CluRoL: Clustering based Robust Localization in Wireless Sensor Networks

Satyajayant Misra and Guoliang Xue†

***Abstract*— In a wireless sensor network (WSN), the sensor nodes (SNs) generally localize themselves with the help of anchors that know their own positions. In this setting, the localization process has a high risk of being subverted by malicious anchors that lie about their own position and/or distance from the SNs. In this paper, we propose an efficient scheme that helps the SNs identify these malicious anchors and discard them from the localization process. We introduce the concept of the *bound circle* of an anchor with respect to an SN as the circle whose center is at the anchor and whose radius is the estimate of the distance between the anchor and the SN. Two bound circles may intersect, resulting in at most two intersection points, of which at least one point is close to the true position of the SN, such a point is defined as a *proximal point*. Pairwise intersection of bound circles results in a dense cluster of proximal points around the position of the SN. This is true even when some of the anchors used by an SN for localization are malicious and are colluding with an aim to have the SN localized at a false position. We propose *CluRoL*, a technique that helps each SN to localize itself accurately, using a clustering mechanism that performs clustering of these proximal points. Using the resulting cluster the SN is able to identify the false anchors and exclude them from its localization process. Our technique is decentralized and can be easily used by the standard sensors. Simulation results indicate that when the malicious anchors are not colluding CluRoL can identify on an average more than 72% of them. CluRoL performs even better when the malicious anchors are colluding in an attempt to localize an SN at a false position, identifying more than 85% of the malicious anchors. CluRoL also has very low false positives.**

I. Introduction

Large scale distributed wireless sensor networks (WSNs) have become popular in both the military and civilian domains because of their infrastructureless nature and relative ease of deployment [1]. However, there still exist many fundamental problems that need to be addressed [7]. The problem of robust localization of the wireless nodes is one such problem. In an infrastructureless WSN, for cost effectiveness, not all nodes are equipped with self-localizing capabilities. Most sensor nodes (SNs) localize themselves using the position estimates of a group of nodes in the network called *anchors* [10], [11], [13]. Each anchor is a fixed wireless node that knows its own position accurately, either through GPS or from pre-programmed information.

In this paper, we assume that Time Difference of Arrival (TDoA) [13], [15], is the underlying mechanism used for localization. Following the TDoA method, each anchor a_i periodically broadcasts its identifier (ID) and position information in its neighborhood, via a radio signal (RS) and an ultrasound signal (US) at the same time instance. We denote these two components together as the *location reference*. On receipt of the location reference l_i from a_i, a sensor node (SN) u, calculates the time difference in receipt of the signals and uses the constants, speed of light (c) and sound (s), to obtain an estimate $\hat{d}_{iu}$ of its distance (d_{iu}) from a_i. The calculation of the estimate $\hat{d}_{iu}$ is given below.

$$\hat{d}_{iu} = \Delta t \cdot \frac{1}{1/s - 1/c}\ , \tag{1}$$

where Δt is the time difference between the receipt of the RS and the US. We note that the wireless medium is inherently error-prone, hence the value of Δt is inaccurate. This results in u being able to only estimate d_{iu}.

When u gets a sufficient number of location references from anchors in its vicinity, it can use them to estimate its own position. The estimation can be done using the Minimum Squared Error (MSE) (also known as the minimum mean square error) method [10], [15] given by,

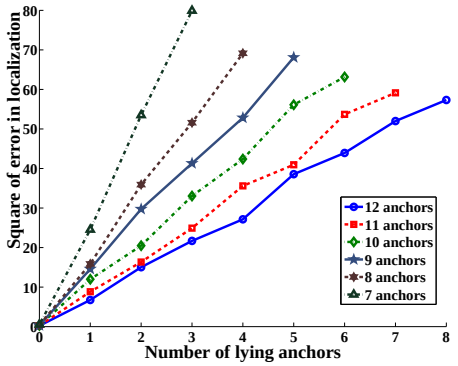
$$\min_{\hat{u}} \sum_{a_i \in \mathcal{A}_u} (||\hat{\mathbf{u}} - \mathbf{a_i}|| - \ \hat{d}_{iu})^2, \tag{2}$$

where $\hat{\mathbf{u}}$ is an estimate of the real position $\mathbf{u} = (u_x, u_y)$ of u, $\mathbf{a_i} = (a_{ix}, a_{iy})$ is the position of anchor a_i, $\hat{d}_{iu}$ is the estimate of the distance between a_i and u, calculated by u using the TDoA method, and $\mathcal{A}_u$ is the set of anchors from whom u receives the location references. In the absence of measurement errors, $\hat{\mathbf{u}}$ is the correct estimate, that is, $||\hat{\mathbf{u}} - \mathbf{u}|| = 0$. In the presence of measurement errors, the error in $\hat{\mathbf{u}}$ is dependent on the measurement error. In this scenario, accurate localization is fairly complex as it is difficult to bound the estimation error. The presence of malicious (lying) anchors makes accurate localization significantly more difficult. We demonstrate this with illustrative simulation results.

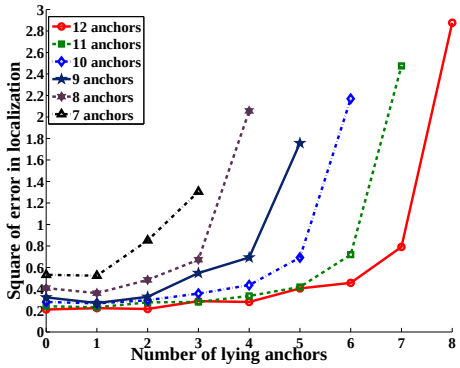
Motivation: In our illustrative simulation set-up, each malicious anchor lied in a way that its distance from an SN it is involved in localizing is in $[d, d \cdot (1 + \epsilon)]$, where d is the true distance and $\epsilon = 0.5$ (for this illustration only). Fig. 1(a) shows the average of the square of the localization error (S_{err}) over 20 iterations, when lying anchors are included in the localization process. Fig. 1(b) shows S_{err} when the lying anchors are not included in the localization process. We would like the reader to note the difference in scale of the Y-axis in the two figures and point out that the value of S_{err} when the number of lying anchors is 0 is the same in both cases. It is easy to see that the error in localization when malicious anchors are included is an order of magnitude higher than when the localization is done with only the true anchors.

