Ad Hoc Networks 46 (2016) 37–47

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

A distributed algorithm for *Ad-hoc* network partitioning based on Voronoi Tessellation

A. Pietrabissa[a,*], F. Liberati[a,b], G. Oddi[a]

[a] *Department Of Computer, Control and Management Engineering "Antonio Ruberti", Università degli Studi di Roma "La Sapienza", Via Ariosto 25, 00162 Rome, Italy*

[b] *Università degli Studi eCampus, Via Isimbardi, 10, 22060 Novedrate, CO, Italy*

ARTICLE INFO

Article history:
Received 2 July 2015
Revised 7 March 2016
Accepted 18 March 2016
Available online 31 March 2016

Keywords:
Voronoi partitioning
Wireless sensor networks
Energy-saving algorithms

ABSTRACT

This paper presents a data sink node election algorithm for multi-hop Wireless Sensor Networks (WSNs) with multiple data sink nodes. For energy-saving considerations, these nodes should be evenly (spatially) distributed on the network area. To achieve this objective, it proposed a distributed and iterative algorithm, which periodically re-assigns the data sink roles to selected WSN nodes. The main innovation of the algorithm is that, even if it does not need to explicitly compute the Voronoi partition of the WSN at each iteration, it eventually partitions the network according to a Centroidal Voronoid Tessellation, which leads to a spatially well-balanced distribution of the data sink nodes. Analytical proofs as well as simulation results validate the approach.

1. Introduction

Management of Wireless Sensor Networks (WSNs) is a widely researched topic in the literature, since an efficient operation of WSNs is relevant to many applications, e.g., just to mention a few, process control monitoring [31], critical infrastructure monitoring [26], health care monitoring [21], environmental sensing [19], air quality monitoring [30]. WSNs are basically composed by sensor nodes and data sink (DS) nodes; the sensor nodes, spread over the monitored area, collect the measures of interest and send them to one or more DS nodes, which elaborate the received measures and convey the elaborated data to a remote data center. Usually, the sensor nodes are directly connected to the DS nodes (one-hop communication), but the application of WSNs with multiple DS nodes and supporting multi-hop communications is becoming more and more frequent: when the area to monitor is wide, multi-hop communications are used to extend the network coverage and each DS node collects data from the sensor nodes nearby, i.e., from the sensor nodes in its cluster (see [22] and references therein).

Two of the main problems that a WSN has to face are the energy autonomy of the nodes and, especially in the case that the WSN is deployed in a harsh environment, its tolerance to node failures. The DS nodes are likely to suffer from faster energy depletion with respect to the other nodes, since they have to collect all the data from the surrounding sensor nodes and to perform scheduling and data fusion tasks [27]. Therefore, in many applications it is crucial that the role of DS can be played by any nodes, so that, if a DS node is becoming depleted or it is experiencing a failure, another node can be selected as the new DS node; this procedure is referred to as *data sink election*, and the fact that the role of DS is exchanged between two nodes is referred to as *data sink migration*. Energy and fault tolerance considerations also favor the deployment of multi-sink multi-hop WSNs. In fact, the presence of multiple DS nodes shortens the paths between each sensor node and the closer DS node, whereas the impact of the additional communications due to the fact that multiple DS nodes have to upload their data to the data center is alleviated by data fusion algorithms, which reduce the size of the exchanged data. Moreover, if multiple DS nodes are deployed, the fault of a single DS node does not imply that data of all the sensors are lost.

The problem dealt with in this paper is the data sink node election problem in WSNs and includes the general case of multi-hop and multi-sink networks[1]. In multi-sink WSNs, the DS migration problem has also to consider that the spatial distribution of the DS nodes over the mission area impacts on the WSN performances.

* Corresponding author. Tel.: +390677274040; fax: +390677274033.
E-mail address: pietrabissa@dis.uniroma1.it, antonio.pietrabissa@gmail.com (A. Pietrabissa).

[1] Even if this paper focuses on the DS election, it is necessary to underline that DS migration requires the 'old' DS nodes to send the collected data to the 'new' DS nodes: the effectiveness of the DS migration strategy is then dependent on the implemented data fusion algorithms, which reduce the size of the exchanged data, and on the frequency with which the DS nodes upload their data to the remote centre, which reduces the amount of data that each DS node has to store.

http://dx.doi.org/10.1016/j.adhoc.2016.03.008
1570-8705/

Then, the problem of selecting the best new DS node to migrate to can be formulated as a network partitioning problem. The proposed algorithm, based on Voronoi partitioning concepts, is performed on-line and converges to a balanced mission area partitioning known in the literature as Centroidal Voronoi Tessellation (CVT).

The research presented in this paper has been performed within the European project SWIPE, aimed at developing a WSN suitable for planetary exploration [23], which has to face (to the highest degree) the problems outlined above, and which, therefore, highly benefits from energy saving and fault tolerant approaches.

The paper is organized as follows: Section 2 presents the state of the art the innovations of the proposed algorithm, Section 3 provides the basic Voronoi partitioning concepts; Section 4 presents the algorithm, the convergence proof and some implementation details; Section 5 shows some simulation results; Section 6 draws the conclusions.

2. State of the art and paper contributions

2.1. Data sinks election and mission space partitioning

Explicit data sinks election algorithms have not been studied intensively in the literature. This is mainly due to the fact that researchers have focused their attention to networks composed by only one DS. In [4], multiple DS nodes are used to optimize the energy consumption of a single-hop sensor network, where the DS nodes are deployed along the periphery of the sensing area. An optimization problem is built and solved off-line to decide the best placement for the DS nodes, in order to minimize the overall transmission power of the sensor nodes. A genetic algorithm optimization is performed in the case of two DS nodes, whereas a heuristic algorithm is used in the case of three DS nodes. Differently, the algorithm proposed in this paper is applicable to multi-hop WSNs as well, and finds on-line a balanced placement of the DS nodes by means of an iterative approach which migrates the DS role among the WSN nodes.

A problem similar to the data sink election and extensively studied in the literature is the Cluster Head (CH) election problem. CH election problems refer to WSNs in which nodes are grouped in clusters, each managed by a CH node. In clustered WSNs, the objective is typically that of efficiently conveying the information (e.g., sensed data) from each node to one or more gateway nodes (GWs) serving as collection points in communication with a remote control center or unit. That is achieved through a combination of intra-cluster (i.e., from a node to the corresponding CH) and inter-cluster communication (i.e., multi-hop CH-CH communication). In this context, the problem of intelligent CH election has been addressed by several works in literature, with the main objective of achieving a balanced configuration of the clusters (in terms of energy consumption, workload distribution, etc.). Moreover, similarly to what done in this paper, it has been widely recognized in literature that the CH role could be periodically rotated among the network nodes, in order to balance and to share in time the additional communication and processing efforts required by the CH role. The key difference between the CH and the DS election problems is that the DS node election problem does not require the DS nodes to form a connected subset of nodes, whereas CHs have to form a connected subnet in order to fully support inter-cluster communication. A brief review of relevant papers dealing with CH election is reported in the following. In LEACH [12], few nodes are randomly selected as CHs. The CH role is rotated periodically to distribute the effort among the nodes in the network. Each CH compresses the data coming from the nodes of the cluster and sends an aggregated packet to the GW. Dealing with the CH election process, at each turn a fraction p of nodes elect themselves as CHs, in the following way. Each node i chooses a random number r between 0 and 1: if $r < T(i)$, then the node becomes a CH for the current round, where $T(i)$ is computed based on (i) the desired fraction p of nodes to become a CH, (ii) the current round and, (iii) the set G of nodes that have not been selected as a CH in the last $1/p$ rounds. The election rule is the following one:

$$T(i) = \begin{cases} \frac{p}{1-p\cdot mod(r,1/p)} & \text{if } i \in G \\ 0, & \text{otherwise} \end{cases}, \quad (1)$$

