there are three cases:

$$l_{ij} = \begin{cases} 0 \rightarrow 0 & \mathrm{SINR}_{ij}^{1} \leq \gamma_0 \rightarrow \mathrm{SINR}_{ij}^{1,2} \leq \gamma_0 \\ 1 \rightarrow 0 & \mathrm{SINR}_{ij}^{1} > \gamma_0 \rightarrow \mathrm{SINR}_{ij}^{1,2} \leq \gamma_0 \\ 1 \rightarrow 1 & \mathrm{SINR}_{ij}^{1} > \gamma_0 \rightarrow \mathrm{SINR}_{ij}^{1,2} > \gamma_0. \end{cases} \tag{5}$$

For instance, a link state l_{ij} changes from 1 to 0, if the SINR from n_i to n_j was larger than γ_0 but drops below γ_0 after J_2 is turned on.

Simultaneously turning on jammers. When two jammers are turned on simultaneously, the SINR at receiver R is similar to the SINR after both jammers are turned on sequentially, and the link state from node n_i to n_j is

$$l_{ij} = \begin{cases} 0 & \mathrm{SINR}_{ij}^{1,2} \leq \gamma_0 \\ 1 & \mathrm{SINR}_{ij}^{1,2} > \gamma_0. \end{cases} \tag{6}$$

3.3.3 Propagation models

We have utilized two propagation models to model the received power of signals: free-space model and the shadowing model. Due to the simplicity of the free space model, we use it to illustrate the theoretical basis of our algorithm. However, our experimental validation leverages the shadowing model, a realistic model that captures the absorption, reflection, scattering, and diffraction in complex propagation environments.

Free Space Model considers a signal propagated through free space without obstructions. The received signal power is,

$$P_{SR} = \frac{P_S G}{4\pi d^2}, \tag{7}$$

where P_S is the transmission power of the sender; G is the antenna field patterns in the line-of-sight (LOS) direction between the sender and receiver; and d is the distance between the sender and the receiver. We note that the sender can either be a jammer J or a network node n_i. The analysis of the free-space model provides insights into understanding the jamming effects and the underlying theoretical basis. However, real wireless communication operates in complex propagation environments full of absorption, reflection, scattering, and diffraction, and it cannot be accurately modeled by the free-space model. In this work, we adopt the shadowing-based signal propagation model which is more realistic in practice.

Shadowing Model captures both path loss *versus* distance and the random attenuation due to blockage from objects in the signal path [29]. Let path loss at the receiver that is at the distance d from the sender be

$$\mathrm{PL}(d) = 10\log_{10}\frac{P_S}{P_{SR}}, \tag{8}$$

then the shadowing model has the following form:

$$\mathrm{PL}(d) = \mathrm{PL}(d_0) - 10 \cdot \eta \cdot \log\left(\frac{d}{d_0}\right) + X_\sigma, \tag{9}$$

where $\mathrm{PL}(d_0)$ is the known path loss at a reference distance d_0, η is the Path Loss Exponent (PLE), and X_σ is a Gaussian zero-mean random variable with standard deviation σ.

The log-normal shadowing model captures both path loss *versus* distance and the random attenuation due to blockage from objects in the signal path [29]. It has the following form:

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

where $\mathrm{PL}(d)$ is the path loss at distance d, $\mathrm{PL}(d_0)$ is the known path loss at a reference distance d_0, η is the Path Loss Exponent (PLE), and X_σ is a Gaussian zero-mean random variables with standard deviation σ.

When there is only one jammer J, the received power from the sender S can be expressed as $P_{SR} = \frac{P_S}{10^{\frac{\mathrm{PL}(d_{SR})}{10}}}$, and from the jammer is $P_{JR} = \frac{P_J}{10^{\frac{\mathrm{PL}(d_{JR})}{10}}}$. Thus, the signal-to-interference-plus-noise ratio under a single jammer scenario is:

$$\mathrm{SINR}^1 = \frac{P_{SR}}{P_{JR} + P_N} = \frac{\frac{P_S}{10^{\frac{\mathrm{PL}(d_{SR})}{10}}}}{\frac{P_J}{10^{\frac{\mathrm{PL}(d_{JR})}{10}}} + P_N}. \tag{11}$$

When multiple jammers are present, we illustrate the SINR jamming model under the shadowing signal propagation model by using two jammers. The SINR at R is calculated by considering the superimposed transmission power from both jammers:

$$\mathrm{SINR}^{1,2} = \frac{\frac{P_S}{10^{\frac{\mathrm{PL}(d_{SR})}{10}}}}{\frac{P_J}{10^{\frac{\mathrm{PL}\left(d_{J_1R}\right)}{10}}} + \frac{P_J}{10^{\frac{\mathrm{PL}\left(d_{J_2R}\right)}{10}}} + P_N}. \tag{12}$$

Further, the link state from node n_i to n_j is decided by equations (5) and (6).

4 Collecting network topology information

To localize multiple jamming attackers, our approach is to estimate the positions of jammers by observing network topology changes caused by their presence. To achieve this, it is crucial to capture topology differences before and after the emergence of jammers. In Section 3, we provide important theoretical insights to understand the impact of jammers on the network. Specifically, the likelihood that node n_i receives messages from node n_j depends on the SINR at n_i when n_j is transmitting. However, measuring SINR is often not feasible in practice. In this section, we describe our experimental study on collecting network topology information in real-time and classifying nodes into three categories: *unaffected nodes, jammed nodes, and boundary nodes.* To conduct our study, we used MicaZ sensor nodes [30], which provide access to the entire network stack. The MicaZ sensor nodes use TinyOS 2.x as the operating system and have a 2.4–2.48 GHz Chipcon CC2420 Radio.

4.1 Link state estimation and information collection

Our approach involves each node updating its neighbor table by locally measuring the link quality to each neighbor and periodically reporting the neighbor table to a designated entity, such as the network sink, which can then localize jammers.

We estimated the link quality by measuring the percentage of packets delivered. Specifically, the instantaneous PDR from node n_j to node n_i at the kth interval can be expressed as $p_{ij}^k = \frac{m_r}{m_t}$, where m_t is the total number of packets transmitted from n_j to n_i and m_r is the total number of packets received at n_i at the kth interval. We defined the link quality as the exponential moving average of instantaneous PDRs.

$$q_{ij}^k = \begin{cases} (1-\alpha)\, q_{ij}^{k-1} + \alpha p_{ij}^k & \Delta_l^k < \beta_1 \\ \alpha q_{ij}^{k-1} + (1-\alpha) p_{ij}^k & \texttt{otherwise}, \end{cases} \tag{13}$$

where α controls the weight of decreasing older link estimations, $\Delta_l^k = \max_{r \in [1,l]} |p_{ij}^k - p_{ij}^{k-r}|$, and β_1 defines the threshold that bounds short-term fluctuations. The condition $\Delta_l^k < \beta_1$ is for expediting link estimations when the link state has indeed been changed.

To improve the accuracy of link estimations, we introduced a small α that discounts older estimations slowly and smooths out short-term fluctuations. However, when a jammer is activated, it introduces delays before the estimations reflect the current network condition, such as being jammed. To address this issue, we examined the instantaneous Packet Delivery Ratios (PDRs) in the past l intervals and assigned a high weight to p_{ij}^k if its changes exceeded the short-term fluctuation range, denoted as β_1. Additionally, we defined the link state from node n_i to n_j as follows:

$$l_{ij}^k = \begin{cases} 1 & q_{ij}^k > \beta_2 \\ 0 & \texttt{otherwise}. \end{cases} \tag{14}$$

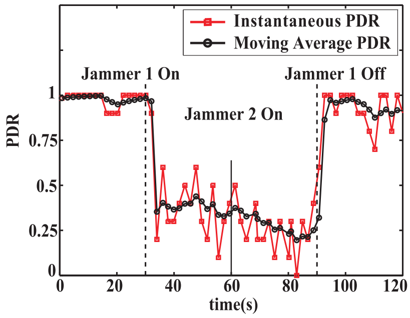


Figure 2. Instantaneous PDR and exponential moving average of PDR from node 11 to node 8, when two jammers were turned on and off in sequence

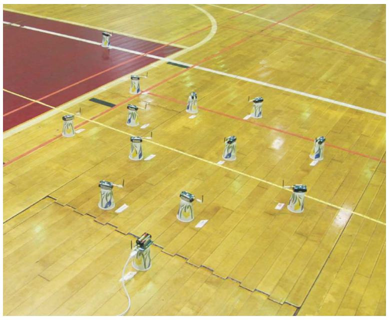


Figure 3. A snapshot of experiment setup

As an example, Figure 2 shows q^k and p^k between a pair of nodes in our experimental network. In our experiment, we set $\alpha = 0.2$, $\beta_1 = 0.7$, $l = 2$, and $\beta_2 = 0.65$. We observed that q^k smoothed out the fluctuations when the network status did not change, but it quickly captured the event that J_1 was turned on at the 30th second and J_2 was turned off at the 90th second. We noted that J_2 has a small impact on the link quality, in this case.

