# A Distributed Cluster-Based Localization Method for Wireless Sensor Networks

Carlos Moreno-Escobar*, Ricardo Marcelín-Jiménez†, Enrique Rodríguez-Colina‡ and Michael Pascoe-Chalke§
*Department of Electrical Engineering*
*Universidad Autónoma Metropolitana*
*Mexico City, Mexico*
*cbi208382775@xanum.uam.mx
†calu@xanum.uam.mx
‡erod@xanum.uam.mx
§mpascoe@xanum.uam.mx

***Abstract*—Node localization is a fundamental capability for several applications of Wireless Sensor Networks (WSN), such as security surveillance, fire detection, animal behavior monitoring, among others. Over the last decade, node localization in wireless sensor networks has evolved from centralized to distributed solutions. Therefore, more demanding conditions have arisen for new applications. These conditions come from massive node deployment and irregular topologies, requiring further analysis. In this paper, we present a method to reduce the signaling overhead due to a distributed localization procedure. This method consists of four stages: Based on the Awerbuch's $\gamma$ synchronizer, the proposal divides the network into clusters. The cluster size is restricted by a growing factor defined by a cluster-head, i.e., a *leader*. Based on connectivity information, the distance between each pair of nodes, belonging to the same cluster, is calculated by the corresponding leader. Next, each leader solves locally a particular instance of the MultiDimensional Scaling (MDS) problem. Finally, a minimum set of beacons is selected on each cluster. This is in order to assemble each region into a global localization solution within a single system of reference. In our method, we turn the initial settlement into several smaller instances of the original problem which can be solved simultaneously and based on local resources. Simulation results show that this approach produces important savings on the required message exchange.**

***Keywords-Localization; Partitioning; Synchronizer; Multidimensional Scaling.***

## I. I

Wireless Sensor Networks (WSN) is an emerging technology offering a wide spectrum of potential applications, and also a source of challenging problems to be solved [1]. Sensor node localization is a fundamental capability supporting most of these applications. A monitoring system, for instance, is able to determine the source of a critical event only if sensor nodes have accurate localization capabilities. Position awareness can also be used to enhance routing decisions because the nodes can send packets to their final destination based only on the position of nearby nodes, i.e., knowing the position of their neighbors. These routing strategies foster local work and, as a consequence, reduce the resource consumption [7], [11], [17].

For a small set of nodes, their individual positions can be programmed manually. In other cases, a Global Positioning System (GPS) may provide a convenient starting point. Nevertheless, the utilization of a GPS is limited due to budget constraints. Alternatively, a mobile node that is aware of its own position may perform a comprehensive tour across the underlying network. This mobile "coordinator" informs to each node about its corresponding position. It is important to recall that GPS is not recommended for indoor deployments, because satellite signal reception could be poor. When neither a GPS-based procedure, nor a manual programming are feasible, an automatic localization procedure is required.

Over the last years, an important number of proposals addressing self-configurable localization procedures have been published. Most of these proposals imply specialized solutions that perform well, merely under particular circumstances. Only a few of them have proved to be useful for general applications. However, even these general methods may show a poor performance under massive node deployment. In the meantime, technology trends show that WSN have permeated in different sector of our lives, as a consequence the number of deployed nodes is growing abruptly. In this context, scalability seems to be a new borderline in localization.

Despite of the fact that there is a well-known set of localization techniques offering general solutions [9], [10], [14], there are pending issues on the subject to be addressed. Scalability is one of these requirements to be fulfilled. The required methods developed to solve localization cannot be directly applied on a massive node deployment, due to their inherent message complexity, which limits the sensors energy budget. Apparently, the implicit agreement among scientists suggests that partitioning is a promising direction to address the scalability issue [18], [19]. From this approach, the underlying network is split up into regions or clusters. Each of the resulting clusters solves a reduced version of the localization problem. Finally, the local solutions are

 ISBN: 978-1-61208-166-3 

assembled between each other, like the pieces of a puzzle, in order to build the global solution.

The partition methods so far developed to address scalability, start selecting a set of nodes; each of these appointed nodes is in charge to build a cluster. A cluster grows inviting its neighbor nodes to join the graph under construction. Nevertheless, to our best knowledge, these procedures do not control neither the cluster growth rate, nor the initial number of appointed nodes. In Shang [16], for instance, each node in the graph is regarded to be a cluster by itself, provided that it is not assimilated by a bigger one. Therefore, the partition message complexity may turn out to be excessive. In addition, the simultaneous construction of clusters may produce an unnecessary condition where neighbor clusters compete for nodes which still are unassigned and, having an impact again, on the number of exchanged messages. In contrast, our proposal provides a control on the number of nodes which are initially appointed to start the graph partitioning. It also offers a parameter $k$, that allows to "modulate" the growth rate and, indirectly, the order of the resulting clusters, which has a deep impact on the message exchange and time complexity.

