Integrated Computer-Aided Engineering 21 (2014) 19–34
DOI 10.3233/ICA-130445
IOS Press

M-cluster and X-ray: Two methods for multi-jammer localization in wireless sensor networks

Tianzhen Cheng[a], Ping Li[a], Sencun Zhu[b,*] and Don Torrieri[c]
[a]*School of Mechatronical Engineering, Beijing Institute of Technology, Beijing, China*
[b]*Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA*
[c]*US Army Research Laboratory, Adelphi, MD, USA*

Abstract. Jamming is one of the most severe attacks in wireless sensor networks (WSNs). While existing countermeasures mainly focus on designing new communication mechanisms to survive under jamming, a proactive solution is to first localize the jammer(s) and then take necessary actions. Unlike the existing work that focuses on localizing a single jammer in WSNs, this work solves a multi-jammer localization problem, where multiple jammers launch collaborative attacks. We develop two multi-jammer localization algorithms: a multi-cluster localization (M-cluster) algorithm and an X-rayed jammed-area localization (X-ray) algorithm. Our extensive simulation results demonstrate that with one run of the algorithms, both M-cluster and X-ray are efficient in localizing multiple jammers in a wireless sensor network with small errors.

Keywords: Multi-jammer localization, wireless sensor networks, clustering, skeletonization

1. Introduction

As sensor nodes with networking capabilities become commercially available in recent years, wireless sensor networks (WSNs) have been increasingly used in industrial and civilian applications as well as in military applications; e.g., industrial process monitoring and control, environment and habitat monitoring, traffic control, and battlefield surveillance [13]. When sensor networks are deployed in unattended or adversarial environments, security becomes a major concern [2]. Jamming, which is one of the most serious security threats in the field of WSNs, occurs when an adversary simply disregards the medium access control (MAC) protocol by continually transmitting on a wireless channel. The jamming is assumed to originate from sources embedded within the WSN or from compromised sensors. Jamming can easily prevent normal devices from communicating with legitimate MAC operations, introduce packet collisions that force repeated backoffs, or even jam transmissions [34]. However, it is challenging to defend against jamming, because WSNs suffer from many constraints, including low computation capability, and limited memory and energy resources.

To protect the communication in WSNs, many algorithms have been proposed to detect and defend against jamming, such as detection by jamming measurements [34], jamming evasion by channel surfing [35], and spread spectrum [28]. Most of these algorithms only detect jamming or try to keep the wireless sensor network working under jamming. Among the countermeasures against jamming, determining a jammer's location within a wireless sensor network is critical to launching certain security actions against the adversary; e.g., deactivating the jamming device, isolating the jammer, or even destroying it. Only a few works have attempted to identify the physical locations

*Corresponding author: Sencun Zhu, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802, USA. Tel.: +1 081 486 5995; E-mail: szhu@cse.psu.edu.

ISSN 1069-2509/14/$27.50 © 2014 – IOS Press and the author(s). All rights reserved

of jammers in a wireless sensor network, and most of them [18,22,29] only focus on the scenario of a single jammer.

A more severe jamming often involves more than one jammer. Multiple adversaries may attack the network at the same time, or even one adversary may use multiple wireless devices to achieve a better jamming effect. This multi-jammer scenario would make the existing jammer localization algorithms inapplicable [18,22,29]. More specifically, we define the multi-jammer scenario as collaborative jamming by multiple jammers whose jamming regions overlap. All the individually jammed areas are then merged and can be regarded as a single jammed area. Any separate jammed area would only be considered as the single-jammer scenario. This multi-jammer scenario raises new challenges to jammer localization, as multiple jammers need to be localized in a more complex setting.

In this work, we develop two algorithms to deal with the multi-jammer localization problem in WSNs: a multi-cluster localization (M-cluster) algorithm and an x-rayed jammed-area localization (X-ray) algorithm. The M-cluster algorithm is based on the grouping of jammed nodes with a clustering algorithm, and each jammed-node group is used to estimate one jammer location. The X-ray algorithm relies on the skeletonization of a jammed area, and uses the bifurcation points on the skeleton to localize jammers. We made a comprehensive study of our algorithms under various conditions determined by node density, jammer transmission power, jammer deployment, and number of jammers. Our simulation results show that M-cluster and X-ray can achieve a mean localization error of 6.5 meters and 5 meters, respectively, in the two-jammer scenario. Compared with the mean error of 38.5 meters in a baseline scheme, M-cluster and X-ray improve the localization accuracy by 80% and 86%, respectively.

The rest of the paper is organized as follows. We review related work in Section 2, and introduce our multi-jammer scenario with network and jammer models in Section 3. Then we describe our algorithms in Section 4, and show the simulation results in Section 5. In Section 6, we give further discussions and talk about future work. We conclude our paper in Section 7.

2. Related work

2.1. Jamming detection

Jamming detection gives one the knowledge of the presence of jammers in a wireless network. The existing jamming detection methods enhance network protection by triggering countermeasures and providing relevant information. In [34], Xu et al. studied four different jamming models based on jamming behaviors (constant, deceptive, random, or reactive), and examined different measurements on detecting jamming, including packet send ratio (PSR), packet delivery ratio (PDR), signal strength, and carrier sensing time. Cakiroglu et al. [3] proposed two algorithms for detecting a jamming, which are based on bad packet ratio (BPR), packet delivery ratio (PDR), energy consumption amount (ECA), and neighboring node conditions. Li et al. [14] proposed a sequential jamming detection technique that works when an increased number of message collisions are observed during an observation window, compared with the previously learned long-term average. Misra et al. [20] selected packets dropped per terminal (PDPT) and signal-to-noise ratio (SNR) as the input to their fuzzy inference system. Based on the Mamdani model, the system outputs the jamming index (JI) of each node. Strasser et al. [25] suggested a method for detecting a reactive jammer through received signal strength (RSS) and bit error rate (BER) sampling.

2.2. Countermeasures against jamming

Spread-spectrum (SS) physical-layer techniques using direct-sequence or frequency-hopping spread spectrum [28] are well-known countermeasures against communication jamming. These solutions require that both the sender and the receiver share the same key and the same pseudorandom function to generate a hopping or spreading sequence. UFHSS [26] and UDSSS [26] deal with the problem of key establishment without pre-shared secret under jamming. Jamming and (unintentional) interference are further thwarted by the application of forward error-correcting codes and special coding methods [28].

For WSNs, Wood et al. [32] proposed DEEJAM, a new MAC-layer protocol for defending against stealthy jammers using IEEE 802.15.4-based hardware. Xu et al. [35] presented two techniques, called channel surfing and spatial retreats. The main idea is to increase the resistance to jamming by avoiding the interference as much as possible in either the transmission frequency domain or the spatial domain.

Chiang and Hu [7] introduced a technique to mitigate jamming for broadcast applications by adopting a binary key tree to control code sequences in direct-sequence spread spectrum. Dong and Liu [9] intro-

duced a jamming-resistant broadcast system that organizes receivers into multiple channel-sharing broadcast groups and isolates malicious receivers using adaptive re-grouping. Jiang et al. [12] proposed a compromise-resilient anti-jamming scheme called split-pairing to deal with insider jamming in a one-hop network setting. Liu and Ning [17] proposed an encoding method called BitTrickle to defend against broadband and high-power reactive jamming. Tague et al. [27] proposed a framework for control-channel access schemes using the random assignment of cryptographic keys to hide the location of control channels in the presence of insider jammers. Wang et al. [30] proposed a delay-bounded adaptive online UFH algorithm for anti-jamming wireless communication.

2.3. Localization against jamming devices

A few jammer-localization algorithms have been proposed for WSNs. Pelechrinis et al. [22], based on packet delivery ratio (PDR) and gradient descent methods, designed and implemented a lightweight jammer localization algorithm. Their approach can find out the nearest node to the jammed area. Liu et al. developed a jammer localization algorithm called Virtual Force Iterative Localization (VFIL) [16] and another algorithm [18] that exploits nodes' hearing ranges. Torrieri proposed a direction-finding and localization method based on the special characteristics of spread-spectrum communications and multiple antennae [29]. Cheng [6] proposed a new jammer localization algorithm, called *Double Circle Localization (DCL)*. DCL uses two classic concepts in geometry, minimum bounding circle (MBC) and maximum inscribed circle (MIC), to solve the jammer localization problem. All these works focused on jammer localization in the context of a single jammer, and did not study the multi-jammer scenario in WSNs.