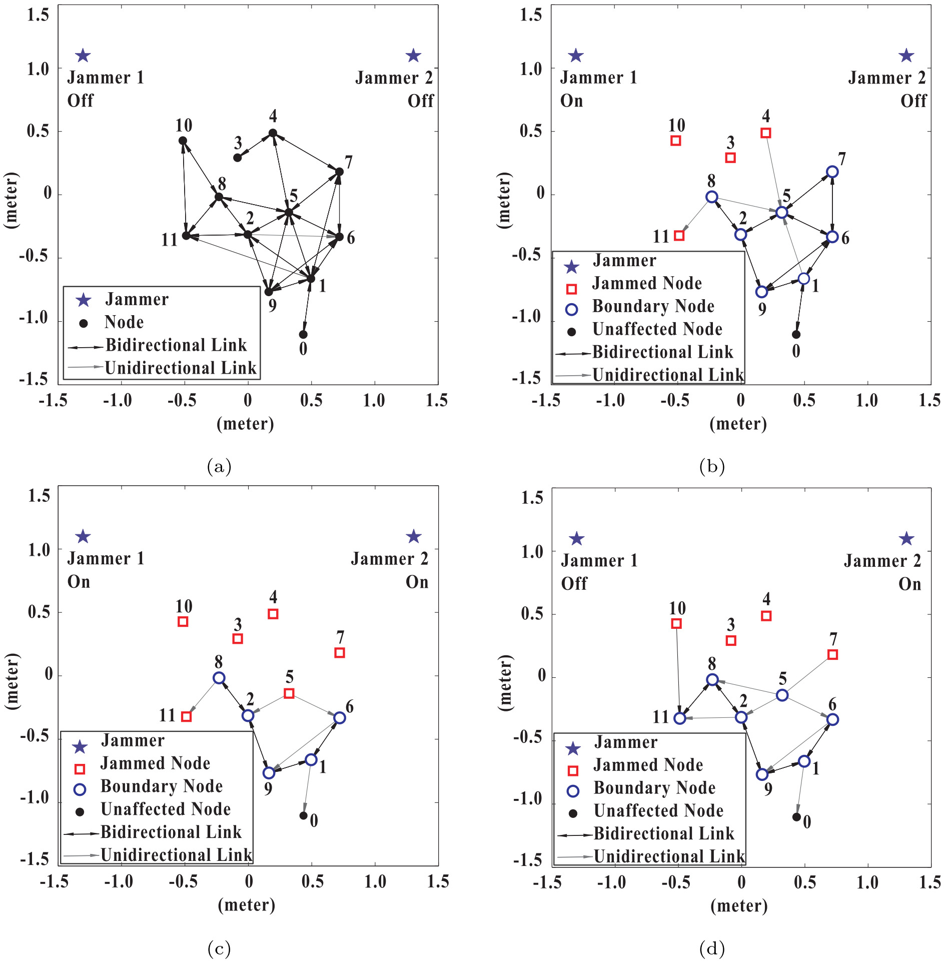
To collect information about the network topology, a customized protocol was used based on the Collection Tree Protocol (CTP) implemented in TinyOS 2.x. Each node was able to monitor the link quality with its neighbors and send this information to the designated node for jamming localization. A routing tree was built after the network was deployed, with the designated node as the root. Each node periodically sends a data packet containing its latest neighbor list to the designated node *via* unicast.

4.2 An example walk-through using MicaZ

To investigate the effects of jamming in an indoor environment, we deployed a network of 12 Micaz nodes with a communication range of approximately 0.8 m, achieved by installing a 10 dB attenuator on each node. The jamming devices, J_1 and J_2, were positioned at the upper-left and upper-right corners, respectively, as illustrated in Figure 3. Node 0 was designated as the root of a routing tree, and all nodes periodically reported their neighbor lists to node 0, which learned the network topology. Figure 4a shows the network topology before the jammers were activated, with all links being bidirectional. Single-headed arrows indicate a unidirectional link from node n_i to n_j ($l_{ij} = 1$), while double-headed arrows represent bidirectional links ($l_{ij} = l_{ji} = 1$). It should be noted that in the absence of jammers, all links in the network were bidirectional.

We activated the first jammer J_1 at the 30th second and the second jammer J_2 at the 60th second. We then turned off J_1 at the 90th second. Figure 4 illustrates the network topologies at each stage. From these observations, we noted that:

- The presence of jammers has caused certain links to become unidirectional. For instance, when J_1 is active, the link from node 8 to node 11, $l_{8,11}$, is connected as shown in Figure 4a, but $l_{11,8}$ is not.



(a)

(b)

(c)

(d)

Figure 4. Topology changes as turning on and off J_1 placed at the upper-left corner and J_2 at the upper-right corner in sequence: (a) No jammers were on. (b) J_1 was turned on at the 30th second. (c) J_2 was turned on at the 60th second. (d) J_1 was turned off at the 90th second

However, node 11 can still occasionally deliver a few packets to node 8 to report its neighbor list, so node 0 is aware that $l_{8,11} = 1$.

- The experiments confirmed our analysis of how multiple jammers affect network topology changes. Furthermore, the experiments suggested that the network can identify the order in which the jammers become active. Specifically, when J_1 was turned on in the upper-left corner, as shown in Figure 4b, nodes {5, 7} became boundary nodes since they lost some of their neighbors but could still receive messages. After J_2 was turned on, as shown in Figure 4c, nodes {5, 7} changed into jammed nodes since they could no longer receive messages from any of their neighbors. Once we turned off J_1, node 5 regained its ability to deliver messages to others and became a boundary node again, but node 7 remained jammed, as shown in Figure 4d.
- Interestingly, we discovered that node 5 was not jammed when only one jammer was active, but became jammed when both jammers were turned on. Therefore, $N_J^1 \cap N_J^2 \neq N_J^{1,2}$, making the task of localizing multiple jammers challenging.

After activating J_1 on the upper-left corner, as depicted in Figure 4b, nodes 3, 4, 10, and 11 became jammed nodes and could no longer receive or send messages to other nodes. Meanwhile, nodes 1, 2, 5, 6, 7, 8, and 9 became boundary nodes, as they were still able to receive messages but lost some of their neighbors. Node 0 remained unaffected. Upon turning on J_2, shown in Figure 4c, nodes {5, 7} changed from boundary nodes to jammed nodes as they were no longer able to receive messages from any of their

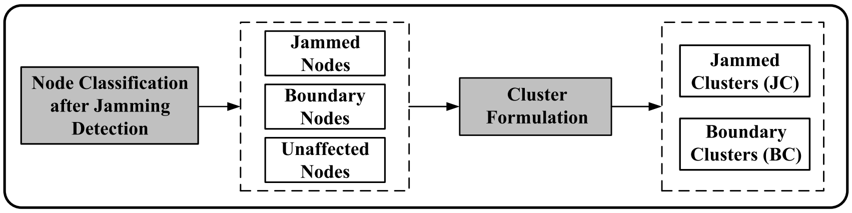


Figure 5. Framework for automatic network topology partitioner.

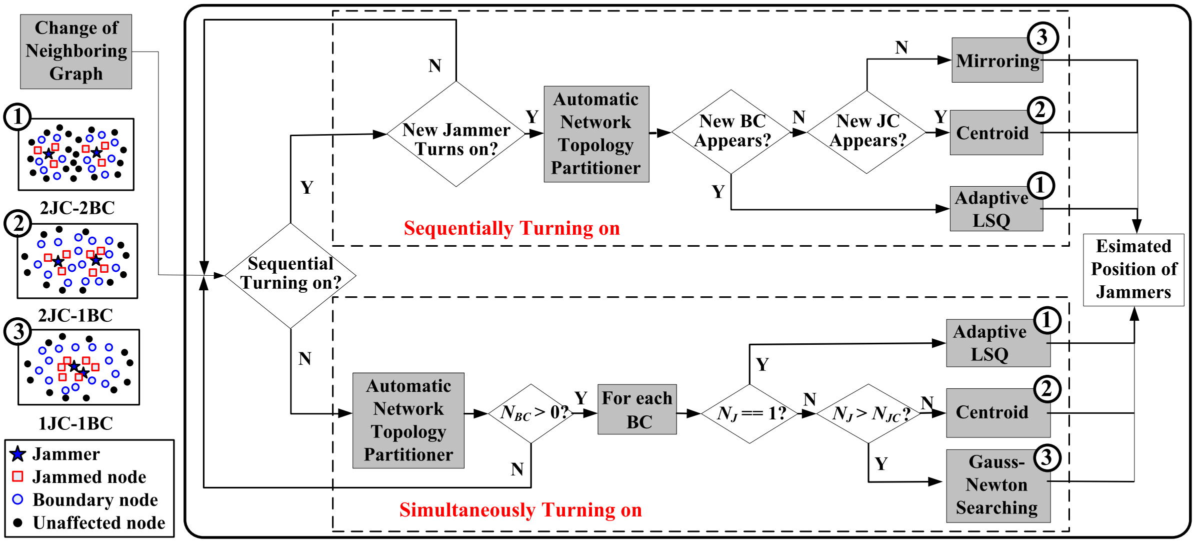


Figure 6. Framework for intelligent multi-jammer localizer

neighbors. Finally, after turning off J_1 as illustrated in Figure 4d, node 11 regained its ability to deliver messages and became a boundary node.

5 Framework for localizing multiple jammers

Our experimental results suggest that we are able to collect the network topology changes in spite of the disturbed network communication under jamming. In this section, we propose a framework that can localize multiple jammers by exploiting the collected network topology changes. We note that this framework can be implemented at the network sink where all the network neighborhood information is reported, but is not limited to it. Furthermore, each node monitors the network topology changes by measuring the beacons required by most routing protocols, and our localization algorithm involves each affected node reporting its neighbor changes in one message. Thus, the additional communication overhead is proportional to the affected nodes.

Our framework consists of two components, *the automatic network topology partitioner* and *intelligent multi-jammer localizer*, as shown in Figures 5 and 6 respectively. The automatic network topology partitioner systematically divides the nodes into three categories by examining the PDR on each node: unaffected nodes, jammed nodes, and boundary nodes. Then it forms two types of clusters *via* the Minimum Spanning Tree technique: *jammed clusters (JCs)* and *boundary clusters (BCs)*. The detailed approach will be discussed in Section 6.