In this work, we have addressed the localization problem for a wireless sensor network with arbitrary topology, where the nodes are deployed at fixed but unknown positions. It is also assumed that the nodes do not have implemented a complementary device to estimate either, power range or distance. The method that we introduce consists of four consecutive stages: in the first stage, the underlying graph associated with the network is partitioned with our modified method. In the second stage, for each of the resulting clusters, the appointed starting node calculates the distance in terms of hops, between every couple of nodes belonging to the same cluster. In the third stage, each leader solves locally a particular instance of the multidimensional scaling problem. Finally, in the last stage, a minimum set of three beacons is deployed on each cluster, to assemble each region into a global solution within a single system of reference.

The rest of this document includes the following parts: In Section II, we formally define the problem and introduce the related work. In Section III, we describe the stages of our method and present a collection of performance assessments. In Section IV, we present the analysis of the results. Finally, we present our final remarks in Section V.

## II. D R W

From the point of view of graph theory , a network is modeled by a graph $G = (V, E)$, with an edge between any two nodes that can communicate directly with each other. In most of the cases, the multi-hop radio network is modeled as a Unit Disk Graph (UDG). In a UDG $G = (V, E)$, there is an edge $u, v \in E$ if and only if the Euclidean distance between $u$ and $v$ is less than or equal to 1.

An embedding of a graph $G = (V, E)$ in the Euclidean plane is a mapping $f : V \to \mathbb{R}^2$, i.e., each vertex $v_j$, $j = 1, 2, \ldots, n$ is identified by a point $x_j \in \mathbb{R}^2$ in the plane. A realization of a unit disk graph $G = (V, E)$, in the Euclidean plane is an embedding of $G$ such that $u, v \in E \leftrightarrow d(f(v), f(u)) \leq 1$, where $d$ is the Euclidean distance between two points. Therefore, localization consists of the realization of a unit disk graph in the Euclidean plane.

Localization is also considered as an optimization problem because given a set of measured distances between nodes that build a network, it is necessary to estimate the position of each node on a plane, up to rotations and translations. This is, while the error between the measured distances and the resulting distances from the estimated positions should be minimized. Practitioners introduce nodes with fixed and known locations, called *beacons* or *anchors*, in order to help the system to settle the reference coordinates.

In a sensor network in $\mathbb{R}^2$ there are two types of nodes: common sensors and anchors. A common sensor $j$ is a node which position has to be estimated and, it is denoted by $x_j \in \mathbb{R}^2$, $j = 1, 2, \ldots, n$. In contrast, each anchor $k$, has a well known position $a_k \in \mathbb{R}^2$, $k = 1, 2, \ldots, m$. Let $d_{ij}$ be the Euclidean distance between a pair of common nodes $i$ and $j$, and let $d_{jk}$ the Euclidean distance between a common node $j$ and an anchor $k$.

There are unknown pairs of distances for some cases, so the pairs of nodes, for which mutual distances are known, are denoted as $(i, j) \in N_x$ distance between sensor and sensor and $(j, a) \in N_a$ between sensor and anchor pair, respectively. The localization problem in $\mathbb{R}^2$ can be stated as: given $m$ anchor locations $a_k \in \mathbb{R}^2$, $k = 1, 2, \ldots, m$ and some distance measurements $d_{ij}, (i, j) \in N_x$, $d_{jk}, (j, k) \in N_a$, find the locations of common sensors, such that (ideally)

$$|x_i - x_j|^2 = d_{ij}^2, \quad \forall (i, j) \in N_x \tag{1}$$

$$|x_j - x_k|^2 = d_{jk}^2, \quad \forall (j, k) \in N_a \tag{2}$$

In many instances of the problem, noisy measurements introduce uncertainty on the calculations. Under such conditions, the problem can be reformulated as follows,

$$\min \{|x_i - x_j|^2 - d_{ij}^2\} \tag{3}$$

$$\min \{|x_j - x_k|^2 - d_{jk}^2\} \tag{4}$$

Notice that, anchors provide to the system with a fixed and absolute reference. Otherwise, when there are not anchors at all, the solution shows only relative positions. In other words, the "drawing" of the solution of the original network can be rotated, reflected or translated.