This research was supported in part by ARO grant W911NF-04-1-0385 and NSF grants CNS-0524736 and CCF-0431167. The information reported here does not reflect the position or the policy of the federal government.

† Both the authors are with the Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85287-8809. Email: {satyajayant, xue,}@asu.edu.



(a) Malicious anchors included in localization



(b) Only true anchors used for localization

Fig. 1. Localization error in MSE method

For instance, when there are 10 anchors in the range of an SN and 5 of them are malicious, the inclusion of malicious anchors in localization results in a value of $S_{err} > 50$ sq. m. However, when localization is done without the malicious anchors, the value of $S_{err} < 0.8$ sq. m. Thus, we can conclude that the presence of malicious anchors is detrimental to accurate localization and their revocation is necessary to increase accuracy.

In this paper, we propose CluRoL, a distributed clustering technique that helps SNs identify the malicious anchors and performs accurate localization without these anchors, using MSE. Simulation results indicate that CluRoL successfully identifies on an average more than 72% of the malicious anchors in the network when they are not colluding and more than 85% of the malicious anchors when they are colluding. The subsequent localization has significantly improved accuracy.

In Section II, we present related work. In Section III, we present the system and threat models along with our assumptions. Section IV presents our proposed mechanism, while Section V presents the simulation results. We conclude our paper in Section VI.

II. Related Work

Localization schemes in WSNs may be classified as range-based and range-free. The range-based mechanisms [3], [5], [10], [16], perform localization by measuring properties such as point-to-point distance or angle estimates, whereas the range-free mechanisms [6], [8], [9], [11], [15] do not require any physical measurements to perform localization. Range-free mechanisms may use hop count or area-based estimation to localize a node [6]. Generally, range-based mechanisms lead to more accurate localization. However, they tend to be resource intensive and may require specialized hardware [13], [16]. The method used for position estimation may be based on minimum mean/median square estimation [10], [15], convex programming [2], [4], or triangulation [16].

Many schemes have been proposed [5], [8], [9], [10], [11], [16] to increase security and robustness of localization by performing secure localization, location anomaly detection, or location verification. Accurate localization in the presence of malicious anchors that are transmitting erroneous estimates has been dealt with in [5], [10], [11]. The schemes in [5], [10] attempt to identify the anomaly and perform compromise-resistant localization, whereas the scheme in [11] attempts to detect and remove the malicious anchors from the network.

Our scheme CluRoL, is fully distributed. It allows the SNs to localize themselves with high accuracy, in the presence of colluding malicious anchors, without any external assistance. The Least Median Square (LMS) based scheme proposed by Li *et al.* [10] is the only other scheme with similar objectives. In this paper, we compare the result of CluRoL with the LMS scheme. CluRoL results in more accurate localization than LMS and also has much lower time complexity. In LMS, to achieve high accuracy, the parameters have to be estimated accurately. This requires a search over an exponential number of subsets of the location references, resulting in higher time complexity. CluRoL, however, is polynomial in the number of anchors.

III. System Model and Assumptions

The network consists of a set of anchors $\mathcal{A} = \{a_i, i = 1, \ldots, K\}$ and a set of sensors $\mathcal{S} = \{s_i, i = 1, \ldots, L\}$ that are deployed randomly and are fixed after deployment. Each anchor a_i knows its own position $\mathbf{a_i}$ ($\mathbf{a_i} = (a_{ix}, a_{iy})$). The transmission range of the SNs is $r > 0$ and that of the anchors is $R \geq r$. The anchors are equipped with radio/ultrasound transmitters and can transmit both signals simultaneously. The SNs are equipped with both radio and ultrasound receivers. All devices have omnidirectional antennas. The anchors broadcast their location references periodically. The measurement error in the distance estimate is proportional to the actual distance of the anchor from the SN u. The measurement error proportion for the estimate of the distance between a_i and u is a uniform random variable, $\delta_{iu} \backsim \mathcal{U}[-\delta_{max}, \delta_{max}]$. The anchors transmit their references encrypted using a key from a hash chain. The key is released at a later time instant (delayed key disclosure mechanism), similar to the μTESLA scheme [14].

A. Assumptions

When the malicious anchors are not colluding, each malicious anchor a_i from whom an SN u receives location reference lies independently, with the resulting false distance estimate $\hat{d}_{iu} = d'_{iu}(1 + \epsilon_{iu})$, where d'_{iu} is the measured distance between a_i and the SN u and $\epsilon_{iu} \sim U[-\epsilon_{max}, \epsilon_{max}]$ is the lying proportion, where ϵ_{max} is an unknown constant. The malicious anchors can also collude by changing their distance estimates so that the SN localizes itself at a false position ($\mathbf{x_f}$). For a malicious anchor a_i, if $d_{if} =$

$\|\mathbf{a_i} - \mathbf{x_f}\|$, then the false distance estimate $\hat{d}_{if}$ of a_i is such that $\hat{d}_{if} \sim U[d_{if}(1 - \delta_{iu}), d_{if}(1 + \delta_{iu})]$. If the number of anchors from which an SN receives location references is N, then an upper bound on the number of malicious anchors M that can be handled is given by $M \leq \lfloor N/2 \rfloor - 2$. This upper bound on the number of malicious anchors for accurate localization was proved in [12], when the measurement errors were absent. This bound also holds when measurement errors exist. The SNs are pre-deployed with the position information of all the anchors in the network. Since the number of anchors in the network is generally small, this is feasible. The anchors have the ability to generate and store hash chains of the keys used to authenticate the location references. The deployment authority (DA) knows the last value (K_0) of the hash chain for each anchor, which can be used to verify any element of the chain. The DA installs K_0 of each anchor in each SN during pre-deployment.