The intelligent multi-jammer localizer estimates the positions of multiple jammers based on the results from the partitioner component, by utilizing different localization algorithms. Particularly, the localizer should deal with two cases: sequentially and simultaneously turning on jammers. For each case, there are three localization algorithms, dealing with different jamming scenarios, are developed for localizing jammers. In particular, for sequentially turning on the case, we have the mirroring method, centroid method, and adaptive LSQ method for estimating the positions of jammers. The algorithm starts localization

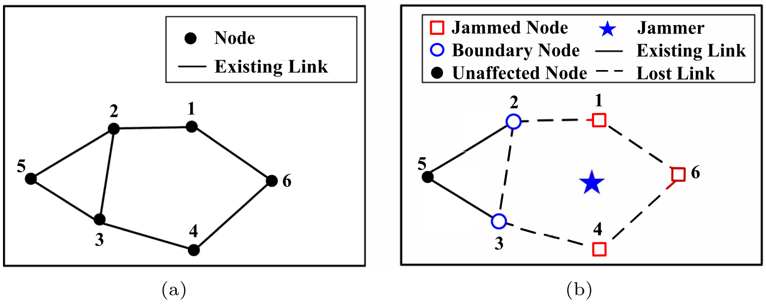


Figure 7. A network example to show the neighborhood adjacency matrix. (a) Without jamming. (b) With jamming

when there is a new jammer turning on. The adaptive LSQ method is exploited if a new boundary cluster (BC) appears; Centroid method deals with the scenario where new jammed cluster (JC) appears but no new BC; Mirroring method addresses the jamming scenario where neither new BC or new JC appears.

For simultaneously turning on the case, the adaptive LSQ method, centroid method, and Gauss-Newton searching method are exploited. The jammers are localized by examining each BC in the jamming scenario. If only one jammer is included in one particular BC, the Adaptive LSQ method is deployed; if the number of jammers in the BC under examination is larger than the number of JC formed in this BC, Gauss-Newton searching method is utilized, otherwise, we choose centroid method. We will discuss the detailed algorithms in Section 7.

6 Automatic network topology partitioner

Upon detection of jamming using existing detection approaches, our automatic network topology partitioner categorizes network nodes into distinct clusters, namely *jammed clusters (JC)* and *boundary clusters (BC)*. Typically, there is one JC and one BC formed around a jammer. To identify these clusters, we developed a topology partitioning method based on Minimum Spanning Trees (MST).

MST-based topology partitioning. We formulate the network into a connected, undirected graph, which is represented by a *neighborhood adjacency matrix* G [31]. The graph is undirected because each communication link between nodes is bi-directional under normal situations without jamming. In the matrix G, an element is set to 1 if two nodes are neighbors, otherwise, it is set to 0. Given a connected, undirected graph, a spanning tree of that graph is a subgraph which is a tree connecting all the vertices together. Further, an **MST** [31] is a spanning tree with a total weight less than or equal to the weight of every other spanning tree. Any undirected graph (not necessarily connected) has a minimum spanning forest, which is a union of minimum-spanning trees for its connected components. The minimum-spanning trees can be obtained using Prim's algorithm.

To identify BCs and JCs, we define subgraphs G_{N_J} and G_{N_B} containing all jammed and boundary nodes, respectively. In the presence of multiple jammers in the network, some nodes will be identified as either jammed nodes or boundary nodes. These nodes can form connected and undirected graphs, which we denote as G_{N_J} and G_{N_B} for jammed and boundary nodes respectively. To better understand the relationship among these graphs, we use a network with 6 nodes as shown in Figure 7. Node 2 has three neighbors, 1, 3 and 5; Node 3's neighbors are 2, 4, and 5; Node 3 and 6 are Node 4's neighbors; Node 5 has 2 and 3 as its neighbors; Node 6 connects with 1 and 4. The 6×6 neighborhood matrix G of this network with respect to the unaffected node vector $[1, 2, 3, 4, 5, 6]$ is:

$$G = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}. \tag{15}$$

The rows and columns in G correspond to the node IDs. For example, $G(1, 2) = 1$ represents a link existing between Node 1 and 2, whereas $G(1, 3) = 0$ indicates Node 1 and 3 are disconnected. Under

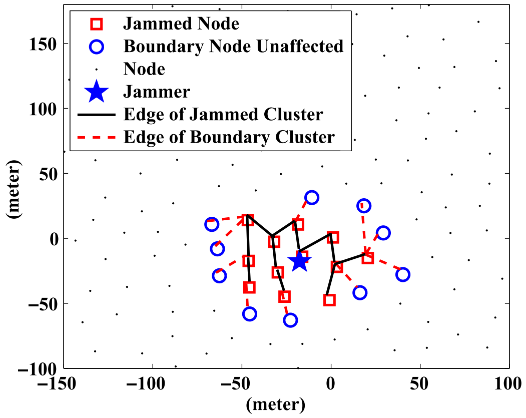


Figure 8. Clusters for jammed nodes and boundary nodes: the nodes connected by black solid lines form a jammed cluster through an MST, whereas those nodes connected by red dashed lines belong to the boundary cluster

jamming, there are 3 jammed nodes *i.e.*, 1, 4, and 6, and 2 boundary nodes, 2 and 3. Thus, the jammed node ID vector $I_{N_J} = [1, 4, 6]$, and boundary node ID vector $I_{N_B} = [2, 3]$. G_{N_J} and G_{N_B} are sub-matrices of G formed by selecting rows and columns indexed by I_{N_J} and I_{N_B}, respectively.

$$G_{N_J} = G\left[I_{N_J}; I_{N_J}\right] = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{16}$$

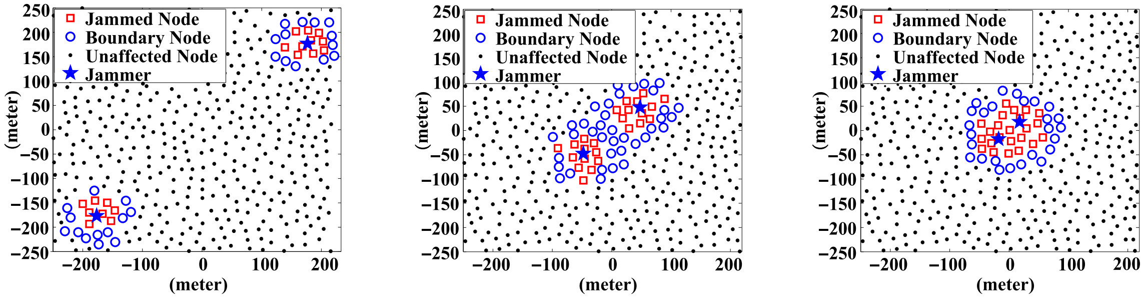
$$G_{N_B} = G[I_{N_B}; I_{N_B}] = \begin{bmatrix} 1\,1 \\ 1\,1 \end{bmatrix}. \tag{17}$$

Since boundary nodes are mostly surrounding jammed nodes, they may not form an appropriate cluster by themselves. To derive the proper number of BCs, instead of using G_{N_B}, we use $G_{N_J\&N_B}$, which is a submatrix of G formed by selecting rows and columns indexed by $I_{N_J} \cup I_{N_B}$:

$$G_{N_J\&N_B} = G\left[I_{N_J} \cup I_{N_B}; I_{N_J} \cup I_{N_B}\right] = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{bmatrix}. \tag{18}$$

Algorithm 1 outlines the MST-based topology partitioning approach. The JC is obtained through Prim's algorithm applied to G_{N_J}. To establish the BC, we create a combined MST comprising the jammed and boundary nodes in $G_{N_J\&N_B}$, beginning with either a jammed or boundary node. We then remove the jammed nodes from the combined MST to obtain the BC. Figure 8 illustrates the JC and BC obtained *via* the MST-based topology partitioning method. The JC consists of nodes connected by black solid lines, while the red dashed lines connect the nodes belonging to the BC. It is worth noting that the BC may not be a suitable cluster on its own, and may require the involvement of jammed nodes to form an appropriate cluster. Overall, two clusters, JC and BC, are formed based on the proposed MST-based approach to facilitate the later localization process. According to [31], the computational complexity of Prim's algorithm is $O((V + E)\log V)$, where V and E represent the number of vertices and edges in a graph. Since the number of edges is uncertain in the G under jamming, the complexity of the proposed approach in the worst case is $O(N_J^2 + (N_J + N_B)^2) = O(2N_J^2 + 2N_J N_B + N_B^2)$.

Node clustering analysis based on jammers' distance. After applying our MST-based topology partitioning method when multiple jammers are present in the network, there will be one jammed cluster and one boundary cluster formed around each jammer as depicted in Figure 9a. These clusters aid in localizing the position of each jammer. For instance, the jammed cluster can be used with the traditional centroid method [9] to obtain an initial position estimate of the jammer. The boundary cluster is also utilized to examine changes in node hearing range caused by jamming, which assists in improving the localization accuracy using the AdaptiveLSQ method, developed in our prior work [11].



(a)

(b)

(c)

Figure 9. Illustration of the clustering results obtained from the MST-based topology partitioning method when two jammers are placed at various distances. (a) 2 jammed clusters, 2 boundary clusters. (b) 2 jammed clusters, 1 boundary cluster. (c) 1 jammed cluster, 1 boundary cluster

We examine a two-jammer example to illustrate our methodology, noting that cases with more jammers can produce a wider range of clustering outcomes but share the same fundamental concept as the two-jammer scenario. When the two jammers are far apart, each one forms a distinct jammed cluster and boundary cluster resulting in *2 jammed clusters and 2 boundary clusters* (2JC-2BC), as shown in Figure 9a. However, when two jammers reside close to each other, they may have overlapping jamming regions and form only one connected jammed area. Depending on how close the two jammers are, two scenarios are possible. The first one is *two jammed clusters and one boundary cluster* (2JC-1BC), as shown in Figure 9b. When two jammers are located close by, the two boundary clusters are merged into one, but the two jammed clusters are still distinguishable. The second one is *one jammed cluster and one boundary cluster* (1JC-1BC), as depicted in Figure 9c. When two jammers are placed even closer, the two jammed clusters merge into one jammed cluster.