Recently, Liu et al. [15] tried to address the case of two jammers coexisting in wireless networks by leveraging the network topology changes caused by jamming. They studied the jamming effects under two jammers and developed an approach to localize jammers under comprehensive simulations. However, their work does not have details about how to address the scenario with more than two jammers.

In our recent work [5], we also made a preliminary study of this multi-jammer localization problem by proposing the X-ray method. In this work, we significantly extended our previous work in the following ways. First, we introduce a new technique called M-clustering for multi-jammer localization. A comparative study has been made for M-clustering, X-ray and a baseline scheme (as shown in Section 5). Pros and Cons of M-clustering and X-ray are also discussed (in Section 6). Second, we propose a method for jammer number estimation without the knowledge of jammer transmission range (in Section 3.3). Further, we show through simulation that our algorithms can tolerate well the errors caused by false estimation of jammer number.

3. The case of multiple jammers in wireless sensor setworks

3.1. Effects of multiple jammers

Jamming is defined as the act of intentionally directing electromagnetic energy towards a communication system to disrupt or prevent signal transmission [21]. Wireless devices can successfully receive information based on the signal-to-noise ratio (SNR) (SNR $= P_{signal}/P_{noise}$), where P is the average power. Noise represents the undesirable electromagnetic spectrum collected by antennas including the electromagnetic energy from jamming. If the SNR of a received message is below a minimum SNR threshold, the message cannot be correctly extracted and decoded from radio signals. Accordingly, the receiver is considered to be jammed.

The multi-jammer scenario can be described as follows. Multiple jammers perform collaborative jamming in WSNs, and they have overlapping jammed areas. We do not consider the scenarios of a single jammer or multiple jammers with non-overlapping jammed areas. Indeed, these latter two scenarios can be simply solved by an existing single-jammer localization algorithm [16,18].

Compared with a single-jammer strategy, a multi-jammer strategy has several advantages to an adversary. First and the foremost, under the same overall budget of power, the multi-jammer strategy is more power-efficient than the single-jammer strategy because of the rapid attenuation of jamming signals. In Fig. 1, we compare the size of jammed areas in four strategies *under the same overall power*: one, two, three and five jammers. For simplicity, the jammed area in a multi-jammer strategy is the sum of jammed areas caused by all the jammers. As the overall power is fixed, if more jammers are used, each jammer will be assigned less power ($1/n$ of the overall power, where

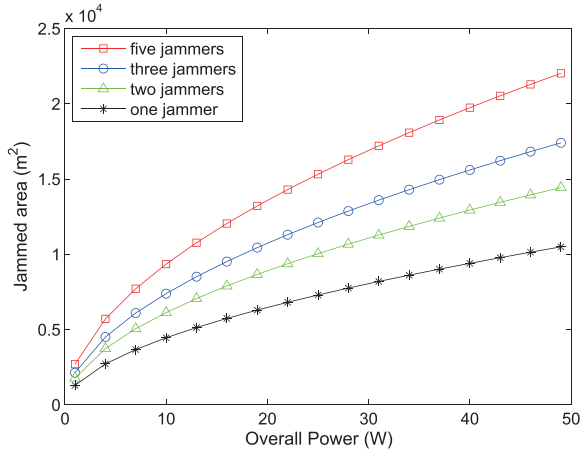

Fig. 1. Jammed area comparison of the single-jammer and multi-jammer strategies. X axis represents the overall jamming power used by the jammer strategies. Jammers in one strategy share out the overall jamming power.

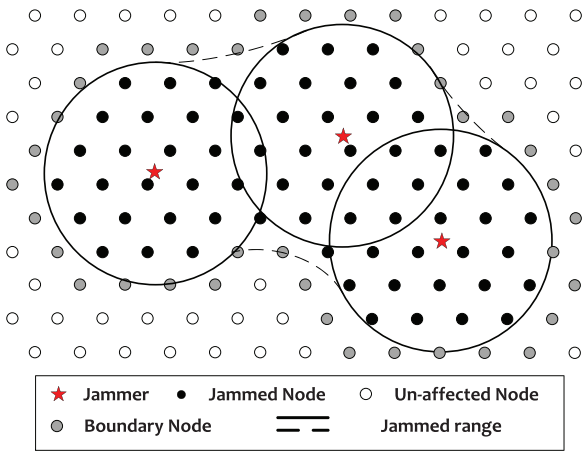

Fig. 2. An illustration of three jammers with overlapping jamming region in the wireless sensor network.

n is the number of jammers). Figure 1 shows that, under the same overall power budget, more jammers result in a larger jammed area (i.e., a better jamming effect). However, from the perspective of a defender, more jammers mean more jamming-attack strategies, and hence more challenges to protect communication and to localize jammers.

We further show a possible topology in a multi-jammer scenario. In Fig. 2, according to their connectivity, sensor nodes in the network are divided into three categories: (a) jammed node, which has no communication with its neighbor nodes; (b) unaffected node, whose connectivity has not been affected by jamming; (c) boundary node, whose neighbor nodes

Layer	Entity	Roles in our algorithms
L1	Base station	Collect node state information and perform our localization algorithms;
L2	Unaffected nodes	Provide node state information to layer L1;
L3	Boundary nodes	Provide jammer information to layer L1 and layer L2
L4	Jammed nodes	Transmit and/or listen to channels for transmission opportunity

Fig. 3. An overview of the network entities and their roles in our algorithms.

are partially jammed. These nodes are represented in black, white and grey dots, respectively. We note that, in the region where two jammed areas meet, the electromagnetic environment is very complex. In our work, we use the sum of received signal strength of all jamming signals as the jamming power at each node. Therefore, as the powers of the different jamming signals accumulate, some nodes near to the intersection areas of jammed regions also become jammed.

3.2. *Network and multi-jammer models with assumptions*

Network model. In this paper, we first assume that WSNs are randomly deployed and static, and a base station is deployed to gather information and runs our multi-jammer localization algorithms. Nodes are assumed to know their own locations and can detect jamming. Some existing techniques might be used to provide such information [19,34]. Also, every node maintains a neighbor list and has such information as their locations and activeness (jammed or not). By periodically broadcasting a beacon message, this list can be easily obtained, and any change of the status of neighbors will be updated. In Fig. 3, we show the different types of entities and their roles in the network at the time of jamming.

Radio propagation channel model. In our work, we use the radio propagation channel model in which the ratio of received to transmitted power in dB is given by

$$\frac{P_r}{P_t}dB = 10\log_{10} K - 10\gamma \log_{10}\frac{d}{d_0} - \psi_{dB}, \quad (1)$$

where ψ_{dB} is a zero-mean Gaussian-distributed random variable that models the shadowing, K is a unitless constant that depends on the antenna characteristics and the average channel attenuation, d_0 is a refer-

ence distance for the far-field antenna, γ is the power-law attenuation or path-loss exponent, and fading is neglected [10,28]. In the simulations, the shadowing is neglected by setting $\psi_{dB} = 0$ dB, but both shadowing and fading will be considered in future work. The power-law attenuation is $\gamma = 3.71$, and $K = 31.54$. The radio transmission is assumed to be omnidirectional.

Multi-jammer model. Multiple jammers work together and transmit either the same or different jamming power levels. All jammers are deployed with their locations fixed. To model the deployment of multiple jammers in WSNs, we impose a constraint from the adversary's viewpoint. Assume there are n jammers in a wireless sensor network, the distance D_{ij} between jammer i and its nearest-neighbor jammer j should follow this condition:

$$D_{ij} \in [\omega(R_i + R_j), (R_i + R_j)], \tag{2}$$

where R_i and R_j are the transmission ranges of jammer i and jammer j, respectively. $\omega \in (0, 1)$ is a variable. In general, a smaller ω implies closer distance between two nodes. Unless specified, ω is set to 0.5 in this work, and we will study how this parameter affects performance in our simulation section. Under this condition, jamming regions are merged, and the jamming impact is optimized. Note that although multiple jammers can be deployed very closely to one another (i.e., with a small ω), such deployments would waste the potential ability to jam more nodes. Moreover, when deploying multiple jammers very closely, localizing one of them would be more likely to expose the other ones that are close. Indeed, the jammers that are overly close can be treated as a single jammer. Jammers without overlapping jammed areas (i.e., $\omega = 1$) will be not considered in this paper, as each of them can be localized individually as a single jammer [16,18].

