Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2010, Article ID 605658, 7 pages
doi:10.1155/2010/605658

Research Article

Fully Decentralized and Collaborative Multilateration Primitives for Uniquely Localizing WSNs

Arda Cakiroglu[1] and Cesim Erten[2]

[1] *Computer Science and Engineering, Işık University, Sile 34980, Turkey*
[2] *Computer Engineering, Kadir Has University, Istanbul 34083, Turkey*

Correspondence should be addressed to Cesim Erten, cesim@khas.edu.tr

Received 31 August 2009; Accepted 21 September 2009

Academic Editor: Benyuan Liu

We provide primitives for uniquely localizing WSN nodes. The goal is to maximize the number of uniquely localized nodes assuming a fully decentralized model of computation. Each node constructs a cluster of its own and applies unique localization primitives on it. These primitives are based on constructing a special order for multilaterating the nodes within the cluster. The proposed primitives are fully collaborative and thus the number of iterations required to compute the localization is fewer than that of the conventional iterative multilateration approaches. This further limits the messaging requirements. With relatively small clusters and iteration counts, we can localize almost all the uniquely localizable nodes.

1. Introduction

Many applications and systems from areas such as environment and habitat monitoring, weather forecast, and health applications require use of many sensor nodes organized as a network collectively gathering useful data; see [1] for a survey. In such applications it is usually necessary to know the actual locations of the sensors. Sensor network localization is the problem of assigning geographic coordinates to each sensor node in a given network. Although Global Positioning System (GPS) can determine the geographic coordinates of an object, a GPS device has to have at least four lines of sight communication lines between different satellites in order to locate itself. Thus, in cluttered space or indoor environments GPS may be ineffective. Additional disadvantages including the power consumption, cost, and size limitations allow only a small portion of nodes in a large scale sensor network having GPS capabilities. It is important to design methods that achieve localization with limited use of such systems. A common paradigm introduced to overcome this difficulty is the *iterative multilateration* where only a small portion of the nodes are assumed to be *anchor* nodes with a priori location information. Every node uses its ranging sensors to measure distances to the neighbors and shares the measurement and location information if available. Multilateration techniques are then employed in an iterative manner to collectively estimate node locations from such information [2]. Assuming no measurement errors, "unique" localization of a sensor node is achieved via trilateration from three anchor positions if distance to those anchors is known. Although this is a sufficient condition for uniqueness, it is rarely necessary in wireless sensor networks settings. A finite number of possible locations for a set of neighbor nodes for which range information is available may also provide unique location for a sensor node. In this paper we propose primitives for unique localization of nodes in a sensor network. Assuming measurements with no noise, the goal is to maximize the number of uniquely localized nodes employing iterative multilateration. The proposed primitives are fully decentralized, fully collaborative. The suggested collaboration model gives rise to a high rate of unique localization. Moreover, this is achieved with reasonably low energy requirements for message exchanges as the average number of iterations per node is low.

2. Related Work

We can classify the previous work based on where the computation takes place and the type of localization solution produced. Centralized methods assume the availability of global information about the network at a central computer where the computation takes place [3–5]. Whereas in decentralized methods each node usually processes some local information gathered via a limited number of message exchanges [6, 7]. A second distinction between different methods is the localization output produced. In *unique localization* a single position is assigned to each node in the network [8]. Even though the topology of the underlying graph may not give rise to a unique localization, it may still be possible to assign a finite set of positions for each node, called *finite localization* [4]. There has been recent interest and theoretical results on uniquely localizing networks. However, the suggested methods usually rely on centralized models [4]. To overcome this difficulty, cluster-based approaches or collaborative multilateration have been suggested. Moore et al. propose the idea of localizing clusters where each cluster is constructed using two-hop ranging information [7]. Only trilateration is employed as the localization primitive which results in fewer nodes being uniquely localized as compared to the primitives we propose. Savvides et al. propose n-hop collaborative multilateration. The nodes organize into *collaborative subtrees*, then using simple geometric relationships each node constructs a position estimate and finally the estimate is refined using a least-squares method. The collaboration is on sharing distance and position information within n-hops but each node is still responsible for locating itself. With our model of full collaboration, not only does a node receive information to compute its position but it may also receive information regarding its own location from neighboring nodes as well.