B. Threat Model and Security Assumptions

In a WSN, the adversary may be classified as, either an *outside adversary* or an *inside adversary*. An outside adversary is not part of the network and generally is more powerful than the SNs. It has bounded abilities to jam or eavesdrop on communication, compromise legitimate nodes, and inject false nodes in the network. An inside adversary on the other hand, is a node in the network that has been compromised. The inside adversary is also a potent attacker as it is a part of the system and hence is privy to the shared secrets. We assume that the malicious anchors may be compromised by a powerful external adversary to lie about their distance references. Hence the malicious anchors are internal adversaries. In this subsection, we use a_i to denote a malicious anchor.

The use of delayed key disclosure by the anchors for transmitting their location references ensures that malicious anchors in the neighborhood cannot change or replay the references. In addition, malicious anchor a_i cannot revoke a true anchor a_j by masquerading as a_j and broadcasting false location references. CluRoL is not affected by wormholes. A wormhole is created when two adversaries have a communication link with latency that is much lower than the other links in the network. A wormhole is dangerous as it may be used to subvert the localization process. A malicious anchor far from an SN can be made to appear very close using a wormhole. A reference that is affected by wormhole is identified as malicious and is not used in the localization process. The malicious anchors information obtained by the SNs during the operation of CluRoL can also be relayed to the BS. The BS can analyze this information to identify the wormholes. We do not discuss the technique in detail in this paper. It suffices to say that CluRoL ensures that wormholes do not affect localization accuracy.

In this set-up, there are only three possible mechanisms by which a malicious anchor can subvert accurate localization, namely by lying about its position, its distance (by not transmitting the RS and the US simultaneously), or by lying both ways. For a_i, lying about its position is not viable as the SNs know the positions of the anchors in the network and can easily detect the location discrepancy. The only other feasible attack is distance enlargement/reduction attack. We note that, if the true distance of a true anchor a_j from the SN u is d_{ju}, due to measurement error, the distance estimate $\hat{d}_{ju}$ can become $\hat{d}_{ju} = d_{ju} \cdot (1 + \delta_{ju})$. If malicious anchor a_i lies about its estimate, then $\hat{d}_{iu} = d_{iu} \cdot (1 + \delta_{iu}) \cdot (1 + \epsilon_{iu})$.

Another potent attack is collusion. The malicious anchors can collude to localize the SN at a false position, $\mathbf{x_f}$. The colluding anchors can choose $\mathbf{x_f}$ in the network and manipulate the value of their distance estimates such that $\mathbf{x_f}$ is the most likely position obtained by the SN when it performs localization. CluRoL addresses all these possible attack scenarios and helps the SNs perform robust localization.

We note here that distance enlargement/reduction attacks may also be caused by denial of service (DoS) attacks. These attacks may be prevented by using error correcting codes or spread spectrum techniques [17]. We do not consider DoS attack in our threat model.

IV. Description of the Scheme

Let u be an SN, and $\mathcal{A}_u$ be the set of anchors from whom u receives location references. The *bound circle* of an anchor $a_i \in \mathcal{A}_u$ with respect to SN u is the circle with $\mathbf{a_i}$ as the center, and $\hat{d}_{iu}$ (the estimate of the distance between u and a_i) as the radius. For each u, CluRoL performs clustering of the intersection points of the bound circles of the anchors in $\mathcal{A}_u$ and identifies the malicious anchors by performing some additional operations as detailed later. We motivate the use of clustering using Fig. 2.

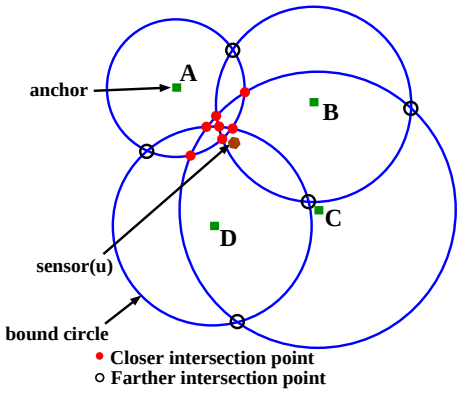


Fig. 2. Motivation for using clustering

The figure shows SN u and all the anchors in $\mathcal{A}_u = \{A, B, C, D\}$. The corresponding bound circles are illustrated by the solid (blue) circles. We assume that all the anchors are true, and that the distance estimates of the anchors are error-prone. Anchor A has negative measurement error (its distance estimate is less than its correct distance from u), while the rest of the anchors have positive measurement errors (their distance estimates are greater than their correct distances from u). If there were no measurement errors, all four bound circles would intersect at node $\mathbf{u}$. However, due to the presence of measurement errors, not all of the bound circles may pass through $\mathbf{u}$. The bound circles of any two true anchors can intersect at at most two points. Let the set of all such