3.3. Jammer number determination

In this paper, we will mainly focus on determining the locations of multiple jammers at one time with the number of jammers already known. How to accurately detect the number of jammers in WSNs in our multi-jammer scenario could be an interesting but undecidable problem as jammers can vary their transmission powers. As such, we address this problem with heuristic approaches under different knowledge assumptions. We will also show the impact of false number estimations in the simulation section.

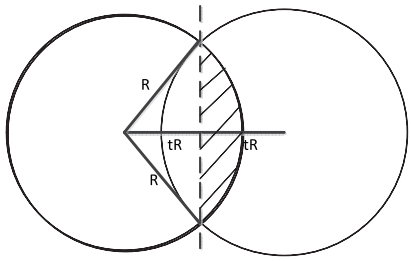

Fig. 4. Determining the parameter λ which is to be used in estimating the number of jammers.

The scenario with the jammer-transmission-range knowledge. Under the approximate propagation model, the jammed area of each individual jammer is circular. We first roughly estimate the average jammed area of one jammer, then use it to estimate the number of jammers existing in the actual jammed area. In Fig. 4, R is the average transmission range of a jammer, and $2tR$ is the average distance between two jammers. The average jamming area of a single jammer without considering overlapping is $S_{SingleJammer} = \pi R^2$. The average jamming area of one jammer considering overlapping is $S_{avg} = \lambda S_{SingleJammer}$, which is calculated as $S_{SingleJammer}$ minus the area of the shadowed zone in the figure. Hence, we can derive $\lambda = (1 - arccos(t)/\pi + t\sqrt{1-t^2}/\pi)$. By correlating t with the concept of ω in Eq. (2), we can express it as $t = (1 + \omega)/2$. Using the parameter λ, a rough estimate of the jammer number is

$$N_{jammer} = \left\lceil \frac{S_{JammedArea}}{\lambda S_{SingleJammer}} \right\rceil \tag{3}$$

where $\lceil x \rceil$ denotes the smallest integer greater than or equal to x. After the number of jammers in the network is estimated, we run our multi-jammer localization algorithms.

The scenario without the jammer-transmission-range knowledge. First, if multiple jammers sequentially turn on to launch jamming [15], the position of the first jammer could be estimated by a single-jammer localization algorithm, and the jammer's transmission range could be obtained. Second, if jammers simultaneously turn on, based on the shape of the jammed area, we can analyze the width of multiple parts of the jammed area and estimate the extent of jammer's transmission range. Because of the rapid attenuation of radio signals, the range of significant jamming power should have an upper limit, R_{Jmax}, (indeed, this is why we

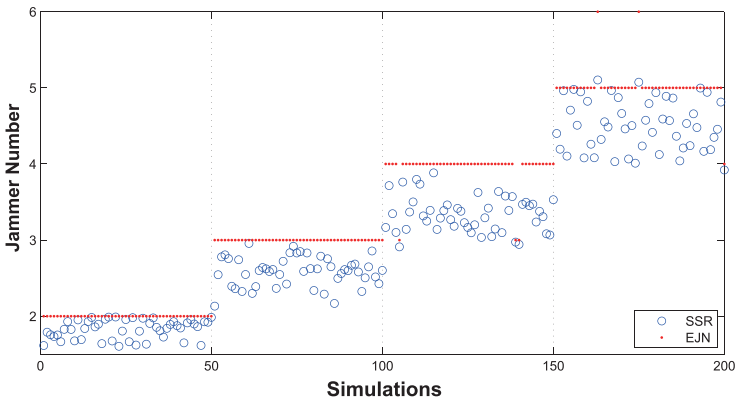

Fig. 5. Jammer-number determination in different multi-jammer scenarios. Here dots (EJN) are the estimated jammer numbers based on Formula (3). Circles (SSR) stand for the part of formula inside the ceilings.

have multiple jammers) as well as a lower limit, R_{Jmin}. To jam the regular wireless communication between two good nodes, the lower limit of jammers' transmission range generally should be larger than good node's transmission range, R_{node}. Here we set $R_{Jmin} \geqslant 1.5R_{node}$. Then we can estimate the jammer number as follows:

$$N_{jammer} \in \left[\frac{S_{JammedArea}}{\pi R_{Jmax}^2}, \frac{S_{JammedArea}}{\lambda \pi R_{Jmin}^2} \right]. \quad (4)$$

Figure 5 shows our simulation results about our jammer-number detection method. The x axis represents the number of simulations in different jammer number scenarios, while the y axis is the number of jammers. Here $\omega = 0.5$. In Fig. 5, dots represent the estimated jammer numbers based on Eq. (3). Circles (SSR) stand for the part of Eq. (3) inside the ceilings. Altogether, four multi-jammer scenarios are simulated (i.e., 2, 3, 4 and 5 jammers), where each scenario shows 50 simulation results. From the simulation results, we observe that our method works correctly in both the 2- and 3-jammer scenarios. It has only 3 false detections among all 50 simulations in the 4-jammer scenario, where the estimated numbers become 3 instead of 4. There are also 3 false detections in the 5-jammer scenario, where two numbers become 6 instead of 5. We observe that circles spread out when jammer number increases. This is because the jammed areas are overlapped in more complex ways when jammer number increases, making our previous way of estimating λ less accurate. As it is very difficult, if not impossible, to estimate the jamming number perfectly under various jamming strategies, we will instead show the impact of false number detection on the performance of our jammer localization algorithms in Section 5.

4. Two multi-jammer localization algorithms

In this section, we present two algorithms for multi-jammer localization, *M-cluster* and *X-ray*. As we will see through algorithm descriptions and simulations: *M-cluster* is an efficient algorithm, but its localization accuracy is affected by the distribution of sensor nodes in the network. Hence, we design *X-ray* to improve the localization accuracy, at the cost of higher computational complexity and the assumption that a jammed-area mapping protocol such as JAM [31] is available in the network.

4.1. *The multiple cluster localization (M-cluster) algorithm*

We will describe our first multi-jammer localization algorithm for WSNs, called Multi-cluster Localization Algorithm (M-cluster), which divides jammed nodes into different clusters based on clustering algorithms and the number of jammers in the network.

Motivation. In a multi-jammer scenario, jammed areas covered by different jammers overlap and they together form a large jammed area. If we directly run an existing single-jammer localization algorithm, the result will be inaccurate. However, if we can first find out the jammed area belonging to each jammer, we can then apply a single-jammer localization algorithm to localize each jammer. Accordingly, we develop our M-cluster algorithm, which groups jammed nodes into clusters and applies the centroid localization (CL) algorithm [16] to estimate each jammer's location. The M-cluster algorithm involves the following three steps, as depicted in Fig. 6.

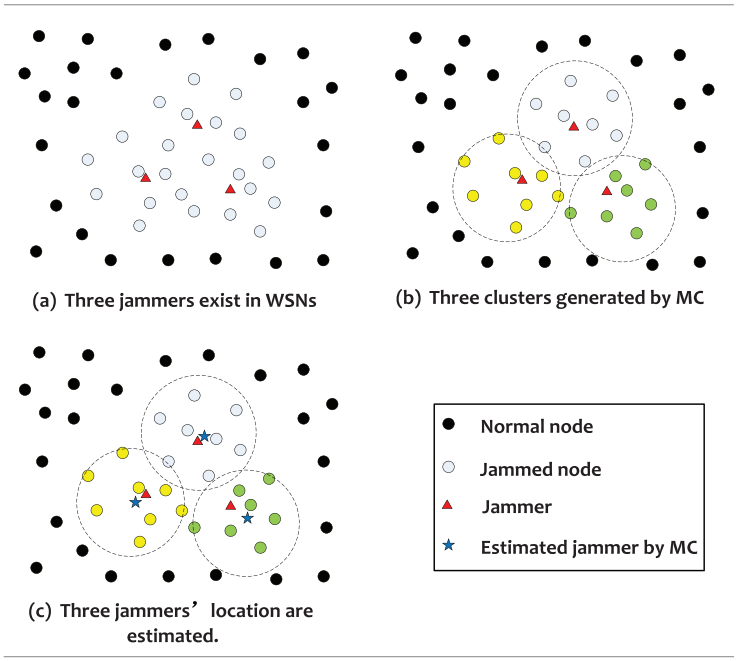

Fig. 6. Overview of the multi-cluster localization algorithm in WSNs.