where $mod(\cdot,\cdot)$ denotes the modulo operator. An extension of LEACH is HEED [29]. It introduces limits on the communication range and cost terms in the intra-cluster communication. Moreover, the probability to become a CH in HEED also depends on the residual battery of the nodes. The main problem of LEACH-based approaches is that, since the election is random, without position or distance considerations, the elected CHs are not necessarily well-distributed over the network. In this sense, EECS [28] is a LEACH-like algorithm for single-hop WSN, which introduces a novel distance-based method to balance load among CHs. EECS is divided in two main phases: the CH election phase and the cluster formation phase. In the CH election phase, WSN nodes become CH candidate nodes according to a certain probability. Then, each CH candidate broadcasts its candidacy to nodes falling within a certain radius R; each candidate actually assumes the CH role only if no other CH candidate he is aware of (i.e. falling within R) has higher battery level (thus ensuring an energy-fair election). In the cluster formation phase, the remaining nodes are assigned to the CHs to form the clusters. In EECS, both the distances of the nodes from their CH and from the GW are considered; in this way, clusters which are far from the GW have less nodes compared to clusters closer to the GW (*unequal clustering*). As a result, in [28] it is shown that network lifetime, defined as the time until one of the nodes runs out of energy, is significantly prolonged compared to LEACH and HEED. In [10], a *centralized* extension of the LEACH algorithm is proposed, called partition-based LEACH (pLEACH), in which each node sends to the sink node information on its location and battery level, and the sink node then partitions the network into a given number of clusters and, for each cluster, selects as CH the node with the highest energy level. The drawback of this algorithm lies in the communication burden implied by its centralized nature, which may be unsustainable in case of large WSNs. Advancements of HEED have been proposed as well, as for instance UHEED (G. [5]), which is a clustering modification of HEED aimed at optimizing HEED in multi-hop WSNs. To this end, UHEED modifies HEED formula so that clusters close to the GW are smaller than the ones far from it (the opposite behavior of EECS, which is designed for single-hop communication). Advanced CH election strategies are continuously being developed based on the basic concepts explained above. The interested reader may find additional surveys on WSN clustering in [1,24].

The algorithm proposed in this paper takes some ideas from the CH election strategies described above, namely the round-based approach and the strategy of network partitioning, and relies on graph-theory concepts, namely, on Voronoi partitioning [7]. Some works recently appeared in the literature propose the Voronoi partition as the fundamental structure for WSN clustering, mainly because: (i) if the number of Voronoi regions is appropriately selected, data inside the same region tend to be homogeneous and thus can be fused, (ii) in single-hop networks, Voronoi partitioning is functional to transmission power control (i.e., the nodes adjust their transmission power in function of the diameter of the Voronoi cell).

A fundamental problem in this chapter of literature regards the computation of the Voronoi partition of a WSN. The paper [2] presents a distributed approach for computing the Voronoi par-

Table 1
Paper contributions.

Paper	Main features	Key differences/improvements brought by the proposed approach
[2]	Distributed computation of the Voronoi partitioning of the WSN sensor field by means of geometrical techniques.	It is not required to compute explicitly the Voronoi partitions.
[25]	The CHs are the Voronoi generator points and they move across the WSN under the action of virtual forces in order to keep the energy consumption balanced.	WSN nodes are fixed and the data sink role is played by different WSN nodes in time.
[17]	Centralized algorithm. Voronoi partition is computed by considering a distance metric including both nodes' position and energy level.	Distributed algorithm. A similar metric is considered, but it is used to determine the probability that a node becomes a candidate to the data sink role, not to compute the Voronoi partition.
[14]	Heterogeneous WSN with fixed CHs considered. The cluster region extension is varied in time to balance CHs' energy consumption.	Homogeneous WNS considered. Both the DS roles and the cluster regions deriving from the DS node locations are varied in time.

tition of the WSN sensor field (i.e., of the area monitored by the WSN) based on geometrical considerations and aimed at minimizing the energy consumption. In (J. [6]), the authors propose a distributed algorithm for achieving energy-aware Voronoi partitioning of the WSN. The approach consists in an off-line partitioning of the network, based on the knowledge about the nodes' deployment. In [25], a Voronoi partitioning-based clustering algorithm for WSNs is presented, in which mobile CH nodes move across the sensor field under the action of so-called virtual forces, computed so as to minimize the variance of the load (i.e., the number of nodes) of each cluster. The clustering algorithm is energy-aware since the virtual forces depend also on the energy level of CHs. The algorithm proposed in [25] requires the explicit computation of the Voronoi diagrams after each iteration. Recently, [17] presented a fuzzy C-means clustering algorithm in which membership functions are calculated based on the Voronoi partitioning of the WSN. To compute the Voronoi partition, a distance metric is used which takes into account both Euclidean distance and the residual energy of the nodes. The algorithm is centralized, since clustering is computed by the (single) DS node.

Reference [13] proposes an energy-aware distributed topology control for the optimization of the network lifetime. The algorithm assumes a heterogeneous WSN (i.e., CHs have different hardware and software properties than regular nodes). CHs are then assumed to be fixed and the objective is to control the cluster regions in order to preserve CHs' energy. That is achieved in [14] by iteratively varying the extension of the cluster regions to balance CHs' energy consumption.

2.2. Paper contributions

Summarizing, the present paper proposes a new approach to elect the DS nodes in such a way that the network is partitioned according to the so-called Centroidal Voronoi Tessellation. Distinctive characteristics of the algorithm are that it is decentralized, it does not need to know the node positions at the deployment phase, it does not need to explicitly compute the Voronoi partition and it is performed on-line by means of migrations of the DS nodes.

The main contributions to Voronoi-based WSN clustering reviewed in Section 2.1 are summarized in Table 1, as well as the key differences with respect to the approach proposed in this paper.

3. Preliminaries on Voronoi partitioning

Mission space partitioning relies on the definition of Voronoi partition. Let us consider a Euclidean domain $\mathcal{A} \in \mathbb{R}^2$, and n points indexed by the set $\mathcal{S} = \{1, 2, .., n\}$, with positions $x(s) \in \mathcal{A}, s \in \mathcal{S}$. The Voronoi regions are therefore defined as:

$$\mathcal{P}(s) = \{p \in \mathcal{A}: \|p - x(s)\| \leq \|p - x(s')\|, \forall s' \in \mathcal{S}\}, \ s \in \mathcal{S}, \tag{2}$$

where $\|p - x(s)\|$ is the Euclidean distance between p and $x(s)$. Eq. (2) states that each point $p \in \mathcal{A}$ belongs to the Voronoi region $\mathcal{P}(s)$ such that the distance between p and $x(s)$ is the minimum among all the distances between p and the points in $\mathcal{S}$. The Voronoi regions $\mathcal{P}(s)$ are such that $\cup_{s\in\mathcal{S}}\mathcal{P}(s) = \mathcal{A}$ and $\cap_{s\in\mathcal{S}}\mathcal{P}(s) = \emptyset$, i.e., they form a partition of the mission space $\mathcal{A}$, referred to as Voronoi partition or tassellation and denoted with $\mathcal{P}(\mathcal{S}) = \{\mathcal{P}(s)\}_{s\in\mathcal{S}}$. The points $x(s)$ are the *generating points* of the Voronoi partition.

Depending on the position of the generating points in the mission space $\mathcal{A}$, specific partitions can be generated. In particular, we are interested in the generalized Voronoi Centroidal Tassellation (CVT), whose generating points are the centers of mass of the Voronoi regions, and which is regarded as an optimal partition corresponding to an optimal distribution of generators [7]:

Definition 1. The generalized centroid of a set $\mathcal{P}(s)$ with respect to a density function φ is $m = \arg\min_{p\in\mathbb{R}^2} \int_{\mathcal{P}(s)} p - q^2\varphi(q)dq$.