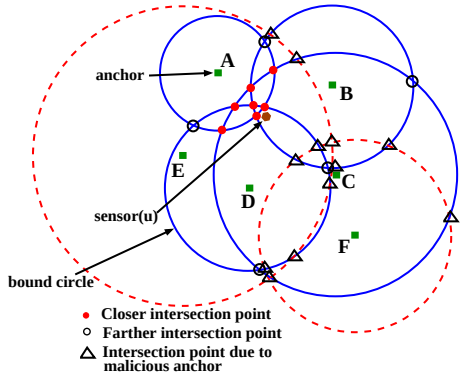


Fig. 3. Clustering in presence of malicious anchors

intersection points be $\mathcal{S}$. Due to the geometry of the bound circles, at least one of the intersection points shall be close to the position of the SN, occasionally both the intersection points may be equidistant from **u**. As a result, the density of the intersection points is highest close to **u**, as indicated by the solid red points in Fig. 2. We refer to these points close to **u** as the *proximal points* and denote them by the set $\mathcal{S}_\mathcal{P}$. We note that $\mathcal{S}_\mathcal{P} \subset \mathcal{S}$. Even when some of the anchors are malicious, intersections of the true bound circles still results in the creation of the proximal points, with the intersection of every pair of true bound circles contributing at least one point to $\mathcal{S}_\mathcal{P}$. This can be inferred from Fig. 3. In Fig. 3, u receives location references from six anchors: A, B, C, D, E, and F, where A, B, C, and D are true anchors (solid blue bound circles) and E and F are malicious anchors (dashed red bound circles) that lie about their distance estimates. E lies by increasing its distance estimate while F lies by decreasing its distance estimate. The solid red circles depict the proximal points, the hollow black circles represent the intersection points of the true bound circles that are farther from **u**, and the black triangles represent the intersection points resulting from the intersection of the bound circles of the malicious anchors. We can see that the presence of the false anchors generates more intersection points. However, due to the enlargement/reduction of their distance estimates, the resultant additional points of intersection are farther from **u** than the proximal points. Thus, using $\mathcal{S}_\mathcal{P}$ we can effectively differentiate between the true and malicious anchors. Given $\mathcal{S}_\mathcal{P}$, the true anchors can be identified as the anchors on whose bound circles at least one of the points $x \in \mathcal{S}_\mathcal{P}$ exists. The anchors that are not identified are malicious.

We note that due to measurement errors, **u** is not inferable from the intersection points of the bound circles, but the points in $\mathcal{S}_\mathcal{P}$ can be used as representative of **u**, to identify the true anchors. However, note that since the correct position of u is not known, identifying $\mathcal{S}_\mathcal{P}$ is also difficult. The next subsection details a mechanism that uses clustering to approximate $\mathcal{S}_\mathcal{P}$ by a cluster of the intersection points, C_{max}.

A. Clustering the intersection points

If SN u receives location references from N anchors, then the number of pairwise intersections between the corresponding bound circles is C_N^2. For each pair of bound circles, BC_i and BC_j, there exist at most two intersection points, hence $|\mathcal{S}| \leq 2 \cdot C_N^2$. At least one of the two intersection points is a proximal point, so $|\mathcal{S}_\mathcal{P}| \geq C_N^2$. However, if some of the anchors are malicious, the above inequality may no longer be true. For an SN u, if $|\mathcal{A}_u| = N$ and the measurements are accurate, Misra *et al.* [12], identified the upper bound on the number of malicious anchors that can be involved in the localization of u while still not being able to subvert the localization process. This upper bound holds even when the malicious anchors are colluding. According to the upper bound, the number of malicious anchors is, $M \leq \lfloor N/2 \rfloor - 2$. When the measurements are error-prone, the bound becomes more strict, as accurate localization is even more difficult to achieve. In this paper, we use the above upper bound on the number of malicious anchors. If M satisfies the upper bound, then the number of true anchors, T $\geq \lceil N/2 \rceil + 2$. Each malicious anchor reduces/dilates its distance estimates. The distance of the intersection points, resulting from the intersection of its bound circle with other bound circles, from **u** depends on how much the malicious anchor lies. The bigger the lie the greater is the distance, hence the intersection points are less likely to form a part of $\mathcal{S}_\mathcal{P}$. Note that $|\mathcal{S}| = 2C_N^2$,while $|\mathcal{S}_\mathcal{P}| \geq C^2_{\lceil N/2 \rceil + 2}$. In what follows, we describe a clustering procedure which generates a cluster C_{max} of $\mathcal{S}$, as an approximation to $\mathcal{S}_\mathcal{P}$.

Let $\mathcal{D} = \{(x, y, d(x, y))|\ x, y \in \mathcal{S}\}$, where $d(x, y) = \|x - y\|$. Also, let $\mathcal{D}' = \{(u, v, d(u, v))|\ u, v \in \mathcal{S}_\mathcal{P}\}$, the set containing the tuples for every pair of points in $\mathcal{S}_\mathcal{P}$. Let $\alpha = C^2_{\lceil N/2 \rceil + 2}$ and $\beta = 2 \cdot C_N^2$. Then the ratio between $|\mathcal{D}'|$ and $|\mathcal{D}|$ satisfies $|\mathcal{D}'|/|\mathcal{D}| \geq C_\alpha^2/C_\beta^2$. Although we know that $T \geq \lceil N/2 \rceil + 2$, the true value of T is unknown. So we approximate the ratio $|\mathcal{D}'|/|\mathcal{D}|$ using C_α^2/C_β^2. Let