Different techniques have been proposed to *measure the distances* that make up the input set of the localization

 ISBN: 978-1-61208-166-3 

problem. These techniques can be classified into two main categories: *range-based* and *connectivity-based* (also called range free). The former depends on a physical signal exchanged between two points which value is a function of the length, or relative position, of the line of sight from transmitter to receiver. e.g., Angle of Arrival (AoA), Time of Arrival (ToA), and Received Signal Strength (RSS).

The downside of range-based techniques is that they require additional hardware that may impact on the price of individual nodes. Besides, they can be very sensitive to environmental conditions. In contrast, *connectivity-based techniques* depend on the number of hops separating any pair of nodes. In this case, it is assumed that two nodes sharing an edge are separated, at most, by one distance unit. For both categories, indirect measurements may be propagated to other nodes in the network using a distributed procedure, such as the Distance-Vector algorithm (DV), where each node successively sends all the distances and the paths to reach the destinations that it already knows.

Research on localization methods has produced reasonable methods that offer excellent performance when the deployed sensors make up a dense and globally uniform network. Among the most relevant proposals, we found that Shang et al. [15] demonstrated the use of a data analysis technique called "MultiDimensional Scaling" (MDS) in estimating positions of unknown nodes. First, using basic connectivity or distance information, a rough estimate of relative node distances is made. Then, classical MDS (which basically involves using eigenvector decomposition) is used to obtain relative maps of the node positions. Finally, an absolute map is obtained by using the known node positions. This technique works well with few anchors and reasonably high connectivity. For instance, for a connectivity level of 12 and 2% anchors, the error is about half of the radio range.

Suppose we could count on the matrix $\mathbf{X}$, where each of its rows codes the position of a point on an Euclidean space. It is possible to calculate the square of the distances between any pair of points in this collection, according to the following expression

$$\mathbf{D(X)}^2 = \mathbf{c1}^T + \mathbf{1c}^T - 2\mathbf{XX}^T = \mathbf{c1}^T + \mathbf{1c}^T - 2\mathbf{B} \qquad (5)$$

where $c$ is a vector made up with the elements from the diagonal of $\mathbf{XX}^T$. Then, we left and right multiply by a centering matrix and by the factor $-1/2$ to obtain

$$\begin{aligned} -\frac{1}{2}\mathbf{HD(X)}^2\mathbf{H} &= -\frac{1}{2}\mathbf{H}(\mathbf{c1}^T + \mathbf{1c}^T - 2\mathbf{XX}^T)\mathbf{H} \\ &= -\frac{1}{2}\mathbf{Hc1}^T\mathbf{H} - \frac{1}{2}\mathbf{H1c}^T\mathbf{H} + \frac{1}{2}\mathbf{H}(2\mathbf{B})\mathbf{H} \\ &= -\frac{1}{2}\mathbf{Hc0}^T - \frac{1}{2}\mathbf{H0c}^T + \mathbf{HBH} = \mathbf{B} \end{aligned} \qquad (6)$$

The first two parts of the equation are canceled since centering a vector made up with 1's produces a vector made up with 0's only ($\mathbf{1}^T\mathbf{H}=\mathbf{0}$). In turn, as we assume that the columns in $\mathbf{X}$ have a mean equal to 0, the centering matrices around $\mathbf{B}$ can be dismissed. Now we can see that if were able to factorize $\mathbf{B}$, according to an eigendecomposition, it will turn out that $\mathbf{Q}\Lambda\mathbf{Q}^T = (\mathbf{Q}\Lambda^{1/2})(\mathbf{Q}\Lambda^{1/2})^T = \mathbf{XX}^T$. There exist a tool that carries out this decomposition: the so-called power method, which is an iterative algorithm of complexity $O(n^3)$, where $n$ is the number of unknown positions. We also tested an optimization approach, called the majorization method, which also is an iterative algorithm of complexity $O(n^2)$, but it is not based on eigendecomposition [3], [8].

## III. M

A synchronizer is a set of techniques that enables an asynchronous system to emulate a synchronous behavior. To support this emulation, each node should be able to proceed with the next step of the given algorithm, only when it is granted that all the participants have accomplished the preceding step [13]. A node under these condition is said to be "safe".

Awerbuch [2] introduced three types of synchronizers: the $\alpha$ type, where each node exchanges messages with all its neighbors to let them know that it is safe. The $\beta$ type, where a spanning tree is previously built. Here, a node sends a message to the root when the current step has finished. Once the root has collected these messages from each node, it broadcasts back a new message to the nodes on the tree, in order to notify the overall safety.

