2011 31st International Conference on Distributed Computing Systems

Localizing Multiple Jamming Attackers in Wireless Networks

Hongbo Liu*, Zhenhua Liu†, Yingying Chen*, Wenyuan Xu†
*Dept. of ECE, Stevens Institute of Technology
Castle Point on Hudson, Hoboken, NJ 07030
{hliu3, yingying.chen}@stevens.edu
† Dept. of CSE, University of South Carolina
Columbia, SC 29208
{liuz,wyxu}@cse.sc.edu

***Abstract*—Jamming attacks and unintentional radio interference are 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. 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.**

I. 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 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 [4].

Those defense technologies provide useful methods to alleviate jamming. However, they primarily reply 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 an unintentional radio interferer that is turned on accidentally. In light of the benefits, in this paper, we address the problem of localizing the position of jammers when multiple jamming attackers coexist in a wireless network.

Although there have 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.

In the area of localizing jammers, a few algorithms [9], [10] have been proposed to localize one jammer. Without presenting performance evaluation, Pelechrinis *et al.* [10] proposed to localize the jammer by performing gradient descent search based on packet delivery rate. 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. In this case,

1063-6927/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDCS.2011.38

prior work cannot identify the locations of all jammers. 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 *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 simulation 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 the prior knowledge of the order that the jammers are turned on.

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

II. Related Work

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

Countermeasures for coping with jamming in commodity wireless networks have been intensively investigated. Defense strategies include the use of error correcting codes [3] to increase the likelihood of decoding corrupted packets, channel hopping [4] to adapt the working channel to escape from jamming, and wormhole-based anti-jamming techniques [13].

Wireless localization has been an active area, attracting many attentions. Based on localization infrastructure, infrared [5] and ultrasound [8] are employed to perform localization, both of which need to deploy specialized infrastructure for localization. Further, using received signal strength (RSS) [7] is an attractive approach because it can reuse the existing wireless infrastructure. Based on the localization methodology, the localization algorithms can be categorized into range-based and range-free. Range-based algorithms involve estimating distance to anchor points with known locations by utilizing the measurement of various physical properties, such as RSS [7], Time Of Arrival [14], and Time Difference of Arrival [8]. Range-free algorithms [15] use coarser metrics to place bounds on candidate positions. However, most of these approaches are inapplicable to localize jammers as the jammer will not cooperate and the regular radio signal is disturbed under jamming.

Recently, a few work has been focused on determining the location of one jammer. Without presenting performance evaluation, Pelechrinis et al. [10] proposed to localize the jamming by measuring packet delivery rate (PDR) and performing gradient decent search. Liu *et al.* [9] utilized the network topology changes caused by jamming attacks and estimated the jammer's position by introducing the concept of virtual forces. The virtual forces are derived from the node states and can guide the estimated location of the jammer towards its true position iteratively. Both algorithms [9], [10] require to search for the jammer location iteratively. To localize the jammer in one round, Liu *et al.* [11] developed a lease-squares-based algorithm that leverages the change of hearing range caused by jammming. Aforementioned algorithms can only localize one jammer and may fail to yield jammers' positions when multiple ones are present. In this paper, we address the problem of localizing multiple jamming attackers.

III. Model

In this section, we first present the adversary model and network model that our work focuses on. We then provide an analysis of jamming effects when multiple jammers are present in the network.

A. Adversary Model

We consider multiple jammers present in the network and focus on localizing each of them. Regardless of the attack strategies, the consequences of various jammers are the same: For those nodes which are located near a jammer, their communication is severely disrupted, whereas a node which is far away from a jammer may not be affected by the jammer at all. Thus, we do not investigate diverse jamming attack philosophies but assume that jammers transmit at the same and fixed power level. Without loss of generality, we consider the scenarios that *two* jammers transmit at the same power level and become active either sequentially or simultaneously. The case that two jammers are turned on simultaneously presents greater challenges than the sequential case, since the former does not reveal any information of the individual jammer but all of them as a whole. We note that our strategies can be extended easily to localize more than two jammers.

B. Network Model

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

- **Stationary.** The network nodes that we consider in this work are stationary after their deployment. We will investigate mobile nodes in our future works.

- **Neighbor-Aware.** Each node is equipped with an omnidirectional antenna and transmits at the same transmission power level. Thus, every node shares the same communication range and can only receive messages from nodes that are within its communication range. We call the nodes that can send messages to node B B *s neighbors*. Each node maintains a neighbor table containing its neighbor IDs and tracks the changes in its neighbor table. Such a neighbor table 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 of wireless devices are equipped with GPS or some other approximate but less burdensome localization algorithms [6], [7].
- **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], [16].

C. Analysis of Jamming Effects

1) General Jamming Effects: To provide a complete depiction of the complex relationships between the transmission power of a wireless node and a jammer, we consider the signal-to-noise-ratio (SNR) model. In a jamming scenario, the "noise" includes ambient noise P_N and jamming signals P_J, and the SNR can be expressed as following by using a sender-receiver pair (S, R):

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

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

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

where SNR_{ij} is the SNR measured at node n_j when node n_i is transmitting and all other network nodes remain silent. γ_o is the threshold SNR, above which packets can be received successfully, and we call it **Decodable SNR threshold** [11].

When jammers are present in the network, the network nodes can be classified into three categories according to the impact of jamming: *unaffected node* N_U, *jammed node* N_J, and *boundary node* N_B. Let $Nbr\{n_i\}$ be the set of neighbors of node n_i before any jammer becomes active: and we formally derive the SNR-based jamming model as follows,

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

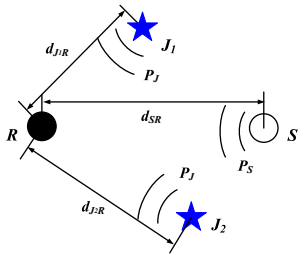


Fig. 1. An illustration of a multi-jammer scenario in a wireless network.