$$\eta = \lceil C_\alpha^2/C_\beta^2 \rceil \cdot 100. \tag{3}$$

Then given two points $x, y \in \mathcal{S}$ they can belong to the same cluster if $d(x, y)$ satisfies the condition, $d(x, y) \leq d_{th}$. d_{th} is the *distance threshold*, which is the value of the distance element of the η^{th} percentile tuple of $\mathcal{D}$, when the tuples of $\mathcal{D}$ are sorted in ascending order of the third element of the tuple (pairwise distance). The intuition for using the η^{th} percentile tuple's distance value is as follows. The points in $\mathcal{S}_\mathcal{P}$ account for at least η of all the pairwise distances. Since they are packed close to the position of the SN with high density, as seen in Figs. 2 and 3, most of the tuples in the first η^{th} percentile of the sorted set $\mathcal{D}$ would be $(x, y, d(x, y))$, where $x, y \in \mathcal{S}_\mathcal{P}$. Algorithm 1, *findMaxCluster*, utilizes the above property to cluster the points in $\mathcal{S}$ into clusters, $C_i, i = 1, \ldots, k$, and return C_{max}, the cluster with the maximum cardinality.The algorithm takes the sorted set $\mathcal{D} = \{D_1, \ldots, D_{|\mathcal{D}|}\}$ and the value of d_{th} as inputs. Lines 1 and 2 of Algorithm 1 initialize the first cluster C_1. The two points x, y belonging to the first tuple $D_1 \in \mathcal{D}$ are added to C_1. Lines 3 to 27 perform the clustering operation for each subsequent tuple $(x, y, d(x, y)) \in \mathcal{D}$. For the pair of points $\{(x, y)|(x, y, d(x, y)) \in \mathcal{D}\}$ such that $d(x, y) \leq d_{th}$, there are four possible conditions. Lines 5 to 7 handle the case where x, y do not belong to any cluster. Then x and y are added to a new cluster. Lines 8 to 12 handle the case where x belongs to a cluster C_p while y does not belong to any

Algorithm 1 findMaxCluster

Input: $\mathcal{D} = \{D_1, D_2, \ldots, D_{|\mathcal{D}|}\}$, $d_{th} > 0$.
Output: $\mathbf{C}_{max}$.

1: k = 1; {Counter for the number of clusters used}
2: $C_1 = \{x, y\}$; $\{(x, y, d(x, y)) \in D_1\}$
3: **for** $i = 2$ to $|\mathcal{D}|$ **do**
4: $x, y \in D_i$;
5: **if** $(x \notin C_p) \wedge (y \notin C_p)$, for $p = 1, \ldots, k$ **then**
6: k++; {x, y not in any cluster, add to a new cluster}
7: $C_k = \{x, y\}$;
8: **else if** $(x \in C_p) \wedge (y \notin C_q)$, for $p, q = 1, \ldots, k$ **then**
9: {$x \in C_p$ and y does not belong to any cluster}
10: **if** $d(x, y) \leq d_{th}$ **then**
11: $C_p = C_p \cup \{y\}$;
12: **end if**
13: **else if** $(x \notin C_p) \wedge (y \in C_q)$, for $p, q = 1, \ldots, k$ **then**
14: {x does not belong to any cluster and $y \in C_q$}
15: **if** $d(x, y) \leq d_{th}$ **then**
16: $C_q = C_q \cup \{x\}$;
17: **end if**
18: **else if** $\{(x \in C_p) \wedge (y \in C_q)\} \wedge \{C_p \neq C_q\}$, $p, q = 1, \ldots, k$ **then**
19: {Need to check if C_p and C_q can be merged}
20: M_{C_p} = $\{\sum_{j=1}^{|C_p|} p\}/|C_p|$; {Centroid of C_p}
21: M_{C_q} = $\{\sum_{j=1}^{|C_q|} q\}/|C_q|$; {Centroid of C_q}
22: **if** $\|M_{C_p} - M_{C_q}\| \leq d_{th}$ **then**
23: *Merge* C_p and C_q;
24: k = k – 1; {One less cluster}
25: **end if**
26: **end if**
27: **end for**
28: $C_{max} = \{C_i|\ |C_i| = \max\{|C_i|, i = 1, \ldots |\mathcal{C}|\}\}$;
29: **return** C_{max};

cluster. Then y is added to the cluster C_p. Lines 13 to 17 handle the exactly opposite case. Lines 18 to 26 handle the case where the x and y belong to two different clusters. In that case, the centroids of the two clusters, M_{C_p} and M_{C_q} are used to identify if the two clusters can be combined. Clusters C_p and C_q can be combined if $d(M_{C_p}, M_{C_q}) \leq d_{th}$. The centroid technique prevents the undesirable case where two clusters C_t, containing only points from $\mathcal{S}_\mathcal{P}$ and C_m, containing no points from $\mathcal{S}_\mathcal{P}$ are merged because there exist two points $x \in C_t$ and $y \in C_m$ such that $d(x, y) \leq d_{th}$. This is undesirable because C_m may contain points of intersection of bound circles of malicious anchors. Hence those points will become a part of C_{max}, resulting in large number of false negatives, that is, malicious anchors not being caught. The centroid technique guarantees that C_p and C_q are merged only when there exist many pairs $\{(x, y)|\ x \in C_p, y \in C_q\}$, such that $d(x, y) \leq d_{th}$. After the clustering procedure, the algorithm returns the cluster with the maximum cardinality C_{max}. The running time of Algorithm 1 is $\mathcal{O}(N^4)$.

We note that C_{max} shall contain a large number of proximal points, as the pairwise distance of most of these points are below the threshold. As a result, C_{max} can be used to approximate $\mathcal{S}_\mathcal{P}$. However, it is likely that $\mathcal{S}_\mathcal{P} \setminus C_{max} \neq \emptyset$. We prove in Theorem 1 that this shall not result in significant false positives, that is, true anchors identified as malicious. The theorem computes the probability that at least one proximal point resulting from a true anchor exists in C_{max} and hence the probability that the anchor is identified as true.

Theorem 1: Let $\gamma = \lceil N/2 \rceil + 1$ and $k' = |C_{max}|$. Let $k \leq k'$ be the number of proximal points in C_{max}, where $k \leq |\mathcal{S}_\mathcal{P}|$. Then the probability that at least one proximal point of a true anchor belongs to C_{max} is at least $1 - \left(\frac{\gamma-1}{\gamma+1}\right)^k$.