Finally, in the $\gamma$ synchronizer the underlying graph is partitioned into a forest. Each of the resulting trees, also called cluster, runs a local version of the $\beta$ synchronizer. However, when the nodes of a given cluster have finished the current step, the root exchanges messages with its neighbor trees to let them know of its local condition. When a root recognizes this condition on each of its neighbor subgraphs, it broadcasts back a new message to the nodes of its cluster to notify the overall safety.

The $\gamma$ synchronizer requires an initialization procedure to split up the underlying graph in a set of disjoint clusters. The construction of a cluster starts when a given node, still unexplored, is appointed as a leader. The new leader begins aggregating layers to the cluster under construction. It is expected that a new layer that joins the cluster should contain, at least, as many nodes as, $k$ times the total number of nodes already in the cluster. When this condition is not met, the cluster construction stops. Then, a new leader is found and the procedure starts again. Here, there is a special link, called "preferred", between the former tree and the new one about to be settled. This link fixes a relationship between the "ancestor" cluster and its "successor". When the cluster stops growing and a new node cannot be appointed, the leader in charge turns the control back to its ancestor tree.

 ISBN: 978-1-61208-166-3 

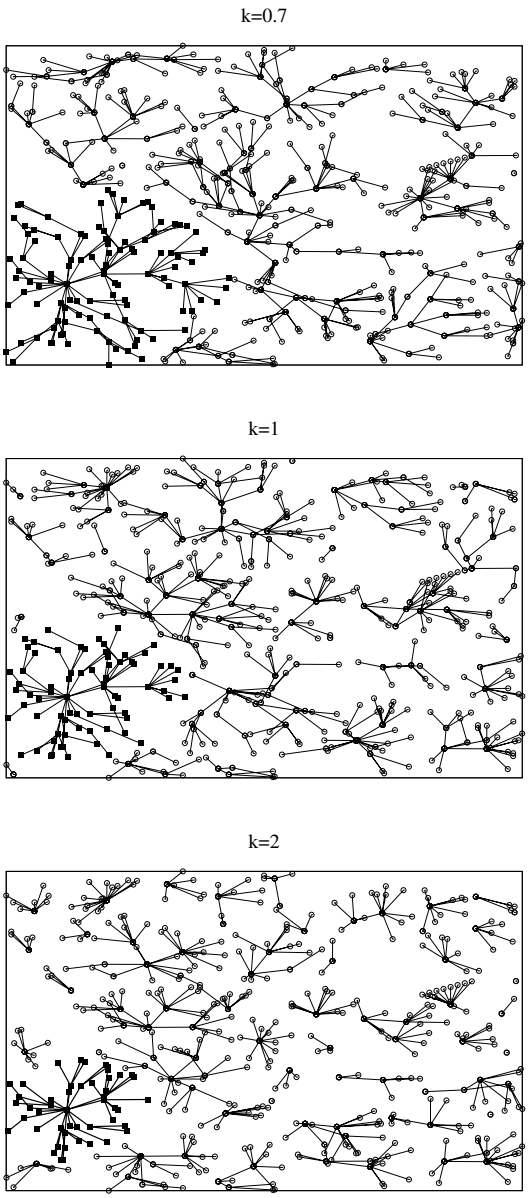


Figure 1. Building clusters with different values of $k$.

In due time, the receiving leader looks for a node to start a new successor tree, otherwise it also turns the control back to its own ancestor tree. According to this rule, the initial leader is able to recognize the moment when the partition is finished. The graph has been exhaustively explored and each node has been incorporated to a given tree. Our partitioning technique is based on a cluster growth parameter $k$. While the original work does not consider the values of $k < 1$, our implementation supports any value of $k > 0$. Nevertheless, when the partition process works under these "suboptimal" growth rate, each cluster grows with a very slow pace and the average cluster order increases.

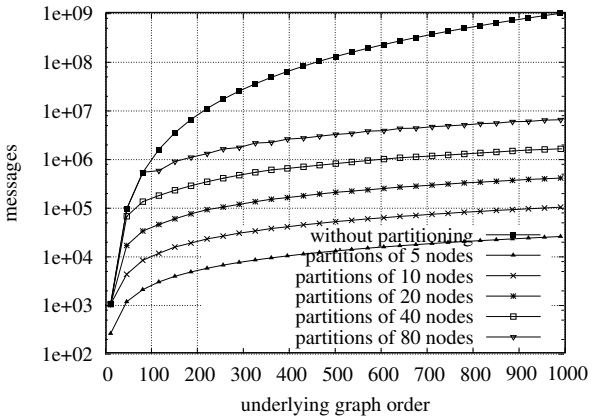


Figure 2. The benefits of partitioning

Besides this original partition procedure, herein called the "serial" partitioning, in this work we propose a new approach called the "concurrent" partitioning which once a cluster stops growing, each node in the border of the cluster selects a neighbor not yet assigned. Each of the newly selected nodes concurrently receives a signal permission to build a new cluster and as a result, preferred links between clusters are implicitly defined. In contrast to the original procedure, a given cluster, does not turn back the control to its ancestor when there is not any further place to explore. This feature does not preclude the further start of the next stage of our global localization method.

