LOCALIZATION OF MULTIPLE JAMMERS IN WIRELESS SENSOR NETWORKS

Maria Juhlin and Andreas Jakobsson

Centre for Mathematical Sciences, Lund University, Sweden.
email: {juhlin, aj}@maths.lth.se

ABSTRACT

Wireless sensor networks are susceptible to jamming attacks that can result in communication breakdowns. Preemptive measures to prevent jamming attacks is an active research field, but to stop an ongoing attack often requires that one is able to locate jammers in order to neutralize them. Several methods exist for the case when the network is corrupted by a single jammer, although these generally do not allow for cases when more than one jammer is present. In this work, we introduce an iterative procedure that determines the number of jammers corrupting the network as part of the localization of the jammers. The performance of the method is illustrated using numerical examples.

***Index Terms*—** Jammer detection

I. INTRODUCTION

In most wireless networks, the risk of jamming attacks is ever present, and if successful, a jamming attack can result in information corruption or even in the complete breakdown of communication. This can result in, for instance, erroneous decisions in GPS or satellite navigation systems, potentially with fatal outcomes. Popular actions to defend against jamming attacks are to use preemptive measures such as a clever and robust network design, often in combination with the use of communication protocols such as e.g. MAC, and other security measures, such as, e.g., user authentication. Notable research conducted in jammer attack prevention can be found in e.g. [1]–[4]. When a successful jamming attack does occur, active action must be taken to fend off the attack. In order to be able to counter jamming attacks, it is generally critical to accurately locate the jammer's location, whereafter it may be neutralized in a suitable way. Thus, an important problem is to accurately estimate the location of jammers, and given the prevalence of wireless sensor networks (WSNs), jammer localization in WSNs is a problem currently attracting notable interest.

There are two main branches of jammer localization/detection, namely *range based* and *range free* [5] approaches. The range based methods utilize the fact that the strength of the jamming signal depends on the distance between the jammer and the jammed sensor. By estimating the jamming signal strength at different locations, the jammer's location may then be deduced. The range free methods on the other hand only rely on the information about which nodes that are jammed, not the extent of the jamming, which is often practically more feasible. Range free methods require information about the network topology and all sensor positions, which may be known from deployment or estimated in a pre-processing stage. The range free methods on the other hand are attractive due to their relative simplicity, and they also have the benefit of not requiring *a priori* knowledge of the signal strength at all the sensors. This makes them more versatile than range based estimators, even though the latter theoretically may have preferable performance [5].

To date, most of the literature has focused on jammer localization in the single jammer scenario, see e.g. [5]–[9]. One main weakness with these methods is their inability to handle multiple jammers with overlapping jamming regions. A notable contribution to the problem of multi-jammer localization was made in [10], wherein the authors present the so-called X-ray and M-cluster methods. The X-ray method determines the entire jammed area by forming the convex hull of the jammed sensors. Then, the different cluster centers used in the jammer localization are found by finding the branching points of the skeleton that maps the topology of the jammed region. The M-cluster approach on the other hand groups the jammed nodes by means of a so-called fuzzy c-means clustering, followed by an iterative centroid localization for each of the clusters [10]. For both methods, the number of jammers is estimated at the start of the procedure, using *a priori* assumed knowledge of the jammers' average transmission range. The methods are efficient if the transmission range is known, or can be well estimated. However, when the number of jammers is inaccurately estimated, or for non-circular transmission patterns, the average transmission range does not accurately reflect the jammed area, the methods performs poorly.

In this paper, we present a novel method for multi-jammer localization with overlapping regions without requiring any prior knowledge about the average jammer transmission range. Our approach follows an iterative procedure. In the first step, we assume the presence of only a single jammer (N_j=1). Then, in the following step, we estimate the locations of the assumed N_j jammers as the clustering centers produced by a Gaussian mixture model. The jammer estimates are then improved by iteratively finding the N_j