2) Effects of Multiple Jammers: In the 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). We 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 SNR at receiver R in the presence of Jammer J_1 becomes:

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

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 SNR-based jamming model at R is changed to:

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

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

$$l_{ij} = \begin{cases} 0 \to 0 & SNR^1_{ij} \leq \gamma_0 \to SNR^{1,2}_{ij} \leq \gamma_0 \\ 1 \to 0 & SNR^1_{ij} > \gamma_0 \to SNR^{1,2}_{ij} \leq \gamma_0 \\ 1 \to 1 & SNR^1_{ij} > \gamma_0 \to SNR^{1,2}_{ij} > \gamma_0 \end{cases} \tag{5}$$

For instance, a link state l_{ij} changes from 1 to 0, if the SNR 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 SNR at receiver R is similar to the SNR after both jammers are turned on sequentially, and the link state from node n_i to n_j is

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

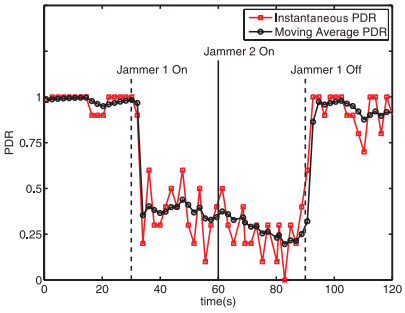


Fig. 2. Instantaneous PDR and exponential moving average of PDR from node 11 to node 8, when two jammers were turned on and off in sequence.

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 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 are the transmission power of the sender; G is the antenna field patterns in the line-of-sight (LOS) direction between 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.

Shadowing Model captures both path loss versus distance and the random attenuation due to blockage from objects in the signal path [17]. Let path loss at the receiver that is at the distance d from the sender be

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

then the shadowing model has the following form:

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

where $PL(d_0)$ 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.

IV. Collecting Network Topology Information

Our basic idea of localizing multiple jamming attackers is to estimate the positions of jammers utilizing network topology changes caused by jammers. Thus, it is essential to capture the topology differences prior to and after the emergence of jammers. Section III presents important theoretical underpinnings to understand the impact of a jammer to the network. For instance, the likelihood that n_i can receive messages from n_j is determined by signal-to-noise-ratio (SNR) at n_i when n_j is transmitting. In practice, however, few wireless devices can measure SNR. In this section, we present our experimental study on collecting network topology information in real time

Fig. 3. A snapshot of experiment setup.

and on classifying nodes into three categories: *unaffected nodes, jammed nodes, and boundary nodes*. We chose to perform our study using MicaZ sensor nodes [18] because they provide access to the entire network stacks. MicaZ sensor nodes have a 2.4-2.48 GHz Chipcon CC2420 Radio and use TinyOS 2.x as the operating system.

A. Link State Estimation and Information Collection

We envision that each node updates its neighbor table locally by measuring the link quality to each individual neighbor, and reports the neighbor table periodically to a dedicated entity (e.g., the network sink) that will localize jammers.

We estimated the link quality by measuring the percentage of packets delivered. Let the instantaneous Packet Delivery Ratio (PDR) from n_j to n_i at the kth interval be $p_{ij}^k = \frac{m_r}{m_t}$, 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. The link quality can be defined as the exponential moving average of instantaneous PDRs. The link quality from n_j to n_i at the kth interval is

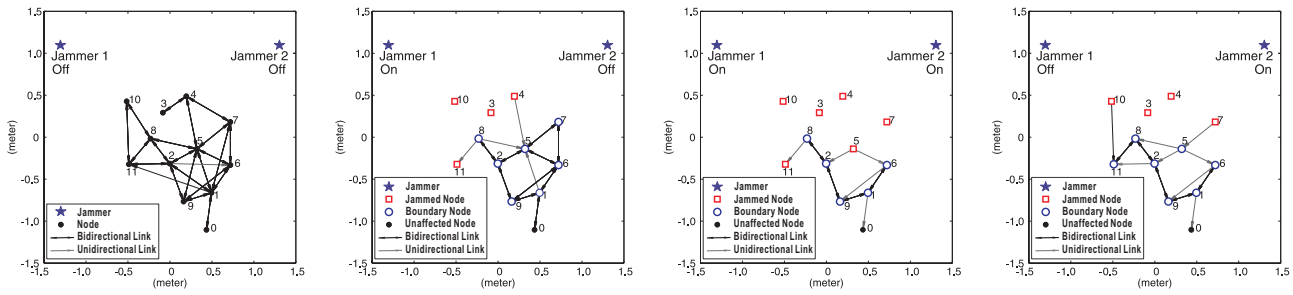
$$q_{ij}^k = \begin{cases} (1-\alpha)q_{ij}^{k-1} + \alpha p_{ij}^k & \Delta_l^k < \beta_1 \\ \alpha q_{ij}^{k-1} + (1-\alpha)p_{ij}^k & \texttt{otherwise}, \end{cases} \tag{10}$$

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.

A small α discounts older link estimations more slowly, and can smooth out short term fluctuations. However, when a jammer is turned on, it also imposes delays before the estimation reflects the latest network condition, e.g. jammed. To address this problem, we examined the instantaneous PDRs in the past l intervals, and give p_{ij}^k a high weight if the changes of p_{ij}^k have exceeded the short term fluctuation range, e.g., β_1. Furthermore, we defined the link state from node n_i to n_j as

$$l_{ij}^k = \begin{cases} 1 & q_{ij}^k > \beta_2 \\ 0 & \texttt{otherwise} \end{cases} \tag{11}$$

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.



(a) (b) (c) (d)

Fig. 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, and (d) J_1 was turned off at the 90th second.

We noted that J_2 has a small impact to the link quality, in this case.

