Large-Scale Networked Systems: From Anarchy to Geometric Self-structuring

Anne-Marie Kermarrec[1], Achour Mostefaoui[2], Michel Raynal[2], Gilles Tredan[2],
and Aline C. Viana[3]

[1] INRIA Rennes-IRISA
[2] Université de Rennes 1-IRISA
[3] INRIA Saclay, France

Abstract. We define geometric self-structuring in a large-scale networked system as the ability of the participating nodes to collaboratively impose a geometric structure to the network. Self-structuring is hard to achieve when no global positioning information about the network is available. Yet this is an useful capability in networked autonomous systems such as sensor networks. In this paper, we present the design and the evaluation of a fully decentralized geometric self-structuring approach. This approach heavily relies on the ability of each node to estimate its position in the network. The contribution of the paper is twofold: *(i)* a simple and fully decentralized virtual coordinated system (VINCOS) is proposed, relying only on local connectivity information and per-neighbor communication; *(ii)* a network geometric self-structuring approach (NetGeoS) is presented that enables a large set of nodes to configure themselves in arbitrary geometric structures. The evaluation shows that the approach is both efficient and accurate while achieving the geometric structuring.

1 Introduction

Context. Self-structuring refers to the ability of a networked system to let emerge a specific structure, from scratch, without requiring external information. Self-structuring is an important dimension of system autonomy (especially in terms of scalability issues [14]). In sensor networks for example, self-structuring represents an important requirement for common operations such as forwarding, load balancing, leader election, or energy consumption management (see again [14]). Examples include the partitioning of an area in several zones for monitoring purposes or the selection of sensors to ensure specific functions for energy saving.

Motivation. The complexity of a self-structuring mechanism strongly depends on the amount of knowledge that is initially provided to nodes in the network. If all nodes in the system have a complete knowledge of the system, structuring the network is trivial. Otherwise, if nodes are only aware of their own neighborhood, ensuring that a given structure emerges from individual decisions is challenging. Let the *external knowledge* be the information provided to a node by an external

V. Garg, R. Wattenhofer, and K. Kothapalli (Eds.): ICDCN 2009, LNCS 5408, pp. 25–36, 2009.

entity or device. This is to oppose to *intrinsic knowledge* that consists of the information that each entity gathers itself from its observation of the network. The more external knowledge is required, the less robust a system is (especially in environments where human intervention is difficult). Instead, approaches relying mostly on intrinsic knowledge definitively increases system autonomy at the price of a higher communication overhead. In short, the autonomy degree of a networked system is inversely proportional to the external knowledge required to structure the network. It is however, crucial to come up with a reasonable trade-off between autonomy and overhead.

This paper presents a robust structuring mechanism, leading to geometric organization that can be deployed from scratch in a networked system where the initial knowledge of each node is limited to its own identity and communication range. Nodes can then be assigned to different adaptive behaviors based on the established organization. In the context of this work, we focus on wireless sensor networks (WSNs). To the best of our knowledge, this is the first *geometric structuring autonomous system* deployed upon those conditions in the literature.

Network structuring becomes a very challenging goal, as soon as neither positioning referential, boundary delimitation, nor density distribution is provided. One solution for this is to allow the nodes to access a coordinate system from which they can obtain a coordinate assignment. Such a coordinate system represents the basic layer on top of which adaptive behaviors can be designed. This constitutes our second contribution.

Most of positioning approaches in the literature [1, 6, 8, 10, 12] rely on some specific assumptions and this shows striking evidence that even more autonomy is required. This has been coped by solutions that use *no position-aware referential points* and that result in *virtual coordinates* being assigned to nodes, instead of geographic coordinates [4,6,15]. Virtual coordinates better reflect the real network connectivity, and can consequently provide more robustness in the presence of obstacles. Despite having clearly defined outlines and presenting good approximation solutions, previous works on virtual coordinates are computationally- (and message) costly, or hardly practical in wireless sensor networks [4, 6]. This situation is slowly changing though [16]. In addition, the solutions presented in [4, 8, 15], although not accounting for position-aware landmarks, require the nomination of well placed entities in the systems to work as anchors or bootstrap beacon nodes.