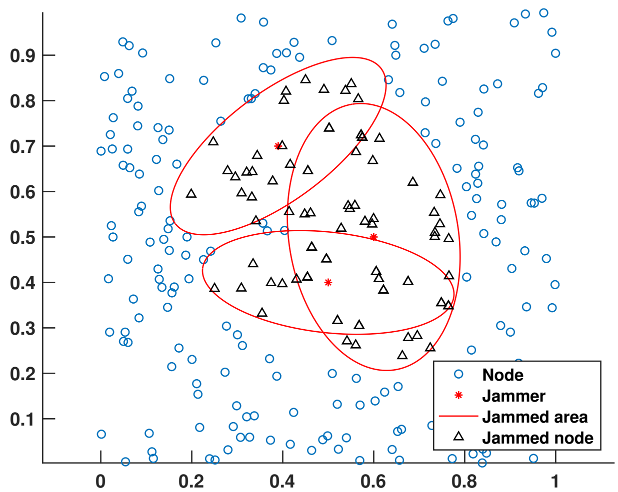

Fig. 1. An illustration of a WSN containing three elliptical jammers. The jammer position is marked as red stars.

maximally inscribed ellipsoids in the jammed area. The method then tests the obtained result for plausibility to determine if the found jammers can well explain the pattern of jammed sensors. If the test fails, the assumed number of jammers, N_j, is increased by one and the localization step is repeated, followed by a new plausibility test, and so on.

II. PROBLEM FORMULATION

One may generally model a wireless sensor network by means of a graph, where each sensor is represented by a node. The network is then described by $G = \{V, E\}$, where V denotes the set of nodes and E the set of edges linking the nodes together. In this setup, E is a binary set; if two nodes can communicate directly, they are considered to be neighbours and their common edge gets the value one, otherwise it is set to zero. The network is here considered to be two dimensional, with the ℓth sensor assumed located at position (x_ℓ^s, y_ℓ^s). Moreover, the nodes are assumed to be stationary, such that their locations are fixed. The node placement may be modelled as having been made in a stochastic manner, or methodically, e.g, on a uniform grid. The nodes are typically divided into three categories; jammed, boundary node, and unaffected. A node is considered jammed if it is inside the jammed area, a boundary node if at least one neighbouring node is being jammed, and unaffected if it is not jammed and has no jammed neighbours.

We are herein considering the case when the jammers are also stationary and may be placed anywhere in the network, with the kth jammer being located at position (x_k^j, y_k^j). Furthermore, different from most other approaches in the literature that makes an isotropic assumption on the jammed region, i.e., that the jammed region is circular with radius equal to the jammers transmission range, we here also allow for jamming signals that are symmetric but not necessarily isotropic, thereby allowing for elliptical jamming regions reflecting also the use of directional jammers.

All nodes inside the jammed region are assumed to be completely jammed. In order to model overlapping regions, the jammers are in our simulations placed in such a way that their respective jamming regions overlap, but are allowed to have different transmission ranges. Figure 1 illustrates an example of a network topology with three elliptical jammers with the resulting jammed and unaffected nodes.

III. JAMMER LOCALIZATION METHODS

A conceptually simple localization method is the centroid localization (CL) technique [6]. The method assumes a single jammer and that the positions of all the jammed sensor nodes are known, allowing the jammer's location to be estimated as the arithmetic mean of the jammed sensors' locations. In a two dimensional setup, let the ith jammed sensor be located at $z_i^s = (x_i^s, y_i^s)$, for $i = 1, \ldots, N_j$, where N_j is the total number of jammed nodes. A single jammer's location, (x_1^j, y_1^j), may then be estimated as

$$\hat{x}_1^j = \frac{1}{N_j} \sum_{i=1}^{N_j} x_i^s \tag{1}$$

$$\hat{y}_1^j = \frac{1}{N_j} \sum_{i=1}^{N_j} y_i^s \tag{2}$$

Clearly, the localization accuracy of CL will be highly dependent on the node distribution in the jammed area, performing best when the nodes are evenly spread in the jammed area; if the nodes are spread asymmetrically, this will cause a shift towards the high node density area in the resulting jammer location estimate.

One way to improve the CL estimate is to introduce weights on the jammed node coordinates, as is done in weighted centroid localization (WCL) [7]. The introduction of weights enables different emphasis on the different nodes. If the distance between a node and the jammer can be estimated, a commonly used weight is one that is inversely proportional to this distance. The resulting jammer location is then calculated as

$$\hat{x}_1^j = \frac{1}{N_j} \sum_{i=1}^{N_j} w_i x_i^s \tag{3}$$

$$\hat{y}_1^j = \frac{1}{N_j} \sum_{i=1}^{N_j} w_i y_i^s \tag{4}$$

where w_i is the weight for node z_i^s, for example selected as $w_i = d_i^{-1}$, where d_i is the distance between node z_i^s and the jammer. Examples of other weight functions that are used may be, e.g., the hop count between different nodes [11].