Figure 1 shows the behavior of the proposed partitioning algorithm for three values of $k$: 0.7, 1 and 2, respectively.

Our partitioning approach shows similarities with the work presented in [4], [12]. In contrast, our method does not have as many cluster construction rules as they do. Potential conflicts on the nodes' assignation are solved with a very simple rule: a free node, i.e., a node not yet assigned to a cluster, decides to be part of the first cluster that accepts it. Otherwise it will eventually turn into a new cluster leader on its own.

We developed a first assessment assuming that the system runs the localization procedure without a previous partitioning, then it is run by choosing a partitioning with different orders, i.e., the number of nodes on the resulting clusters. Figure 2 shows the overall message complexity associated to each test. Results show that partitioning saves expenses by several orders of magnitude.

In a second evaluation, we decided to compare the serial and the concurrent partitioning stages, for a value of $k = 1$. We test both over 50,000 different networks with 600 nodes each, which have been generated randomly. Our results provide a 95% confidence.

In the second stage of our localization procedure, a local instance of the Bellman-Ford [6] algorithm is executed on each of the resulting clusters to calculate the shortest

 ISBN: 978-1-61208-166-3 

Table I
R

| Variables | Serial | concurrent |
|---|---|---|
| Finalization Time[a] | 809.25 | 213.01 |
| Messages Transmitted[b] | 28146.73 | 25445.32 |
| Transmitted Messages per Node[c] | 49.77 | 45.81 |
| Leader's Transmitted Messages[b] | 1920.82 | 2091.70 |
| Leader's Transmitted Messages[c] | 111.79 | 83.07 |
| Avg. Number of Resulting Clusters | 4.37 | 11.98 |
| Avg. Cluster Size | 19.336 | 11.11 |

[a] assuming that a message is transmitted using a time unit
[b] average total number of messages
[c] average individual number of messages

path between every couple of nodes belonging to the same cluster. The length of each path is considered a substitute for the Euclidean distance between nodes, which is required in the following stage. The routing algorithm requires the whole set of links that make part of the induced subgraph. In the original work, each node exchanges messages with all its neighbors, to calculate the shortest path between any couple of nodes. This approach produces a message complexity $O(|V|^3)$, where $|V|$ is the order of the underlying graph, i.e., the total number of nodes that make part of the graph. However, in wireless sensor networks, message transmission is an event that has a major impact on the nodes' energy supply. For this reason, we developed an alternative approach: using the tree that spans its cluster, each node sends to the root its neighbors list. The leader which is appointed as the root, collects this information to build a model of the underlying graph and then it runs a centralized version of the routing algorithm. This method has a complexity $O(|V|)$ on the number of exchanged messages.

When a leader has estimated the distances between any couple of nodes belonging to its cluster, it starts the third stage of our procedure: it solves a local instance of the MultiDimensional Scaling problem (MDS) i.e., it transforms a distance matrix into a list of vectors coding the positions where nodes can be preliminary located. We evaluated three alternatives, see Figure 3: a) the classical eigendecomposition, b) a second iterative procedure called the majorization method, i.e., Scaling by Majorizing a Complicated Function (SMACOF), and c) the combination of both. In this last procedure, we build a preliminary solution using method a) which is further supplied as a new input to method b). As it could be expected, this combined approach offers the best results. Nevertheless, the second alternative offers nearly the same quality under a lower price. Let us recall that eigendecomposition has a complexity order equal to $O(n^3)$, where $n$ is the number of unknown positions. In contrast, majorization's complexity is $O(n^2)$.