The MST-based topology partitioning method can identify all three cases: *2JC-2BC, 2JC-1BC,* and *1JC-1BC*. However, the diversity of clustering results makes it challenging to localize multiple jammers. While in the case of 2JC-2BC, algorithms for localizing one jammer can be applied to determine both jammers' location, this is not applicable when two jammers are close to each other and form 2JC-1BC or 1JC-1BC. To overcome this challenge, we developed an intelligent multi-jammer localizer that uses the topology partitioning results to localize jammers regardless of whether they have overlapping jammed areas.

7 Intelligent multi-jammer localizer

Continuing with the two-jammer example, to localize multiple jammers, our framework is designed to perform intelligent localization based on three possible classification outcomes returned from the automatic network topology partitioner: 2JC-2BC, 2JC-1BC, and 1JC-1BC. For the cases with more jammers, the localization strategy can also refer to the two-jammer example. We will discuss it in a later section. Moreover, we develop two sets of solutions, one possesses the prior knowledge of the order in which the jammers are turned on, referred to as *sequentially* turning on; and the other does not have any prior knowledge about the order in which the jammers are turned on, which makes the system consider that the jammers are turned on *simultaneously*. The jamming effects of sequentially turning on jammers and

Algorithm 1. MST-based topology partitioning

```
Require: INPUT:
    $G_{N_J}, G_{N_J \& N_B}$
    OUTPUT:
    $JC, BC$
    PROCEDURES:
    $JC = MST(G_{N_J})$
    $\{BC\&JC\} = MST(G_{N_B \& N_J})$
    $BC = \{BC\&JC\}|JC$
```

simultaneously turning on jammers are presented in Section 3 and are used as prior knowledge for the intelligent multi-jammer localizer.

7.1 Basic algorithms to localize single jammer

We start by introducing several localization algorithms used to estimate the position of a single jammer: *Centroid-based* [9], *Adaptive LSQ* [11], mirroring method and Gauss-Newton searching method. All these algorithms work even when the network communication is disturbed by jamming, and they utilize the affected network topology to perform localization.

Centroid-based. The Centroid-based localization method estimates a single jammer's position $(\hat{x}_J, \hat{y}_J)$ by averaging over the coordinates of all jammed nodes belonging to the corresponding jammed cluster formed around the single jammer. Consider that there are M jammed nodes $\{(x_m, y_m)\}_{m=1,\ldots,M}$, the position of the jammer can be estimated by Centroid-based localization as:

$$\hat{J} = (\hat{x}_J, \hat{y}_J) = \left(\frac{\sum_{m=1}^{M} x_m}{M}, \frac{\sum_{m=1}^{M} y_m}{M}\right). \tag{19}$$

Adaptive LSQ. The Adaptive LSQ exploits the formation of the boundary cluster and uses a node (*e.g.*, boundary node)'s neighbor list changes under jamming to estimate the jammer's location. We describe the main component of Adaptive LSQ in this paper and refer readers to our prior work [11] for a complete algorithm description.

In summary, we formed a least squares problem to estimate the position and transmission power of the jammer:

$$\hat{\mathbf{v}} = \left[\hat{x}_J, \hat{y}_J, \hat{P}_J\right]^T = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b} \tag{20}$$

where $\mathbf{A}$ and $\mathbf{b}$ are matrices depending on the position of M boundary nodes $\{(x_m, y_m)\}_{m=1,\ldots,M}$ and their hearing range $\{r_{h_m}\}_{m=1,\ldots,M}$, *i.e.*, the range within which they can receive packets from other nodes, respectively.

$$\mathbf{A} = \begin{pmatrix} x_1 - \frac{1}{M}\sum_{m=1}^{M} x_m & y_1 - \frac{1}{M}\sum_{m=1}^{M} y_m & \frac{1}{2}(C(r_{h_1}) - C_\Sigma) \\ \vdots & \vdots & \vdots \\ x_M - \frac{1}{M}\sum_{m=1}^{M} x_m & y_M - \frac{1}{M}\sum_{m=1}^{M} y_m & \frac{1}{2}(C(r_{h_M}) - C_\Sigma) \end{pmatrix} \tag{21}$$

$$\mathbf{b} = \begin{pmatrix} \left(x_1^2 - \frac{1}{M}\sum_{m=1}^{M} x_m^2\right) + \left(y_1^2 - \frac{1}{M}\sum_{m=1}^{M} y_m^2\right) \\ \vdots \\ \left(x_M^2 - \frac{1}{M}\sum_{m=1}^{M} x_m^2\right) + \left(y_M^2 - \frac{1}{M}\sum_{m=1}^{M} y_m^2\right) \end{pmatrix} \tag{22}$$

and

$$C\left(r_{h_m}\right) = \frac{\gamma_0 r_{h_m}^2}{P_S - \frac{4\pi\gamma_0 P_N}{G} r_{h_m}^2}, \quad C_\Sigma = \frac{1}{M}\sum_{m=1}^{M} C\left(r_{h_m}\right). \tag{23}$$

Based on equation (7), we can get:

$$\frac{\frac{P_S G_{SR}}{4\pi d_{SR}^2}}{P_N + \frac{P_J G_{SR}}{4\pi d_{JR}^2}} = \gamma_0$$

$$(x - x_J)^2 - (y - y_J)^2 = P_J C(d_{SR}), \tag{24}$$

with $d_J^2 = (x - x_J)^2 - (y - y_J)^2$ where $(x$,$y)$ and $(x_J$,$y_J)$ are the coordinates of a boundary node and the jammer J respectively, and $C(d_{SR}) = \frac{\gamma_0 d_{SR}^2}{P_S - \frac{4\pi\gamma_0 P_N}{G_{SR}} d_{SR}^2}$. d_{SR} is the range within which the boundary node R can receive packets from other nodes, *e.g.*, any transmitter S within d_{SR} of R has $\mathrm{SNR}_{S\to R} \geq \gamma_0$ where γ_0 is the decodable SNR threshold.

Under jamming, suppose that the range of a boundary node m that can hear packets is changed to d_{SR_m} with $m = \{1, \dots, M\}$. Then, we can construct M equations for M boundary nodes:

$$\begin{gathered}
(x_1 - x_J)^2 + (y_1 - y_J)^2 = P_J C\left(d_{SR_1}\right) \\
(x_2 - x_J)^2 + (y_2 - y_J)^2 = P_J C\left(d_{SR_2}\right) \\
\vdots \\
(x_M - x_J)^2 + (y_M - y_J)^2 = P_J C\left(d_{SR_M}\right).
\end{gathered} \tag{25}$$

In Adaptive LSQ, d_{SR_m} can be estimated for each boundary node m and the jammer's position can be estimated by solving the above equations. Moreover, the non-linear equation groups can be linearized into the form of $\mathbf{Az} = \mathbf{b}$ with

$$\mathbf{A} = \begin{pmatrix} x_1 - \frac{1}{M}\sum_{m=1}^{M} x_i & y_1 - \frac{1}{M}\sum_{m=1}^{M} y_i & \frac{1}{2}(C(d_{S_1}) - C_\Sigma) \\ \vdots & \vdots & \vdots \\ x_M - \frac{1}{M}\sum_{m=1}^{M} x_M & y_M - \frac{1}{M}\sum_{m=1}^{M} y_M & \frac{1}{2}(C(d_{S_M}) - C_\Sigma) \end{pmatrix} \tag{26}$$

and

$$\mathbf{b} = \begin{pmatrix} \left(x_1^2 - \frac{1}{M}\sum_{i=1}^{M} x_i^2\right) + \left(y_1^2 - \frac{1}{M}\sum_{i=1}^{M} y_i^2\right) \\ \vdots \\ \left(x_M^2 - \frac{1}{M}\sum_{m=1}^{M} x_M^2\right) + \left(y_M^2 - \frac{1}{M}\sum_{m=1}^{M} y_i^2\right) \end{pmatrix} \tag{27}$$

where $C_\Sigma = \frac{1}{M}\sum_{m=1}^{M} C(d_{SR_m})$. Thus, we can calculate the estimation of the jammer's position and the jammer's transmission power by solving the least squares:

$$\mathbf{z} = [x_J, y_J, P_J]^T = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b}. \tag{28}$$

Moreover, the radio propagation in real-world scenarios is complex with random attenuation and multipath effects, Adaptive LSQ further combines the Centroid-based method with the basic LSQ algorithm to handle the radio irregularity. After introducing the basic algorithms that we adopt to use in our intelligent multi-jammer localizer, we next present how to localize multiple jammers based on the classification from network topology partitioning by continuously using the two-jammer example.

Centroid-based versus Adaptive LSQ. According to our prior work, the Adaptive LSQ is more likely to provide a better estimation of the jammer's location than the Centroid-based method, because Centroid-based is sensitive to the distribution of jammed nodes. However, such a conclusion is only valid under the assumption of a single jammer. When multiple jammers are present, it is sometimes difficult, even impossible, to identify which jammer or jammers disturb the communication of the boundary nodes. Thus, it is non-trivial to construct $\mathbf{A}$ and $\mathbf{b}$ for jammer localization using Adaptive LSQ. In those cases, the Centroid-based method will perform better. Regarding the computational complexity, Centroid-based method is proportional to the number of jammed nodes as $O(M_J)$, while Adaptive LSQ is approximately $O(M_B^3)$ due to 3 times of matrix (or its transpose, inverse matrix) multiplication, where M_J and M_B are the jammed and boundary nodes involved in the localization process.