The virtual iterative force location (VFIL) estimator was introduced in [5] to mitigate the dependence on node density and distribution limiting the performance of CL and WCL, as well as to avoid the need to estimate the weights in WCL.

The multi-jammer scenario was addressed in [10], where the X-ray method and M-cluster method were introduced. In the X-ray approach, the jammed area is calculated as the convex hull of the jammed nodes. Next, a skeletonization of the boundary region of the jammed sensors is performed to find the different cluster centers used in the jammer localization. This step will map the topology of the jammed region, and the cluster centers are found as the centers of the maximal (circular) disks inside the jammed area. It should be noted that the method estimates the number of jammers as an initial step of the algorithm, assuming *a priori* knowledge of the jammer's average transmission range in order to do so; this is a notable weakness of the method as this range is generally difficult to determine prior to an attack.

The M-cluster method instead aims at clustering the jammed nodes into M clusters using the fuzzy c-means algorithm. Next, the jammer localizations are estimated by applying CL to the different clusters. Assuming knowledge of the transmission range, an estimated jammed area is formed. If this area corresponds to the real jammed area, all jammed nodes should be covered and no boundary nodes. If the jammed areas do not match, some cluster centers are moved, and the process iterated. The fuzzy c-means process requires the user to define the user-defined "fuzzyness" parameter α, which determines the allowed overlap between the clustered sets. As the amount of overlap is not known *a priori*, and the location estimates are highly dependent on the cluster points, the choice of α will notably affect the resulting estimate.

IV. PROPOSED METHOD

In this work, we aim at extending the earlier works by proposing an iterative method, schematically described in Figure 2. Initially, it is assumed that the number of jammers is one, such that $N_j = 1$. An initial localization estimate is then formed using the CL method. In order to determine if all jammers have been found, we form an intuitive test by considering the jammed region generated by a jammer at the estimated location. Denote the set of jammed nodes, $\mathbb{S}_j^t$, and the set of neighbour nodes, $\mathbb{S}_n^t$. If the correct number of jammers, and their positions, have been determined accurately the estimated jammed area should enclose all of the jammed nodes and none of the neighbouring nodes, i.e.,

$$\mathbb{S}_j^t = \mathbb{S}_j^e \tag{5}$$

$$\mathbb{S}_n^t = \mathbb{S}_n^e \tag{6}$$

where $\mathbb{S}_j^e$ and $\mathbb{S}_n^e$ denote the estimated set of jammed and neighbouring nodes, respectively. If the neighbouring nodes are within the jammed region, or jammed nodes are outside the jammed region, it may be concluded that the number of jammers was not correctly determined, and we proceed to increase the number of assumed jammers by one.

In this case, one needs to decompose the jammed region into smaller overlapping regions spanned by the individual

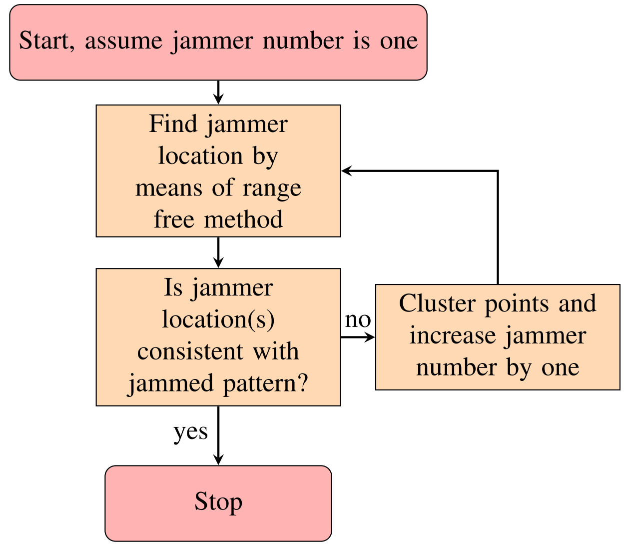

Fig. 2. Flowchart of the proposed algorithm.