To the authors' best knowledge, all the preceding work based on range free distance estimation assume a fixed hop

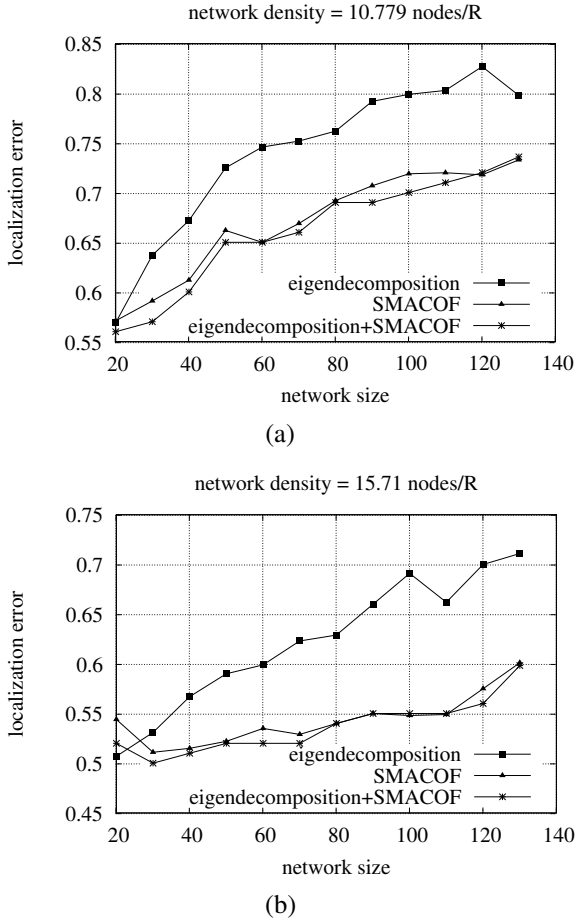


Figure 3. Localization error for two network densities.

length equal to one. Our contribution in this stage also consists on a test varying the hop length between 0 and 1. We found that, depending on the density of the underlying graph there is a hop value that optimizes the outputs of the MDS decomposition. These results are shown in Figure 4. We plotted the reconstruction error for ten different hop values and for three network average densities: 4, 8 and 12 $Nodes/Range$. In each case exists a hop value that minimizes the MDS reconstruction error. Also, note that the mean error decreases due to the network density increment.

When each leader has solved the local instance of the MDS, the geometric center of the cluster is considered at the position $(0, 0)$, or $(0, 0, 0)$, whether the nodes deployment are in 2D or 3D, respectively. This means that all clusters are logically overlapped. In the last stage of our procedure, we install a minimal set of beacons on each cluster in order to perform an isometric transformation that fixes the final coordinates of each region. And thus, a global and coherent picture of the system has been built.

Figure 5 shows the results of localization without and with partitioning, respectively. It is worth mentioning that

 ISBN: 978-1-61208-166-3

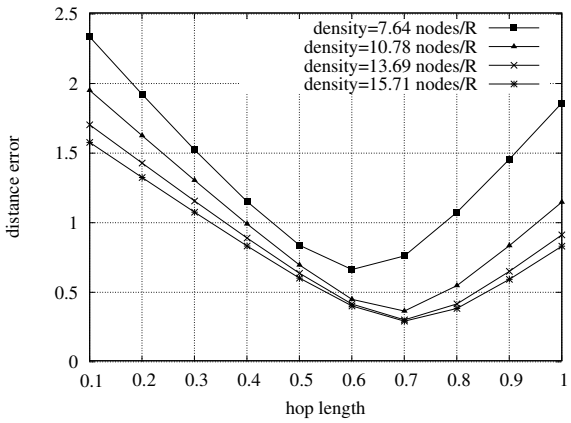


Figure 4. Reconstruction error varying the hop length in normalized units

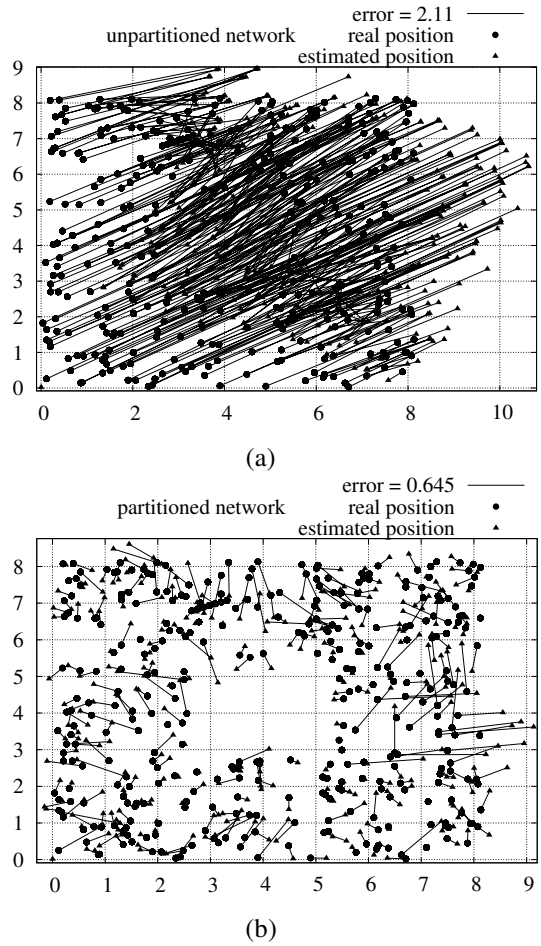


Figure 5. A network reconstruction. Fig. (a) reconstruction without partitioning. Fig. (b) reconstruction with partitioning.

compared to the proposal of Shang, our method achieves similar results. Nevertheless, from Shang's point of view in [15], each node starts working considering itself a cluster on its own and therefore, exchanging an excess of messages. We bound this message complexity by growing clusters during the first stage of our method. The underlying tree that spans the nodes of each cluster provides an efficient message exchange.