In Definition 1, the density function $\varphi : \mathcal{A} \to \mathbb{R}_{\geq 0}$ is an absolutely continuous spatial distribution, with bounded and convex support within $\mathcal{A}$ (i.e., there is a bounded and convex subset $\mathcal{Q}$ of $\mathcal{A}$ such that $\varphi(q) > 0$ if $q \in \mathcal{Q}$, and $\varphi(q) = 0$ if $q \in \mathcal{A} \setminus \mathcal{Q}$).

The resulting Voronoi partition is the generalized CVT:

Definition 2. A Voronoi Tessellation $\mathcal{P}(\mathcal{S}) = \{\mathcal{P}(s)\}_{s\in\mathcal{S}}$ of a set $\mathcal{A}$ is called a Centroidal Voronoi Tessellation with respect to the density function $\varphi(q), q \in \mathcal{A}$, if each generator $s \in \mathcal{S}$ is equal to the generalized centroid of its partition $\mathcal{P}(s)$ with respect to φ.

4. Data sink election algorithm

Section 4.1 describes the data sink node election algorithm; Section 4.2 presents the proof of convergence to the CVT; Section 4.3 describes some practical implementation details.

4.1. Algorithm description

Let the mission space be a finite, convex Euclidean domain $\mathcal{A} \subset \mathbb{R}^2$. The sensor network is deployed on the area $\mathcal{A}$, and the sensor nodes connections build a communication graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{1, 2, \ldots, n\}$ is the set of the n vertices or nodes of the network, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges or links between node couples: $(i, j) \in \mathcal{E}$ if nodes i and j are connected. We assume that the graph is connected (i.e., there exists at least a path between every node couple), and that the node position on the mission space is not time-varying (as common in sensor networks). The position of node i is defined by the mapping function $x : \mathcal{V} \to \mathcal{A}$. Let $\mathcal{X} = \{x(i)\}_{i\in\mathcal{V}}$ be the set of node positions. The degree of node $i \in \mathcal{V}$, i.e., the number of its neighbor nodes, is denoted with $d(i)$.

Let n^{ds} be the number of DS nodes to be set-up in the sensor network. The value of n^{ds} is decided at planning level, based, e.g., on fault tolerance considerations. The algorithm relies on control agents which, at any time of the mission, are associated to each DS node. The set of control agents is denoted with $\mathcal{S} = \{1, 2, \ldots, n^{ds}\}$. We consider that the DS functionality is not permanently associated to a given node: at each time t, the DS control agents are associated to a given set of n^{ds} nodes. We then define a time-varying mapping function π_t from the control agent set $\mathcal{S}$ to the node set $\mathcal{V}$. This mapping function determines the set of DS nodes at time t: $\pi_t(s)$ is the network node which, at time t, is associated to the DS agent $s \in \mathcal{S}$. The set of the n^{ds} DS nodes is denoted with $\mathcal{V}_t^{ds} = \{\pi_t(s)\}_{s\in\mathcal{S}} \subseteq \mathcal{V}$. At different time instants, the set of the n^{ds} DS nodes, is, generally, different; at time t, let $\mathcal{D}_t = \{x(\pi_t(s))\}_{s\in\mathcal{S}} \subset \mathcal{A}$ be the set of the positions of the DS nodes.

In sensor networks, the DS nodes are likely to suffer from faster energy depletion, due to the fact that they have to collect and process all the data sent by the sensor nodes. Then, there is the need of changing the DS nodes, i.e., of varying the mapping π_t. The role of the data sink election algorithm is then to dynamically map the set of DS nodes $\mathcal{S}$ onto the subset of nodes $\mathcal{V}_t^{ds}$, i.e., to compute the mapping function π_t, at any time t. The proposed algorithm is distributed, since each DS agent takes the decisions independently from the other agents. However, the agents communicate and, in particular, the assumption is that, each time a migration of a given DS agent occurs, the agent in question communicates to the other agents the position of its new DS node.

To choose a sensor node as a candidate new DS, we consider three normalized metrics:

1. *Uniform metric*: all the nodes have the same weight: $w(i) = 1,\ i \in \mathcal{V}$.
2. *Connectivity degree metric*: each node i computes its own weight as $w(i) = \frac{d(i)}{d_{MAX}} \in (0, 1]$, where d_{MAX} is a maximum metric value selected for normalization purposes; for instance, d_{MAX} can be selected as the maximum degree of the graph.
3. *Leftover energy metric*: each node i computes its own weight as $w(i) = \frac{e_t(i)}{e_{MAX}} \in [0, 1]$, where $e_t(i)$ is the leftover energy of node i at time t, and e_{MAX} is a maximum metric value selected for normalization purposes; for instance, e_{MAX} can be selected as the maximum node energy. Note that this metric is time-varying due to the progressive energy depletion of nodes.

Different metrics may be considered; for instance, in [11] the metrics 2 and 3 proposed above as well as a linear combination of those are evaluated in the framework of cluster-head election algorithms.

At given time instants, each node $i \in \mathcal{V}$ decides whether to be a candidate DS node or not, based on its metric $w(i)$ (the larger $w(i)$, the higher the probability). This paper does not specify how the decision time of each node is determined; the algorithm can use any of the analogous mechanisms used in the similar cluster-head election problems, e.g., periodic, random with a Poisson time distribution, decided by a back-off algorithm. In the following, to say that a node decides to be a candidate node, we will write that the candidate node *appears* (for similarity to the target appearance in the routing problems).

The DS nodes election algorithm is described hereafter. The algorithm is inspired by the routing algorithm for multiple vehicles developed in [3].

For each DS agent $s \in \mathcal{S}$, let $\mathcal{B}_t(s)$ be the set of the $k+1$ positions of the nodes which played the role of DS s up to time t:

$$\mathcal{B}_t(s) = \{x(\pi_{t_{s,0}}(s)), x(\pi_{t_{s,1}}(s)), \ldots, x(\pi_{t_{s,k}}(s)) | 0 = t_{s,0} < t_{s,1} < \ldots < t_{s,k} \leq t\} \subset \mathcal{A},$$

where $\pi_{t_{s,0}}(s)$ is the position of the initial DS node, with $t_{s,0} = 0$, $t_{s,j}$ denotes the time when the j^{th} migration of the DS node s occurs, and $t_{s,k} \leq t$ is the time of last migration, i.e., k is the number of migrations of DS agent s up to time t. Let $\mathcal{T}_{s,k} = \{t_{s,i}\}_{i=1,\ldots,k}$ denote the set of the first k^{th} migration instants of DS agent s.

Let $i \in \mathcal{V}$ be the new candidate DS node appearing at time t'. The DS agents decide which DS node has to migrate to the new node based on the position of the current candidate nodes and on the sets of the past positions, collected in the sets $\mathcal{B}_t(s),\ s \in \mathcal{S}$. Let $y_t(s) \in \mathcal{A}$ denote the *reference point* of the set $\mathcal{B}_t(s), s \in \mathcal{S}$, defined as the point which minimizes the average squared distance to the past positions of the DS agent s up to time t:

$$y_t(s) = \arg\min_{p\in\mathbb{R}^2} \frac{1}{k+1} \sum_{q\in\mathcal{B}_t(s)} \|p - q\|^2, s \in \mathcal{S}. \quad (3)$$

Given that the functions $y_t(s) : \mathcal{S} \to \mathbb{R}^2$ defined by Eq. (3) are strictly convex in $\mathbb{R}^2$, there is a unique reference point $y_t(s)$ for each $s \in \mathcal{S}$, and, since $\mathcal{B}_t(s) \subset \mathcal{A}$, with $\mathcal{A}$ convex, we have also that $y_t(s) \in \mathcal{A}$. The DS election algorithm is based on the Voronoi partition generated by the reference points (3) of the DS agents. Let $\mathcal{P}_t(s) \subset \mathcal{A}$ denote the Voronoi region generated by the reference point $y_t(s)$ of the agent s at time t. At time t, the candidate node i belongs (necessarily) to one of the Voronoi regions, say region $\mathcal{P}_t(s')$. Then, it becomes the new DS node of the agent s', i.e., the DS agent s' migrates to node i.