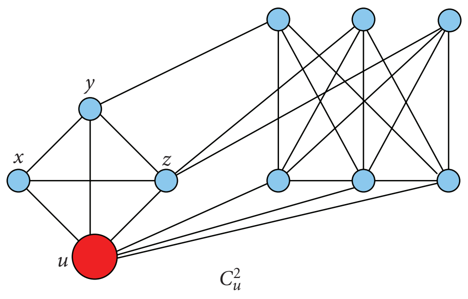
3. Unique Localization Primitives

Our unique network localization method relies on two low level primitives: Reliable internode distance measurement and internode communication mechanisms. Several techniques such as TDoA, RSSI, ToA may be used to obtain distance between two sensor nodes [9, 10]. The gathered distance information is assumed to be error-free. We also assume that the communication between sensor nodes is through broadcasting and that the broadcasted data is received by all neighboring nodes within the sensing range of the broadcaster.

The main unit of localization is a *cluster*. Each node u in the sensor network $\mathbb{N}$ goes through an initial setup phase constructing C_u^r, the r-radius cluster centered at u. Let N_u denote the immediate neighborhood of node u. We assume that u has already gathered $d(u, v_i)$, distance to $v_i \in N_u$. With this information every u creates its own cluster C_u^1, where u is at the center; there is an edge (u, v_i) with weight $d(u, v_i)$ for every $v_i \in N_u$. At the second round of information exchange, every node shares its cluster with all the nodes in its neighborhood. As a result each node u can construct C_u^2, and so on. After the final iteration of broadcasting/gathering of packets, the cluster C_u^r is constructed with the collected data. With this construction, C_u^1 is a *star* graph centered around u, and C_u^2 is a *wheel* graph centered at u, together with the edges connecting to the wheel. We note that with this model a pair of clusters may share many nodes. Although this may seem like a waste, we show that these overlaps give rise to efficient collaboration in terms of unique localization. The collaboration is modelled via the iterative localization. At each iteration, each node constructs a new set of uniquely localized nodes in its cluster C_u^r using our proposed primitives, broadcasts this set to the neighbors, gathers analogous information, and iterates the same process.

3.1. Bilaterations and Unique Localization. The network localization problem can be converted into that of *graph realization problem*. Let $\mathbb{G}(V, E)$ be the graph corresponding to a physical sensor network $\mathbb{N}$. Each vertex $v_i \in V = \{v_1, \ldots, v_m, v_{m+1}, \ldots, v_n\}$ corresponds to a specific physical sensor node i in $\mathbb{N}$. Vertices $v_1, \ldots, v_m$ are the nodes with known positions, called *anchors*. There exists an edge $(v_i, v_j) \in E$ if nodes i and j are within sensing range or both $i, j \leq m$. Each edge (v_i, v_j), where $v_i, v_j \in V$ and $i \neq j$ is associated with a real number which represents the Euclidean distance between the two nodes i, j. Formally, the graph realization problem is assigning coordinates to the vertices so that the Euclidean distance between any two adjacent vertices is equal to the real number associated with that edge [11, 12]. The graph realization problem has intrinsic connections with the graph rigidity theory. If we think of a graph in terms of bars and joints, a *rigid* graph means "not deformable" or "not flexible" [12]. Formally, the rigidity of a graph can be characterized by Laman's condition: A graph with n vertices and $2n - 3$ edges is rigid if no subgraph with n' vertices contains more than $2n' - 3$ edges [13]. Obviously if the graph is not rigid, infinite number of realizations are possible through continuous deformations. However, even when the graph is rigid there may be ambiguities that give rise to more than one possible realization. In order to formalize these ambiguities, the term *globally rigid* is introduced [14]. A graph is globally rigid if and only if it is *3-connected* and *redundantly rigid*. Global rigidity is a necessary and sufficient condition for unique realizations. We note that the discussions of rigidity and global rigidity apply to "generic" frameworks, that is, those with algebraically independent vertex coordinates over the rationals [8, 14]. As almost all point sets are generic and the generic global rigidity can be described solely in terms of combinatorial properties of the graph itself, in what follows the term globally rigid assumes the genericity of the frameworks.