## IV. A

We introduced a localization procedure which consists of four consecutive stages: in the first stage, the underlying graph is partitioned. In the second stage, each appointed starting node calculates the distance in hops, between every couple of nodes belonging to its cluster. In the third stage, each leader solves a local instance of the multidimensional scaling problem. Finally, in the last stage, we introduce a set of beacons on each cluster, in order to assemble each region into a coherent solution within a single system of reference.

Our partitioning technique is based on a cluster growth parameter $k$. We realized that a value of $k > 1$ offers better solutions in terms of: i) time to solve stage one, ii) it dramatically reduces the amount of resources involved on the overall procedure, iii) the reduced overall expenses are shared among a bigger number of participants, and iv) it produces more accurate solutions.

In the downside, we consider that the last stage limits the applicability of our method, but we have also identified that in order to overcome this limitation it is necessary to review the connection step between neighbor clusters, during partitioning. It is known that the rigidity of a graph is a desired property that facilitates its realization in an Euclidean space. Therefore, the more connections there are between neighbor clusters, the more rigid is the resulting combined graph [5]. If the number of links between clusters is maximized, then it is possible to use a minimal number of beacons to fix a global coordinated system.

The partitioning method works on any network, independently from its topology and size. We found, in fact, that this partitioning stage can cope with irregularities and obstacles and, it is a necessary step to scale up any localization algorithm. This is a well-known approach called "divide and conquer". The initial settlement turns in several local instances of the original localization problem, where it is assumed that these local instances are easier to solve than the initial settlement and can be solved simultaneously. In addition, this approach enforces the organization based on local resources and being also possible to achieve coordination in a global context.

The coordination is a key capability whose complexity depends on partitioning. If each node of the network were a cluster by itself, it would require to exchange messages with its immediate neighbors to achieve a coordinated action as it is proposed in the $\alpha$ synchronizer. Although it is a very fast strategy for the coordination, it can be very expensive in terms of the overall number of messages sent on each link of the underlying graph. In contrast, the $\beta$ synchronizer can be used where a single spanning tree could be previously

 ISBN: 978-1-61208-166-3 

built on the graph. And as a result, it would be necessary a minimal number of messages to coordinate the whole system. Whereas, the time required to achieve coordination may be as much as the necessary to travel the tree's longest path.

The $\gamma$ synchronizer and the concurrent version proposed in this work find a trade-off between the number of messages and the time complexity in a coordination procedure, including the localization process.

## V. C

The method that we introduce consists of four consecutive stages.

In the first stage, our solution comprises partitioning the underlying communications graph as proposed in [16], [18], [19]. However our method has a significant improvement in reducing computational resources, since we control the cluster grow rate, as well as the number of simultaneous clusters under construction. Indeed, our approach shares many ideas with the work of [4], [12].

In the second stage, for each of the resulting clusters, there exists an appointed starting node called leader, that calculates the distance in hop units between every couple of nodes belonging to its cluster. This operation can be solved using the distance-vector protocol. Nevertheless, this method requires a message exchange that has a major impact on the energy supply. Therefore, we developed an alternative method which reduces the message complexity from $O(|V|^3)$ to $O(|V|)$.

In the third stage, each leader solves locally a particular instance of the multidimensional scaling problem. An estimation of the distances between any couple of nodes lying on the same cluster is required as the input for this stage. In many cases the distance in hop units is regarded as a good alternative. Most of the authors, cited in the references, fixed the hop length to one. We found that the density of the underlying graph determines the optimum hop length for the MultiDimensional Scaling (MDS) decomposition method. Therefore, for each case there is a hop length that minimizes the MDS reconstruction error and for the examples shown here, the optimum hop length is around 0.7 instead of one.

Finally, in the last stage, a minimum set of three beacons is deployed on each cluster. Beacons provide a global reference that supports an isometric transformation of the cluster position. This means that the cluster can be rotated or translated to its final position within a single system of reference.