4.2. Convergence of the reference points to the CVT

We are interested at showing that the sequences of the reference points converge to the generating points of the CVT, i.e., to the generalized centroids, as the set of migration instants $\mathcal{T}_{s,k} = \{t_{s,i}\}_{i=1,\ldots,k}$ of each DS agent $s \in \mathcal{S}$ grows for $k \to \infty$.

The generalized centroids, in the discrete spatial distribution case, are defined as follows:

Definition 3. The generalized centroid of a region $\mathcal{P} \in \mathbb{R}^2$ with respect to a discrete density function φ defined in $\mathbb{R}^2$ with support given by the discrete set $\mathcal{X}_\mathcal{P} \subset \mathcal{P}$ is

$$m_{\mathcal{X}_\mathcal{P}} = \arg\min_{p\in\mathbb{R}^2} \sum_{q\in\mathcal{P}} \|p - q\|^2 \varphi(q) = \arg\min_{p\in\mathbb{R}^2} \sum_{q\in\mathcal{X}_\mathcal{P}} \|p - q\|^2 \varphi(q). \quad (4)$$

Note that a given Voronoi partition $\mathcal{P}(\mathcal{S}) = \{\mathcal{P}(s)\}_{s\in\mathcal{S}}$ induces an associated graph node partition, denoted with $\mathcal{V}_\mathcal{P}(\mathcal{S}) = \{\mathcal{V}_\mathcal{P}(s)\}_{s\in\mathcal{S}}$, and defined as:

$$\mathcal{V}_\mathcal{P}(s) = \{i \in \mathcal{V} | x(i) \in \mathcal{P}(s)\},\ \forall s \in \mathcal{S}. \quad (5)$$

Then, we will denote the generalized centroid of a Voronoi region $\mathcal{P}(s)$ with respect to a discrete density function φ with support given by the discrete set of the positions of the nodes in $\mathcal{V}_\mathcal{P}(s)$ as

$$m_{\mathcal{V}_\mathcal{P}(s)} = \arg\min_{p\in\mathbb{R}^2} \sum_{q\in\mathcal{P}(s)} \|p - q\|^2 \varphi(q) = \arg\min_{p\in\mathbb{R}^2} \sum_{i\in\mathcal{V}_\mathcal{P}(s)} \|p - q\|^2 \varphi(x(i)). \quad (6)$$

As shown in [3], since the cardinality of $\mathcal{V}$ is finite, the sequences of reference points $y_{t_{s,k}}(s)$ of each DS converge to well-defined limit generation points, denoted as $\hat{y}(s) = \lim_{k\to\infty} y_{t_{s,k}}(s), s \in \mathcal{S}$; the corresponding limit Voronoi regions are denoted as $\hat{\mathcal{P}}(s)$. The following property holds:

Property 1. [3]. The sequence of the Voronoi partitions $\{\mathcal{P}_{t_{s,k}}(\mathcal{S})\}_{k=0,1,2,\ldots}$ generated by the sequences of reference points

$\{y_{t_{s,k}}(s)\}_{k=0,1,2,\ldots}$, $s \in \mathcal{S}$, converges, almost surely, to the Voronoi partition $\hat{\mathcal{P}}(\mathcal{S})$ generated by the limit reference points $\hat{y}(s)$, $s \in \mathcal{S}$.

Thanks to Property 1, to study the steady-state properties of the algorithm we just need to check the properties of the limit Voronoi partition $\hat{\mathcal{P}}(\mathcal{S}) = \{\hat{\mathcal{P}}(s)\}_{s\in\mathcal{S}}$. Also, from Property 1, it follows that, as $k \to \infty$, the network graph is partitioned in n^{ds} subsets of nodes $\mathcal{V}_{\hat{\mathcal{P}}}(s), s \in \mathcal{S}$, defined as in Eq. (5), and that the node set partition $\mathcal{V}_{t_{s,k}}(\mathcal{S}) = \{\mathcal{V}_{t_{s,k}}(s)\}_{s\in\mathcal{S}}$ converges, almost surely, to the limit node set partition $\mathcal{V}_{\hat{\mathcal{P}}}(\mathcal{S}) = \{\mathcal{V}_{\hat{\mathcal{P}}}(s)\}_{s\in\mathcal{S}}$.

Theorem 1. *Under the assumption that the probability distribution of the appearances of the candidate nodes is stationary, the limit reference points $\hat{y}(s)$ of the sequence of reference points (3) coincide with the generalized centroids of the limit Voronoi regions $\hat{\mathcal{P}}(s)$, $\forall s \in \mathcal{S}$.*

Proof. Let φ be the distribution of the candidate node appearances. The distribution, stationary by assumption, is defined over the discrete set of node positions $\mathcal{X}$.

We are interested in the limit generating points of the limit Voronoi partition, which, given the update rule of Eq. (3), are defined as

$$\hat{y}(s) := lim_{k\to\infty} y_{t_{s,k}}(s) = lim_{k\to\infty}\left(\arg\min_{p\in\mathbb{R}^2} \frac{1}{k+1}\sum_{q\in\mathcal{B}_{t_{s,\ k}}(s)} \|p-q\|^2\right), \quad s \in \mathcal{S}. \tag{7}$$

By Property 1, $\mathcal{V}_{t_{s,k}}(s)$ converges to the limit set $\mathcal{V}_{\hat{\mathcal{P}}}(s)$ as k grows. It follows that, since each agent s can migrate only to nodes belonging to its own Voronoi region, as $k \to \infty$, the new DS nodes selected by agent s are all in $\mathcal{V}_{\hat{\mathcal{P}}}(s)$, i.e., the new elements $\pi_{t_{s,k}}(s)$ of the set $\mathcal{B}_{t_{s,k}}(s)$ are all such that $\pi_{t_{s,k}}(s) \in \mathcal{V}_{\hat{\mathcal{P}}}(s)$. Thus, eventually, as $k \to \infty$, the probability that a given node $i \in \mathcal{V}_{\hat{\mathcal{P}}}(s)$ is the new DS node s is described by a density function, denoted with $\hat{\varphi}^s$, with support given by the discrete set $\{x(i)\}_{i\in\mathcal{V}_{\hat{\mathcal{P}}}(s)}$:

$$\hat{\varphi}^s(q) = \varphi\left(q|q \in \hat{\mathcal{P}}(s)\right) = \begin{cases} \frac{1}{\hat{c}^s}\varphi(x(i)) & \text{if } q = x(i) \text{ and } i \in \mathcal{V}_{\hat{\mathcal{P}}}(s) \\ 0 & \text{otherwise} \end{cases}, \tag{8}$$

where $\hat{c}^s = \sum_{i\in\mathcal{V}_{\hat{\mathcal{P}}}(s)} \varphi(x(i))$ is a normalization constant.