It is NP-Hard to find a realization of $\mathbb{G}(V, E)$ even if $\mathbb{G}$ is globally rigid [15, 16]. However, there exists an exceptional graph class called *trilateration graphs* that is uniquely localizable in polynomial time. A graph is a trilateration graph if it has a trilateration ordering $\pi = \{u_1, u_2, \ldots, u_n\}$, where u_1, u_2, u_3 form a K_3 and each u_i has at least three neighbors that come before u_i in π. Although easily localizable, trilateration is a strong requirement for



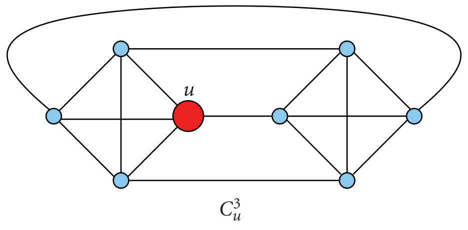


Figure 1: (a) Order should be picked carefully for C_u^2. (b) No order exists for C_u^3.

```
(1) procedure UniqueLocalization($A_u$)
(2)     Let $L$ indicate the set of finitely localized nodes
(3)     for all $a \in A_u$ do
(4)         Update($L$, $a$)
(5)     $\pi \leftarrow$ FindBilaterationOrder
(6)     for all $v \in \pi$ do
(7)         Multilaterate($v$, $N_\pi(v)$),
(8)         where $N_\pi(v)$ are neighbors of $v$ that are to
            the left of it in $\pi$
(9)     Broadcast new anchors in $C_u^r$ to $N_u$
```

Algorithm 1: Iterated at u each time new anchors in A_u are received from N_u.

unique localization. C_u^2, for instance, contains a wheel graph centered at u. A wheel graph is uniquely localizable although it is not a trilateration graph. If the graph is not 3-connected, there exists a pair of vertices whose removal disconnects the graph. The graph can be flipped through the line passing through the disconnecting pair. Although flipping introduces nonunique realizations, the fact that finite possibilities arise as a result is the main motivation behind *bilateration graphs*. A graph is a bilateration graph if it has a bilateration ordering $\pi = \{u_1, u_2, \dots, u_n\}$, where u_1, u_2, u_3 form a K_3 and each u_i has at least two neighbors that come before u_i in π. For the localization application, fixing the initial K_3 allows one to neglect the global rotations and translations. Our main focus is on localizing bilateration graphs that arise within the defined cluster C_u^r. For $r = 1$ the constructed model is no different than the usual trilateration primitive. We provide useful remarks regarding bilateration graphs and the clusters, more specifically for small values of r, since for the sensor network settings those are of special interest.

Remark 3.1 (For $r = 2$). If C_u^r is globally rigid, then it is a bilateration graph. However, even if C_u^r is globally rigid, not every ordering is a bilateration ordering. For instance, C_u^2 in Figure 1 is globally rigid. Starting an ordering with any one of the four triangles formed between u, x, y, z bilateration does not contain C_u^2 completely. However, starting an ordering with any other triangle in the cluster provides a complete bilateration of C_u^2.

Remark 3.2 (For $r = 3$). There exist globally rigid clusters which are not bilateration graphs. C_u^3 in Figure 1 is globally rigid and therefore uniquely localizable, but it is not a bilateration graph.

3.2. *Unique Localization within Iterative Collaboration*

The goal is to finitely localize as many nodes as possible and share the resulting unique node positions with the neighbors. The main localization method is iterative. Each node u executes the localization method on its own cluster. If u creates the unique positions of some new nodes within C_u^r, then it shares this information with the neighbors, some of which may have overlapping clusters with C_u^r. Those neighbors may benefit this exchange if the shared nodes are part of a globally rigid component within their clusters. This procedure continues iteratively until no node creates any new unique positions and gathers any new information from the neighbors. The main localization procedure is shown in Algorithm 1 which is repeated at every iteration.

At the beginning of each iteration node, u gathers a recently discovered set of anchors, A_u, by listening to the broadcasts made by nodes in N_u. We note that we use *anchor* as a more general term in the sense that every node that is uniquely localized throughout the localization process is called an anchor. If a newly discovered anchor node a is not finitely localized, a and its position are appended to L, the set of finitely localized nodes and their positions. Otherwise all positions other than the real one are *removed*. Next, a bilateration order π of the nodes in C_u^r is found. As stated in Remark 3.1, not every ordering covers C_u^r completely. To find a bilateration order π in general, we select a *seed* set as the first level in the ordering. We continue a breadth-first traversal to construct new levels of nodes while making sure every node in a level has at least two neighbors in the preceding levels. As iterations of the localization procedure increase, the set of finitely localized nodes grows, therefore it constitutes a good candidate for the seed set. However, for the initial iterations we try every possible triple as a candidate for the seeds and take the maximum size set. Following the order in π, multilaterations are done to compute position possibilities for each node. The traversal is done in a breadth-first manner rather than a depth-first manner so as to decrease the number of position possibilities as early as possible during this process. The rest of the localization procedure where each node in π is multilaterated in order is the same as the centralized localization method of [4]. Going through π, finite positions are created for each v using two *consistent* positions p_b, p_c of immediate ancestors b, c.