Proof: The bound circle of each true anchor a_j intersects with at most $\lceil N/2 \rceil + 1$ of the remaining true bound circles, resulting in at most $\lceil N/2 \rceil + 1$ proximal points. Given that the number of proximal points in C_{max} is k, the probability $\mathcal{P}$ that none of the k points are points of intersection of a_i with another true anchor is given by, $\mathcal{P} = \frac{\gamma(\gamma-1)}{\gamma(\gamma+1)} \cdot \frac{\gamma(\gamma-1)-1}{\gamma(\gamma+1)-1} \cdots \frac{\gamma(\gamma-1)-(k-1)}{\gamma(\gamma+1)-(k-1)} = \frac{(\gamma-1)^k}{(\gamma+1)^k} \cdot \left(\frac{\gamma-\frac{1}{\gamma-1}}{\gamma-\frac{1}{\gamma+1}} \cdots \frac{\gamma-\frac{k-1}{\gamma-1}}{\gamma-\frac{k-1}{\gamma+1}}\right)$. For all possible values of k, the expression inside the pair of braces is no greater than 1. Therefore $\mathcal{P} \leq \frac{(\gamma-1)^k}{(\gamma+1)^k}$. Thus the probability of atleast one intersection point of the true anchor a_j being present in C_{max} is $1 - \mathcal{P} \geq 1 - \frac{(\gamma-1)^k}{(\gamma+1)^k}$. ■

The theorem indicates that if ***findMaxCluster*** is used, the probability of false positive is low. For illustration, let $N = 8$, then $\gamma = 5$ and the number of proximal points is 15. Even if k is as small as 5, the probability that a true anchor is identified as true, is 0.87, which is a fairly high probability. This indicates that using ***findMaxCluster*** will result in low false positives.

It is also possible that C_{max} contains more than one $x \in \mathcal{S} \setminus \mathcal{S}_\mathcal{P}$. In that sense our ***findMaxCluster*** is pessimistic, as C_{max} could potentially contain some points of intersection of malicious bound circles thus resulting in false negatives. However, we note that our emphasis is on the reduction of false positives at the expense of some false negatives.

We would like to note that even when the malicious anchors are colluding, the set C_{max} would comprise of a large number of proximal points. When the malicious anchors collude this results in the creation of another set of closely placed intersection points near the false position of u. However, the cardinality of the cluster resulting from these points cannot be larger than the cardinality of the cluster containing the proximal points, as $M < T$.

B. Identification of malicious anchors using CluRoL

Now we are ready to present CluRoL, which uses the clustering algorithm ***findMaxCluster*** to identify the malicious anchors, removes them from the localization process, and localizes the SN u with high accuracy.

We describe CluRoL as Algorithm 2. Line 2 performs initialization of the variables, $\mathcal{S}$, $\mathcal{D}$, and C_{max}. ***ancStatus*** is an array that contains the status of the anchors obtained from CluRoL. If ancStatus[i] = false, then a_i is malicious. Lines 4 to 8 populate $\mathcal{S}$ with the pairwise intersection points and have running time $\mathcal{O}(N^2)$. Lines 9 to 12 populate the set $\mathcal{D}$ and have running time $\mathcal{O}(N^4)$. Line 13 performs the sorting of the pairwise distances, with running time $\mathcal{O}(N^4 \log N)$. Line 14 obtains the value of d_{th} by the procedure described

Algorithm 2 Algorithm for CluRoL at each SN u

Input: Position $\mathbf{a_i}$ of the anchors in range of u and their distance estimates $\hat{d}_{iu}, i = 1, \ldots, N$, $|T|$, the lower bound on number of true anchors, δ_{max}, and ϵ_{max}.

Output: $\hat{\mathbf{u}}$ and the malicious anchors;

1: $BC_i = \{$Bound circle of anchor $a_i\}$,
2: $\mathcal{S} = \emptyset, \mathcal{D} = \emptyset, \mathcal{C}_{max} = \emptyset$;
3: ancStatus[N] = false; {An anchor is true or malicious}
4: **for** $i = 1$ to $N - 1$ **do**
5: **for** $j = i + 1$ to N **do**
6: $\mathcal{S} = \mathcal{S} \cup \{x |\ x \in (BC_i \cap BC_j)\}$; {$x$ is an intersection point of BC_i and BC_j}
7: **end for**
8: **end for**
9: **for all** $x, y \in \mathcal{S}$ **do**
10: $d(x, y) = \|x - y\|$; {Pairwise distance tuple}
11: $\mathcal{D} = \mathcal{D} \cup (x, y, d(x, y))$;
12: **end for**
13: Sort $\mathcal{D}$ in ascending order of the pairwise distances;
14: $d_{th} = \eta^{th}$ percentile tuple's pairwise distance value;
15: $\mathcal{C}_{max}$ = *findMaxCluster*($\mathcal{D}, d_{th}$);
16: **for** each $x \in \mathcal{C}_{max}$ **do**
17: **for** $i = 1$ to N **do**
18: UB = $\hat{d}_{iu}(1 + \delta_{max})^2$;
19: LB = $\hat{d}_{iu}/(1 + \delta_{max})^2$;
20: **if** $\{(\|\mathbf{x} - \mathbf{a_i}\| \leq \text{UB}) \wedge (\|\mathbf{x} - \mathbf{a_i}\| \geq \text{LB})\}$ **then**
21: ancStatus[i] = true;
22: **end if**
23: **end for**
24: **end for**
25: $\mathcal{A} = \{(\mathbf{a_i}, \hat{d}_{iu}) |$ ancStatus[i] = true$\}$, $i = 1, \ldots, N$.
26: $\hat{\mathbf{u}}$ = MSE($\mathcal{A}$);
27: **return** $\hat{\mathbf{u}}$, ancStatus;