Each node can monitor its link qualities with all its neighbors and can deliver them to the designated node for jamming localization. To collect data, we customized a protocol based on existing Collection Tree Protocol(CTP) implemented in TinyOS 2.x. to glean nodes' neighbor lists. In particular, a routing tree rooted at the designated node was built after the network was deployed, and each node periodically unicasted a data packet containing its latest neighbor list to the designated node.

B. An Example Walk-Through Using MicaZ

To study the impact of jamming, we deployed 12 nodes in an indoor environment. To make the network fit the room, we reduced the communication range to approximately 0.8 meter by installing a 10 dB attenuator on each Micaz node. We placed J_1 at the upper-left corner and J_2 at the upper-right corner, as shown in Figure 3. We selected node 0 as the designated node. All nodes periodically report their neighbor list to node 0 through a routing tree rooted at node 0, and node 0 learned the network topology from the received neighbor lists. As an example, the network topology prior to turning on any jammer is depicted in Figure 4(a). A single-headed arrow pointing from n_i to n_j indicates $l_{ij} = 1$ and a double-headed arrow represents a bidirectional link, e.g., $l_{ij} = l_{ji} = 1$. We note that all links in the networks are bidirectional without the interference from jammers.

We turned on the first jammer J_1 at the 30th second, the second jammer J_2 at the 60th second, and turned off J_1 at the 90th second. The network topologies for each stage are depicted in Figure 4, from which we had the following observations:

- Some of the links are no longer bidirectional because of jamming. For instance, the link state from node 8 to node 11, $l_{8,11}$, is connected as shown in Figure 4 (a), but $l_{11,8}$ is not after J_1 is on [1].
- The experiments confirmed our analysis about the effects of multiple jammers on network topology changes, and suggested that the network is able to identify the order that two jammers become active. In particular, once we turned on J_1 in the upper-left corner, as shown in Figure 4 (b), nodes $\{5, 7\}$ become boundary nodes, since they can still receive messages but lost some of their neighbors. After J_2 was turned on, as shown in Figure 4 (c), nodes $\{5, 7\}$ changed into jammed nodes, because they can no longer receive messages from any of their neighbors. After we turned off J_1, node 5 regained its ability to deliver messages to others and became a boundary node, but node 7 remained to be jammed, as shown in Figure 4 (d).
- Interestingly, we found that node 5 was not a jammed node when only one jammer was active, and became a jammed node when both jammers were turned on. Thus, $N_J^1 \cap N_J^2 \neq N_J^{1,2}$, which makes the task of localizing multiple jammers challenging.

[1] Although the link from node 11 to node 8 is not connected when J_1 is active, node 11 can still deliver few packets to node 8 occasionally to report its neighbor list. Thus, node 0 is aware that $l_{8,11} = 1$

V. 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 discuss our 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.

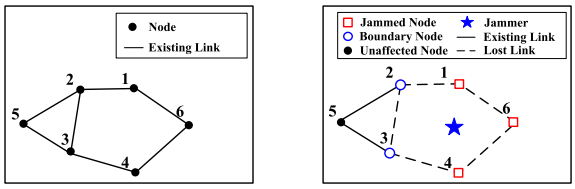
Our framework consists of two components: *automatic network topology partitioner* and *intelligent multi-jammer localizer*. The automatic network topology partitioner systematically divides the nodes into three categories: unaffected nodes, jammed nodes, and boundary nodes. Then it forms two types of clusters: *jammed clusters (JCs)* and *boundary clusters (BCs)*. The intelligent multi-jammer localizer localizes multiple jammers by utilizing the results from the network topology partitioner. We detail these two components in the following subsections and summarize the notations in table I.

A. Automatic Network Topology Partitioner

Once the jamming is detected utilizing one of the existing jamming detection approaches, our automatic network topology partitioner will classify network nodes into different categories

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;
$Gauss-Newton()$	Gauss-Newton searching method;

TABLE I
NOTATION SUMMARY.



(a) without jamming (b) with jamming

Fig. 5. A network example to show the neighborhood adjacency matrix.

and further form two types of clusters: *jammed clusters (JCs)* and *boundary clusters (BCs)*. Typically, there will be one distinct *jammed cluster (JC)* and one distinct *boundary cluster (BC)* formed around a jammer, and we developed a Minimum Spanning Tree (MST) based topology partitioning method to identify them.

MST-based topology partitioning. We formulate the network into a connected, undirected graph, which is represented by a *neighborhood adjacency matrix* G [19]. The graph is undirected because each communication link between nodes is bi-directional under normal situations without jamming. In the neighborhood adjacency matrix G, if two nodes are neighbors, the corresponding element in the matrix is set to 1, otherwise, it is 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, a **MST** [19] is a spanning tree with total weight less than or equal to the weight of every other spanning trees, and the Dijkstra's algorithm can obtain the minimum spanning trees.

To identify BCs and JCs, we define subgraphs G_{N_J} and G_{N_B} containing all jammed and boundary nodes, respectively. To illustrate the relationship among G, G_{N_J} and G_{N_B}, we use a network with 6 nodes depicted in Figure 5. And the 6×6 neighborhood matrix G of this network is:

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

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 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.