7.2 Algorithms for localizing multiple jammers

When jammers have overlapped jamming regions, a single JC contains more than 1 jammer. Simply applying Adaptive LSQ or centroid method is not working. We develop Mirroring and Gauss-Newton searching methods to address this case sequentially and simultaneously turning on the case respectively.

Mirroring algorithm. When two jammers are sequentially turned on, the first jammer's location can be estimated by applying Adaptive LSQ to the portion of BC when only the first jammer is on. To localize the second jammer's position after it is on, since only one connected jammed area is formed, our assumption about the omni-direction characteristic of the propagation model and the uniform distribution of the nodes in the network implies that the two jammers will be at symmetric positions with respect to the center of the jammed region. Thus, the Mirroring algorithm uses the location estimation of the first jammer $\hat{J}_1$, and applies the Adaptive LSQ to the whole jammed area to obtain a position estimation $\hat{J}$ based on the single boundary cluster. Based on our assumption, the second jammer's position $\hat{J}_2$ can be estimated as a symmetric position of $\hat{J}_1$ with respect to the position estimation $\hat{J}$:

$$\hat{J}_2 = \hat{J} - \left(\hat{J}_1 - \hat{J}\right). \tag{29}$$

It can be found from the above equation that the computation complexity of the Mirroring algorithm is about $O(M_B^3+M_J)$, where M_J and M_B are the jammed and boundary nodes involved in the localization process.

Gauss-Newton Searching algorithm. When the jammers' turning-on sequence is not available, our framework treats two jammers being turned on simultaneously. 1JC-1BC makes estimating each jammer's position especially hard. We propose a method grounded on Gauss-Newton Searching to localize each jammer's position.

To localize more than one jammer within one single jammed cluster, we derive a general form of Gauss-Newton searching method for localizing these jammers. According to the SNR model, we can formulate the SNR at one particular boundary node m from multiple jammer's transmission power as follows:

$$\begin{gathered}\frac{\frac{P_sG}{4\pi r_{SR}^2}}{P_N + \sum_{m=1}^{M} \frac{P_JG}{4\pi r_{J_mR}^2}} \leq \gamma_0 \\ \sum_{m=1}^{M} \frac{\gamma_0 P_J}{r_{J_mR}} + \frac{4\pi\gamma_0 P_N}{G} + \frac{P_S}{r_{SR}^2} \geq 0 \\ \gamma_0 P_J \sum_{m=1}^{M} \prod_{n=1,n\neq m}^{M} r_{J_nR}^2 + C \prod_{m=1}^{M} r_{J_mR}^2 \geq 0\end{gathered} \tag{30}$$

where $C = \frac{4\pi\gamma_0 P_N}{G} - \frac{P_S}{r_{SR}^2}$.

The objective function minimizes the left term of the above equation and if there exists M such boundary nodes.

$$\arg\min_{\mathbf{v}} S(\mathbf{v}) = \arg\min_{\mathbf{v}} \left(\gamma_0 P_J \sum_{m=1}^{M} \prod_{n=1,n\neq m}^{M} r_{J_nR}^2 + C \prod_{m=1}^{M} r_{J_mR}^2\right). \tag{31}$$

To obtain the searching gradient, we can obtain the Jacobi matrix by calculating the derivative of the objective function as follows:

$$J_{\mathbf{f}} = \begin{bmatrix} \frac{\partial f_{r_1}(\mathbf{v})}{\partial x_{J_1}} & \frac{\partial f_{r_1}(\mathbf{v})}{\partial y_{J_1}} & \frac{\partial f_{r_1}(\mathbf{v})}{\partial x_{J_2}} & \frac{\partial f_{r_1}(\mathbf{v})}{\partial y_{J_2}} & \cdots & \frac{\partial f_{r_1}(\mathbf{v})}{\partial P_J} \\ \frac{\partial f_{r_2}(\mathbf{v})}{\partial x_{J_1}} & \frac{\partial f_{r_2}(\mathbf{v})}{\partial y_{J_1}} & \frac{\partial f_{r_2}(\mathbf{v})}{\partial x_{J_2}} & \frac{\partial f_{r_2}(\mathbf{v})}{\partial y_{J_2}} & \cdots & \frac{\partial f_{r_2}(\mathbf{v})}{\partial P_J} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{r_M}(\mathbf{v})}{\partial x_{J_1}} & \frac{\partial f_{r_M}(\mathbf{v})}{\partial y_{J_1}} & \frac{\partial f_{r_M}(\mathbf{v})}{\partial x_{J_2}} & \frac{\partial f_{r_M}(\mathbf{v})}{\partial y_{J_2}} & \cdots & \frac{\partial f_{r_M}(\mathbf{v})}{\partial P_J} \end{bmatrix}. \tag{32}$$

Each element in the Jacobi Matrix is represented as below:

$$\begin{aligned}\frac{\partial f_1(\mathbf{v})}{\partial x_{J_u}} &= \left(2\gamma_0 P_J \sum_{m=1}^{M} \left(x_{J_u} - x\right) \prod_{n=1,n\neq m}^{M} r_{J_nR}^2 + 2C\left(x_{J_u} - x\right) \prod_{m=1}^{M} r_{J_mR}^2\right) \\ \frac{\partial f_1(\mathbf{v})}{\partial y_{J_u}} &= \left(2\gamma_0 P_J \sum_{m=1}^{M} \left(y_{J_u} - y\right) \prod_{n=1,n\neq m}^{M} r_{J_nR}^2 + 2C\left(y_{J_u} - y\right) \prod_{m=1}^{M} r_{J_mR}^2\right)\end{aligned} \tag{33}$$

$$\frac{\partial f_1(\mathbf{v})}{\partial P_J} = \gamma_0 \sum_{m=1}^{M} \prod_{n\neq m} r_{J_nR}^2.$$

Therefore, by using the Gauss-Newton searching method, we can locate more than one jammer, $x_{J_m}, y_{J_m}, m = 1, \cdots, M$, in a single jammed cluster. Since the Gauss-Newton searching method is an iterative method in the search for regression parameters in a nonlinear regression model, its computational complexity is unknown, but it usually converges in less than 10 iterations according to our empirical study.

Algorithm 2. Localizing multiple jammers

```
Require: INPUT:
    $JC, BC$;
    OUTPUT:
    $\hat{J_i}, i \geq 1$;
PROCEDURES:
Sequentially Turning On:
while $DetectNewJammer()$ do
    if $DetectNewBC()$ then
        $BC_{New} = obtainNewBC()$;
        $\hat{J_{New}} = AdaptiveLSQ(BC_{New})$.
    else if $DetectNewJC()$ & $!DetectNewBC()$ then
        $JC_{New} = obtainNewJC()$;
        $\hat{J_{New}} = Centroid(JC_{New})$.
    else
        $\hat{J_{New}} = Mirroring(BC)$.
    end if
end while
Simultaneously Turning On:
for each $BC_i, i = 1, \cdots, \|BC\|$ do
    if $containJammer(BC_i) == 1$ then
        $\hat{J_i} = AdaptiveLSQ(BC_i)$.
    else
        for each $JC_i, i = 1, \cdots, \|JC\|$ do
            if $containJammer(JC_i) == 1$ then
                $\hat{J_i} = Centroid(JC_i)$
            else
                $\hat{J_i} = Gauss\text{-}Newton(JC_i)$
            end if
        end for
    end if
end for
```

7.3 Localization strategy

After introducing the localization algorithms, we present our intelligent multi-jammer localizer, which can localize multiple jammers based on the clustering results from the automatic network topology partitioner. The algorithmic flow of our intelligent multi-jammer localizer is displayed in Algorithm 2.

Based on the topology partitioning results, we will choose the appropriate localization methods introduced above to localize jammers. To illustrate the application of jamming algorithms chosen for dealing with different jamming scenarios, we categorize the jamming scenarios based on the jammedd and boundary cluster classification.

7.3.1 Sequentially turning on

For sequentially turning on the case, the localization algorithm starts working only when a new jammer turns on.

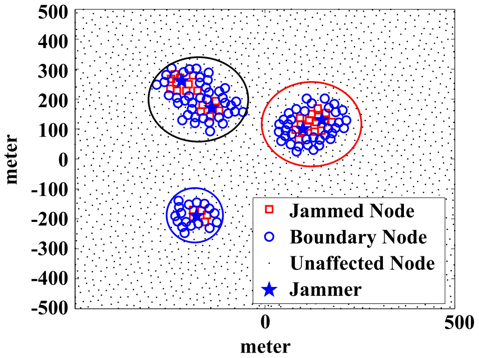


Figure 10. Illustration for different jamming scenarios

- **New BC appears:** new jammer resulting in new BC indicates that the new BC includes a single jammer. The new jammer inside the particular boundary cluster is shown as the blue ellipse in Figure 10. The jammer in a distinguishable BC has less impact from other jammers, therefore, we can directly apply the Adaptive-LSQ method for localizing this jammer.
- **New JC appears, no new BC appears:** new jammer resulting in new JC indicates that the new jammed cluster JC_1 only includes a single jammer shown as the black ellipse in Figure 10. The jammer in a distinguishable JC also has less impact from other jammers, therefore centroid method based on the jammed cluster can be used for localizing the jammer.
- **Neither new JC nor new BC appears:** new jammer results in no new BC and JC indicates that the new jammer is close to previous jammers. Both new jammers and previous jammers are included in single JC. Multiple jammers in a single JC are shown in the red ellipse in Figure 10. The estimated jammer gets impact from other jammers, hence the new jammer is localized leveraging the previously localized jammers' positions *via* Mirroring method.