From our point of view, the saving achieved with our distributed method comes from different sources; evidently, the most important is that the method works simultaneously on the construction of several clusters. In addition and, in contrast with our method, the $\gamma$ synchronizer spends more time on both, the selection of the next leader and appointing the preferred links.

The proposal presented in this work shows simple but significant contributions in each stage of the localization method. The results show that our solution significantly reduces the number of messages exchanged, which is indeed an important operation condition for wireless sensor networks.

For future work we are planning the implementation of our method on a real massive node deployment.

## R


[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 12, no. 2, pp. 291–301, June 1991.

[2] B. Awerbuch, "Complexity of network synchronization," *Journal of the ACM (JACM)*, vol. 32, no. 4, pp. 745–770, October 1985.

[3] T. Cox and M. Cox, *Monographs on Statistics and Applied Probability 59: Multidimensional Scaling*. London: Chapman and Hall, 1994.

[4] B. Derbel and M. Mosbah., "A fully distributed linear time algorithm for cluster network decomposition." in *16th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS04)*, 2004, pp. 548–553.

[5] T. Eren, D. K. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. O. Anderson, and P. N. Belhumeur, "Rigidity, computation, and randomization in network localization," in *In Proceedings of IEEE INFOCOM '04, Hong Kong*, 2004, pp. 2673–2684.

[6] L. R. Ford and D. Fulkerson, *Flows in Networks*. Princeton University Press, 1962.

[7] C. Georgiou, E. Kranakis, R. Marcelín-Jiménez, S. Rajsbaum, and J. Urrutia, "Distributed Dynamic Storage in Wireless Networks," *International Journal of Distributed Sensor Networks*, vol. 1, no. 3, pp. 355–371, 2005. [Online]. Available: http://dx.doi.org/10.1080/15501320500330695

[8] P. Groenen and I. Borg, *Modern Multidimensional Scaling, Theory and Applications*. New York: Springer-Verlag, 1997.

[9] G. Mao and B. Fidan, *Localization Algorithms and Strategies for Wireless Sensor Networks*. New York: Information Science Reference, 2009.

[10] R. Marcelin-Jimenez, M. Ruiz-Sanchez, M. Lopez-Villasenor, V. Ramos-Ramos, C. Moreno-Escobar, and M. Ruiz-Sandoval, *Emerging Technologies in Wireless Ad Hoc networks: Applications and Future Development*. IGI Global, 2010, ch. A survey on Localization in Wireless Sensor Networks.

[11] R. Marcelín-Jiménez, "Locally-constructed trees for ad-hoc routing," *Telecommunication Systems*, vol. 36, no. 1-3, pp. 39–48, January 2007.


 ISBN: 978-1-61208-166-3 

[12] M. Mosbah, B. Derbel, and A. Zemmari, “Fast distributed graph partition and application.” in *In 20th IEEE International Parallel & Distributed Processing Symposium (IPDPS06). IEEE Computer Society Press*, April 2006.

[13] D. Peleg, *Distributed Computing. A Locality-Sensitive Approach*. Philadelphia: Society for Industrial and Applied Mathematics, 2000.

[14] L. M. Pestanao de Brito and L. M. Rodriguez Peralta, “Collaborative localization in wireless sensor networks,” in *Proceedings of the 2007 International Conference on Sensor Technologies and Applications*. IEEE Computer Society, 2007, pp. 94–100.

[15] Y. Shang and W. Ruml, “Improved MDS-based localization,” in *In proceedings of IEEE INFOCOM ’04, Hong Kong*, 2004, pp. 2640–2651.

[16] Y. Shang, W. Ruml, and Y. Zhang, “Localization from mere connectivity,” in *International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, March 2003, pp. 201–212.

[17] J. Urrutia, *Routing with guaranteed delivery in geometric and wireless networks*. New York, NY, USA: John Wiley & Sons, Inc., 2002, pp. 393–406. [Online]. Available: http://portal.acm.org/citation.cfm?id=512321.512339

[18] L. Wang and Q. Xu, “GPS-free localization algorithm for wireless sensor networks,” *Sensors*, vol. 10, no. 6, pp. 5899–5926, 2010. [Online]. Available: http://www.mdpi.com/1424-8220/10/6/5899/

[19] C.-Y. Wen, Y.-C. Hsiao, and F.-K. Chan, “Cooperative anchor-free position estimation for hierarchical wireless sensor networks,” *Sensors*, vol. 10, no. 2, pp. 1176–1215, 2010. [Online]. Available: http://www.mdpi.com/1424-8220/10/2/1176/

 ISBN: 978-1-61208-166-3 