This paper proposes a versatile virtual coordinate system for WSNs. The main difference with related works is that the proposed approach does not rely on any anchors, position-aware landmarks, or signal measurement. Yet, nodes get assigned virtual coordinates in a fully decentralized way. Nodes derive local connectivity information, solely leveraging their per-neighbor communication.

Contribution. In summary, the contributions of the paper are the following ones.

- A simple and fully decentralized, *VIrtual Networked COordinate System* (VINCOS) that achieves a good coordinate assignment.
- A *Networked Geometric Structuring* approach (NetGeoS) for autonomous systems. Here, we show how NetGeoS builds upon VINCOS.

Outline. After introducing our system model in Section 2, we present the design rationale of our approach and give an overview of related works in Section 3. The virtual coordinate system (VINCOS) is described in Section 4. In Section 5, we show how VINCOS can be used to obtain specific geometric structuring. Performance results are presented in Section 6. We give an overview of related works and conclude this work in Section 7. More development can be found in [7].

2 System Model

Area monitoring is one of the most typical applications of WSNs. It consists in deploying a large number of sensors in a geographic area, for collecting data or monitoring events. It is not unusual that human intervention is hardly feasible in such settings. Sensors are then deployed in mass and must be able to form a network and to operate in a decentralized self-organized manner, maintaining connectivity and area monitoring for as long as possible. These are typical applications considered here.

Nodes. We consider a system composed of a (finite) set of N resource-limited wireless sensor devices scattered on a geographical area. Each node gets assigned a unique identifier i. This represents both the only difference between two nodes and the only information that nodes have about the configuration. Nodes are all “equal” with respect to their capabilities. No synchronization is required.

Communication. Each node i is able to directly communicate wirelessly with a subset of nodes that are reachable (we refer to this subset as the *neighbors* of the node i). No node is provided with geographical topology information (such as physical obstacles). We assume *bidirectional* communication and that the density of nodes is such that the resulting communication network is connected (*i.e.*, the network is not partitioned). The presented hereafter approaches relies solely on node connectivity.

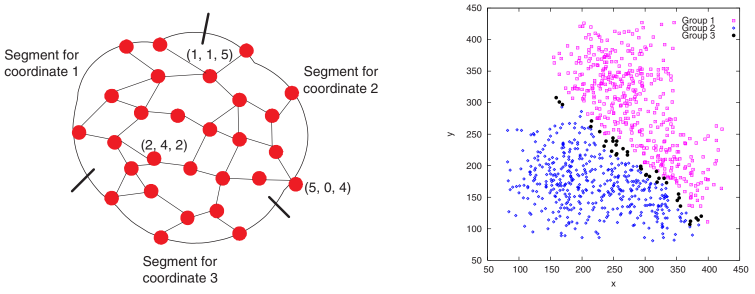
Initial knowledge. Initially a node only knows its identity, the fact that no two nodes have the same identity, and a parameter d that will define the size (or dimension) of the virtual coordinate space. The design of the parameter d depends on the application requirements.

3 Design Rationale

This section discusses the fundamental principles that underlie each of our contributions.

3.1 VINCOS: Virtual Coordinate System

Position awareness is a key functionality to build and maintain autonomous networked systems. A coordinate system provides each node with a “position”



(a) (b)

Fig. 1. (a) An example of virtual coordinates. (b) “Line partitioning” in a topology with three hot spots of different nodes densities.

that is both individual and globally consistent. As already observed, a node network awareness may come from two sources: an *intrinsic knowledge* resulting from the algorithm execution (*e.g.*, neighbors set, hop distance to a specific node, etc.), and *external knowledge* supplied by external devices (*e.g.*, satellite or radio signal device), or design hypothesis (such as “all sensors have a unique *id*”, “the network is initially connected”, etc.).