in subsection IV-A. Line 15 calls Algorithm 1 to obtain C_{max}. Lines 16 to 24 illustrate the use of the maximum measurement error proportion to distinguish true anchors from the malicious anchors. As mentioned earlier, it is difficult to obtain $C_{max} = \mathcal{S}_{\mathcal{P}}$, since $\mathbf{u}$ is not known. Thus $\mathcal{S}_{\mathcal{P}} \setminus C_{max}$ may not be $\emptyset$. Hence there exists a small probability that *findMaxCluster* does not identify any proximal point of a true bound circle, thus resulting in the anchor being identified as malicious. Consider the scenario where there are M malicious anchors and T true anchors ($N = T + M$). Let $T - 1$ of the true anchors ($t_1, \ldots, t_{T-1}$) be subject to very little measurement error. Hence, the proximal points resulting from the intersection of the bound circles of these anchors would be located closely about $\mathbf{u}$. Further, let t_T be subject to the maximum measurement error (δ_{max}). Without loss of generality, let two malicious anchors, m_1 and m_2, lie by increasing their distance estimates by a very small proportion, that is, $(1+\delta_{m_i u})(1+\epsilon_{m_i u}) << (1+\delta_{max}), i = 1, 2$. Then the intersection points of BC_{m_1} and BC_{m_2} with the other bound circles is going to be close to $\mathbf{u}$, in fact closer than the points of intersection of B_{t_T}. As a result, C_{max} shall contain no proximal points obtained from B_{t_T}, but may contain some of the points of intersection of B_{m_1} and B_{m_2}. Thus, t_T shall be incorrectly identified as malicious, resulting in a false positive. At the same time, m_1 and m_2 might be identified as true. Lines 16 to 24 attempt to reduce such false positives even further. Since there is no way of identifying if a measurement is subject to positive/negative measurement error, we use the estimate of each anchor to define two conservative estimate bounds, the Upper Bound (UB) and the Lower Bound (LB) as shown in the algorithm. If an anchor is true, the position of the SN u is inside the circle drawn with UB as the radius, while it is outside the circle with LB as the radius. CluRoL uses the points of C_{max} to approximate $\mathcal{S}_{\mathcal{P}}$, whose points are representative of $\mathbf{u}$. If at least one of the points in C_{max} is such that it falls within the UB and LB circles of t_T, then t_T is identified as true, even if none of its intersection points belong to C_{max}. This technique also results in malicious anchors that lie very little, such as m_1 and m_2, to be identified as true. However, this is unavoidable, as any malicious anchor a_i that lies such that $(1 + \delta_{iu})(1 + \epsilon_{iu}) << (1 + \delta_{max})$ cannot be distinguished from a true anchor subjected to measurement errors, thus cannot be identified by any mechanism.

If an anchor a_i is not identified by CluRoL. This implies that there exists no $x \in C_{max}$ such that x lies within the UB and LB circles of a_i. This is possible if a_i is lying, thus a_i is identified by CluRoL as malicious with high likelihood. In line 2, the position of the SN is estimated using the MSE method. Only the estimates from the anchors identified as true are used in the procedure. The running time of Algorithm 2 is $\mathcal{O}(N^4 \log N)$. We note that in general the number of anchors from whom a SN receives location references is fairly small, hence the running time of Algorithm 2 is reasonable.

The simulation results demonstrate the effectiveness of CluRoL in identifying malicious anchors increasing the accuracy of localization.

V. Simulation Results

The WSN is deployed in a 100×100 sq. m. field. The transmission range of the anchors is set to 30m, $|\delta_{max}| = 0.05$, and $|\epsilon_{max}| = 1.0$. The location references are broadcast every second. The number of anchors in range of the SN is between 4 and 12. For each value of N, the number of malicious anchors M varies between, $\{0, \ldots, \lfloor N/2 \rfloor - 2\}$. For the case where the malicious anchors are colluding, the anchors collude to localize the SN u 30m away from its original position $\mathbf{u}$. When the malicious anchors do not collude, each anchor lies about its distance estimate independently. Fig. 4 shows the percentage of malicious anchors identified by CluRoL and the percentage of false positives resulting from CluRoL.

For the results in Fig. 4, for each configuration of number of anchors and number of malicious anchors, we perform 50 simulation runs. Our scheme displays very little false positives. The percentage of false positives is less than 2% for both when the malicious anchors collude and when they do not. CluRoL results in a significant percentage of malicious anchors being caught. When the malicious anchors are not colluding we catch anywhere between 65% to 82% of them. When they collude we catch more than 85% of them. The increase in percentage

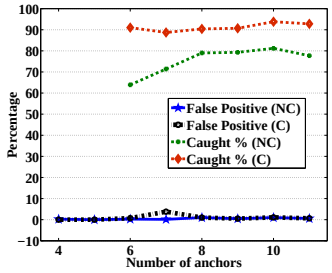
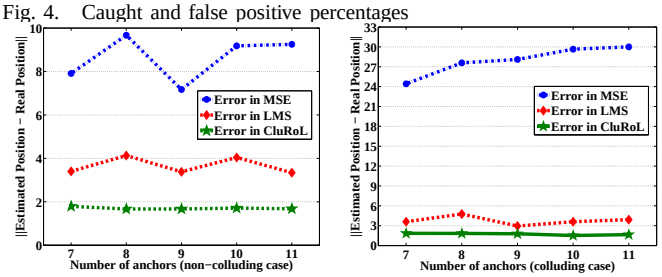


Fig. 4. Caught and false positive percentages



(a) Malicious anchors not colluding (b) Malicious anchors colluding

Fig. 5. $\sqrt{MSE}$ when no scheme is used, CluRoL, or LMS are used.

of malicious anchors caught when they collude is because the intersection points of their bound circles are farther from the position of the SN due to their collusion, thus they are not identified as true in lines 2 to 2 of Algorithm 2. This demonstrates the effectiveness of our scheme. In the figure, the curves depicting the malicious anchors caught start at $N = 6$, as for $N < 6$, $M \leq 0$. If a malicious anchor lies such that the false estimate is within the allowed maximum error, it cannot be caught. We do not consider such anchors as malicious in our simulations study.