This is via *bilateration*, computing the intersection points of two circles centered at p_b, p_c with appropriate radii. Each generated position has a localization history in the form of an *ancestors* list which stores the consistent positions of b, c, and the ancestors of those positions, check the consistency of a pair of points, then involves comparing their ancestries. If a node exists in the ancestries of both, but with different positions, then they are not consistent. Finally positions of the rest of the immediate ancestors of v in π are checked for consistency with the new positions of v. Throughout bilateration and update bilateration, every position that has been found inconsistent is *removed* immediately and thus further localizations do not take into account any unnecessary data. We note that all *removals* in the unique localization are recursive in the sense that the removal of a point p_b causes the removal of positions containing p_b in their ancestries as well.

3.3. Analysis. Let k denote the average degree of a node in $\mathbb{N}$. The size of a packet broadcasted by a node in the jth iteration of the initial setup is $O(k^j)$. The total size of all packets broadcasted throughout the network in this phase is $O(n \times k^r)$, where n is the number of nodes in the network. During the unique localization within iterations, each recently discovered anchor is broadcasted at most once. The number of nodes in a cluster C_u^r is bounded by $O(k^r)$. Therefore, the total size of all packets broadcasted throughout the network is the same as the first phase, $O(n \times k^r)$ which is the total size overall. Assuming k, r are constants, the total packet size is linear in terms of the size of the network. In terms of running time, a single execution of the localization takes $O(2^{k^r})$ time in the worst case. Although exponential on the size of a cluster, assuming the cluster sizes are constant, each iteration requires constant time. Same argument holds for the memory requirements of a sensor node. For practical considerations, the value of r is of crucial importance in determining the efficiency in terms of the messaging overhead, time and space requirements. For most of the experiments the maximum number of position possibilities for the whole cluster rarely exceeds 2^{10} for all practical values of r.

Assuming the iterative model of collaboration, the value of r is also important for determining the unique localizability as the next lemma shows.

Lemma 3.3 (For $\forall r \geq 2$). *Within the defined model of collaboration between clusters, there exists a class of graphs that have $O(1)$ uniquely localizable nodes for $r-1$, whereas $\Theta(n)$ uniquely localizable nodes for r.*

Proof. The *flower* graph of Figure 2 is such a class. The middle part called the *sepal* is a circulant graph of vertices $x_1, \ldots, x_c$. Within the sepal, each x_i has edges to x_{i-2}, x_{i-1}, x_{i+1}, x_{i+2}. Thus the sepal is the circulant graph of c vertices, $Ci_c(1,2)$, which is globally rigid. Corresponding to each x_i there is a *petal* P_{x_i} which is a wheel centered at x_i. P_{x_i} itself is globally rigid and shares exactly two vertices with the two neighboring petals $P_{x_{i-1}}$ and $P_{x_{i+1}}$. Vertices x_i, x_{i-1} are shared with the petal of x_{i-1}, and x_i, x_{i+1} are shared with that of