In VINCOS, each node is initially configured with the parameter d indicating the dimension of the coordinate system (number of coordinates). The virtual coordinates of a node i are consequently a tuple $(x_1, \ldots, x_d)$, where x_j is the projection of i on the jth axis of the d-dimensional virtual space.

The virtual d-dimensional space is defined as follows. The border of the geographical area covered by the nodes is partitioned into d “segments”. These border segments can have the same size or different sizes (we assume in the paper that they have the same size). This depends only on the algorithm and may easily be changed. Let us consider any axis j, $1 \leq j \leq d$, of the coordinate system. The coordinate x_j is the length, in hops, of a shortest path from the node i to the border segment j (*i.e.*, to the closest node on that border segment). This is illustrated on Fig. 1(a) through a simple example. The coordinate system is 3-dimensional ($d = 3$). The virtual coordinates of three nodes are indicated. The coordinates $(2, 4, 2)$ mean that the corresponding node is at distance 2 of both borders 1 and 3, and at distance 4 of border 2.

The coordinate assignment of VINCOS depends on an accurate definition of the *border segments*, *i.e.*, the d “segments” dividing the border of the considered geographical area. To define the border segments, VINCOS relies on a *belt* construction mechanism. The resulting belt, defined as a set of *border-belt nodes* (nodes located on the perimeter of the area), is a connected structure that (*1*) enables communication among border-belt nodes along two different paths; (*2*) allows an order assignment to these nodes; (*3*) is one-hop wide; and (*4*) has proportional size wrt the network size[1]. This ensures that, given the broadcast

---

[1] A too small belt could generate inaccuracies in the coordinate system resulting in many non-neighbor nodes being assigned to the same coordinates.

communication pattern, any message forwarded along the belt (complete round) reaches every border-belt node.

Nevertheless, discovering a connected border at a low cost and in an accurate way is difficult. We quickly describe at the Section 4, how VINCOS computes such a belt, referred hereafter as *border-belt*.

3.2 NetGeoS: Geometric Structuring

In the literature, network coordinates are mainly used for routing. We argue that their interest goes far beyond. When appropriately manipulated, network coordinates represent a powerful tool for different kinds of network management. A main contribution of this paper lies in showing how to rely on such a coordinate system for a different purpose: geometric structuring.

NetGeoS defines organization laws that, once applied to the virtual coordinates of each node, let emerge a specific *geometric structuring* so that nodes get assigned different behaviors/functionalities depending on their position in the network. Geometric structuring can be (1) defined as a logical partitioning of the network and (2) used as a powerful tool for structuring wireless networks and assigning different functionalities to nodes for clustering, data aggregation, or energy consumption management. Although this can be easily done upon any polar coordinates, we show here how NetGeoS is deployed upon VINCOS.

As an example, consider sensors disseminated in a large geographical area and that are partitioned in three groups, the *North* group, the *South* group, and the *Equator* group. The Equator group is a simple “straight line” of nodes that separating the north and south groups. Let us consider the application scenario where the nodes in the North and South groups are in charge of collecting some information, subsequently sent (via a routing protocol) to the Equator group. Periodically, a plane flies over the Equator line and collects the relevant data stored in the Equator sensors [3]. To implement such an application, the sensor network should be partitioned in such three groups. It turns out that this can be easily achieved with a 2-dimensional virtual coordinate system. Let (x_1, x_2) be the VINCOS coordinates of node i. i belongs to the North group if $x_1 < x_2$, the South group if $x_1 > x_2$ and to the Equator group if $x_1 = x_2$ (see example in Fig. 1(b)).

4 VINCOS: From Anarchy to Virtual Coordinates

To build a meaningful coordinate system, the nodes need to acquire some consistent knowledge of the network. This approach provides nodes with a novel and fully decentralized way of acquiring that knowledge. The protocol is composed of four consecutive phases summarized in Table 1 and described in a nutshell in the following paragraph. A full description of the protocol is available in [7].