jammers. We propose doing this by assigning a likelihood that each node belongs to a given cluster, such that the node assignment may be viewed as a Gaussian mixture model (GMM). To form the estimated posterior probabilities, we here employ the Expectation Maximization (EM) algorithm (see, e.g., [12]). Next, we apply a K-means clustering with a soft thresholding[1] to the estimated posterior probabilities of each node in order to form the optimal cluster partitioning, $\hat{C}$. This is done by iteratively partitioning the data into N_j overlapping clusters by minimizing [13]

$$\hat{C} = \underset{C,\{m_k(C)\}}{\text{minimize}} \sum_{k=1}^{N_j} N_k \sum_{i=1}^{N_s} \mathbb{1}\{C(i) = k\} ||z_i - m_k(C)||^2 \tag{7}$$

where C is the cluster partitioning, i.e., $C(\ell)$ indicates which cluster the ℓth node belongs to, $m_k(C)$ is the kth cluster mean as formed using the clustering partitioning C, N_k is the number of points in cluster k, N_s the number of sensor nodes to cluster, and $\mathbb{1}\{C(i) = k\}$ is the indicator function, specifying that node i belongs to cluster k.

Having computed the cluster centers, we proceed by estimating the maximally inscribed ellipsoids in the jammed area centered at the cluster centers. For each cluster center, the maximally inscribed ellipsoid is iteratively estimated by growing a circle centered at the cluster center. As soon as the circle reaches a neighbour node, the growth in that direction stops, and the vector from the cluster center to the encountered node is termed $\mathbf{b}_1$. The ellipsoid then continues to grow in the direction $\mathbf{b}_2$ perpendicular to $\mathbf{b}_1$, until the next neighbour node is encountered. When $\mathbf{b}_2$ encounters

[1]The soft threshold is included in order to allow nodes with posteriors in a given range to belong to more than one cluster.

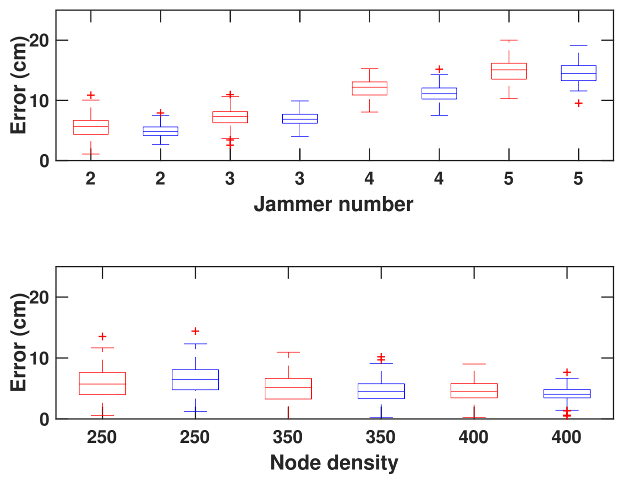

Fig. 3. Top: The rMSE for different number of jammers, showing the X-ray estimate (red, left) and the proposed method (blue, right). Bottom: The rMSE for different node densities, showing the X-ray estimate (red, left) and the proposed method (blue, right)

a neighbour the ellipsoid stops growing. When all the ellipsoids have been calculated, the covered area is evaluated. If all jammed nodes are covered, while no neighbour nodes are, the procedure is terminated. If not, the cluster centers are moved by a predetermined step size, λ, in the direction anti-parallel to the sum of the vectors, $\mathbf{b} = \mathbf{b}_1 + \mathbf{b}_2$. Then, the new cluster center is found as

$$\hat{x}_i^{j_{new}} = \hat{x}_i^{j_{old}} - \lambda\langle \mathbf{b}, \boldsymbol{e}_x\rangle \quad (8)$$

$$\hat{y}_i^{j_{new}} = \hat{y}_i^{j_{old}} - \lambda\langle \mathbf{b}, \boldsymbol{e}_y\rangle \quad (9)$$

where $(\hat{x}_i^{j_{new}}, \hat{y}_i^{j_{new}})$ and $(\hat{x}_i^{j_{old}}, \hat{y}_i^{j_{old}})$ represent the new and old estimates of the jammer's coordinates, $\langle\cdot\rangle$ represents the inner product, and $\boldsymbol{e}_x$ is the canonical basis vector. The process is then reiterated. If all the ellipsoids have been maximized and the estimated jamming pattern is still not consistent, the number of assumed jammers is increased by one, and full procedure is repeat as outlined above (see also Figure 2).