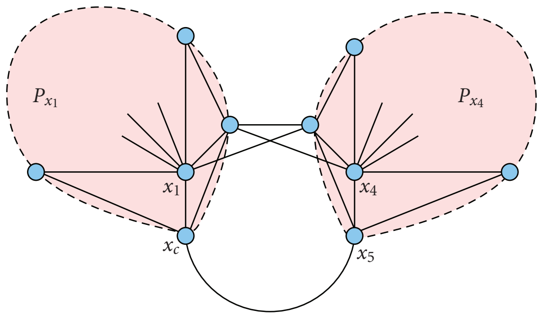


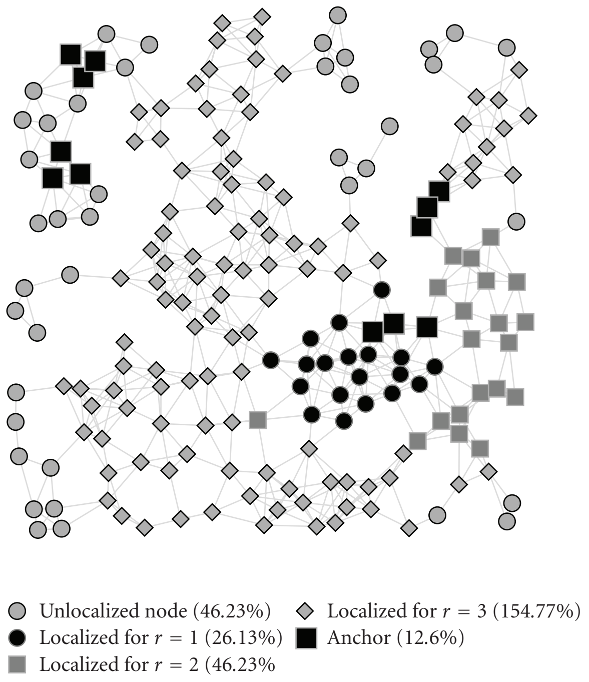
Figure 2: The flower graph not localizable for $r-1$, uniquely localizable for r.

x_{i+1}. We set $c = 4r - 2$. The cluster $C_{x_i}^r$ includes the sepal in its entirety and is globally rigid. If three anchors belong to the same petal, the center of the petal can collapse its petal, and therefore the sepal completely, which further enables the unique localization of every petal in the graph and the whole network is uniquely localized. However, $C_{x_i}^{r-1}$ is not globally rigid. Unless each petal contains at least three anchors, that is, it is independently localizable, unique localization is not possible. □

We can assign equal sizes to the petals, such that each petal consists of almost (n/c) nodes. An immediate consequence then is that, assuming a random assignment of anchors, the probability of localizing $n/(4r-2)$ nodes under the C_u^{r-1} cluster model is the same as that of localizing the complete network under the C_u^r cluster model. This is true since in C_u^{r-1} a petal is uniquely localizable either in its entirety or none at all. Moreover, a uniquely localizable petal in C_u^{r-1} gives rise to uniquely localizable network in C_u^r. It implies that, deploying the same number of anchors, with $r = 3$, we can localize the complete network, whereas only ten percent of the network is localizable for $r = 2$. We note that in practical deployment scenarios the likelihood of flower-like configurations is higher for small values of r. Therefore, the contrast between the uniquely localizable node ratios for $r = 2, 3$ is quite remarkable and is verified with the experiments of the following section.

4. Experiments and Discussion of Results

The implementation is coded in C++ using LEDA library [17]. The implementations are available at http://hacivat.khas.edu.tr/~cesim/uniloc.rar. Experiments are performed on a computer with the configuration of AMD X2 3800+ of CPU and 3 GB of RAM. Because we propose a distributed algorithm, a discrete event simulation system has been designed. We start by generating a random network with parameters n and k. Firstly, n random points in the plane are generated using the random number generation of LEDA [17]. All nodes are assumed to have equal sensing range which is increased iteratively until average degree equals k. All experiments are reproducible in any platform. Network size is fixed at $n = 200$ and four randomly chosen K_3s

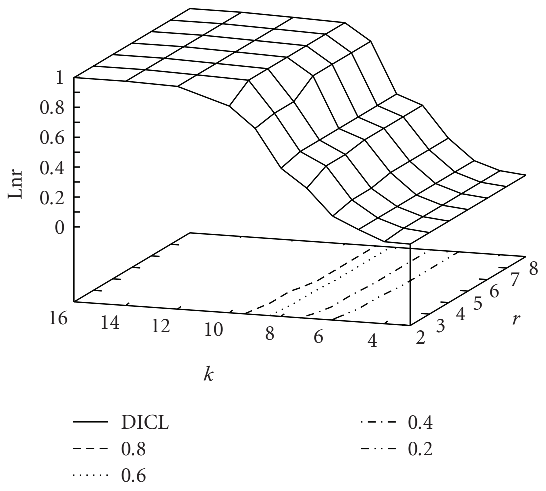


○ Unlocalized node (46.23%)
● Localized for $r = 1$ (26.13%)
■ Localized for $r = 2$ (46.23%
◇ Localized for $r = 3$ (154.77%)
■ Anchor (12.6%)