Bootstrap. The first phase consists in identifying a set of initiators (act as landmarks). In short, nodes chosen as initiators are more connected than any of

Table 1. Summary of the four phases of VINCOS

P.	Input	Output
1	*none*	*initiators*: nodes with more neighbors
2	*initiators*: flood their id	*all*: learn score (average distance to initiators) *perimeter bootstrap nodes*: nodes that are for sure on the border
3*a*.	*perimeter bootstrap nodes*: send probes *highest scoring nodes*: relay the probes	*segment definer node*: node that knows a connected border belt with the probes it received
3.*b*	*segment definer nodes*: split the belt	*border-belt nodes*: nodes that own the border-belt
4	*border-belt nodes*: flood their segment number	*all*: learn the smallest distance to each segment (*i.e.* coordinates)

their neighbors therefore representing a local density maximum. During the second phase, each initiator then floods the network. This allows each node to learn its *distance to each initiator*[2]. Nodes then compute a *score* that is the average of all distances to the initiators.

Border belt construction. This phase identifies a belt of nodes on the border. To this end, a set of nodes is in charge of sending *probes*. Probes are assumed to never travel backwards: let node i receive a probe from node j. i chooses the next probe destinator among i's neighbors that are not j's neighbors. In addition, among the eligible neighbors a node always chooses the node with the highest score. This ensures that the probe remains on the border. Eventually, the probe reaches the node that originated it. The probe path thus defines a suitable border belt: it is connected and thin (less than one hop wide). Note that at this point, more than one border could be detected. In this case, the following phases happen concurrently on each detected border. Nodes only adopt coordinates originated from the longer border.

Segment definition. The goal is here to devide the border in d segments. The belt allows border nodes to easily elect one of them as leader during Phase 3.*a*. The leader computes the total length of the border and sets the direction of the border. A node is able to uniquely decide to which of the segments it belongs based on its distance to the leader and the information about the direction (Phase 3.*b*).

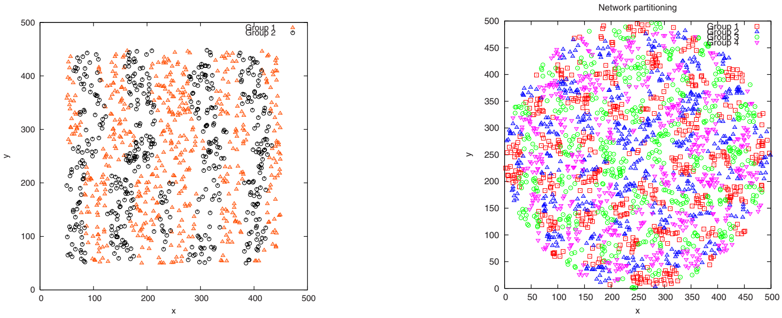
Coordinate definition. Finally, each border node floods the rest of the system (Phase 4), allowing all nodes to get their distance to each border segment. This results in each node i being provided with a set of coordinates $(x_1, \ldots, x_d)$ in the d-dimensional space. Section 5 shows how these coordinates can be used to easily define geometric structuring.

5 From Virtual Coordinates to Geometric Structuring

Geometric structuring can be defined as a logical partitioning of the network into geographical zones for application-dependent purposes. The aim of NetGeoS is

---

[2] In the following, a distance always refers to the minimal distance between two nodes in terms of number of hops.



(a) (b)

Fig. 2. Geometric partitioning examples

to provide in a fully decentralized way and based only on a local observation of the neighborhood, each node with a partition number. Partitions are then used to fulfill specific applications requirements or systems properties, for example. Formally, let $\mathcal{K}$ be the coordinate space (in our application, $\mathcal{K} \in \mathbb{N}^d$), and let p be the number of partitions. Let c_i and p_i be the coordinates and the partition number of node i, respectively. Then a geometric structuring function is a function f s.t. $f : \mathcal{K} \rightarrow \{0, \ldots, p\}$, where $f(c_i) \mapsto p_i$. Let us observe that such a definition allows any node to compute the partition number of any other node whose coordinates are known: *each node has a global foresight of the system layout*.