V. NUMERICAL EXAMPLES

We proceed to evaluate the performance of the proposed method, using the Euclidean distance between the jammers true and estimated locations as the measure for the accuracy of the estimates. As a comparison, we show the performance of the X-ray method, which in [10] was shown to offer preferable performance to the M-cluster method.

We initially evaluate how well the methods perform when the number of overlapping jammers in a cluster grows. In this setup, we use a sensor node density of 300 nodes placed randomly in the unit square ($1m \times 1m$), and place the sought number of jammers in such a way that their jamming regions will overlap, while still being within the unit square. Thus, no jammers will be placed at the edges of the square in the simulations.

We randomly select the half semi-axes a and b, and rotation θ, of the jammer transmission ellipses[2] by drawing these from uniform distributions, such that a jammer located at (x_c, y_c) will have a transmission range

$$\frac{(d_x \cos\theta - d_y \sin\theta)^2}{a^2} + \frac{(d_x \sin\theta + d_y \cos\theta)^2}{b^2} = 1 \quad (10)$$

where $d_x = x - x_c$, $d_y = y - y_c$, with $a \in \mathcal{U}(0.15, 0.30)$, $b \in \mathcal{U}(0.10, 0.25)$, and $\theta \in \mathcal{U}(-\pi, \pi)$. The step size is set to be $\lambda = 0.001$.

Figure 3 shows the resulting root mean squared error (rMSE) of the location estimates as a function of the number of jammers in the cluster (top figure), where

$$\text{RMSE} = \frac{1}{MC}\frac{1}{N_j}\sum_{m=1}^{MC}\sum_{i=1}^{N_j}\epsilon_i \quad (11)$$

with ϵ_i denoting the estimated location error for the i:th jammer, i.e.,

$$\epsilon_i = \sqrt{(x_i^j - \hat{x}_i^j)^2 + (y_i^j - \hat{y}_i^j)^2} \quad (12)$$

where (x_i^j, y_i^j) and $(\hat{x}_i^j, \hat{y}_i^j)$ denote the true and estimated location of the ith jammer, respectively. Here, we use $MC = 1000$ Monte-Carlo simulations. The figure illustrates the preferable performance for the proposed method (blue boxes, to the right of each pair), as compared to the X-ray estimate (red boxes, to the left in each pair). As can be seen in the figure, the proposed method performs better than the X-ray estimate. It may also be noted that the average error increases as the number of jammers increase, which should be expected as the difficulty of the localization problem increases with the number of jammers present.

Proceeding, we investigate how the methods' performance depends on the node density. For this procedure, we fix the number of jammers to $N_j = 2$, and randomly place 250, 350, and 400 nodes in the area. The results are shown in the bottom plot of Figure 3, wherein it may also be noted that the average error decreases as the node density increases, as may be expected. Again, we note that the presented method offers preferable performance as compared to the X-ray estimate.

Lastly, we investigate the methods' probability of estimating the correct number of jammers. The results for the discussed methods are presented in Figure 4. It should be noted that the X-ray method uses a heuristic based on the average transmission range, which will perform worse the more elliptical the transmission pattern is. Denoting A_{tot} the total jammed area and A_{avg} the average area covered by

[2] It should be noted that the presented method offers a similar performance gain as compared to the X-ray method if the jamming transmission is restricted to be circular.