Eq. (8) states that, eventually, a node $i \in \mathcal{V}_{\hat{\mathcal{P}}}(s)$ is a new element of the set $\mathcal{B}_{t_{s,k}}(s)$ with probability $\frac{\varphi(x(i))}{\hat{c}^s}$, whereas the probability that a node $i \in \mathcal{V} \setminus \mathcal{V}_{\hat{\mathcal{P}}}(s)$ is a new element of $\mathcal{B}_{t_{s,k}}(s)$ is null. Hence, the following equation holds for the limit reference point $\hat{y}(s)$ of Eq. (7):

$$\hat{y}(s) = \arg\min_{p\in\mathbb{R}^2}\left(\lim_{k\to\infty}\frac{1}{k+1}\sum_{q\in\mathcal{B}_{t_k}(s)}\|p-q\|^2\right) = \arg\min_{p\in\mathbb{R}^2}\sum_{i\in\mathcal{V}_{\hat{\mathcal{P}}}(s)}\|p-x(i)\|^2\frac{\varphi(x(i))}{\hat{c}^s}. \tag{9}$$

This last equality holds thanks to the assumptions that the spatial density distribution φ is stationary.

By comparing the definition in Eq. (6) with the last term of Eq. (9), it turns out that the limit reference point $\hat{y}(s)$ is the generalized centroid $m_{\mathcal{V}_{\hat{\mathcal{P}}}(s)}$ of the limit set $\mathcal{V}_{\hat{\mathcal{P}}}(s)$. □

Theorem 1 demonstrates that the limit Voronoi partition obtained by the proposed algorithm is a generalized CVP, and shows that, eventually, each DS agent s, $s \in \mathcal{S}$ migrates only among the nodes of the Voronoi region generated by its limit reference point $\hat{y}(s)$ (see Eq. (9)).

Remark 1. In practice, there is no need to solve the optimization problem (3) at each migration or to store all the values of the past DS node positions, since iterative algorithms exist, as, for instance, the MacQueen's k-means method described hereafter. Let $c_t(s)$ be a counter of the number of migrations of the DS agent s up to time t; the following iterative algorithm eventually converges to the same minimizer of (9) (see [7, 8]):

$$y_{t_{s,\ k+1}}(s) = \frac{c_{t_{s,\ k}}(s)y_{t_{s,\ k}}(s) + x\left(\pi_{t_{s,\ k+1}}(s)\right)}{c_{t_{s,\ k}}(s)+1}e = (1-\gamma_k)y_{t_{s,\ k}}(s) + \gamma_k x(\pi_{t_{s,\ k+1}}(s)), \quad s \in \mathcal{S}, \tag{10}$$

with $\gamma_k = \frac{1}{c_{t_{s,\ k}}(s)+1}$ and $c_{t_{s,k}}(s) = k$. To compute Eq. (10), it is sufficient that each agent s stores the last reference point $y_{t_{s,\ k}}(s)$ and the last counter value $c_{t_{s,k}}(s)$.

Remark 2. If a time-varying candidate metric is used (as the left-over energy metric), the distribution of the candidate node appearances (i.e., the function φ in the proof of Theorem 1) is non-stationary, and, according to Definition 3, the generalized CVT is time-varying as well. In this case, Eq. (10), can be modified in order to weight the more recent reference points more than the older ones, e.g.:

$$y_{t_{s,\ k+1}}(s) = (1-\gamma)y_{t_{s,\ k}}(s) + \gamma \cdot x\left(\pi_{t_{s,\ k+1}}(s)\right), s \in \mathcal{S}, \tag{11}$$

where γ is now a constant real number between 0 and 1. Depending on the dynamics of the distribution, this new rule might be able to 'follow' the variations of the CVT. A similar remark holds if the network graph is time-varying, e.g., due to mobility and/or due to the occurrence of node failures.

4.3. Practical implementation

The proposed algorithm is distributed, since it is executed separately by each DS node. In practice, each sensor is eligible as a DS node; when a node is elected, beside the procedures needed to perform the DS tasks, it also runs the algorithm for the DS election (i.e., it acts as a control agent), which, as detailed below, requires negligible computational and storage resources and limited communications among the agents.

As detailed below, the DS election procedure assures that, at any time $\bar{t}$, each DS agent knows the positions of all the current reference points; moreover, each DS agent is aware of the positions of the current candidate DS nodes, since they communicate their position to all the DS nodes. Let k' be the number of migrations of agent s' up to time $\bar{t}$, let us assume that a new migration occurs at time $\bar{t}$, i.e., $\bar{t} = t_{s',\ k'+1}$, and let the old DS node be $\pi_{t_{s',\ k'}}(s') = i$. The proposed algorithm requires the following tasks to be performed by DS node i:

1. *New DS node selection.* DS node i selects the new DS node $\pi_{\bar{t}}(s')$, without computing the Voronoi partition explicitly. In fact, for each candidate node p, the DS node i just checks if it lies within its own Voronoi region or not: if $s' = \arg\min_{s\in\mathcal{S}} y_{\bar{t}}(s) - x(p)$, it means that the candidate node p lies in the Voronoi region $\mathcal{P}_{\bar{t}}(s')$ generated by the reference point of agent s' at time $\bar{t}$. DS node i simply selects the best candidate node (according to the used metric, e.g., one of the metrics defined in Section 4.1) among the ones in $\mathcal{P}_{\bar{t}}(s')$.
2. *New reference point computation.* DS node i computes the new reference point $y_{\bar{t}}(s')$ by using Eq. (10), which, as discussed in Remark 2, has negligible storage and computational requirements. The new reference point is then communicated to the other DS nodes.
3. *Migration of the information.* DS node i communicates its data (likely, fused data) and the list of the current reference points $\{y_{\bar{t}}(s)\}_{s\in\mathcal{S}}$ to the new DS node $\pi_{\bar{t}}(s')$.

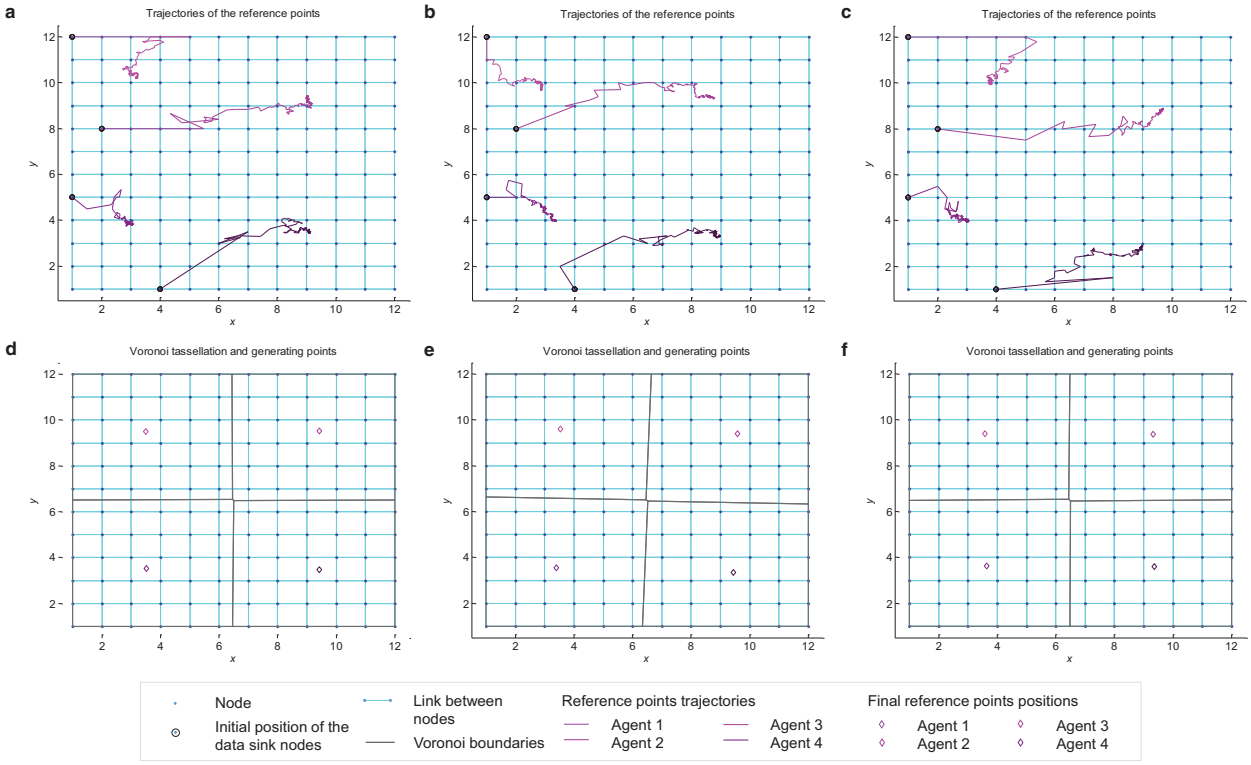

Fig. 1. Simulation 1, upper plots: trajectories of the generating points until time-step $h = 200$: (a) uniform metric, (b) node degree metric, (c) leftover energy metric. Lower plots: reference points and Voronoi regions at the end of the simulation: (d) uniform metric, (e) node degree metric, (f) leftover made energy metric.