Let (x_1, x_2) be the VINCOS coordinates of node i. In Section 3.2, the following *line partitioning function* was used to produce the introductory example, where nodes were structured into *North*, *South*, and *Equator* groups ($f : \mathbb{N} * \mathbb{N} \rightarrow \{1, 2, 3\}$):

$$f(x_1, x_2) \mapsto \begin{cases} 1 \ when \ \ x_1 > x_2 \\ 2 \ when \ \ x_1 = x_2 \\ 3 \ when \ x_1 < x_2, \end{cases}$$

where node i belongs to the North group if $x_1 < x_2$, to the South group if $x_1 > x_2$ and to the Equator group if $x_1 = x_2$ (see Fig. 1(b)).

Another useful geographic partitioning is the *target-like partitioning*. This is a straightforward structuring to achieve using $d = 1$. In this structure, each node gets as a partition number, its minimum hop distance to the border ($f : \mathbb{N} \rightarrow \mathbb{N}$): $f(x_1) \rightarrow x_1$. This can be a useful structure, for example, for tracking applications where all the inner rings could be set in sleep mode, the only active partition being the border. Whenever a node from a partition p senses something, it wakes up the partition $p + 1$, so that the network is gradually woken up.

Using the line partitioning, one can also create parallel vertical lines at each j_v hop in the network. In this structure, each node i belongs to the vertical line resulted from ($f : \mathbb{N}^4 \rightarrow \{0, 1\}$): $f(x_1, x_2, x_3, x_4) \mapsto max(x_1, x_3) \mod j_v$ (see Fig. 2(a)). In this way, parallel vertical or horizontal lines can be used to select well distributed nodes in the network to be responsible for performing data aggregation. In addition, a lattice partitioning as shown in Fig. 2(b) can be used to distribute nodes between awake and sleep states for energy consumption management.

6 Performance Evaluation

This section describes the experiments we have conducted to assess both the performance and the accuracy of the proposed approach. The experiments have been done using a discrete event simulator implemented in Java. Note that, as we are mostly interested in the algorithmic evaluation, our simulator deliberately does not model all the details of a realistic MAC protocol. Instead, we considered a simplified MAC layer where neither messages losses, nor collisions and duplications are considered. We argue they will not affect the correctness of the proposed algorithms. This is mainly due to the fact that the well performance of the presented mechanisms are independent of the order of messages' arrival, even if it may cause a longer convergence time. A realistic MAC protocol would have the major effect of introducing arbitrary delays on messages due to messages re-transmission, which will be studied in a future work.

Experimental setup. We implemented the system model described in Section 2. Our simulations involve scenarios where the number of nodes varies from 250 to 2600. The nodes are distributed over a 2-dimensional plane, in an area of 500×500 square units. A node range is simulated as a circle area. Radio ranges from 30 to 50 distance units have been used in the simulations. Nodes broadcast `Hello` messages within their radio range, containing information required in each phase of VINCOS.

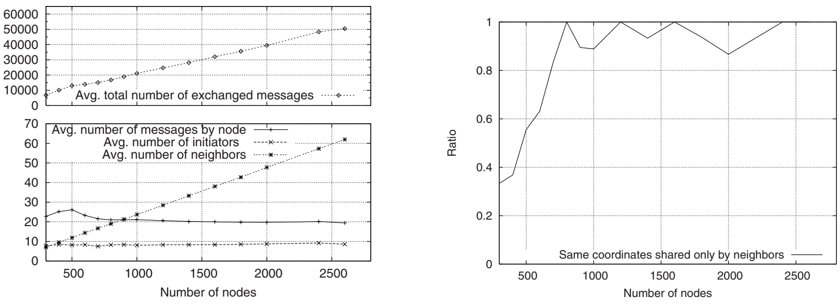
We conducted experiments under various node distributions in the network (*i.e.*, uniform, normal, multi-centered normal distribution) and various border shapes, like topologies with concave shapes (*e.g.* rectangle- and cross-shaped topologies) or that contains a large void in the center (*e.g.* donuts-shaped[3] topologies). Each point of an experimental curve results from 20 independent experiments.