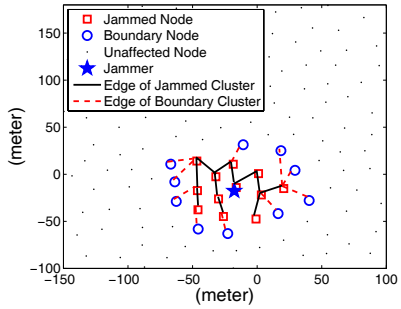


Fig. 6. Clusters for jammed nodes and boundary nodes: the nodes connected by black solid lines form a jammed cluster through an MST, whereas those nodes connected by red dashed lines belong to the boundary cluster.

$$G_{N_J} = G[I_{N_J}; I_{N_J}] = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

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

Since boundary nodes are mostly surrounding jammed nodes, they may not form a proper 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 $I_{N_J} \cup I_{N_B}$:

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

The MST-based topology partitioning method is shown in Algorithm 1. In particular, we derive the JC by applying Dijkstra's algorithm to G_{N_J}. To form the BC, we start with either a boundary node or a jammed node in $G_{N_J \& N_B}$ and construct a combined MST containing both jammed nodes and boundary nodes. Then, we obtain the BC by eliminating the jammed nodes from the combined MST. Figure 6 presents an example of the formed JC and the BC after applying the MST-based topology partitioning method. The nodes connected by black solid lines form a JC, and those nodes connected by red dashed lines belong to the BC. As Figure 6 indicates, boundary nodes may not form a proper cluster by themselves, and the BC has to be formed with the assistance of jammed nodes.

Algorithm 1 MST-based topology partitioning.

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

Node clustering analysis based on jammers' distance. We consider a two-jammer example as a case study. We note that the cases of more than two jammers may result in a richer set of clustering results but share the same basic idea as two-jammer ones. When the distance between two jammers are large, the jamming regions of each jammer are disjoint. One jammed cluster and one boundary cluster are formed around each jammer, and resulting in *two jammed clusters and two boundary clusters* (2JC-2BC) as depicted in Figure 7 (a). However, when two jammers are resided close to each other, they may have overlapping jamming regions and form only one connected jammed area. Depending on how close two jammers are, two scenarios are possible. The first one is *two jammed clusters and one boundary cluster* (2JC-1BC), as shown in Figure 7 (b). 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 7 (c). When two jammers are placed even closer, the two jammed clusters merge into one jammed cluster.

MST-based topology partitioning method can identify all aforementioned cases: *2JC-2BC, 2JC-1BC,* and *1JC-1BC*. The diversity of clustering results makes it challenging to localize multiple jammers. In the case of 2JC-2BC, algorithms for localizing one jammer can be applied to determine both jammers' location. However, when two jammers are resided close to each other and form 2JC-1BC or 1JC-1BC, algorithms for single-jammer scenarios are no longer applicable. To address this challenge, we build intelligent multi-jammer localizer which takes the topology partitioning results and can localize jammers regardless whether two jammers have overlapping jammed areas or not.

B. 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. Moreover, we develop two sets of solutions, one possesses the prior knowledge of the order that the jammers are turned on, referred as *sequentially* turning on; and the other does not have any prior knowledge about the order that the jammers are turned on, which makes the system consider that the jammers are turned on *simultaneously*. The jamming effects with sequentially turning on jammers and simultaneously turning on jammers are presented in Section III and are used as prior knowledge for the intelligent multi-jammer localizer.

Basic algorithms to localize a single jammer. We start by introducing two localization algorithms used to estimate the position of a single jammer: *Centroid-based* [9] and *Adaptive LSQ* [11]. Both algorithms work even when the network communication is disturbed by jamming, and they utilize 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 $\{(x_m, y_m)\}_{m=1...M}$, the position of the jammer can be estimated by Centroid-based localization as:

$$\hat{J} = (\hat{x}_J, \hat{y}_J) = (\frac{\sum_{m=1}^{M} x_m}{M}, \frac{\sum_{m=1}^{M} y_m}{M}). \tag{12}$$

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:

$$\hat{\mathbf{v}} = [\hat{x}_J, \hat{y}_J, \hat{P}_J]^T = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}. \tag{13}$$

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

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

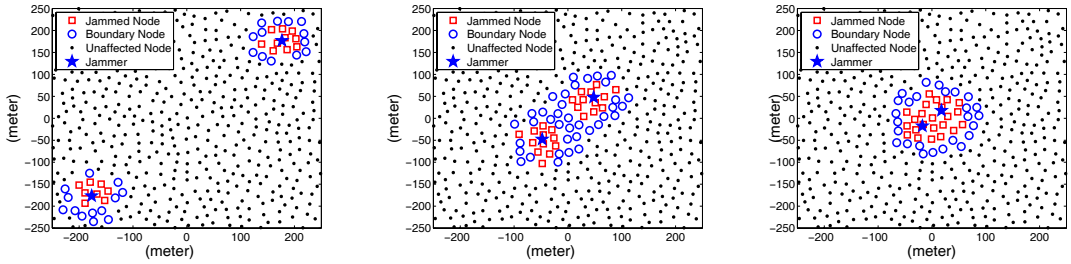
$$\mathbf{b} = \begin{pmatrix} (x_1^2 - \frac{1}{M}\sum_{m=1}^{M} x_m^2) + (y_1^2 - \frac{1}{M}\sum_{m=1}^{M} y_m^2) \\ \vdots \\ (x_M^2 - \frac{1}{M}\sum_{m=1}^{M} x_m^2) + (y_M^2 - \frac{1}{M}\sum_{m=1}^{M} y_m^2) \end{pmatrix}$$

and

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

Centroid-based vs. Adaptive LSQ. According to our prior work, the Adaptive LSQ is more likely to provide 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, Centroid-based method will perform better.

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



(a) 2 jammed clusters, 2 boundary clusters (b) 2 jammed clusters, 1 boundary cluster (c) 1 jammed cluster, 1 boundary cluster

Fig. 7. Illustration of the clustering results obtained from the MST-based topology partitioning met hod when two jammers are placed at various distances.

1) 2JC-2BC: When distinct jammed cluster and boundary cluster for each individual jammer are returned from the MST-based topology partitioning method, the Adaptive LSQ method can be directly applied to localize each jammer when multiple jammers are present. In particular, two independent least square problems can be formed using the information of either BC to localize two jammer independently.