7.3.2 Simultaneously turning on

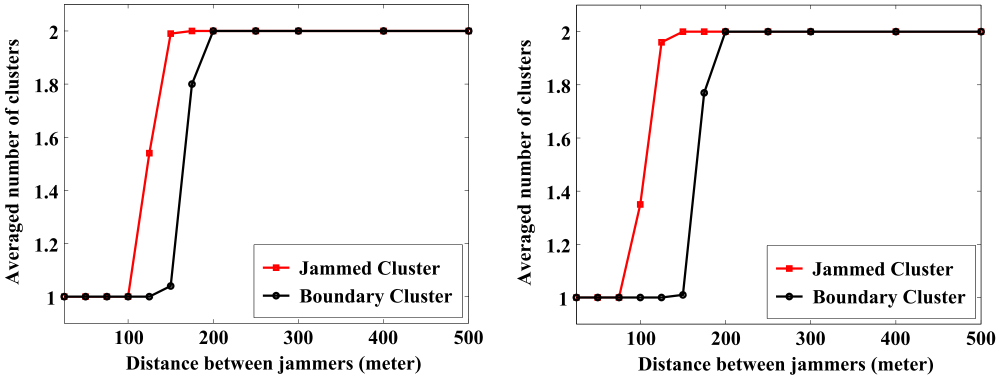
For simultaneously turning on the case, we localize each jammer from each individual BC:

- **One BC with one JC which includes only one jammer:** each jammer has only one jammed cluster, and this jammed cluster includes one single jammer shown as the blue circle in Figure 10. A jammer inside one particular boundary cluster has less impact from other jammers, which is distinguished by the corresponding BC as the blue ellipse in Figure 10. The jammer corresponds to only BC, the Adaptive-LSQ method is appropriate for localizing the involved jammer.
- **One BC with multiple JCs:**
  (i) **Each JC with one jammer:** each jammed cluster includes only one single jammer shown as the black circle in Figure 10. A particular jammed cluster within a boundary cluster has only one jammer inside, which is shown as the black ellipse in Figure 10. The jammer can be distinguished by the corresponding JC, which indicates that the centroid method based on the jammed cluster can be applied.
  (ii) **Each JC with multiple jammers:** each jammed cluster includes multiple jammers shown as the red circle in Figure 10. Multiple jammers are included in the single jammed cluster of a particular boundary cluster shown as the red ellipse in Figure 10, no single jammer can be distinguished based on either JC or BC. Therefore, all the jammers within one particular JC should be localized *via* Gauss-Newton searching method.

8 Simulation evaluation

8.1 Simulation setup and performance metrics

In an area of 1000-by-1000 square meters, we generated 1000 different network topologies with 2000 and 3000 nodes, respectively, to validate the effectiveness of our framework. The nodes were uniformly placed



(a)

(b)

Figure 11. Node topology partitioning study: average number of clusters as a function of the distance between two jammers (2000-node and 3000-node deployment with node transmission range set to 30 m). (a) 2000-node. (b) 3000-node

to cover the entire deployment region with a minimum distance bounded by a threshold. To study the effects of multiple jammers, we presented the results of two jammers with a transmission range of 60 m and a decodable SNR threshold γ_0 of 1.1. To emulate real-world scenarios, we developed our simulation under the shadowing model and tuned the parameters using those obtained from our empirical experimental study [11]. Specifically, we set the path loss exponent η to 2.11 and the standard deviation σ to 1.0.

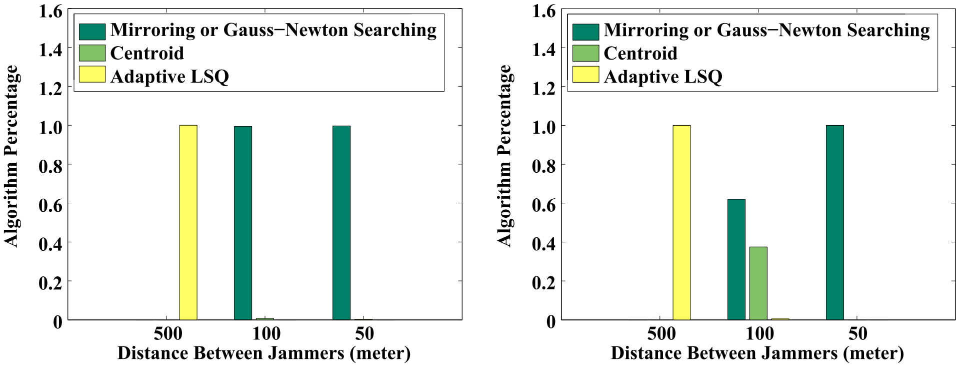
We evaluated the accuracy of localizing the jammers by defining the localization error as the Euclidean distance between the estimated jammer's location and the true location. To capture the statistical characteristics, we studied the average errors under multiple experimental rounds and used the median error and the Cumulative Distribution Function (CDF) of the localization error as our validation metrics.

8.2 Node topology partitioning study

We first studied the results of the automatic network topology partitioner when varying the distance between two jammers. Figure 11 depicts the average number of clusters obtained from our network topology partitioner as a function of the distance between jammers in a 2000-node network and a 3000-node network when setting each node's transmission range to 30 m. We observed that both the number of JC and BC starts from 1 when two jammers are placed closely, and then jumps to 2 when two jammers are moving away from each other. Particularly, 2JC-2BC is returned when the distance $L_J > 200$ m in both networks, and 1JC-1BC is returned when $L_J < 100$ m in the 2000-node network and $L_J < 75$ m in the 3000-node network, respectively. Finally, all three clustering results are possible when L_J falls in between. Particularly, under the 2000-node deployment, only 1JC and 1BC are obtained when L_J is less than 100 m. Then the number of JC and BC goes up to 2 when the distance L_J is over 150 m and 200 m respectively. Whereas for the 3000-node deployment, when the distance L_J between two jammers is less than 75 m, there is always only 1JC and 1BC returned. When the two jammers' distance L_J is over 125 m, the number of the jammed cluster first jumps to 2, and later the number of the boundary cluster becomes 2 when the two jammers are about 200 m away. When L_J is beyond 200 m, the topology partitioning always returns 2JC and 2BC. This observation confirms that our network topology partitioner works flexibly when the distance between jammers varies.

8.3 Localization algorithm selection study

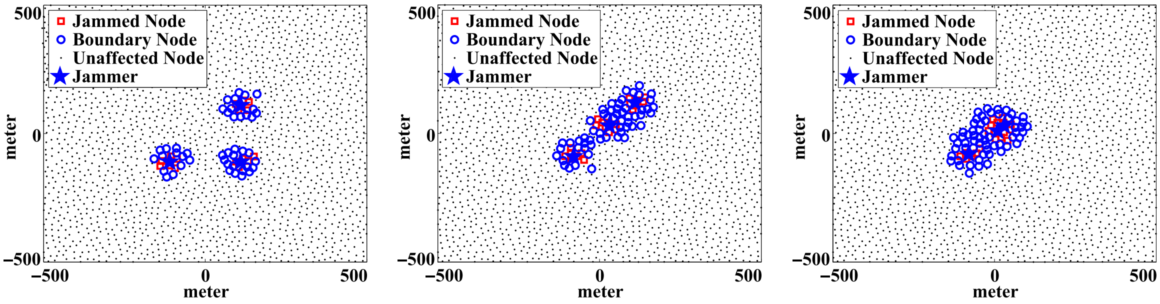
We investigated how our multi-jammer localizer behaves when the distance between two jammers changes. Figure 12 shows the percentage of the basic localization algorithm usage within our multi-jammer localizer for three different distances: 500 m, 100 m, and 50 m. For the case when the partitioner returns 2JC-2BC, which occurs when the jammers are 500 m away, our multi-jammer localizer always uses Adaptive LSQ to perform localization.



(a)

(b)

Figure 12. Usage of localization algorithms in our multi-jammer localizer with node transmission range setting to 30 m. (a) 2000-node. (b) 3000-node



(a)

(b)

(c)

Figure 13. Jamming topology partition results under 3 Jammers. (a) 3J-3B. (b) 3J-1B. (c) 2J-1B

However, when the jammers are closer to each other, such as around $L_J = 100\,\text{m}$, the usage of basic localization algorithms changes. For example, in the 2000-node deployment, the mirroring algorithm or Gauss-Newton Searching method dominates and conducts localization 98% of the time because 1JC-1BC is classified by topology partitioner for 98% of the jamming scenarios. Similarly, in the 3000-node deployment, the usage of the Centroid-based method is about 60%, and the mirroring algorithm and Gauss-Newton Searching method is about 40%, as shown in Figure 12b, where over half of the jamming scenarios are classified as 2JC-1BC and less than half is classified as 1JC-1BC. Finally, when the jammers are very close to each other (*i.e.*, $L_J = 50\,\text{m}$), the usage of the mirroring algorithm and Gauss-Newton Searching method is over 99%, matching with the 1JC-1BC topology partitioning case.

8.4 Localization performance study

For localizing multiple jammers, we study the localization performance by taking the example with 3 jammers both sequentially and simultaneously turning on for illustration. The different locations of multiple jammers would result in different jamming topology partitions, the algorithms selected are also different. For different jamming topology partitions, our proposed jammer localization scheme produces good performance. Note that for the jammers sequentially turning on, the first jammer is always localized with the Adaptive LSQ method.