4.1.1. Feature selection or extraction

Clustering algorithms divide a group of objects into subgroups based on similarity measures. Every clustering algorithm is based on the index of similarity or dissimilarity between data points, where the similarity or dissimilarity measures rely on descriptions of data points with features. To classify jammed nodes, M-cluster first needs to choose features. Feature selection chooses distinguishing features from a set of candidates, while feature extraction utilizes some transformations to generate useful and novel features from the original ones. Both feature selection and feature extraction are very important to the effectiveness of clustering applications. A different clustering criterion or clustering algorithm, even for the same algorithm but with different selection of parameters (features), may cause completely different clustering results.

In our M-cluster algorithm, for N jammed nodes with d features, we build an $N \times d$ pattern matrix to represent the pending data, and use the Euclidean distance to describe quantitatively the similarity of two data points or two clusters. The Euclidean distance between nodes $n_x = (x_1, x_2, \ldots, x_d)^T$ and $n_y = (y_1, y_2, \ldots, y_d)^T$ is calculated as

$$D(n_x, n_y) = \left(\sum_{j=1}^{d} (x_j - y_j)^2 \right)^{1/2}. \tag{5}$$

where x, y are features belonging to n_x and n_y, respectively. In this paper, we select the coordinates of jammed nodes as one feature for the similarity measure. Although some other features may also be applied, such as received signal strength (RSS), packet deliver ratio (PDR) and so on, they are difficult to be obtained in a jammed wireless sensor network. As such, we do not choose them in this work.

4.1.2. Selection of clustering algorithms

The next step is to choose an appropriate clustering algorithm to do the grouping by optimizing a criterion function. A criterion function is constructed with the similarity measures of features selected or extracted in the previous step. Clustering techniques are generally classified as partitional clustering and hierarchical clustering, based on the properties of the generated clusters [33]. Partitional clustering directly divides data points into some pre-specified number of clusters without the hierarchical structure, while hierarchical clustering groups data with a sequence of nested partitions, either from singleton clusters to a cluster including all individuals or vice versa. In hierarchical clustering the goal is to produce a hierarchical series of nested clusters, ranging from clusters of individual points at the bottom to an all-inclusive cluster at the top, generating a tree structure called a dendrogram. However, in the scenario of jammed-nodes

clustering, we want to produce a one-layer partition of the jammed nodes without any hierarchical structures. Hence, we choose the partitional clustering approaches to grouping nodes. Meanwhile, as jammed areas have overlaps with each other in the multi-jammer scenario, nodes in the jammed area may be affected by more than one jammer. As such, jammed nodes may also belong to more than one cluster.

Therefore, the fuzzy partitional clustering is much more suitable for this special situation, as all data points in the data set are allowed to belong to all clusters with a degree of membership. In our multi-jammer localization problem, we use a classic clustering algorithm, Fuzzy c-Means (FCM) [33], which performs clustering in a fuzzy way (i.e., objects can belong to multiple clusters in a certain degree). FCM attempts to find a partition, represented as c fuzzy clusters, for a set of data objects $x_j \in \Re^d, j = 1, \ldots, N$, while minimizing a cost function:

$$J(U, M) = \sum_{i=1}^{c} \sum_{j=1}^{N} (u_{ij})^{\alpha} D_{ij}^2, \tag{6}$$

where

- $U = [u_{ij}]_{c \times N}$ is the fuzzy partition matrix and $u_{ij} \in [0, 1]$ is the membership coefficient of the jth object in the ith cluster that satisfies the following two constraints: $\sum_{i=1}^{c} u_{ij} = 1, \forall j$, which assures the same overall weight for every data point, and $0 < \sum_{j=1}^{N} u_{ij} < N, \forall i$, which assures no empty clusters;
- $M = [m_1, \ldots, m_c]$ is the cluster prototype (mean or center) matrix;
- $\alpha \in [1, \infty)$ is the fuzzification parameter and a larger α favors fuzzier clusters;
- $D_{ij} = D(x_j, m_i)$ is the Euclidean distance between x_j and m_i.

4.1.3. Localization calibration

Next, M-cluster computes cluster centers according to the output of FCM partition. Only nodes in one cluster can be used in the centroid localization (CL) algorithm to compute the centroid of this cluster. Based on the knowledge of jammer transmission range or its scope, M-cluster improves results by producing an imitation for each jammed area, which is a circular area centered at the estimated location of each jammer. If the estimation perfectly matches the real location of the jammer, then no boundary nodes should be covered by the imitated jammed area. Hence, if some boundary nodes are covered, M-cluster moves some cluster centers to improve the final estimation accuracy.

More specifically, let us assume that a set of boundary nodes $\{(X_i, Y_i)\}$ are covered by the imitation, a corresponding jammer location estimation is (X_e, Y_e), and node (X_m, Y_m) is a boundary node m in $\{(X_i, Y_i)\}$ that is nearest to the estimation. Then, the new coordinate of the jammer is estimated as

$$(\hat{X}_e, \hat{Y}_e) = \left(X_e + D_{step} \times \frac{X_e - X_m}{D_{em}} , \right.$$
$$\left. Y_e + D_{step} \times \frac{Y_e - Y_m}{D_{em}} \right),$$

where D_{step} is a constant step distance, and D_{em} is the distance between node m and the estimation. M-cluster does this improvement iteratively until no boundary node is falsely covered. After this improvement, M-cluster outputs the final estimations of jammers' locations.

4.2. The X-rayed jammed-area localization (X-ray) algorithm

In this section, we describe our second multi-jammer localization algorithm, called an x-rayed jammed-area localization (X-ray) algorithm, which skeletonizes jammed areas and estimates jammer locations based on the bifurcation points on skeletons of jammed areas. Here, we show an overview of the x-rayed jammer localization algorithm. As Fig. 7 shows, this X-ray algorithm can be divided into three phases: jammed-area mapping, jammed-area skeletonization and jammer-location determination. A more detailed algorithmic flowchart is shown in Fig. 8.

4.2.1. Jammed-area mapping

Wood et al. [31] have presented "JAM", a jammed-area mapping service, which can roughly produce a jammed area. However, since their work mostly focuses on jammer detection, node communication and protocol design, the precise jammed area is not defined clearly. As sensor nodes in WSNs are interspersed in a target field, and there are a lot of blank spaces between sensors, it is a challenge to generate the jammed area simply and precisely. In X-ray, we compute a *convex polygon* of jammed nodes as a jammed area for the process that will follow. A convex polygon, also called a convex hull or a convex envelope in mathematics, is defined as a polygon with all its interior angles less than $180°$; that is, all the vertices of the polygon will point outwards, away from the interior of the shape.

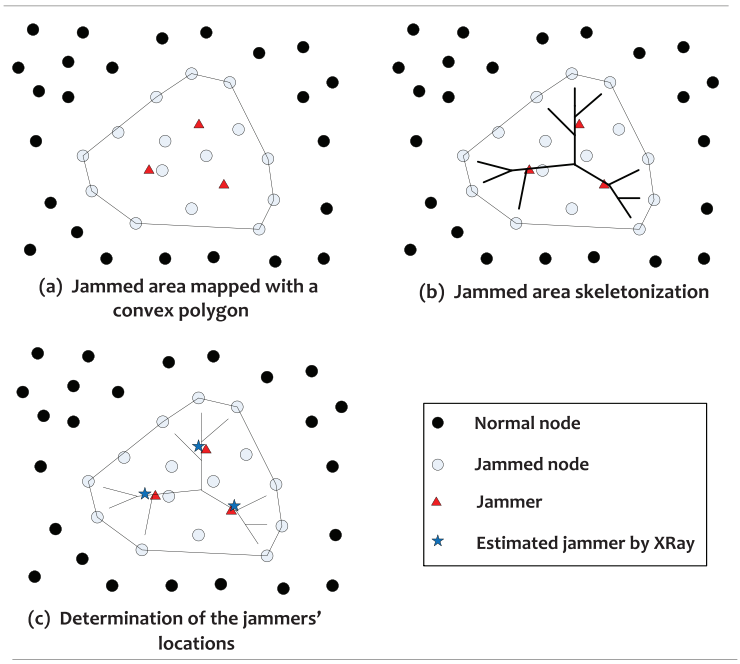

Fig. 7. An overview of X-Ray multi-jammer localization algorithm in WSNs.