2) 2JC-1BC: When two jammers are nearby, only one BC is returned from the network topology partitioner. Depending on the availability of the prior knowledge of the timing that jammers become available, we consider two cases,

- Sequentially turning on: The position of the first jammer is estimated by the Adaptive LSQ method utilizing the returned first portion of the BC when the first jammer is on. When the second jammer is turned on, because only one BC is formed which contains all boundary nodes caused by two jammers, the Adaptive LSQ method cannot be applied to estimate the position of the second jammer. Our framework turns to examine the two JCs and calculates the second jammer's position by using the Centroid-based algorithm based on the JC related to the second jammer.
- Simultaneously turning on: Adaptive LSQ cannot be used to determine the positions of two jammers since there is only one BC returned. Our framework then resorts to utilize the information from the two returned JCs and apply the Centroid-based algorithm to each JC and obtain the location estimation of each jammer.

3) 1JC-1BC: The most challenging scenario is that only one jammed cluster and one boundary cluster are returned by the MST-based topology partitioning when two jammers appear in a close proximity. This makes neither Adaptive LSQ nor Centroid-based methods sufficient for localizing either jammer's position. To solve this problem, we develop two new algorithms: the *Mirroring* algorithm for the scenario that our framework knows the information about sequentially turning on jammers, and the *Gauss-Newton Searching* algorithm when our framework does not have the prior knowledge, instead, treating two jammers simultaneously turned on. We next describe these two algorithms under two different cases of the order that jammers are turned on.

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 $\hat{J}$:

$$\hat{J}_2 = \hat{J} - (\hat{J}_1 - \hat{J}). \tag{14}$$

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.

Let $\mathbf{v}$ be the variable vector of the two jammers' positions and transmission power, i.e., $\mathbf{v} = (x_{J_1}, y_{J_1}, x_{J_2}, y_{J_2}, P_J)$. Given M boundary nodes $\{(x_m, y_m)\}_{m=1\ldots M}$, we define m residual functions f_{r_m} of $\mathbf{v}$, i.e.,

$$f_{r_m} : \mathbb{R}^5 \rightarrow \mathbb{R}$$

Let $\hat{\mathbf{v}}$ be the estimated value of $\mathbf{v}$. When the estimated positions of the jammers are equal to their true locations, i.e., $\hat{\mathbf{v}} = \mathbf{v}$, all M residual functions become 0 under the assumption of free space model. Thus, estimating the position of jammers is equivalent to minimizing the sum of squares of $f_{r_m}(\mathbf{v})$:

$$\hat{\mathbf{v}} = \arg\min_{\mathbf{v}} S(\mathbf{v}) = \arg\min_{\mathbf{v}} \sum_{m=1}^{M} f_{r_m}^2(\mathbf{v}) \tag{15}$$

Now, we discuss the definition of f_{r_m}, and show $f_{r_m} = 0$ if $\hat{\mathbf{v}} = \mathbf{v}$ using the free-space propagation model. Let r_{h_m} be the range within which the boundary node m can receive packets from other nodes, e.g., any transmitter i within r_{hm} of m has $SNR_{mi} \geq \gamma_0$ where γ_0 is the decodable SNR threshold. Applying the free space propagation model to Equation 4, we obtain M equations for $m = 1 \ldots M$:

$$\frac{\frac{P_S G}{4\pi r_{h_m}^2}}{P_N + \frac{P_J G}{4\pi d_{J_1 R_m}^2} + \frac{P_J G}{4\pi d_{J_2 R_m}^2}} = \gamma_0, \tag{16}$$

where

$$\begin{aligned} d^2_{J_1R_m} &= (x_m - x_{J_1})^2 + (y_m - y_{J_1})^2 \\ d^2_{J_2R_m} &= (x_m - x_{J_2})^2 + (y_m - y_{J_2})^2. \end{aligned} \tag{17}$$

After manipulating Equation 16, we obtain the following equation:

$$P_J(d^2_{J_1R_m} + d^2_{J_2R_m}) - C_m d^2_{J_1R_m} d^2_{J_2R_m} = 0, \tag{18}$$

where $C_m = \frac{P_S}{r^2_{h_m}\gamma_0} - \frac{4\pi P_N}{G}$. By applying Equation 17 to Equation 18, we have the following equations,

$$\begin{aligned} &P_J((x_m - x_{J_1})^2 + (y_m - y_{J_1})^2 + (x_m - x_{J_2})^2 + (y_m - y_{J_2})^2) \\ &- C_m((x_m - x_{J_1})^2 + (y_m - y_{J_1})^2)((x_m - x_{J_2})^2 + (y_m - y_{J_2})^2) = 0 \end{aligned} \tag{19}$$

Motivated by the above equation, we define $f_{r_m}(\mathbf{v})$ as,

$$\begin{aligned} f_{rm}(\mathbf{v}) =& P_J((x_m - x_{J_1})^2 + (y_m - y_{J_1})^2 + (x_m - x_{J_2})^2 + (y_m - y_{J_2})^2) \\ &- C_m((x_m - x_{J_1})^2 + (y_m - y_{J_1})^2)((x_m - x_{J_2})^2 + (y_m - y_{J_2})^2). \end{aligned} \tag{20}$$

In ideal free space, when the hearing range of the boundary node m is a circle and we can accurately estimate it, f_{r_m} equals 0. However, in practice, the hearing range is irregular and the estimated range $\hat{r}_{h_m}$ for boundary node m is inaccurate. Thus, typically $f_{r_m} \neq 0$ and the objective function $S(\mathbf{v})$ cannot reach 0 as well, and we search for the best location estimation that minimizing $S(\mathbf{v})$.