8.4.1 Sequentially turning on

For sequentially turning on jammers, if one new jammer only results in one new jammed cluster and one new boundary cluster due to the large distance between each other, we always apply the Adaptive-LSQ method to localize it. As a result, for 3 jammers, we have 3 jammed clusters and 3 boundary cluster (3JC-3BC), and the localization results is shown in Figure 13a. For sequential cases, the localization

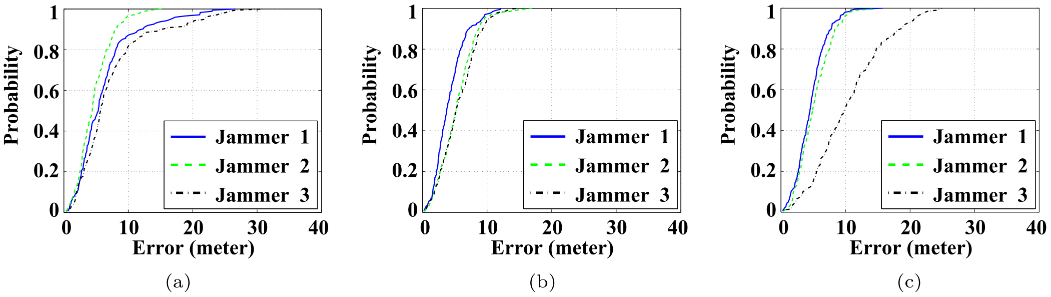


Figure 14. Localization error for different cluster combinations for 3 sequentially turning on jammers. (a) 3J-3B. (b) 3J-1B. (c) 2J-1B

accuracy of a particular jammer has little impact from its previous jammer due to the large distance between jammers, *i.e.*, the median errors for all three jammers are around 4.1 m as shown in Figure 14a.

If one new jammer results in one new jammed cluster and no new boundary cluster when it turns on, the centroid method is suitable for this case. For 3JC-1BC, the boundary clusters for all the jammers merge into one boundary cluster, since all three jammers are closer to each other than 3JC-3BC scenario. However, each jammer still maintains its own jammed cluster as shown in Figure 13b. Under such a jamming scenario, except for the first jammer, Adaptive LSQ will not be applicable here for the two following jammers, therefore we turn to the centroid method. The previous jammer will have more impact on the localization accuracy of the following jammers than the 3JC-3BC scenario, so the localization performance would degrade. Further, since the centroid method is also sensitive to node distribution, that is also one reason leading to worse performance.

From Figure 14c, we observe that the jammer localized with the centroid method has a median error of 5.2 m, which is almost 1 m worse than the first jammer localized with the Adaptive LSQ method. If one new jammer does not incur a new jammed cluster and boundary cluster, which means the new jammer shares the same jammed cluster and boundary cluster with its previous jammers as shown in Figure 13c, the Mirroring method will be exploited. For the Mirroring method, since the new jammer is estimated based on the estimated positions of its previous jammers, the impact from the previous jammer on localization accuracy would be even larger than in 3JC-1BC scenarios. We find in Figure 14c, that the median localization error for the jammer localized with the Mirroring method is about 9.9 m, and the other two jammers also have a median error of around 4.1 m.

8.4.2 Simultaneously turning on

Different from the sequential case, the jamming topology is the only information we obtained for simultaneously turning on jammers. If both the number of jammed clusters and boundary clusters equal to the number of jammers, that means each jammer forms an independent jammed and boundary cluster. Adaptive-LSQ would be the most accurate localization method for this case. As shown in Figure 15a, due to little impact with each other, the median errors for 3 jammers have a similar median error of around 4.1 m, which is almost the same as the sequential case with 3JC-3BC.

If the number of jammed clusters is equal to the number of jammers, but the number of boundary clusters is less, the Centroid method will be applied to the boundary cluster including multiple jammers. A typical case is 3JC-1BC as shown in Figure 15b, only centroid method is employed for localizing all three jammers. We observe that the median errors for the jammers localized with the centroid method are around 4.9 m.

If both the number of jammed cluster and boundary clusters are less than the number of jammers, some jammers must share the same jammed and boundary cluster. For 2JC-1BC scenarios, two of the jammers are within the same jammed and boundary cluster, so the Gauss-Newton searching method will be applied to these two jammers. In addition, the other jammer is localized with centroid method due to no distinct boundary cluster. Figure 15c shows the median localization error of about 14.5 and 16.6 m with the Gauss-Newton method, and the third jammer with a median error of around 4.7 m.

The observations above confirm the feasibility of our proposed multiple jammer localization strategies. For different network topology partitioning results, an appropriate jammer localization algorithm can be

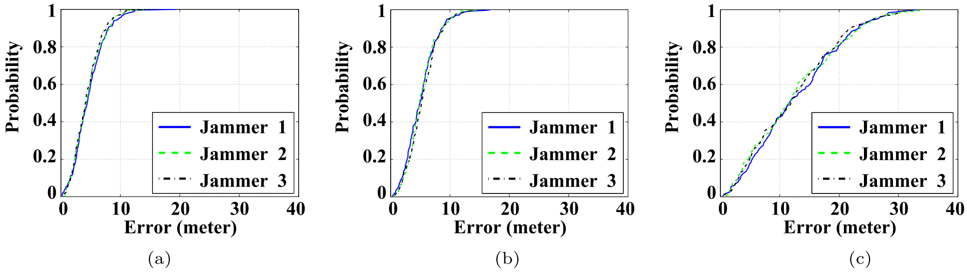


Figure 15. Localization error for different cluster combinations for 3 simultaneously turning on jammers. (a) 3J-3B. (b) 3J-1B. (c) 2J-1B

applied to achieve good localization accuracy. Particularly, for sequentially turning on jammers, previous jammers would affect the localization accuracy of the following jammers, but our proposed method would mitigate such impact to get better localization results for the following jammers. Whereas for simultaneously turning on jammers, the topology partitioning becomes much more complex due to mutual impact among multiple jammers, however, our strategy still works on providing good location estimation for each jammer. Further, our algorithm is generic to the localization problem for any number of jammers with or without the knowledge of the turning-on order.

8.4.3 Distance study between two jammers

We studied the effect of node transmission range on the localization accuracy by setting the distance between jammers to $\{500\,\mathrm{m}, 100\,\mathrm{m}, 50\,\mathrm{m}\}$, respectively, with the node transmission range $\{35\,\mathrm{m}, 45\,\mathrm{m}, 55\,\mathrm{m}\}$. Table 2 summarizes the distribution of clustering results, *i.e.*, 2JC-2BC, 2JC-1BC, and 1JC-1BC, for all settings. The corresponding median error of localization is presented in Table 2.

In general, changing the node transmission range does not significantly affect the localization error 2 when jammers are turned on sequentially, with localization error always between 3 m and 4 m. However, when jammers are simultaneously turned on, the localization accuracy under $L_J = 500\,\mathrm{m}$ degrades with increasing node transmission range, while the performance under $L_J \leq 100\,\mathrm{m}$ improves with increasing node transmission range.

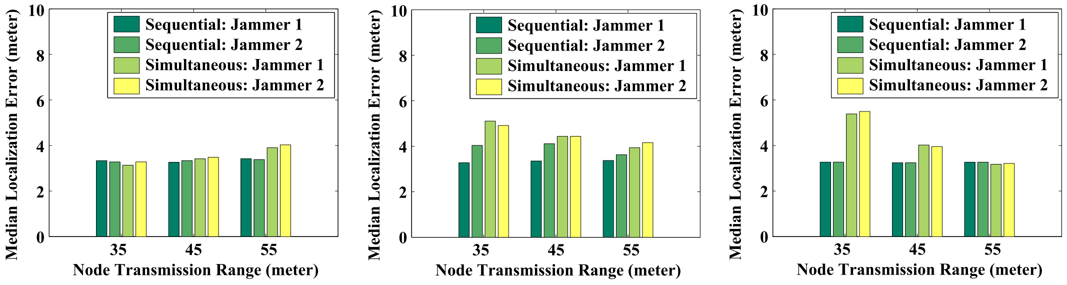
This is because, under larger distances L_J, the intelligent multi-jammer localizer uses the Adaptive LSQ method to localize jammers. When the node transmission range increases, the number of boundary nodes reduces, and the number of constraints for the Adaptive LSQ method also decreases, leading to a degradation in localization accuracy. On the other hand, under smaller distances L_J, the dominant algorithms selected by the intelligent multi-jammer localizer are Centroid-based, Mirroring, and Gauss-Newton Searching methods. As the node transmission range increases, the interference between each jammer's JC (or BC) decreases, leading to an improvement in localization performance.

When $L_J = 500\,\mathrm{m}$, where two jammers are far away from each other, it is more likely that they are located in two different boundary clusters, which indicates that there is less mutual impact on location estimation for these two jammers. The Adaptive LSQ method will dominate, and high localization accuracy will be achieved for both jammers. Particularly, the localization accuracy is around 3.5 m when the node transmission range varies for both cases of sequentially and simultaneously turning on jammers as shown in Figure 16a.

When $L_J = 100\,\mathrm{m}$, where two jammers approach each other, their boundary clusters will be integrated into one, but jammers clusters remain separate. Without a third jammer, it is appropriate to choose the centroid method for localizing these two jammers. However, the position estimation accuracy will decrease a little. Particularly, the centroid-based method dominates and performs localization for over 75% of scenarios as displayed in Table 2. When two jammers are sequentially turned on, the localization error is between 3.2 m and 4 m as shown in Figure 16b, and the localization accuracy of the second jammer underperforms that of the first jammer. This is because the jammed cluster formed by the second jammer is interfered with by the first jammer when two jammers are placed close by. As the node transmission range increases, the interference is weakened. When two jammers are simultaneously turned on, the localization error presents a decreasing trend, from 5 m to 4 m, when the node transmission range increases. This indicates that the increasing node transmission range improves the localization performance.