Finally, also the routing algorithm has to be updated; this point is however specific to the routing algorithm implemented in the WSN.

5. Simulations

Simulation results of the proposed data sinks election algorithm are presented in this Section. The simulations are aimed at comparing the proposed metrics and at showing the qualitative results of the algorithm in terms of network partitioning.

Simulations are organized in time-steps. In each simulation run, a given number of nodes n and a given number of DS nodes n^{ds} are defined; the links between nodes are set-up depending on the position of the nodes on the mission space: if the distance between two nodes is below a given transmission radius r_{tx}, the link is set-up. The node positioning is either random or deterministic, to obtain a random network topology and a grid network topology, respectively.

The node energy depletion due to the routing of packets heavily depends on the routing algorithm implemented (see, e.g., [20] and references therein), whereas this paper focuses on the evaluation of the migration policy; therefore, the implemented routing algorithm uses a simple hop-count metric. Similarly, since the algorithm does not make any assumption on the node energy model, the energy model of the nodes is kept as simple as possible. Each node has an initial energy level equal to 1. At every time-step, the energy level of the DS nodes is decreased by a constant quantity $\Delta e = 0.01$. Also, at every time-step, the min-hop path from every node to the nearest DS node is computed: the energy depletion of a node belonging to a given number n_p of paths (i.e., a node that has to relay packets for n_p paths during a round) is equal to $n_p \cdot \Delta e_r$, where Δe_r is the energy depletion for relaying packets and is set equal to $\Delta e/100$. As the energy of a node becomes 0, say at time-step τ_f, the simulation terminates and the *minimum node lifetime* is τ_f.

The metrics presented in Section 4 are used by the data sinks election algorithm: the weights $w(i), i \in \mathcal{V}$, are computed according to the uniform metric, the node degree metric and the leftover energy metric.

The nodes take the decision on being a candidate node periodically, with period h, and the probability that node i is a candidate node is $0.5 \cdot w(i)$: in practice, each node i randomly extracts a number $\xi \in [0, 1]$ from a uniform probability distribution: if $\xi < 0.5 \cdot w(i)$, the node is eligible as a new DS node[2]. Thus, at every time-step h, the nodes decide about candidating as a new DS node. In the simulation, at time-step h, either no nodes or one or more nodes are candidate nodes. If there are no candidate nodes, nothing happens. If there is one candidate node, it is (clearly) in the Voronoi region of a given DS agent, which therefore simply migrates to the new node. If there are more candidate nodes, each agent has to decide which candidate node to migrate to, among the ones in its own Voronoi region (recall that each agent s considers a node i as a candidate sink node s if it lies in its Voronoi region). In this last case, the probability of a candidate node of being selected is set as proportional to its weight.

The reference points of the DS nodes are updated according to the iterative Eq. (10).

In the first two simulations, the proposed algorithm is compared to the performance of a DS node election algorithm which uses the LEACH rule (1) to select the DS nodes at every time-step; the parameter p of Eq. (1) is set in order to have an expected number of DS nodes equal to n_{ds}.

The results of the simulations are evaluated in terms of:

[2] In a real implementation, a different policy is needed: with the described one, e.g., in the uniform metric case, with $w(i) = 1, \ \forall i \in \mathcal{V}$, on average half of the fully charged nodes are candidate ones, which leads to excessive control traffic.

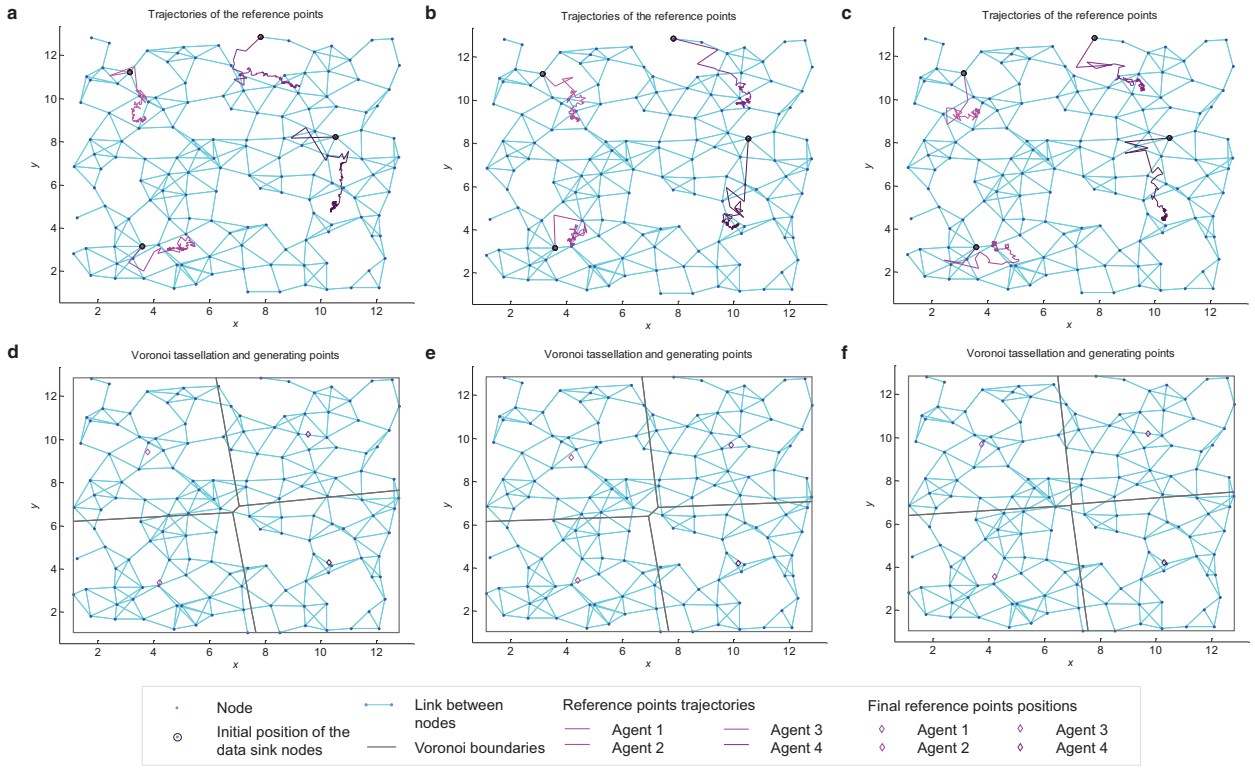

Fig. 2. Simulation 2, upper plots: trajectories of the generating points until time-step $h = 200$: (a) uniform metric, (b) node degree metric, (c) leftover energy metric. Lower plots: reference points and Voronoi regions at the end of the simulation: (d) uniform metric, (e) node degree metric, (f) leftover energy metric.

Table 2
Simulation 1 results.