Metrics. Fig. 3(a) presents the average results of 20 experiments as the network size increases from 250 to 2600 nodes, for a 40-unit radio range. Network size and radio range are the two main parameters of the simulations since they impact the average number of neighbors, the system size, and the density (which globally impact initiators detection). Another key parameter is the system shape: the regularity of the border is important to ensure a correct border detection.

Communication Costs. The $O()$ communication cost imposed by each phase during the execution of the VINCOS algorithm has been evaluated in [7] for a square grid shaped network of N nodes. With y and z being respectively the number of initiators and the perimeter bootstrap nodes, and d being the size of the coordinate system, the total cost of VINCOS is : $O((2+y+d)*N+4*z\sqrt{N})$.

---

[3] *I.e.*, circle-shaped topologies in presence of voids.



(a) (b)

Fig. 3. (a) Cost analysis as a function of the network size, for $d = 4$. (b) Ratio of neighbors with the same coordinates.

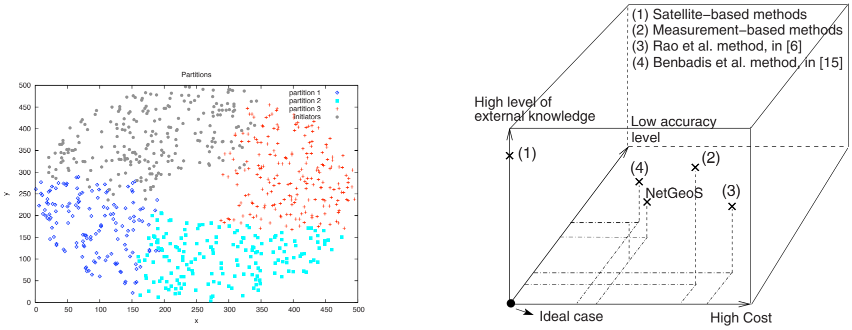
Figure 3(a) confirms the theoretical results by showing that the message communication cost per node for the VINCOS construction is low and independent of the size of the network. It can be observed that for increasing network size and neighbors density, the number of initiators is constant and remains low. In addition, even if the total number of messages exchanged in the network grows nearly linearly with the number of nodes, the average number of messages per node is kept low and presents a slight decrease with the increase of the network size. This can be explained by the fact that inner nodes have a constant number of messages to exchange and that only border-belt nodes exchange additional messages[4]. The proportion of border-belt nodes, however, decreases with the size of the system, which by consequence, decreases the average cost.

Time of convergence. Similarly to communication cost, the theoretical time cost was discussed through the VINCOS extended description available in [7]. The total expected convergence time is $2 + (2\sqrt{2} + 4)\sqrt{N}$ interactions.

Accuracy. Even though we do not focus on the routing capabilities over VINCOS, it is well known in the literature [15] that accurate coordinate systems lead to correct greedy routing. This is highly dependent on the number of nodes which get identical coordinates as well as their respective physical positions. In order to assess the accuracy of VINCOS, and consequently its routing capabilities, we measured the number of nodes in the system sharing the same coordinates. Among those, we considered the following metric at the granularity of an experiment: *"any two nodes having the same coordinate are neighbors"*. Systems that meet this metric are perfectly suitable for greedy routing: a message reaching the destination's coordinates also reaches the final destination. Fig. 3(b) shows the ratio of experiments that respect the considered metric, for different number of nodes and 50-unit radio range. The figure shows that for some node densities, the considered metric is nearly always respected. For example, consider the point of the curve $x = 1500$ nodes, it

---

[4] This does not happen in [6], since perimeter nodes flood all the network to discover each other.