Table 2. Distribution of number of clusters with various node transmission ranges under different jammers' distances

$L_J(m)$	Node Trans. Range (m)	2JC-2BC	2JC-1BC	1JC-1BC
500	35	100%	0%	0%
	45	100%	0%	0%
	55	100%	0%	0%
100	35	0%	77%	23%
	45	0%	92.4%	7.6%
	55	0%	87.9%	12.1%
50	35	0%	0%	100%
	45	0%	0%	100%
	55	0%	0%	100%



(a) (b) (c)

Figure 16. Median localization error as a function of the node transmission range under the 3000-node deployment. (a) $L_J = 500\,\text{m}$. (b) $L_J = 100\,\text{m}$. (c) $L_J = 50\,\text{m}$

Finally, when $L_J = 50\,\text{m}$, where two jammers get even closer, the jammed clusters also merge into one. Only the Mirroring or Gauss-Newton method can be chosen, and the localization performance will be degraded. Such performance difference is caused by the fact that both the jammed cluster and the boundary cluster of two jammers are largely overlapping due to the close proximity of the two jammers, making it extremely hard to locate each individual jammer without prior knowledge. This is the case of 1JC-1BC, whereby Mirroring algorithms are selected for sequentially turning-on cases and the Gauss-Newton Searching method is selected for simultaneously turning-on cases. As shown in Figure 16c, when jammers are sequentially turned on, the localization conducted by the dominating Mirroring algorithm achieves a similar performance (around 3.5 m) to that of when $L_J = 500\,\text{m}$. When jammers are simultaneously turned on, the median localization error *via* the Gauss-Newton Searching method reduces from 5.5 m to 3.1 m when the node transmission range increases from 35 m to 55 m, suggesting that the Gauss-Newton Searching method also benefits from larger node transmission ranges. We note that in practice the attackers may not desire to place two jammers in the vicinity as it reduces the overall jamming effects in the network. Instead, the attackers would prefer to place two jammers farther away from each other to cause network communication disturbance in a large area.

9 Conclusion

In this paper, we addressed the problem of localizing jamming attackers when multiple jammers are present in a wireless network. Our jammers can be intentional jamming attackers and unintentional radio interfers coexisting in the network. We proposed to identify the physical position of jammers by leveraging the network topology changes caused by jamming. In particular, we studied the jamming effects under multiple jammers and developed a framework that can perform critical tasks of automatic network topology partitioning and intelligent multi-jammer localization. Our approach does not depend on measuring signal strength inside the jammed area, nor does it require delivering information out of the jammed area. Instead, our framework uses disturbed network communication and derives node clusters for jammer localization grounded on network topology changes. Our experimental results on a multi-hop network using MicaZ sensor nodes showed that we can successfully collect real-time network topology changes under jamming, and thus confirmed the feasibility of applying our approach in practice. In addition to utilizing the existing jammer localization algorithms, *e.g.*, Adaptive LSQ and Centroid-based methods, we developed two new algorithms, namely Mirroring and Gauss-Newton Searching algorithms, that are particularly effective when multiple jammers create one connected jamming area. We evaluated the performance of our multi-jammer localizer through simulation using large-scale network setups with various distances between jammers. Our simulation results indicated that the multi-jammer localizer can intelligently use appropriate localization strategies to estimate the position of jammers and achieve comparable accuracy to localize a single jammer.

Range-free based localization heavily relies on the distribution of network nodes. Our future work will focus on the impact of network nodes distribution on the accuracy of jamming localization leveraging the tool of stochastic geometry. Furthermore, the jamming behaviors (*i.e.*, the variation of jamming power, the duration and interval of jamming signal, and the mobility of jammer, etc.) also cause great trouble to the localization strategies. Therefore, a more intelligent framework integrating jamming detection, identification, and localization is in demand. In addition, we will also take more powerful jammers equipped with directional antennae into consideration.

Conflict of Interest

The author declares no conflict of interest.

Data Availability

No data are associated with this article.

Authors' Contributions

Hongbo Liu proposed the overall multi-jammer localization framework and wrote this paper. Yingying Chen, Wenyuan Xu, and Zhenhua Liu contributed to the experimental design and embellished the whole paper. Yuchen Su participated in manuscript preparation and improved the readability of the paper by grammatical modification and polishing. All authors read and approved the final manuscript.

Acknowledgements

Thanks to the anonymous reviewers for their helpful comments and suggestions.

Funding

This work is partially supported by the National Natural Science Foundation of China under Grant No. 62172080, the Natural Science Foundation of Sichuan Province under Grants 2023NSFSC0478, and the National Key R&D Program of China No. 2022YFB3103404.

References

[1] Xu W, Trappe W and Zhang Y et al. The feasibility of launching and detecting jamming attacks in wireless networks. In: Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc). ACM, 2005, 46–57.

[2] Proakis JG. Digital Communications, 4th ed. McGraw-Hill, 2000.

[3] Noubir G and Lin G. Low-power DoS attacks in data wireless lans and countermeasures. ACM SIGMOBILE Mob Comput Commun Rev 2003; **7**: 29–30.

[4] Xu W, Trappe W and Zhang Y. Channel surfing: defending wireless sensor networks from interference. In: Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN), 2007, 499–508.

[5] Want R, Hopper A and Falcao V et al. The active badge location system. ACM Trans Inf Syst 1992; **10**: 91–102.

[6] Bahl P and Padmanabhan VN. RADAR: an in-building RF-based user location and tracking system. In: Proceedings of the IEEE International Conference on Computer Communications (INFOCOM). IEEE, March 2000, 775–84.

[7] Chen Y, Francisco J and Trappe W et al. 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). IEEE, September 2006.

[8] Priyantha N, Chakraborty A and Balakrishnan H. The cricket location-support system. In: Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom). ACM, August 2000, 32–43.

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

[10] Pelechrinis K, Koutsopoulos I and Broustis I et al. Lightweight jammer localization in wireless networks: system design and implementation. In: Proceedings of the IEEE Global Communication Conference (GLOBECOM). IEEE, December 2009.

[11] Liu Z, Liu H and and Xu W et al. Wireless jamming localization by exploiting nodes' hearing ranges. In: Proceedings of the International Conference on Distributed Computing in Sensor Systems (DCOSS). Springer Berlin Heidelberg, 2010.

[12] Liu H, Liu Z and Chen Y et al. Localizing multiple jamming attackers in wireless networks. In: 2011 31st International Conference on Distributed Computing Systems. Minneapolis, MN, USA: IEEE, 2011, 517–28.

[13] Çakıroğlu M and Özcerit AT. Jamming detection mechanisms for wireless sensor networks. In: Proceedings of the 3rd International ICST Conference on Scalable Information Systems (INFOSCALE). ICST, May 2010, 1–8.

[14] Navda V, Bohra A and Ganguly S et al. Using channel hopping to increase 802.11 resilience to jamming attacks. In: IEEE Infocom Minisymposium. IEEE, May 2007, 2526–30.

[15] Khattab S, Mosse D and Melhem R. Modeling of the channel-hopping anti-jamming defense in multi-radio wireless networks. In: Proceedings of the 5th Annual International Conference on Mobile and Ubiquitous Systems (Mobiquitous). Brussels, Belgium: ICST, 2008, 1–10.

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

[17] Xu W, Trappe W and Zhang Y. Anti-jamming timing channels for wireless networks. In: Proceedings of the First ACM Conference on Wireless Network Security (WiSec). New York, NY, USA: ACM, 2008, 203–13.

[18] Cagalj M, Capkun S and Hubaux J. Wormhole-based anti-jamming techniques in sensor networks. In: IEEE Transactions on Mobile Computing. IEEE, January 2007, 100–14.

[19] Ward A, Jones A and Hopper A. A new location technique for the active office. IEEE Pers Commun 1997; **4**: 42–7.

[20] Chen Y, Kleisouris K and Li X et al. 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). Springer Berlin Heidelberg, June 2006, 546–63.

[21] Chandrasekaran G, Ergin MA and Yang J et al. 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). IEEE, June 2009.

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

[23] He T, Huang C and Blum B et al. Range-free localization schemes in large scale sensor networks. In: Proceedings of the Ninth Annual ACM International Conference on Mobile Computing and Networking (MobiCom). ACM, 2003.

[24] Bulusu N, Heidemann J and Estrin D. GPS-less low-cost outdoor localization for very small devices. IEEE Pers Commun Mag 2000; **7**: 28–34.

[25] Niculescu D and Nath B. Ad hoc positioning system (APS). In: Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM). IEEE, 2001, 2926–31.

[26] Shang Y, Ruml W and Zhang Y et al. Localization from mere connectivity. In: Proceedings of the Fourth ACM International Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc). ACM, Jun 2003, 201–12.

[27] Kim YS, Mokaya F and Chen E et al. All your jammers belong to us – localization of wireless sensors under jamming attack. In: Proceedings of International Communication Conference (ICC). IEEE, 2012.

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

[29] A. Goldsmith, Wireless Communications. Cambridge: Cambridge University Press, 2005.

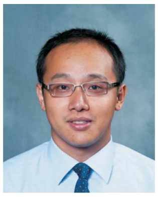
[30] Aceinna, Aceinna. http://www.aceinna.com/wireless-sensor-networks/.

[31] Cormen TH, Leiserson CE and Rivest RL et al. Introduction to Algorithms. Cambridge, MA: MIT Press, 2001.

Hongbo Liu is a Professor with the University of Electronic Science and Technology of China, Chengdu, China. His research interests include wireless sensing and mobile computing, and cyber security and privacy.

Yingying (Jennifer) Chen is a Professor of Electrical and Computer Engineering with Rutgers University, where she is a member of the Wireless Information Network Laboratory, and also leads the Data Analysis and Information Security Laboratory. Her research interests include smart healthcare, cybersecurity and privacy, Internet of Things, and mobile computing and sensing.

Wenyuan Xu is currently a Professor with the College of Electrical Engineering, Zhejiang University. Her research interests include wireless networking, network security, and the IoT security.



Zhenhua Liu is currently working in the Arena for Research on Emerging Networks and Applications (ARENA) lab. His research interests include wireless PHY/MAC layer security, jamming/radio interference, wireless localization/tracking and mobile computing.

Yuchen Su received his bachelor's degree from University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2022. He is pursuing the master's degree in cyberspace security with the UESTC. His research interests include mobile computing and IoT security.