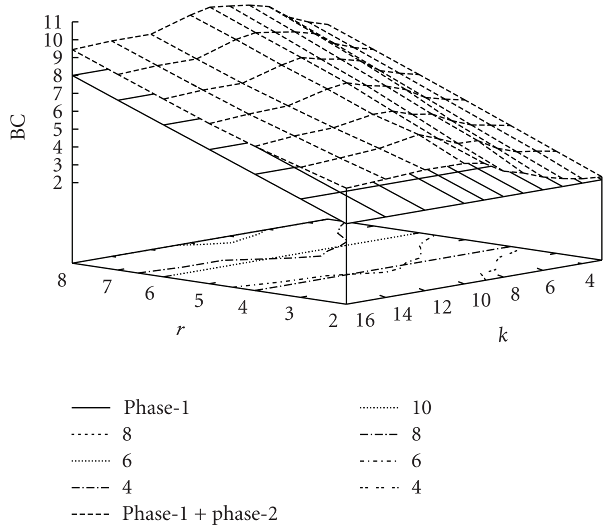
Figure 3: Random network with $n = 200$ and $k = 6$. Squares are localized for $r = 0$ (anchors), purple circles for $r = 1$, dashes for $r = 2$, diamonds for $r = 3$, and gray circles are unlocalizable.

are declared as anchors. The experiments are run for r, k values, where r varies from 1 to 8, and k varies from 4 to 16. Every unique configuration is repeated 10 times. The recorded results are divided into two when appropriate. First phase results are those arising from the cluster construction and the second phase results correspond to those of the actual unique localization.

We select performance measures in order to analyze and construe localization, messaging performances, and running time and space requirements. LNR (Localized Node Ratio) is the number of uniquely localized nodes divided by n. BC (average Broadcast Count per node) is total number of broadcasted messages divided by n. The size of each broadcasted packet differs especially in the second phase. Thus BC alone does not fully represent bandwidth usage per node. BA (average Broadcast Amount per node) is the total size of all broadcasted packets divided by n. AT (Average Time per node) is the average time spent for computations required by the localization algorithm on a physical sensor node. MP (Maximum possibilities per node) is defined as $\text{Max}_{u\in\mathbb{N}}$ $(\text{Max}_{v\in C_u^r}|v.\text{Positions}|)$. In our implementation we bound $|v.\text{Positions}|$ to be at most 1024. Finally, TP (average Total Possibilities per cluster) is $(\sum_{u\in\mathbb{N}}\sum_{v\in C_u^r}|v.\text{Positions}|)/n$, which is a measure of the average space requirements of a sensor node. Figure 3 is a visual illustration of our unique localization method. All the uniquely localizable nodes (almost 80% of the whole network) are uniquely localized with $r = 3$, whereas 6% of the network is uniquely localizable for $r = 0$ (anchors), 13% for $r = 1$, and 23% for $r = 2$. We note that in the simplest case, for $r = 1$, our method is analogous to iterative trilateration [7].



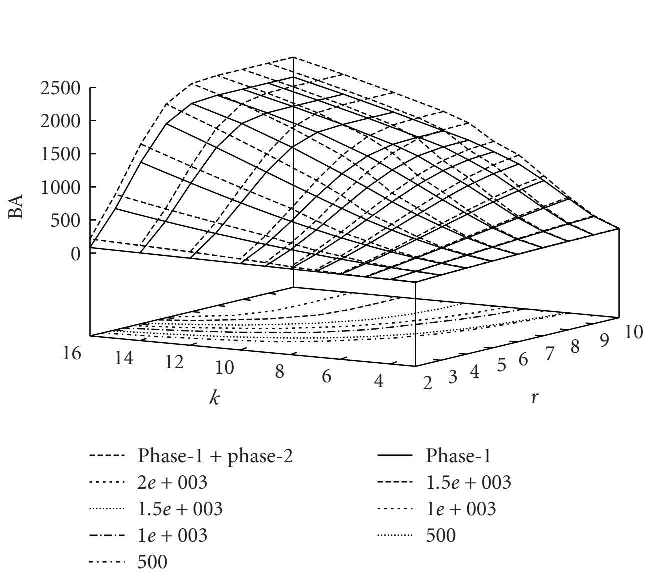
(a)



(b)

Figure 4: (a) The ratio of localized nodes. (b) Average number of broadcasts per node.

Figure 4(a) shows the growth of LNR with respect to r and k. As r increases, LNR grows as expected. There exist partial graphs, such as the one in Figure 2, that are localizable only for a specific r. Howeve, since the occurrence probability of such graphs is inversely proportional to r, for $r > 5$ when $6 \le k \le 10$, LNR does not grow drastically. The change in BC values with respect to r, k is plotted in Figure 4(b). Since the number of broadcasts in the Initial Setup phase is constant, the irregularity of the plot is caused by the broadcasts in the Iterative Localization phase. Number of broadcasts in this phase depends on how many new anchors are localized at each iteration. The BC values are maximum for $6 \le k \le 10$. For sparse networks when $k < 6$, the messaging overhead is low since not that many nodes are localized to be broadcasted in the first place. In contrast, when $k > 10$ each