(a) (b)

Fig. 4. (a) VINCOS's segment definition for a topology with a large void in the center. (b) VINCOS w.r.t. cost, accuracy, and external knowledge.

means that in 96% of the experiments, all nodes sharing the same coordinates are neighbors. For networks with a small number of nodes (less then 800 in the figure), the low ratio is explained by the low density in the system.

Different shapes and node distribution. As previously discussed, the shape of the system is a parameter that can impact the coordinate system correctness. In this section, we explore this issue by showing the resulting segment definition of VINCOS under two specific topologies. Fig. 4(a) depicts the result for a donut-shaped topology with $d = 4$ in a 2000-node network. We observe that even in the presence of voids (*i.e.*, regions inside the network that do not contain any nodes), VINCOS performs well and correctly defines the required $d = 4$ partitions. The presence of the void leads to the detection of multiple border-belts: the biggest one is used for the virtual coordinate definition.

Fig. 2(a) shows a geometric structuring for a rectangle-shaped topology of 2000 nodes and 40-unit radio range. The obtained geometric structuring shows that VINCOS performs well in a topology with sharp angles too (this will be addressed in future). Similarly, Fig. 1(b) shows the partitioning of a 900-node network and 30-unit radio range, where nodes were scattered following a normal distribution. The topology presents three hot spots of different densities. By applying a "line" predicate, the figure shows the correct partitioning of the network in *North*, *South*, and *Equator* groups.

7 Related Work on Positioning Systems and Conclusion

Fig. 4(b) compares the features of related works in the literature compared to the VINCOS coordinate assignment approach. It shows their differences with respect to cost (message complexity), coordinate assignment accuracy (in terms of number of nodes sharing the same coordinate), and initial knowledge. The closest to the origin, the better is the compromise between these three criteria.

The importance of a coordinate system for autonomous systems is demonstrated by the vast literature on the topic [1,2,4,5,6,8,10,11,12,13,15]. Regarding absolute positioning systems, the literature is dominated by the satellite-based methods, like GPS and Galileo. Equipping all entities with a satellite receiver constitutes the best way to provide them with a very high level of external knowledge, resulting in a very accurate coordinate assignment (point (1) in Fig 4(b)). Unfortunately, this approach suffers from important drawbacks when applied to sensor networks: it is expensive, energy-inefficient, and cannot work when the sensors are deployed in a zone that is out of satellites receiver scopes.

This problem can be solved by equipping only a few entities (i.e., position-aware landmarks) with a satellite receiver, and let the other entities infer their position from connectivity information, or signal strength measurements [1,4,9,10,11,12,13]. While allowing systems to be designed with fewer external knowledge (*i.e.*, only few entities know their position), such hybrid approaches are costly and are not accurate from a coordinate assignment point of view[5] (point (2) in Fig. 4(b)). Differently, solutions that are based only on connectivity [1,9,10] are algorithmically simpler. Nevertheless, they have a high communication cost, being their coordinate assignment accuracy strongly dependent of the landmarks density and good positioning [8,15] (point (4) in Fig. 4(b)). A more attractive approach is then, when no position-aware referential is used. Recent researches propose solutions that result in *virtual coordinates* being assigned to nodes, instead of geographic ones [4,6] (point (3) in Fig. 4(b)).

The coordinate assignment procedure of VINCOS is similar to the approaches described in [6,8] with respect to initial knowledge assumptions. In these approaches, perimeter nodes or landmarks are considered to be known, situated on the border of the network topology, or are determined based on the distance to a well centered node. In particular, compared to [6], VINCOS exploits border knowledge in a much different way. Moreover, our segment definition's result gives the perfect landmarks positioning for the coordinated assignment mechanism described in [8]. In addition, the upper layer *NetGeoS* relies also on simple design principles. Its versatility dimension makes it relevant for defining a geometric partitioning on top of which upper layer scalable services can be implemented. Simulation results attest of the accuracy of both VINCOS and NetGeoS. Future work includes considering more realistic MAC layer models and non-convex irregular-shaped topologies.