In Fig. 5, we show the result of the comparisons of CluRoL with the Least Median Square (LMS) scheme [10], and the standard MSE method using all the anchors. For each configuration of number of anchors and number of malicious anchors, we perform 100 simulation runs. The figures show the average error in localization ($\|\hat{\mathbf{u}} - \mathbf{u}\|$) over all possible values of M for a given value of N. Fig. 5(a) shows $\|\hat{\mathbf{u}} - \mathbf{u}\|$ given that the malicious anchors are non-colluding. As already illustrated in Section I, use of MSE without filtering the malicious estimates results in significant error in localization. With the LMS technique the error in localization is reduced significantly, however, with CluRoL the performance is even better. This is because in the LMS scheme, an estimate that deviates from a calculated threshold by a chosen constant factor, is identified as an outlier and not involved in the localization process and the corresponding anchor is identified as malicious. However, inaccuracies in the choice of the factor can increase the false positives or false negatives. The accuracy of LMS depends on the number of subsets of the distance estimates used to identify the parameters. Larger the number of subsets more accurate is the scheme. To get the best possible result requires exponential time in N, as all subsets of the set of estimates need to be enumerated. In CluRoL on the other hand, the use of the points in C_{max} to approximate $\mathcal{S}_{\mathcal{P}}$ and hence to represent the position of the SN and the use of UB and LB help reduce the false positives. More importantly CluRoL has significantly less time complexity in comparison to LMS.

When the malicious anchors collude, the error in localization increases significantly in the unfiltered case. The LMS and the CluRoL schemes fare better, with CluRoL still being better than LMS because of the same reasons as before. This is illustrated in Fig. 5(b). Thus, CluRoL identifies a large proportion of the lying anchors and also improves localization accuracy.

VI. Conclusions and Future Work

In this paper, we have proposed a clustering based scheme that successfully identifies a large number of malicious anchors that may subvert the localization process in a WSN. We have demonstrated the increase in accuracy of subsequent localization without the identified malicious anchors. In the future, we would like to implement this technique on a testbed for further evaluation and perform rigorous quantitative analyses.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks*, 38(4):393–422, 2002.

[2] Pratik Biswas, Tzu-Chen Lian, Ta-Chung Wang, and Yinyu Ye. Semidefinite programming based algorithms for sensor network localization. *ACM Transaction on Sensor Networks*, 2(2):188–220, 2006.

[3] Xiuzhen Cheng, Andrew Thaeler, Guoliang Xue, and Dechang Chen. TPS: A time-based positioning scheme for outdoor wireless sensor networks. In *Proceeding of Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, volume 4, pages 2685–2696, 2004.

[4] L. Doherty, K. Pister, and L Ghaoui. Convex position estimation in wireless sensor networks. In *Proceedings of the IEEE INFOCOM*, pages 22–26, 2001.

[5] W. Du, L. Fang, and P. Ning. LAD: Localization anomaly detection for wireless sensor networks. In *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2005.

[6] Tian He, Chengdu Huang, Brian M. Blum, John A. Stankovic, and Tarek F. Abdelzaher. Range-free localization and its impact on large scale sensor networks. *Trans. on Embedded Computing Sys.*, 4(4):877–906, 2005.

[7] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. *Elsevier's Ad Hoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*, 1(2–3):293–315, September 2003.

[8] L. Lazos and R. Poovendran. HiRLoc: High-resolution robust localization for wireless sensor networks. *IEEE Journal on Selected Areas of Communications*, 24(2):233–246, February 2006.

[9] L. Lazos, R. Poovendran, and S. Čapkun. ROPE: Robust position estimation in wireless sensor networks. In *Proceedings of Information Processing in Sensor Networks (IPSN)*, pages 324–331, 2005.

[10] Z. Li, W. Trappe, Y. Zhang, and B. Nath. Robust statistical methods for securing wireless localization in sensor networks. In *Proceedings of Information Processing in Sensor Networks (IPSN)*, pages 91–98, 2005.

[11] D. Liu, P. Ning, and W. Du. Detecting malicious beacon nodes for secure location discovery in wireless sensor networks. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 609–619, 2005.

[12] S. Misra, S. Bhardwaj, and G. Xue. ROSETTA: Robust and secure mobile target tracking in a wireless ad hoc environment. In *Proceeding of the Military Communications Conference (MILCOM)*, pages 1–7, 2006.

[13] D. Niculescu and B. Nath. Error characteristics of ad hoc positioning systems (APS). In *Proceeding of ACM MobiHoc*, 2004.

[14] A. Perrig, R. Canetti, D. Tygar, and D. Song. The tesla broadcast authentication protocol. *Cryptobytes*, 5(2):2–13, 2002.

[15] A. Savvides, C. Hans, and M. Srivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceeding of ACM MobiCom*, pages 166–179, 2001.

[16] S. Čapkun and J. Hubaux. Secure positioning in wireless networks. *IEEE Journal on Selected Areas of Communications*, 24(2):221–232, February 2006.

[17] A. Wood and J. Stankovic. Denial of service in sensor networks. *IEEE Computer*, 35(10):54–62, oct 2002.