Algebraically, the convex hull of X can be characterized as the set of all the convex combinations of finite subsets of points from X, as in the following formula:

$$H_{convex}(X) = \left\{ \sum_{i=1}^{k} \alpha_i x_i \mid x_i \in X, \alpha_i \in \Re, \right.$$
$$\left. \alpha_i \geqslant 0, \sum_{i=1}^{k} \alpha_i = 1, k = 1, 2, \ldots \right\} \quad (7)$$

We show this step in Fig. 7(a), where the three-jammer jammed area is denoted by a convex polygon enveloping all jammed nodes. Only the space enveloped by the convex polygon is regarded as the jammed area.

Although some contour-tracing algorithms can be used to identify and produce the jammed area, they mostly process integral images in pixels [4]. However, here we only obtain images composed by a cluster of points, and all pixels are separate, so contour tracing algorithms will be unable to identify the jammed area. Hence, in this jammed-area identification scenario, we compute the convex polygon of all jammed-node points in the field as the jammed area. This has a few advantages for our localization algorithm. First, the convex polygon is simple and easily computed. There are many existing schemes to compute the convex hull of points that can be operated unambiguously and efficiently [8,23]. The complexity of the corresponding algorithms is usually estimated in terms of n, which is the number of input points, and h, which is the number of points on the convex hull. Second, using a convex polygon may reduce the impact of much noise and fluctuation on the boundary of the jammed area while conserving most of the information about the jammed area; consequently, the skeletonization algorithm (the next step) will easily process the jammed-area map. This is because most skeleton algorithms are sensitive to boundary deformation; that is, little noise or a variation of the boundary often generates redundant skeleton branches that may seriously disturb the topology of the skeleton's graph [1].

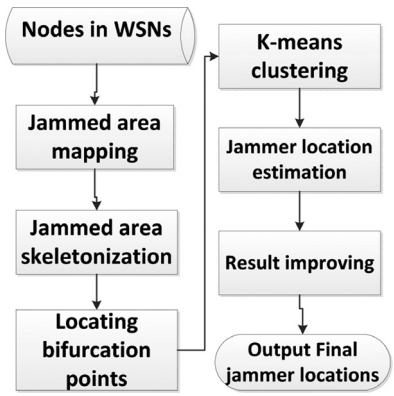

Fig. 8. An algorithmic flowchart of X-Ray.

Through simulations, we also notice that some unjammed nodes may be covered by our convex hull in certain scenarios, which leads to concave cases. To address this concave problem, we compute the convex hull of the miscovered nodes, and subtract this area from the original convex hull of the entire jammed area, as in the following equation:

$$H_{concav} = H_{concex} - H_{miscovered}.$$

4.2.2. Jammed-area skeletonization

In shape analysis, the skeleton (or topological skeleton) of a shape is a thin version of that shape that is equidistant to its boundaries. The skeleton can serve as a representation of the shape (it contains all the information necessary to reconstruct the shape). A formal definition is as follows: a skeleton is the locus of the centers of all maximal inscribed hyper-spheres (i.e., discs and balls in 2D and 3D, respectively). An inscribed hyper-sphere is maximal if it is not covered by any other inscribed hyper-sphere. All points on the final skeleton will have the same distance to more than one boundaries of the jammed area. Specifically, our X-ray algorithm will leverage the skeletonization method proposed by Xiang [1], which can produce a stable skeleton without spurious branches, and therefore provide accurate skeleton information for the following process. More details can be found in the reference [1].

4.2.3. Jammer-location determination and improvement

As shown in Fig. 7(b), we can see the skeleton of the jammed area has multiple *bifurcation points* (or skeleton joints), which are introduced by angles on the convex polygon of the jammed area. Due to the discrete distribution of sensor nodes in a WSN, on the edge of the jammer's influence region, the jammed area has no smooth circular edge; hence, the skeleton of the jammed area has branches (bifurcations) at the extremity of the main skeleton. These branches conserve the location information of the jammers. Based on the coordination information of these bifurcation points on the skeleton, X-ray can roughly localize the multiple jammers in WSNs. Then the bifurcation points can be divided into groups based on a K-means clustering algorithm. Finally, the centroid of the coordinates of all points in one group is considered as the estimated location of a jammer.

Once the locations of jammers are computed, X-ray calibrates the result based on some specific heuristics. First, as in the M-cluster algorithm, we consider the falsely covered boundary nodes and calibrate the result in a similar way. Second, we discover that when many bifurcation points belong to one jammer, the clustering technique may falsely divide them into two clusters, resulting in two jammers. X-ray discovers this error by using a filter that measures the distance between two estimated jammers. As we previously discussed in Section 3, for the purpose of jamming a greater area with the same number of jamming devices, an adversary should separate the jamming devices more. As such, for two estimated jammers i, j, whose distance satisfies the following condition:

$$D_{ij} \in \{d | d < \omega(R_i + R_j)\}, \tag{8}$$

where R is the transmission range of jammers and ω is the constant variable used in Eq. (2), X-ray makes the following calibration. These two estimated jammer locations will be merged into one, whose coordinate is the central of the two estimated jammer locations. Then X-ray generates an imitation of the jamming area with one fewer jammers. Due to the lack of one jammer, this imitation might miss some jammed nodes. If so, X-ray records these jammed nodes that are uncovered by the imitation and computes their average coordinate as another estimated jammer location. After finishing this calibration, X-ray reports the locations of multiple jammers.

5. Performance evaluation

In this section, we first show our simulation setup and performance metrics, and then compare our multi-jammer localization algorithms under various network conditions as a function of network node density, jammer transmission power, jammer deployment scenario, and number of jammers in WSNs.

A random-selection multi-jammer localization scheme. For the purpose of comparison, we propose a naive random-selection multi-jammer localization scheme as the baseline scheme. This random scheme localizes jammers based on the coordinates of jammed nodes. After the number of jammers is estimated, it randomly chooses their locations.

5.1. Simulation setup and performance metrics

In our simulation with MATLAB, we deploy sensor nodes in a 400-by-400 meter region, with the nor-

mal communication range of each node set to 30 meters. Jammers are randomly located in the center of this field, following the distance constraint stated in Section 3. Unless specified, transmission ranges of jammers are set the same (60 meters), and the number of jammers is set to three. For our radio propagation channel model, we set some typical values for the parameters $K = 31.54$ and $\gamma = 3.71$. To find out the performance of our algorithms under different node distributions, we use two network deployments: simple deployment and mesh deployment. In simple deployment, following a uniform distribution nodes are randomly disseminated in the region, so that nodes may congregate at some spots and miss other areas. In mesh deployment, to increase the coverage of the network, the region is meshed into smaller grids, and nodes are divided according to the number of grids. The nodes are uniformly deployed in each grid. For each experiment, we generate 1000 network topologies to obtain higher accuracy.

To measure the performance of the algorithms, we use *localization error* as the metric, which is defined as the Euclidean distance between the estimated jammer locations and the true locations. More specifically, let (x_t, y_t) be the true location of a jammer, and (x_e, y_e) be its estimated location. The localization error is $Err = \sqrt{(x_e - x_t)^2 + (y_e - y_t)^2}$. We show both the cumulative distribution functions (CDF) of average localization errors and bar charts of the mean localization errors of 1000 simulations.

5.2. Evaluation results

Impact of node density. First, we study the impact of node density on the performance of our algorithms. In this part, we adjust the total number of nodes by setting it to 400, 500 and 600, respectively and calculate the mean errors for both M-cluster and X-ray. Two scenarios of simple deployment and mesh deployment are shown in the same figure. From Fig. 9(a), we observe that X-ray has consistently better performance than M-cluster in all node density and node deployment setups. In the mesh deployment scenario, X-ray's mean errors fall between 6 to 9 meters, while M-cluster's mean errors fall between 8 to 11 meters. M-cluster has a greater improvement in the localization accuracy as the node deployment model changes from the simple style to the mesh style; in other words, M-cluster is more influenced by the deployment style of sensor nodes in WSNs. Meanwhile, both M-cluster and X-ray have better performance when node density increases.