References

1. Bischoff, R., Wattenhoffer, R.: Analyzing Connectivity-Based, Multi-hop Positioning System. In: Proc. of IEEE Percom, pp. 165–176 (March 2004)
2. He, T., Huang, C., Blum, B., Stankovic, J., Abdelzaher, T.: Range-free Localization Schemes in Large Scale Sensor Networks. ACM Mobicom, 81–95 (2003)
3. Ben Hamida, E., Chelius, G.: Analytical Evaluation of Virtual Infrastructures for Data Dissemination in Wireless Sensor Networks with Mobile Sink. In: SANET 2007 (September 2007)

---

[5] For example, the signal measurement-based solutions [12] are less efficient in indoor or underground environments besides being algorithmically costless.

4. Moscibroda, T., O'Dell, R., Wattenhoffer, M.R.: Virtual Coordinates for Ad Hoc and Sensor Networks. In: ACM Workshop DIAL-POMC, pp. 8–16 (2004)
5. Nagpal, R., Shrobe, H., Bachrach, J.: Organizing a global coordinate system from local information on an ad hoc sensor network. In: Zhao, F., Guibas, L.J. (eds.) IPSN 2003. LNCS, vol. 2634, pp. 333–348. Springer, Heidelberg (2003)
6. Rao, A., Ratnasamy, S., Papadimitriou, C., Shenker, S., Stoica, I.: Geographic Routing without Location Information. In: Mobicom, pp. 96–108. ACM Press, New York (2003)
7. Kermarrec, A.-M., Mostéfaoui, A., Raynal, M., Trédan, G., Viana, A.C.: Large-scale networked systems: from anarchy to geometric self-structuring. Tech Report #1876, 19 pages, IRISA, Université de Rennes (December 2007)
8. Benbadis, F., Obraczka, K., Cortès, J., Brandwajn, A.: Exploring landmark placement strategies for self-organization in wireless sensor networks. Eurasip Journal on Advances in Signal Processing (to appear, 2008)
9. Benbadis, F., Friedman, T., Amorim, M.D., Fdida, S.: GPS-free-free positioning system for sensor networks. In: Proc. of WOCN, pp. 541–545 (April 2005)
10. Caruso, A., Chessa, S., De, S., Urpi, A.: GPS free coordinate assignment and routing in wireless sensor netwoks. Infocom, 150–160 (2005)
11. Shang, Y., Ruml, W., Zhang, Y., Fromherz, M.: Localization from mere connectivity. In: Proc. of ACM Mobihoc, pp. 201–212 (June 2003)
12. Gustafsson, F., Gunnarsson, F.: Positioning using time-difference of arrival measurements. In: Proc. of ICASSP, vol. 6, pp. VI-553–556 (April 2003)
13. Doherty, L., Pister, K.S.J., El, G.L.: Convex position estimation in wireless sensor networks. In: Proc. of IEEE Infocom, pp. 1655–1663 (April 2001)
14. Dohler, M., Watteyne, T., Barthel, D., Valois, F., Lu, J.-L.: Kumar's, Zipf's and Other Laws: How to Structure an Optimum Large-Scale Wireless (Sensor) Network? In: 13th European Wireless Conference (April 2007)
15. Benbadis, F., Puig, J.-J., Amorim, M.D., Chaudet, C., Friedman, T., Simplot-Ryl, D.: JUMP: Enhancing hop-count positioning in sensor network using multiple coordinates. Ad Hoc and Sensor Wireless Networks Journal (to appear, 2008)
16. Pemmaraju, S.V., Pirwani, I.A.: Good quality virtual realization of unit ball graphs. In: Arge, L., Hoffmann, M., Welzl, E. (eds.) ESA 2007. LNCS, vol. 4698, pp. 311–322. Springer, Heidelberg (2007)