Output	Proposed partitioning (metric)			LEACH
	Uniform	Node degree	Leftover energy	
Minimum node lifetime	1238.9	1175.5	1521.2	873.0
Mean leftover node energy	0.31	0.35	0.14	0.34
Std. dev. leftover node energy	0.14	0.16	0.07	0.19
Average path length	3.61	3.60	3.72	5.74
Average n. of DS nodes	4.00	4.00	4.00	4.07
Std. dev. % of nodes per DS node	2.43%	2.54%	3.24%	–

- *Minimum node lifetime*, which is the time-step when the first node energy is depleted; this parameter shows the effectiveness of the DS election protocol to save the nodes energy.
- *Mean leftover node energy*, which is the average energy level of the nodes at the end of the simulation, and *standard deviation of node energy*, which is the standard deviation of the energy level of the nodes at the end of the simulation; these parameters show how the DS election protocol manages to balance the energy depletion of the nodes.
- *Standard deviation of the % of nodes per data sink node*, which indicates how well balanced is the distribution of the sensor nodes among the DS nodes (this statistic is not applicable to the simulations with the LEACH selection rule since the number of DS nodes varies in time).

In the first simulation, a 12×12 grid topology with $n = 144$ nodes and with $n^{ds} = 4$ DS nodes is simulated. The transmission radius r^{tx} is set to 1.4, and the distance between two nodes is 1; the resulting topology is an equispaced grid (see Fig. 1). The results are averaged over 20 simulation runs. In each run, the initial positions of the DS nodes are chosen randomly, and the simulation is executed with the three proposed metrics.

Table 2 shows the simulation results. With all the metrics, the algorithm manages to distribute the workload of the sensor nodes among the DS nodes, as shown by the small values of the standard deviation of the % of nodes per DS node. Concerning the network lifetime, the leftover energy metric outperforms both the random metric, by 21%, and the node degree metric, by 34%, thanks to the fact that it achieves a better balancing of the leftover energy levels, as shown by the values of the standard deviation of leftover node energy. The LEACH selection rule obtains much smaller lifetimes (about 70%, 74% and 57% of the lifetime of the algorithm with the uniform metric, node degree metric and leftover energy metric, respectively), since it cannot control the spatial distribution of the DS nodes and, therefore, the average path length is larger.

The upper plots of Fig. 1 show, for one of the simulation runs and for the three metrics, the simulated grid topology, the initial position of the DS nodes and the trajectories of the reference points during the first 100 time-steps. The lower plots of Fig. 1 show the final reference point positions and the resulting Voronoi regions.

If the spatial distribution of the candidate node appearances were a continuous and uniform spatial distribution on the mission square area, the CVTs would define 4 regions with similar area

Table 3
Simulation 2 results.

Output	Proposed partitioning (metric)			LEACH
	Uniform	Node degree	Leftover energy	
Minimum node lifetime	751.6	771.0	919.3	416.3
Mean leftover node energy	0.60	0.60	0.50	0.71
Std. dev. leftover node energy	0.16	0.19	0.10	0.18
Average path length	3.33	3.25	3.52	5.09
Average n. of DS nodes	4.00	4.00	4.00	4.04
Std. dev. % of nodes per DS node	2.35%	2.56%	2.93%	–

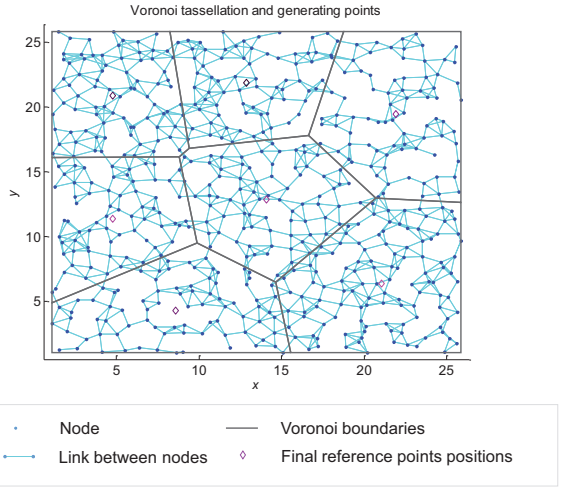

Fig. 3. Simulation 3, reference points of the DS nodes and Voronoi regions at the end of the simulation.

(e.g., the 4 quadrants). The figures clearly show that, even with the time-varying energy metric, the algorithm well approximates a CVT. In this respect, we note that the energy metric is able to produce a balanced node discharging during the mission, which, in turn, makes the probability distribution balanced over the nodes.

Then, non-regular topologies with $n = 136$ nodes and with $n^{ds} = 4$ DS nodes are simulated. The results are averaged over 20 simulation runs; each run is executed with the proposed metrics. In each run, the initial DS node positions are randomly chosen.

Table 3 shows the simulation results, analogous to the ones of the first simulation runs.

The upper plots of Fig. 2 show, for one of the simulation runs and for the three metrics, the simulated grid topology, the initial position of the DS nodes and the trajectories of the reference points during the first 100 time-steps. The lower plots of Fig. 2 show the final reference point positions and the resulting Voronoi regions.

The figure of the result of a third simulation is added just to show qualitatively how the algorithm works in larger networks and with more DS nodes: Fig. 3 shows an example of network partitioning resulting from the DS node election algorithm with the energy metric, with $n = 575$ nodes and with $n^{ds} = 7$ DS nodes. The figure suggests that the dynamic partitioning is effective also in this case; the workload is in fact well balanced among the DS nodes, since the standard deviation of the % of nodes per DS node is about 1.78%.

Finally, we take into account a 12×12 grid scenario, where the initial node energy is different among the nodes (see the upper-left plot of Fig. 4). In this scenario, the energy metric tries to balance the leftover node energy and, by doing so, it changes the relative energy distribution among the nodes; therefore, the spatial distribution φ is time-varying. As discussed in Remark 2, the iterative Eq. (11) is used, with the parameter γ set equal to 0.1 (tuned by simulation runs). Fig. 4 shows the energy distribution among the nodes (which is proportional to the node weights) during the simulation, and the reference point positions: the algorithm is able to 'follow' the distribution variation and, at the end of the simulation at time $h = 1070$ s, the leftover energy is balanced among the nodes and the final Voronoi partition is similar to the one obtained in Fig. 1.

The plots at intermediate time-steps are suitable for showing the algorithm behavior. Since the initial node weights (proportional to the leftover node energy levels) are larger for larger values of y (see the right plot at $h = 0$), the elected DS nodes are initially distributed in such a way that the Voronoi regions including the less charged nodes have a larger number of nodes (see the left plots at $h = 250$ and, especially, at $h = 500$). By so doing, the energy of the nodes is already fairly balanced at $h = 500$, as shown by the corresponding right figure.

6. Conclusions

The proposed data sinks election algorithm manages to organize the data sink node migrations in such a way that, during the network lifetime, the data sink nodes are positioned within a network partition which is proved to converge to the Centroidal Voronoi Tessellation. As shown by numerical simulations, the result is a balanced positioning of the data sink nodes within the sensor network area, which is capable of balancing the load of data sink nodes and, especially if coupled with an energy-based metric, of prolonging the lifetime of the nodes. The problem dealt with in the paper is similar to other problems in WSNs, such as the cluster head election, and with some awareness can be regarded as a generic method to perform partitioning in *ad-hoc* wireless networks.

On-going work is devoted at analyzing the impact of energy harvesting approaches (see, e.g., [9]) and the impact of the proposed procedure in presence of all the other network management algorithms of a WSN, as, in particular, the routing algorithm and the data-fusion techniques proposed in the project SWIPE ([16, 15] and [18, 27], respectively): energy-aware routing algorithms affect the energy depletion of the nodes and may also cause a complex interaction with the data sink election algorithm, and data-fusion techniques affect both the computational requirements of the data sink nodes (and therefore their energy consumption) and the traffic load on the network. Moreover, the proposed algorithm is being implemented in the sensor nodes which the SWIPE project will integrate and demonstrate.