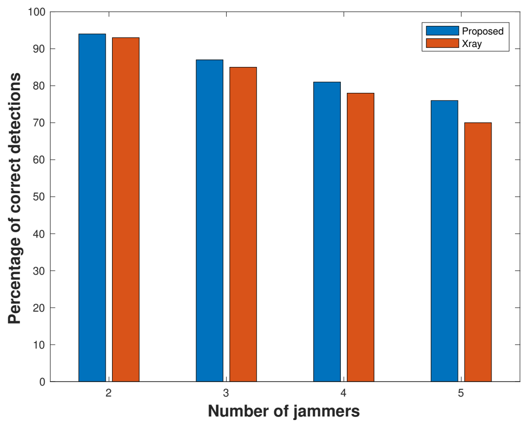

Fig. 4. The probability of correctly estimating the number of jammers present in the network showing the results for the X-ray estimate (red, left) and the proposed method (blue, right).

a single jammer, the number of jammers present in a cluster is then estimated

$$N_j^{est} = \text{ceil}\left(\frac{A_{tot}}{A_{avg}}\right) \tag{13}$$

where ceil rounds the result up towards nearest integer. To allow for a fair comparison with the X-ray method, we for the proposed method estimate the expected area covered by a single jammer as $A_{avg} = \pi\bar{a}\bar{b}$, where $\bar{a}$ and $\bar{b}$ denote the average half semi-axes. As can be seen in the figure, the probability of correctly determining the number of jammers decreases as the number of jammers increase, but the rate of decrease is slower for the proposed method than for the X-ray method, indicating that the proposed method is more robust than the X-ray method for cases when the network is corrupted by more than one jammer.

VI. CONCLUSION

In this paper, we have explored the problem of localizing multiple jammers in wireless sensor networks. The presented algorithm allows for an unknown number of jammers, without making assumptions on the jammers transmission range, nor restricting the jammers to be isotropic. Using an iterative procedure the methods estimates the locations of the assumed jammers, whereafter the plausibility of the estimate is determined; if deemed unsatisfactory, the number of assumed jammers is increased and the procedure repeated. Numerical simulations illustrate the preferable performance of the proposed method, both in terms of the lower localization errors and in the likelihood of correctly determining the number of jammers, as compared to the state-of-the-art X-ray technique.

VII. REFERENCES

[1] F. Tao, T. Wang, and J. Wu, "A novel jammer suppression method based on kaspice-stap," in *International Radar Symposium (IRS)*, 2019.

[2] A. A. Bodkhe and A. R. Raut, "Identifying Jammers in Wireless Sensor Network with an Approach to Defend Reactive Jammer," in *Fourth International Conference on Communication Systems and Network Technologies*, 2014.

[3] W. Xu, T. Wood, W. Trappe, and Y. Zhang, "Channel Surfing and Spatial Retreats: Defenses Against Wireless Denial of Service," in *Conference: Workshop on Wireless Security*, 2014.

[4] Y. S. Kim, F. Mokaya, E. Chen, and P. Tauge, "All your jammers belong to us - localization of wireless sensors under jamming attack," in *IEEE International Conference on Communications (ICC)*, 2012.

[5] H. Liu, W. Xu, Y. Chen, and Z. Liu, "Localizing jammers in wireless networks," in *IEEE International Conference on Pervasive Computing and Communications*, 2009.

[6] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, pp. 28–34, 2000.

[7] J. Blumenthal, R. Grossman, F. Golatowski, and D. Timmermann, "Weighted Centroid Localization in Zigbee-based Sensor Networks," in *IEEE International Symposium on Intelligent Signal Processing*, 2007.

[8] H. Huang, X. Hu X. Zhang, P. Zhang, and Y. Li, "An Optimal Jammer Selection for Improving Physical-Layer Security in Wireless Networks with Multiple Jammers," in *International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2016.

[9] Y. Sun, R. Molva, M. Onen, X. Wang, and X. Zhou, "Catch the Jammer in Wireless Sensor Network," in *IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, 2011.

[10] T. Cheng, P. Li, S. Zhu, and D. Torrieri, "M-cluster and X-ray: Two methods for multi-jammer localization in wireless sensor networks," *Integrated Computer-Aided Engineering*, vol. 21, pp. 19–34, 2014.

[11] A. Kaur, P. Kumar, and G. P. Gupta, "A weighted centroid localization algorithm for randomly deployed wireless sensor networks," *Journal of King Saud University - Computer and Information Sciences*, vol. 31, 2019.

[12] T. K. Moon, "The Expectation-Maximization Algorithm," *IEEE Signal Process. Mag.*, vol. 13, pp. 47–60, November 1996.

[13] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer, 2 edition, 2009.