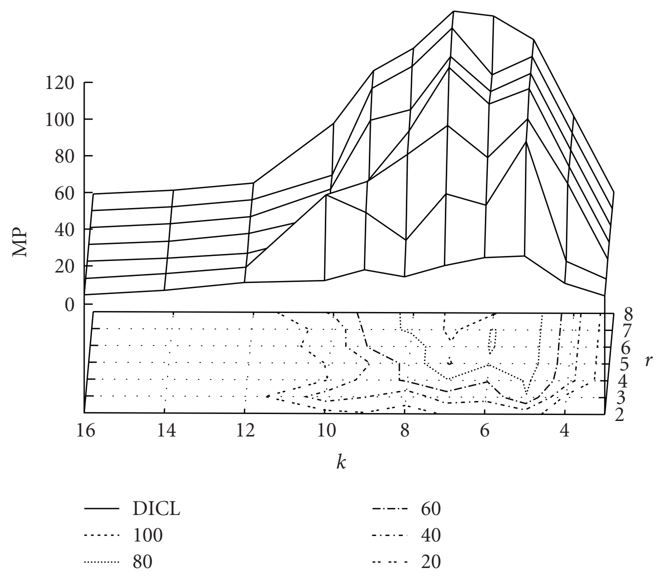
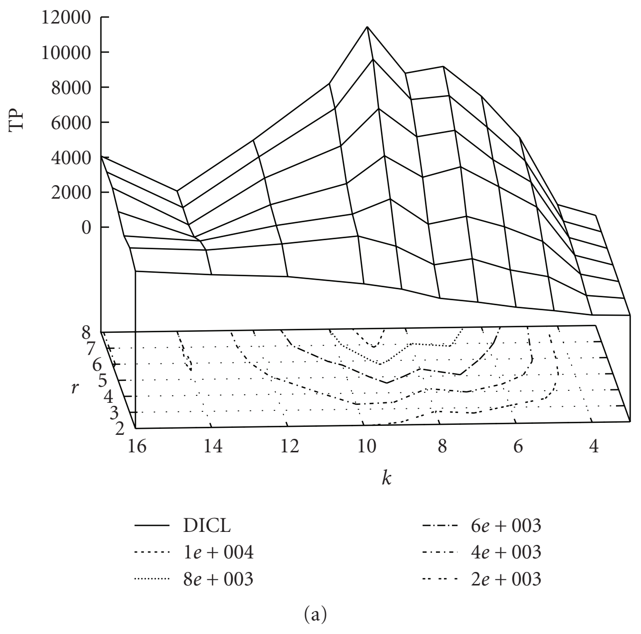


Figure 5: (a) Average amount of data broadcasted in units. (b) Average memory requirements (maximum possibilities per node).



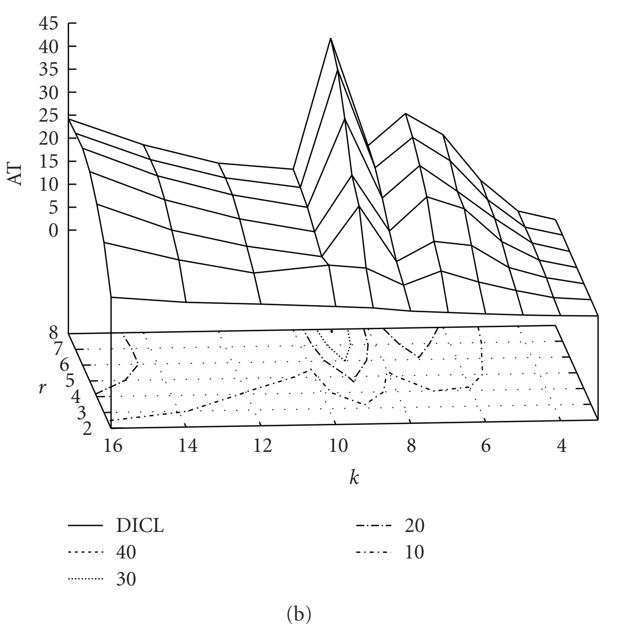


Figure 6: (a) Total possibilities stored in each cluster at any time during unique localization. (b) Simulation running time in seconds during localization.

localization iteration uniquely localizes many nodes at once therefore requires few broadcasts. However, as can be verified in Figure 5(a), the BA values indicating the broadcast size per node grow proportionally in terms of k and r. Figure 5(b) shows the MP values. The peak values are reached at $7 \leq k \leq 10$ for almost all r values, since low connectivity does not enable too many bilaterations, therefore possible locations, whereas high connectivity leads to unique localization too quickly.