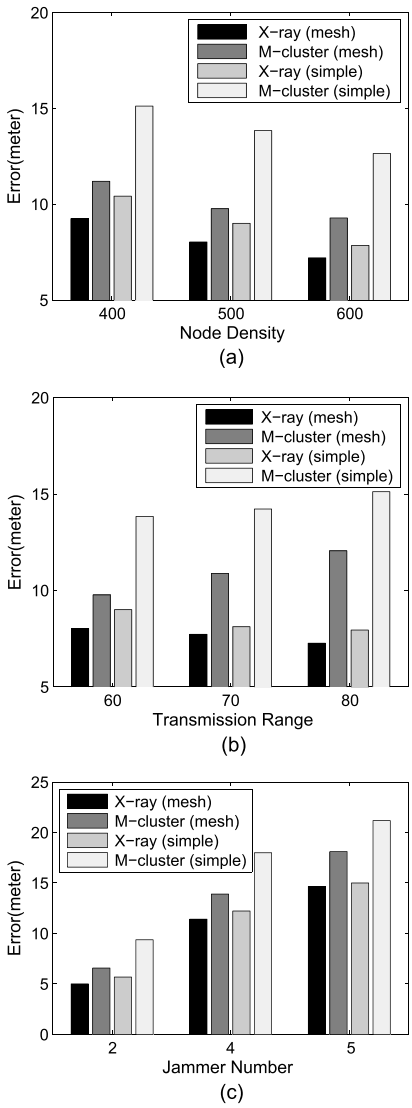

Fig. 9. Impacts of different conditions on X-ray and M-cluster in multi-jammer localization: (a) Node density, (b) Jammer transmission range, (c) Jammer number.

We also provide a view of CDF curves for both our algorithms and the random scheme in Fig. 10. Here, we only show the performances in the 400 and 600 node cases under the simple deployment model. Again we observe that X-ray has the best performance, 90% of the time X-ray can estimate the three jammers' locations with mean errors less than 15 meters, while M-cluster achieves 20 meters 90% of the time. The results of both algorithms improve when node density increases while no clear influence is observed on the random scheme.

Impact of jammer transmission range. To study the

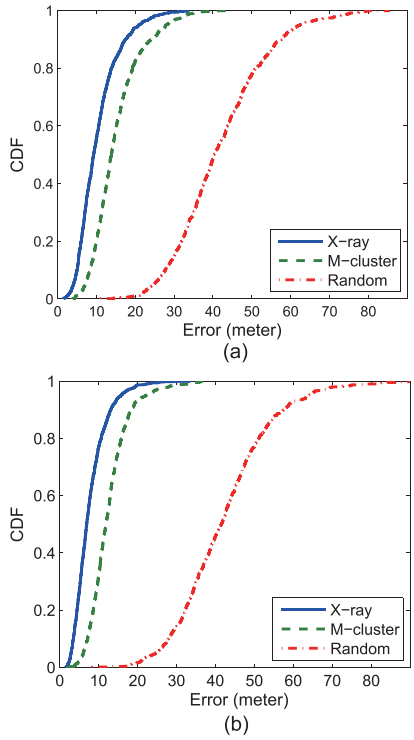

Fig. 10. Impact of different node density with the transmission range set to 60 meter: (a) $N = 400$, (b) $N = 600$.

impact of jammer transmission range on the performance of our algorithms, we evaluate these algorithms in two settings: same fixed transmission ranges and random transmission ranges. In the fixed transmission range scenario, we fix the jammer's transmission ranges at 60, 70 and 80 meters, respectively. In the random transmission range scenario, the jammer transmission ranges are randomly chosen from a certain range in [60, 80], respectively. (Once the transmission range is chosen, it would not change in that simulation). As a result, each jammer has a different transmission range. From Fig. 9(b), the mean-error figure, we observe that, with the increase of jammer transmission range, X-ray has reduced localization errors, while the error of M-cluster increases. This is an interesting observation, and it can be explained as follows. As the jammer transmission range increases, the jammed area becomes bigger; as a result, more sensor nodes are covered in the jammed region. For X-ray, a bigger jammed area will lead to a larger boundary consisting of more jammed nodes, so more branches on the skeleton of the jammed area will be created and X-ray can then obtain more information about the jammers, improving the final estimation accuracy. For M-cluster, a bigger jammed area covers more space and nodes. It is

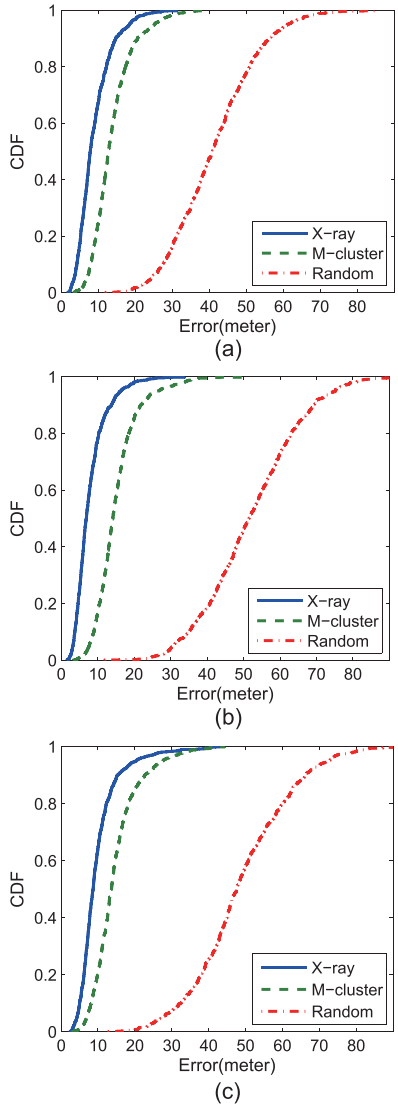

Fig. 11. Impact of different jammer transmission ranges: (a) $R = 60$, (b) $R = 80$, (c) $R \in [60,80]$.

likely that more non-uniformly deployed nodes are introduced into the jammed area, negatively affecting the results of M-cluster.

The CDF performance curves of all algorithms under different jammer ranges are shown in Fig. 11. In Figs 11(a) and (b), the jammer transmission ranges are set to 60 and 80 meters with the simple deployment model. We observe that as the transmission range of the jammers increases to 80 meters, both M-cluster and the random scheme have visible performance declines, while X-ray has a little improvement. As we explained previously, the increment of jammers' transmission range has different impacts on X-ray and M-

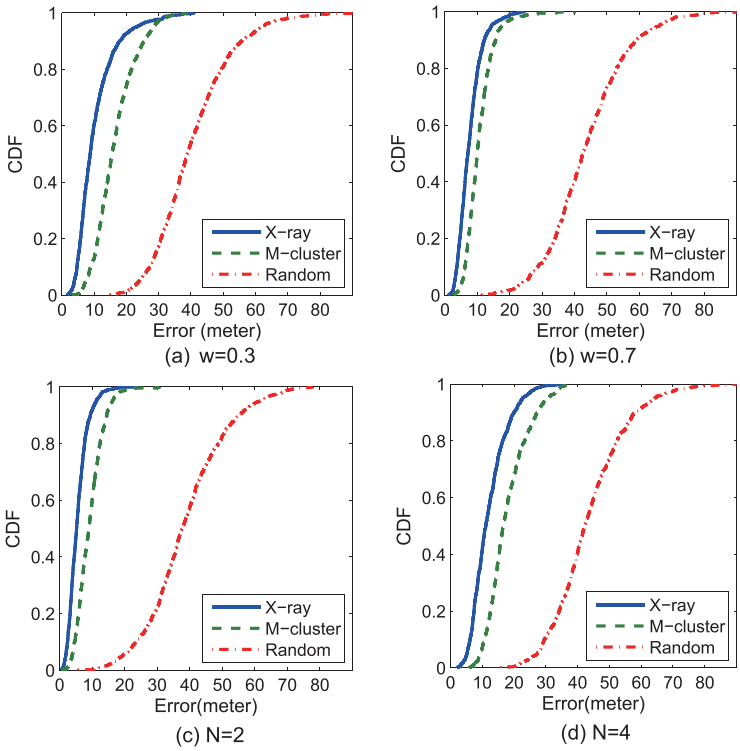

Fig. 12. (1) Impact of different jammer deployments: (a) $\omega = 0.3$, (b) $\omega = 0.7$. (2) Impact of different jammer numbers: (c) $N = 2$, (d) $N = 4$.

cluster. In Fig. 11(c), we show the performance of these algorithms under the condition of random transmission range of jammers, which means each jammer in the network randomly chooses its transmission range ($R \in [60, 80]$). Under this condition, X-ray still achieves a 15-meters estimation error 90% of the time, while M-cluster achieves 20 meters 90% of the time.