We next describe our iterative searching method to find the solution that minimizes the objective function $S(\mathbf{v})$. Starting with an initial guess $\mathbf{v}^0$ for the minimum, the method proceeds by iterations until it converges. For each iteration step:

$$\mathbf{v}^{s+1} = \mathbf{v}^s + \delta \tag{21}$$

where the increment δ is the solution to the normal equations:

$$(J^T_{\mathbf{f}} J_{\mathbf{f}})\delta = -J^T_{\mathbf{f}}\mathbf{f}. \tag{22}$$

Here, $\mathbf{f}$ is the vector of functions $f_{r_m}(\mathbf{v})$, i.e., $[f_{r_1}(\mathbf{v}), \cdots, f_{r_M}(\mathbf{v})]$ and $J_{\mathbf{f}}$ is the $M \times 5$ Jacobian matrix of $\mathbf{f}$ with respect to $\mathbf{v}$ as shown:

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

After each iteration, $\mathbf{v}^{s+1}$ would be substituted into function $S(\mathbf{v})$. Once $S(\mathbf{v})$ becomes less than the predefined threshold, e.g., S_δ, the estimated values $(\hat{x}_{J_1}, \hat{y}_{J_1}, \hat{x}_{J_2}, \hat{y}_{J_2})$ at the last iterative round will be the final estimated position of the two jammers.

Initial estimated jammers position. To start Gauss-Newton searching, the initial estimated positions of the two jammers, $\hat{J}_1$ and $\hat{J}_2$, need to be chosen. We first calculate a temporary position $\hat{J}$ by applying Centroid-based method on the JC. We then find the farthest jammed node J_{Far} from $\hat{J}$. The initial $\hat{J}_1$ and $\hat{J}_2$ can then be obtained as:

$$\hat{J}_1 = \frac{1}{2} \times (\hat{J} + J_{Far}), \hat{J}_2 = \hat{J} - (\hat{J}_1 - \hat{J}) \tag{23}$$

By substituting the initial estimated positions into Equations 20 the Gauss-Newton based searching method will start to iterate until the algorithm converges.

Algorithm 2 Localizing multiple jammers (by using two jammers as an example)

Require: INPUT:
$JC, BC, BC^{Seq,1}$; (Note: $BC^{Seq,1}$ represents the boundary cluster formed by the first jammer in sequentially turning on case)
OUTPUT:
$\hat{J}_i, i = 1, 2;$
PROCEDURES:
if $\|JC\|$ == 2 and $\|BC\|$ == 2 **then**
  For Sequentially Turning On:
  $\hat{J}_1 = AdaptiveLSQ(BC^{Seq,1})$; $\hat{J}_2 = AdaptiveLSQ(BC^2)$.
  For Simultaneously Turning On:
  $\hat{J}_i = AdaptiveLSQ(BC^i)$, i =1, 2.
else if $\|JC\|$ == 2 and $\|BC\|$ == 1 **then**
  For Sequentially Turning On:
  $\hat{J}_1 = AdaptiveLSQ(BC^{Seq,1})$; $\hat{J}_2 = Centroid(JC^2)$.
  For Simultaneously Turning On:
  $\hat{J}_i = Centroid(JC^i)$, i =1, 2.
else
  For Sequentially Turning On:
  $\hat{J}_1 = AdaptiveLSQ(BC^{Seq,1})$; $\hat{J}_2 = Mirroring(BC, \hat{J}_1)$.
  For Simultaneously Turning On:
  Initialization: $\hat{J} = Centroid(JC)$, $\hat{J}_1 = \frac{1}{2} \times (\hat{J} + J_{Far})$, $\hat{J}_2 = \hat{J} - (\hat{J}_1 - \hat{J})$;
  Iteration: $\{\hat{J}_1, \hat{J}_2\} = Gauss - Newton(\hat{J}_1, \hat{J}_2, BC)$.
end if

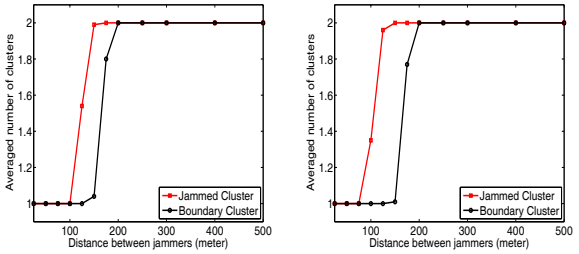
C. Localizing More than Two Jammers

Our localization framework can be extended to localize more than two jammers, and the extended framework will also consist of two components: automatic network topology partitioner and intelligent multi-jammer localizer. We note that localizing more than two jammers requires no modification of the automatic network topology partitioner, but may produce a wider variety of clustering results, e.g., 3JC-3BC. As a result, the intelligent multi-jammer localizer requires slight modification to cope with the new clustering results but the building blocks for jammer localization are the same. For example, in the case of 3JC-3BC, each JC can be paired up with one BC, and Adaptive LSQ can be applied to localize each jammer based on the information of the corresponding JC-BC pair. Detailed modification will be reported in our future work.

VI. Simulation Evaluation

A. Simulation Setup and Performance Metrics

To validate the effectiveness of our framework, in an area of 1000-by-1000 square meters, we generated 1000 different network topologies with 2000 nodes and 3000 nodes, respectively. The nodes are placed to cover the entire deployment region uniformly and the minimum distance between any pair of nodes is bounded by a threshold. To study the effects of multiple jammers, we presented the results of two jammers with the jammers' transmission range set to $60m$, and the decodable SNR threshold γ_0 set to 1.1. To emulate real-world scenarios, we developed our simulation under the shadowing model and we tuned the parameters in the shadowing model



(a) 2000-node (b) 3000-node

Fig. 8. Node topology partitioning study: average number of clusters as a function of the distance between two jamemrs (2000-node and 3000-node deployment with node transmission range setting to $30m$).

with those obtained from our empirical experimental study [11]. In particular, we set the path loss exponent $\eta = 2.11$ and the standard deviation $\sigma = 1.0$.