Similar reasoning could apply to TP except that cluster size plays an important role as well in this case; see Figure 6(a). The average cluster size is k^r, therefore for large values of k ($k \geq 9$), TP is constant or grows slightly even though MP decreases. High connectivity leads to ease of unique localization but also gives rise to large clusters, which seem to cancel out each others' effects in terms of space requirements of a sensor node. It is interesting to analyze the time spent for localization computations at each node, since it depends on both MP and TP. As can be seen in Figure 6(b), the growth seems to be similar to that of MP for $k \leq 9$. However, the large cluster sizes reflected in TP seem to overcome the advantages created by low MP for larger connectivities and the running time required for localization increases.

5. Conclusion

We provided primitives for uniquely localizing nodes in a given sensor network. Assuming measurements with no noise, the suggested fully decentralized and fully collaborative computational model gives rise to a high rate of unique localization. Moreover, this goal is achieved with reasonably low energy requirements for message exchanges as the average number of iterations per node is low. An important direction for future work is to generalize the unique localization framework to handle error in measurements. It would also be useful to conduct experiments to analyze the efficiency of the model when the network is employed in various regions with randomly placed obstacles.

Acknowledgment

This work was partially supported by The Scientific and Technological Research Council of Turkey (TUBITAK) Grant no. 106E071.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] B. Krishnamachari, *Networking Wireless Sensors*, Cambridge University Press, New York, NY, USA, 2006.

[3] A. Efrat, C. Erten, D. Forrester, and S. G. Kobourov, "A force-directed approaches to sensor localization," in *Proceedings of the 8th Workshop on Algorithm Engineering and Experiments (ALENEX '06)*, pp. 108–118, Miami, Fla, USA, January 2006.

[4] D. K. Goldenberg, P. Bihler, M. Cao, and J. Fang, "Localization in sparse networks using sweeps," in *Proceedings of the 15th International Conference on Mobile Computing and Networking (MOBICOM '06)*, pp. 110–121, Los Angeles, Calif, USA, September 2006.

[5] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Proceedings of the 10th International Conference on Mobile Computing and Networking (MOBICOM '04)*, pp. 45–57, Philadelphia, Pa, USA, September-October 2004.

[6] A. Basu, J. Gao, J. S. B. Mitchell, and G. Sabhnani, "Distributed localization using noisy distance and angle information," in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '06)*, pp. 262–273, Florence, Italy, May 2006.

[7] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 50–61, Baltimore, Md, USA, November 2004.

[8] J. Aspnes, T. Eren, D. Goldenberg, et al., "A theory of network localization," *IEEE Transactions on Mobile Computing*, vol. 5, no. 12, pp. 1663–1678, 2006.

[9] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *Computer*, vol. 34, no. 8, pp. 57–66, 2001.

[10] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2007.

[11] J. Aspnes, D. Goldenberg, and Y. Yang, "On the computational complexity of sensor network localization," in *Proceedings of the 1st International Workshop on Algorithmic Aspects of Wireless Sensor Networks*, vol. 3121, pp. 32–44, Turku, Finland, July 2004.

[12] B. Hendrickson, "Conditions for unique graph realizations," *SIAM Journal on Computing*, vol. 21, no. 1, pp. 65–84, 1992.

[13] G. Laman, "On graphs and rigidity of plane skeletal structures," *Journal of Engineering Mathematics*, vol. 4, no. 4, pp. 331–340, 1970.

[14] R. Connelly, "Generic global rigidity," *Discrete and Computanional Geometry*, vol. 33, no. 4, pp. 549–563, 2005.

[15] A. R. Berg and T. Jordán, "A proof of Connelly's conjecture on 3-connected circuits of the rigidity matroid," *Journal of Combinatorial Theory Series B*, vol. 88, no. 1, pp. 77–97, 2003.

[16] B. Jackson and T. Jordán, "Connected rigidity matroids and unique realizations of graphs," *Journal of Combinatorial Theory Series B*, vol. 94, no. 1, pp. 1–29, 2005.

[17] K. Mehlhorn and S. Naeher, "LEDA: a platform for combinatorial and geometric computing," *Communications of the ACM*, vol. 38, no. 1, pp. 96–102, 1999.