Impact of jammer deployment. As introduced in Eq. (2), we use ω to denote the overlapping degree of multiple jammers. To study the impact of jammer deployment, we change the deployment condition, ω, in Eq. (2), from 0.3 to 0.7. The transmission range is set to 60 meters with 500 nodes randomly deployed in the field. In Figs 12(a) and (b), both M-cluster and X-ray have better localization performance when ω increases from 0.3 to 0.7. As ω denotes the overlapping degree of jammers in WSNs, the result indicates that our algorithms have better performance when jammers are less overlapping. This is because when overlapping degree becomes smaller, jammers get farther away from one another. Thus, the overlapping effects become less and the differentiation of jammers becomes easier. Meanwhile, in Fig. 12(b), we observe that M-cluster has a big improvement when the overlapping degree decreases. When jammed regions are less overlapping, clustering algorithm can group jammed nodes more accurately.

Impact of jammer numbers. We also study the effect of the number of jammer numbers by varying it from 2 to 5. In all the cases, we set the jammer transmission range to 60 m and deploy 500 nodes in the field. The results are shown in Fig. 9(c), where X-ray has the best performance in all situations and the mean errors under the mesh deployment are below 15 meters when the jammer number is increased to 5. We observe that the performance of all algorithms decrease with more jammers. Figures 12(c) and (d) show the CDF curves of the algorithms' performance when there are 2 and 4 jammers in the network. We observe that when there are two jammers to be localized, more than 90% of the time X-ray can estimate the jammers' locations with an error less than 10 meters, and M-cluster reaches 13 meters accuracy 90% of the time.

We also show how the performance of our algorithms degrades when the number of jammers further increases. The results are presented in Fig. 13, where the number of jammers is 8 and 10, respectively. We observe that in these cases, the errors of our two algorithms approach 30 m and 40 m, respectively. Whether such big errors are acceptable or not depends on the ap-

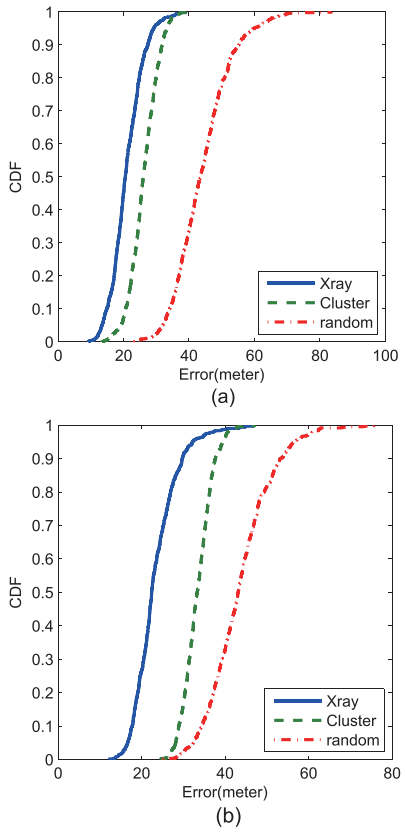

Fig. 13. The limit of our algorithms w.r.t. number of jammers (a) $N = 8$, (b) $N = 10$.

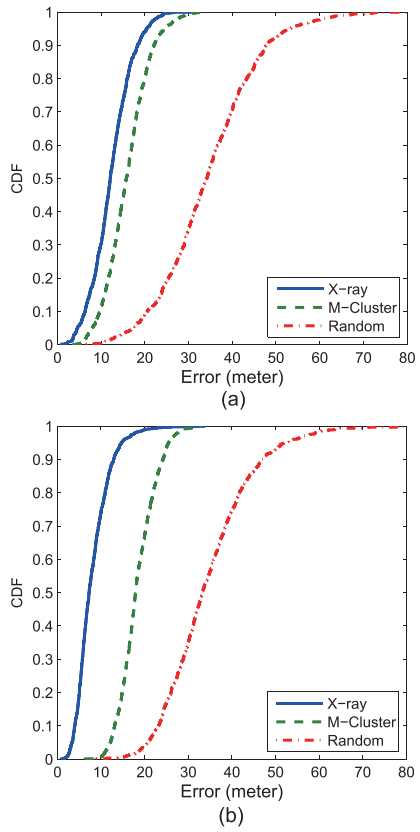

Fig. 14. Impact of jammer-number false detection: (a) $N_e = 2$, (b) $N_e = 4$.

plication scenarios. If one considers localization errors cannot exceed half of transmission range, 10 could be the upper limit for jammer number in our algorithms.

In this part, we also studied the negative impact of false jammer number determination on the performance of our algorithms. In Fig. 14, we show that the CDF of localization errors when the true number of jammers, N_t, is 3, but the estimated jammer number, N_e, is (a) 2, and (b) 4 (in these two cases, we compute the localization errors without considering the missed or double counted one.). From the simulation result, we can observe that due to the false detection of the jammer number, both our algorithms have a decline in localization accuracy, but X-ray still has estimation errors below 19 m for 90% of the time, and M-cluster below 21 m for 90% of the time.

6. Discussion and future work

In this paper, we focus on addressing the multi-jammer localization problem where multiple jammers perform collaborative jamming in WSNs with overlapping jammed areas. While our results show some promise, fully addressing this problem still requires much effort. Next we discuss some of the issues as well as possible solutions.

The first challenge is that attackers may use other types of jamming devices that are more powerful, e.g., using directional antennae. In the case of directional antennae, the jammed area might become more irregular, causing higher localization errors. Torrieri has proposed a direction-finding and localization method based on multiple antennae [29]. His scheme is more resistent to such jamming attacks but currently only works for a single jammer. How to combine his scheme with our algorithms for multi-jammer localization could be an interesting direction for our future work.

The second challenge is how to determine jammer number in WSNs precisely. In our work, through computing the size of a jammed area while assuming an adversary attempts to maximize the coverage of jamming with a fixed number of jamming devices, we use

two simple ways to identify the number of jammers. These algorithms are efficient but likely result in increased errors as the jammer number goes up. Note that when the jammed areas are disconnected, each individual area can be treated separately with our algorithms. Therefore, the question is how to accurately determine the number of jammers within one large jammed area. Clearly, if the adversary does not want to maximize his jamming effect, he may deploy some jamming nodes very closely (i.e., violating our distance constraint). As a result, we may not be able to accurately estimate the number. Indeed, if the jamming nodes simultaneously vary their transmission powers so that they disguise their number or if an unjammed area is totally surrounded by jammed areas (hence no information about the unjammed nodes can be obtained), jammer-number identification could become an undecidable problem.

Fortunately, our ultimate goal is to localize the jammers, not to estimate their numbers. As such, we can have some additional mechanisms to improve the accuracy. First, we may overcome the limitation by scanning the results of the possible jammer numbers (considering not only the calculated jammer number n, but also the numbers $n - 1$ and $n + 1$), and choose the best matched estimation result. Second, the defender is not limited to a single round of localization. It can iteratively localize the jammers in a jammed area until no jammer remains. That is, after each round of localization, the localized jammers will be removed or destroyed immediately and the localization algorithm will be run again when jamming continues. Third, we may improve the localization accuracy of our algorithms. In M-cluster, we only choose the coordinates of jammed nodes as the clustering features; however, some other information may be used to improve final partitions; e.g., RSS of jammed nodes.