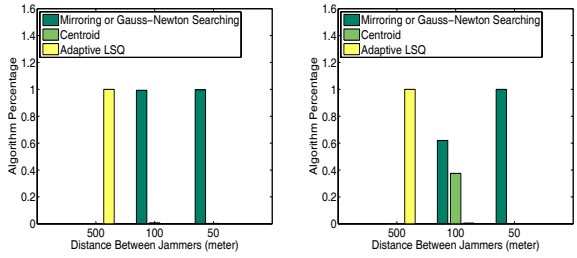
To evaluate the accuracy of localizing the jammers, we defined the localization error as the Euclidean distance between the estimated jammer's location and the true location. To capture the statistical characteristics, we studied the average errors under multiple experimental rounds and we utilized the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

B. Results

1) Node Topology Partitioning Study: We first studied the results of automatic network topology partitioner when varying the distance between two jammers. Figure 8 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 $30m$. 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 $L_J > 200m$ in both networks, and 1JC-1BC is returned when $L_J < 100m$ in the 2000-node network and $L_J < 75m$ in the 3000-node network, respectively. Finally, all three clustering results are possible when L_J falls in between. This observation confirms that our network topology partitioner works flexibly when the distance between jammers varies.

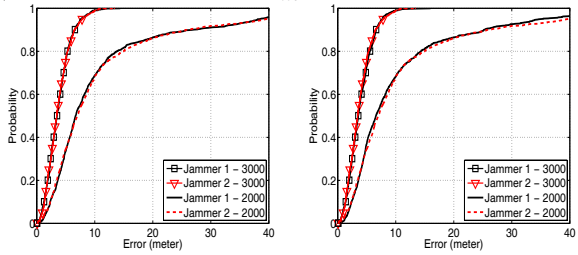
2) Localization Algorithm Selection Study: We next examined how our multi-jammer localizer reacts when the distance between two jammers varies. We plotted the percentage of the basic localization algorithm usage within our multi-jammer localizer in Figure 9 when the distance of two jammers L_J is set to $500m$, $100m$, and $50m$, respectively. When the two jammers are $500m$ away and the partitioner returns 2JC-2BC as indicated by Figure 9, we observed that our multi-jammer localizer uses Adaptive LSQ to perform localization all the time.

When the two jammers are placed close by, around $L_J = 100m$, the usage of basic localization algorithms changes: In the 2000-node deployment, the mirroring algorithm or Gauss-Newton Searching method dominates and conducts localization 98%. This is because 1JC-1BC is classified by topology partitioner for 98% of the jamming scenarios. Whereas under



(a) 2000-node (b) 3000-node

Fig. 9. Usage of localization algorithms in our multi-jammer localizer with node transmission range setting to $30m$.



(a) Sequentially (b) Simultaneously

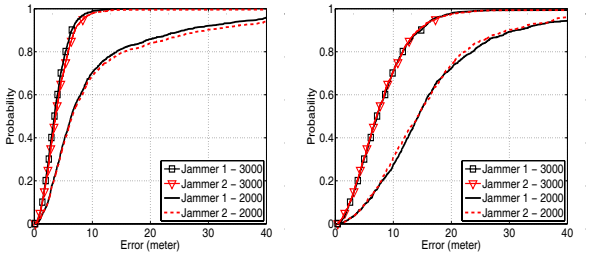
Fig. 10. Cumulative Distribution Function(CDF) of localization error with $L_J = 500m$ and node transmission range setting to $30m$.

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%, which is supported by Figure 9 (b), i.e., 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 close to each other (i.e., $L_J = 50m$), the usage of the mirroring algorithm and Gauss-Newton Searching method is over 99%, matching with 1JC-1BC topology partitioning case.

3) Impact of Distance between Jammers: We next investigate the effectiveness of localization when the distance between jammers varies. In particular, we examined the localization accuracy when $L_J = 500m$, $100m$, and $50m$, and show the results in Figures 10, 11 and 12, respectively.

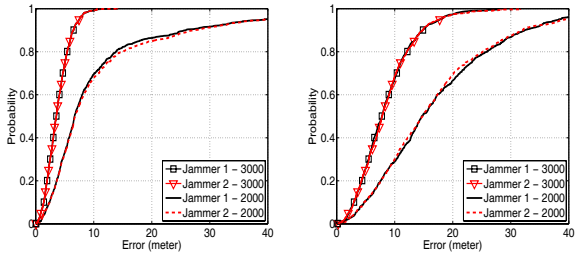
500 meters. This is the case with 2JC-2BC in which the two boundary clusters are distinct. The Adaptive LSQ method is applied by our multi-jammer localizer to perform localization. In general, the accuracy of localizing both jammers is about the same with overlapping curves showing Cumulative Distribution Function (CDF) in Figure 10. The localization performance under 3000-node deployment is 60% better than that under 2000-node one. Particularly, when the jammers are sequentially turning on, the median errors are around $6.9m$ for 2000-node case and $3.7m$ for 3000-node case, respectively. When the order of turning on jammers is unavailable, i.e., simultaneously turning on, the median error of two jammers is similar to the case when the order information is available. This is encouraging as it indicates that our multi-jammer localizer can achieve the similar performance even without the prior knowledge of the order.

100 meters. This is the case when we had a mixture cases of 2J-1B and 1J-1B resulted from the network topology partitioner. Overall, as Figure 11 shows, the localization results



(a) Sequentially (b) Simultaneously

Fig. 11. Cumulative Distribution Function(CDF) of localization error with $L_J = 100m$ and node transmission range setting to $30m$.