In our simulation results, we discover M-cluster is not as good as X-ray under the proposed different conditions. However, compared with X-ray, M-cluster has some considerable advantages on computational complexity (much less computation than X-ray) and practical flexibility (no reliance on the availability of a jammed-area mapping service [31] that is itself very complex). In our M-cluster algorithm, we choose node coordinates as the feature used in clustering algorithms, while other characteristics might be extracted to improve the grouping results. This is our future direction to improve the M-cluster algorithm. In the X-ray algorithm, we choose the convex envelope to compute the jammed area, as it is efficient and suitable to derive a unique simple skeleton by the skeletonization technique. However, some information may be unavailable due to the requirement of convexity. A future improvement of X-ray would be to generate a more accurate jammed area that can preserve the most information of jammers. In conclusion, choosing M-cluster or X-ray is primarily a trade-off between localization accuracy and computational requirements.

7. Conclusion

This paper studied a multi-jammer localization problem in wireless sensor networks and proposed two multi-jammer localization algorithms: M-cluster and X-ray. The algorithms attempt to determine the locations of multiple jammers in WSNs in one run. We made our comprehensive simulation and comparison, and applied our algorithms under variable conditions including different node densities, transmission ranges, overlapping degrees, and jammer numbers. The simulation results show that our algorithms achieve good performance in localizing the jammers under the diverse situations. Future directions include an improved jamming propagation model with shadowing and fading, more accurate determination of jammer number, and further improvement of both X-ray and M-cluster.

Acknowledgments

We thank the reviewers for their valuable feedback which greatly helped improve the work. The work of Sencun Zhu was supported in part by the grant W911NF-11-2-0086 from the Army Research Laboratory (ARL) and the U.S. NSF CAREER 0643906. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of ARL or NSF.

References

[1] X. Bai, L.J. Latecki and W.-Y. Liu, Skeleton pruning by contour partitioning with discrete curve evolution, in: *IEEE Trans Pattern Anal Mach Intell* (2007), 449–462.

[2] Z. Banković, J.M. Moya, A. Araujo, D. Fraga, J.C. Vallejo and J.-M. de Goyeneche, Distributed intrusion detection system for wireless sensor networks based on a reputation system coupled with kernel self-organizing maps, *Integr Comput-Aided Eng* **17**(2) (April 2010), 87–102.

[3] 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), 4:1–4:8.

[4] F. Chang, C.-J. Chen and C.-J. Lu, A linear-time component-labeling algorithm using contour tracing technique, in: *Computer Vision and Image Understanding* (2004), 206–220.

[5] T. Cheng, P. Li and S. Zhu, Multi-jammer localization in wireless sensor networks, in: *Proceedings of International Conference on Computational Intelligence and Security (CIS)* (2011).

[6] T. Cheng, P. Li and S. Zhu, An algorithm for jammer localization in wireless sensor networks, in: *Proceedings of The 26th IEEE International Conference on Advanced Information Networking and Applications (AINA)* (2012).

[7] J.T. Chiang and Y.-C. Hu, Dynamic jamming mitigation for wireless broadcast networks, in: *IEEE Infocom*, (2008).

[8] T.H. Cormen, C. Stein, R.L. Rivest and C.E. Leiserson, Introduction to algorithms, McGraw-Hill Higher Education, 2nd edition, 2001.

[9] Q. Dong and D. Liu, Adaptive jamming-resistant broadcast systems with partial channel sharing, in: *Proceedings of the International Conference on Distributed Computing Systems (ICDCS)*, 2010.

[10] A. Goldsmith, Wireless commulications, Cambridge University Press, 2005.

[11] A. Habib and M. Rupp, Antenna selection in polarized multiple input multiple output transmissions with mutual coupling, *Integrated Computer-Aided Engineering* (2012), 299–312.

[12] X. Jiang, W. Hu, S. Zhu and G. Cao, Compromise-resilient anti-jamming for wireless sensor networks. In *Wirel Commun Mob Comput* Wireless Networks, **17** (2011).

[13] J. Kim, H. Jeon and J. Lee, Network management framework and lifetime evaluation method for wireless sensor networks, *Integrated Computer-Aided Engineering* (2012), 165–178.

[14] M. Li, I. Koutsopoulos and R. Poovendran, Optimal jamming attacks and network defense policies in wireless sensor networks, in: *INFOCOM'07: Proceedings of the 26th IEEE International Conference on Computer Communications.* (2007), 1307–1315.

[15] H. Liu, Z. Liu, Y. Chen and W. Xu, Localizing multiple jamming attackers in wireless networks, in: *ICDCS'11: Proceedings of Int'l Conference on Distributed Computing Systems* (2011), 517–528.

[16] H. Liu, W. Xu, Y. Chen and Z. Liu, Localizing jammers in wireless networks, in: *PERCOM'09: Proceedings of the 2009 IEEE International Conference on Pervasive Computing and Communications* (2009), 1–6.

[17] Y. Liu and P. Ning, Bittrickle: Defending against broadband and high-power reactive jamming attacks, in: *Proceedings of IEEE International Conference on Computer Communications (Infocom)* (2012).

[18] Z. Liu, H. Liu, W. Xu and Y. Chen, Wireless jamming localization by exploiting nodes' hearing ranges, in: *DCOSS'10: Proceedings of the International Conference on Distributed Computing in Sensor System* (2010), 348–361.

[19] G. Mao, B. Fidan and B.D.O. Anderson, Wireless sensor network localization techniques, in: *Comput Netw* **51** (2007), 2529–2553.

[20] S. Misra, R. Singh and S.V.R. Mohan, Information warfare-worthy jamming attack detection mechanism for wireless sensor networks using a fuzzy inference system, in: *Sensors* **10** (2010), 3444–3479.

[21] A. Mpitziopoulos and D. Gavalas, An effective defensive node against jamming attacks in sensor networks, in: *Security and Communication Networks* **2** (2009), 145–163.

[22] K. Pelechrinis, M. Iliofotou and S.V. Krishnamurthy, Attacks in wireless networks: The case of jammers, in: *IEEE Commun Surveys Tuts* **13** (2011).

[23] F.P. Preparata and S.J. Hong, Convex hulls of finite sets of points in two and three dimensions, in: *Commun ACM* (1977).

[24] S. Roger, C. Ramiro, A. Gonzalez, V. Almenar and A. Vidal, An efficient gpu implementation of fixed-complexity sphere decoders for mimo wireless systems, *Integrated Computer-Aided Engineering* (2012), 341–350.

[25] M. Strasser, B. Danev and S. Čapkun, Detection of reactive jamming in sensor networks, in: *ACM Trans Sen Netw* **7** (2010), 16:1–16:29.

[26] M. Strasser, C. Pöpper and S. Čapkun, Efficient uncoordinated fhss anti-jamming communication, in: *MobiHoc'09: Proceedings of the Tenth ACM International Symposium on Mobile ad Hoc Networking and Computing* (2009), 207–218.

[27] P. Tague, M. Li and R. Poovendran, Mitigation of control channel jamming under node capture attacks, in: *IEEE Transactions on Mobile Computing* **8** (2009).

[28] D. Torrieri, Principles of spread-spectrum communication systems, 2nd ed., Springer, 2011.

[29] D. Torrieri, Direction finding of a compromised node in a spread-spectrum network, in: *MILCOM: Proceedings of the IEEE Military Communications Conference* (2012).

[30] Q. Wang, P. Xu, K. Ren and X.-Y. Li, Delay-bounded adaptive ufh-based anti-jamming wireless communication, in: *INFOCOM'11: Proceedings of the IEEE International Conference on Computer Communications* (2011).

[31] A.D. Wood, J.A. Stankovic and S.H. Son, Jam: A jammed-area mapping service for sensor networks, in: *RTSS'03: Proceedings of the 24th IEEE International Real-Time Systems Symposium* (2003), 286–297.

[32] A.D. Wood, J.A. Stankovic and G. Zhou, Deejam: Defeating energy-efficient jamming in ieee 802.15.4-based wireless networks, in: *Proceedings of IEEE SECON'07* (2007).

[33] R. Xu and D. Wunsch, Clustering, Wiley-IEEE Press, 2008.

[34] W. Xu, K. Ma, W. Trappe and Y. Zhang, Jamming sensor networks: Attacks and defense strategies, in: *IEEE Network*, (2006).

[35] 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), 499–508.

Copyright of Integrated Computer-Aided Engineering is the property of IOS Press and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.