(a) Sequentially (b) Simultaneously

Fig. 12. Cumulative Distribution Function(CDF) of localization error with $L_J = 50m$ and node transmission range setting to $30m$.

of the sequentially-turning-on cases outperform those of the simultaneously-turned-on cases by about 50% improvement. The localization accuracy exhibits a better performance of over 40% in the 3000-node deployment than the one in the 2000-node deployment: When jammers are sequentially turned on, the median error is around $3.6m$ for the 3000-node case, where as it becomes $6.3m$ for the 2000-node case. We also found that localization of the second jammer performs slightly worse than that of the first jammer because the JC related to the second jammer is interfered by the first jammer when two jammers are located close by.

50 meters. Finally, we examined the localization accuracy when the distance between two jammers is $50m$. This maps to the case of 1JC-1BC. Again, we observed that localization with prior knowledge achieves better performance than that without the prior knowledge, as indicated in Figure 12. In particular, under the 3000-node deployment, the localization error is only $3.9m$ when the jammers are sequentially turned on, while the error is $7.5m$ when the jammers are simultaneously on. 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.

We note that in practice the attackers may not desire to place two jammers in 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.

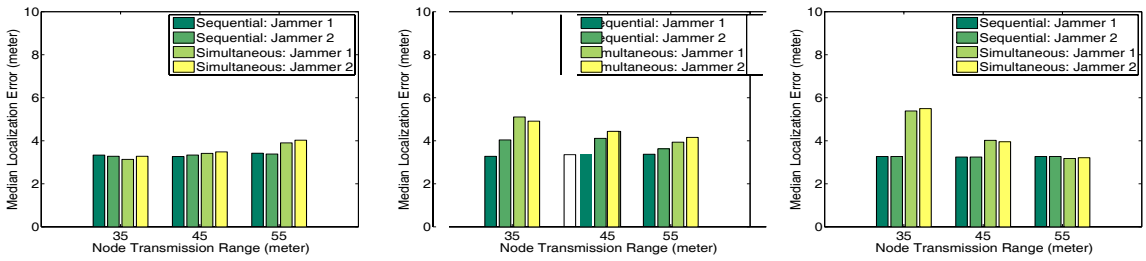
4) Impact of Node Transmission Range: We studied the effect of node transmission range on the localization accuracy by setting the node transmission range to $\{35m, 45m, 55m\}$ and setting the distance between jammers to $\{500m, 100m, 50m\}$, respectively. Table II summarizes the distribution of clustering results, i.e., 2JC-2BC, 2JC-1BC, and 1JC-1BC, for all settings. The corresponding median error of localization is presented in Figure 13. In general, the node transmission range changes do not affect the localization performance much when jammers are treated as sequentially turned on. The localization error is always between $3m$ - $4m$. However, when jammers are treated as simultaneously turned on, the localization accuracy under $L_J = 500m$ degrades as the node transmission range increases, whereas the performance under $L_J \leq 100m$ improves with the increasing node transmission range.

This is because when the distance between jammers is large, the intelligent multi-jammer localizer chooses Adaptive LSQ to localize jammers. When the node transmission range increases, the number of boundary nodes reduces, and the number of constraints for Adaptive LSQ method also decreases, which results in the degradation of localization accuracy. However, under smaller distances L_J, the dominant algorithms selected by the intelligent multi-jammer localizer are Centroid-based algorithm, Mirroring, and Gauss-Newton Searching method. As the node transmission range increases, the interference between each jammer's JC (or BC) decreases. Thus, the localization performance is improved.

Specifically, when $L_J = 500m$, Adaptive LSQ dominates and the localization accuracy is around $3.5m$ when the node transmission range varies for both cases of sequentially and simultaneously turning on jammers as shown in Figure 13 (a).

When $L_J = 100m$, Centroid-based method dominates and performs localization for over 75% of scenarios as displayed in Table II. When two jammers are sequentially turned on, the localization error is between $3.2m$ - $4m$ as shown in Figure 13 (b), 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 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 $5m$ to $4m$, when the node transmission range increases. This indicates that the increasing node transmission range can improve the localization performance.

Finally, when $L_J = 50m$, it is the case of 1JC-1BC, whereby Mirroring algorithms is selected for sequentially-turning-on cases and Gauss-Newton Searching method is selected for simultaneously-turning-on cases. As shown in Figure 13 (c), when jammers are sequentially turned on, the localization conducted by the dominating Mirroring algorithm achieves the similar performance (around $3.5m$) to that of when $L_J = 500m$. When jammers are simultaneously turned on, the median localization error via Gauss-Newton Searching method reduces from $5.5m$ to $3.1m$ when the node transmission range increases from $35m$ to $55m$, suggesting that the Gauss-Newton Searching method can also benefit from larger node transmission ranges.



(a) $L_J = 500m$ (b) $L_J = 100m$ (c) $L_J = 50m$

Fig. 13. Median localization error as a function of the node transmission range under the 3000-node deployment.

$L_J(m)$	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%

TABLE II
DISTRIBUTION OF NUMBER OF CLUSTERS WITH VARIOUS NODE TRANSMISSION RANGES UNDER DIFFERENT JAMMERS' DISTANCES.

VII. 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 to deliver information out of the jammed area. Instead, our framework uses the 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 utilize 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.

REFERENCES

[1] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *MobiHoc 05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, 2005, pp. 46–57.

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

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

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

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

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

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

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

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

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

[11] W. X. Zhenhua Liu, Hongbo Liu and Y. Chen, "Wireless jamming localization by exploiting nodes' hearing ranges," in *Proceedings of the International Conference on Distributed Computing in Sensor Systems (DCOSS 2010)*, 2010.

[12] M. Çakiroğlu and A. T. Özcerit, "Jamming detection mechanisms for wireless sensor networks," in *InfoScale 08: Proceedings of the 3rd international conference on Scalable information systems*, 2008, pp. 1–8.

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

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

[15] T. He, C. Huang, B. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes in large scale sensor networks," in *Proceedings of the Ninth Annual ACM International Conference on Mobile Computing and Networking (MobiCom 03)*, 2003.

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

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

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

[19] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. MIT Press, 2001.