Acknowledgments

This work is supported by the by the European FP7 project SWIPE (Space WIreless sensor networks for Planetary Exploration), funded under Grant Agreement no. 312826.

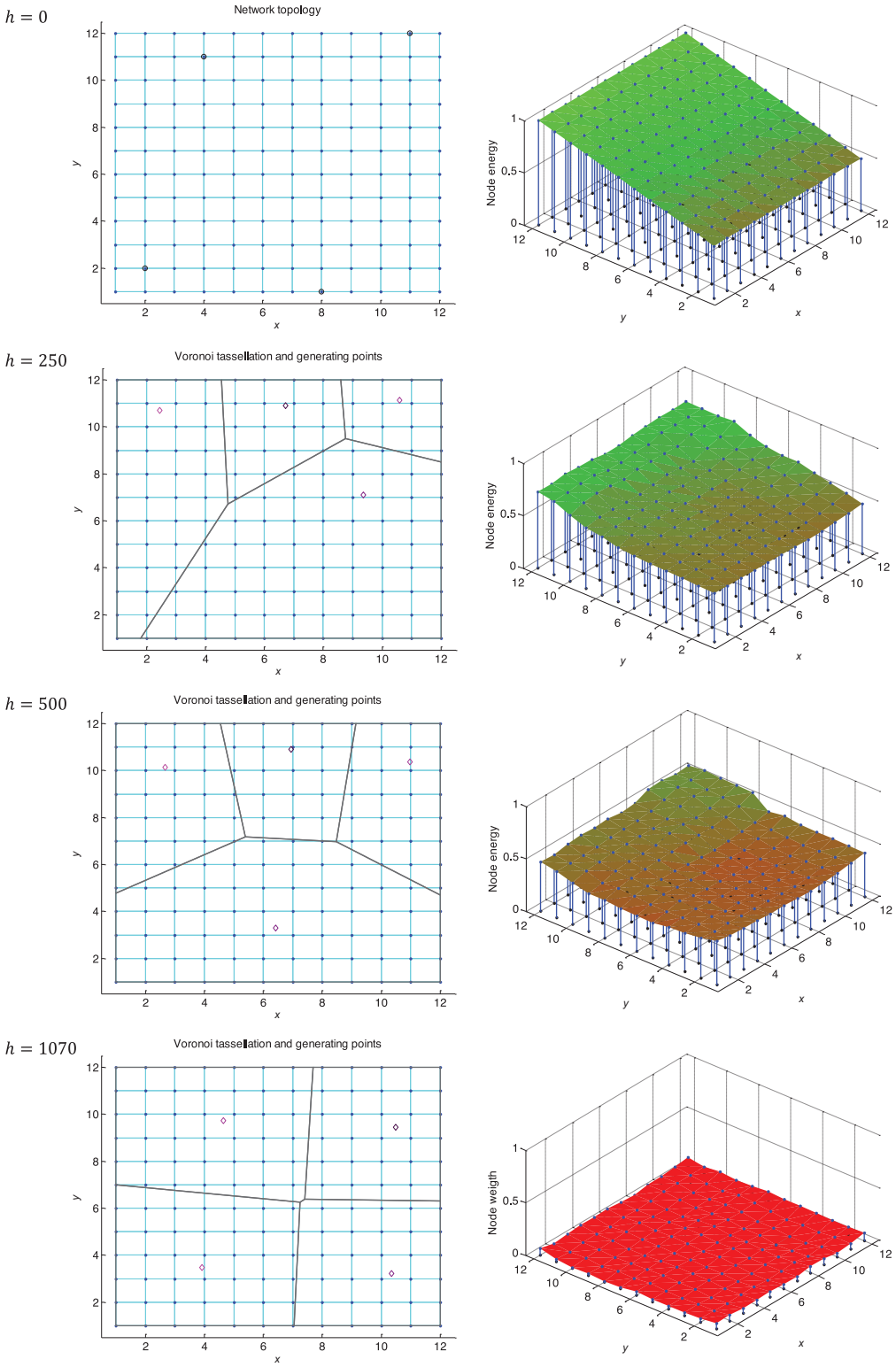

Fig. 4. Simulation 4, left plots: reference points of the data sink nodes and Voronoi regions during the simulation, right plots: leftover node energy during the simulation. In the figure, the node energy level is represented by a bar on the z-axis starting from the node position on the (x, y) plane; the figure also shows an envelope plane for presentation clarity.

References

[1] M.M. Afsar, M.-H. Tayarani-N, Clustering in sensor networks: a literature survey, J. Netw. Comput. Appl. 46 (2014) 198–226, doi:10.1016/j.jnca.2014.09.005.

[2] W. Alsalih, K. Islam, Y. Núñez-Rodríguez, H. Xiao, Distributed voronoi diagram computation in wireless sensor networks, in: Proceedings of the Twentieth Annual Symposium on Parallelism in Algorithms and Architectures - SPAA '08, ACM Press, New York, New York, USA, 2008, p. 364, doi:10.1145/1378533.1378597.

[3] A. Arsie, K. Savla, E. Frazzoli, Efficient routing algorithms for multiple vehicles with no explicit communications, IEEE Trans. Autom. Control 54 (10) (2009) 2302–2317, doi:10.1109/TAC.2009.2028954.

[4] A.P. Azad, A. Chockalingam, Enhancing lifetime of wireless sensor networks using multiple data sinks, Int. J. Sens. Netw 9 (3/4) (2011) 139–157, doi:10.1504/IJSNET.2011.040236.

[5] G. Chen, C. Li, M. Ye, J. Wu, An unequal cluster-based routing protocol in wireless sensor networks, Wirel. Netw. 15 (2) (2009) 193–207, doi:10.1007/s11276-007-0035-8.

[6] J. Chen, C. Kim, F. Song, A distributed clustering algorithm for voronoi cell-based large scale wireless sensor network, in: Proceedings of the 2010 International Conference on Communications and Mobile Computing, IEEE, 2010, pp. 209–213, doi:10.1109/CMC.2010.230.

[7] Q. Du, V. Faber, M. Gunzburger, Centroidal voronoi tessellations: applications and algorithms, SIAM Rev. 41 (4) (1999) 637–676, doi:10.1137/S0036144599352836.

[8] Q. Du, T.W. Wong, Numerical studies of MacQueen's k-means algorithm for computing the centroidal Voronoi tessellations, Comput. Math. Appl. 44 (02) (2002) 511–523, doi:10.1016/S0898-1221(02)00165-7.

[9] A. Frezzetti, S. Manfredi, M. Pagano, A design approach of the solar harvesting control system for wireless sensor node, Control Eng. Pract. 44 (2015) 45–54, doi:10.1016/j.conengprac.2015.07.004.

[10] H. Gou, Y. Yoo, H. Zeng, A partition-based LEACH algorithm for wireless sensor networks, in: Proceedings of the 2009 Ninth IEEE International Conference on Computer and Information Technology, 2, 2009, doi:10.1109/CIT.2009.21.

[11] Z. Guo, S. Malakooti, S. Sheikh, C. Al-Najjar, B. Malakooti, Multi-objective OLSR for proactive routing in MANET with delay, energy, and link lifetime predictions, Appl. Math. Model. 35 (3) (2011) 1413–1426, doi:10.1016/j.apm.2010.09.019.

[12] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, in: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 2000, doi:10.1109/HICSS.2000.926982.

[13] K. Joongheon, C. Jihoon, L. Wonjun, Energy-Aware distributed topology control for coverage-time optimization in clustering-based heterogeneous sensor networks„ in: Proceedings of the 2006 IEEE 63rd Vehicular Technology Conference, Vol. 3, IEEE, 2006, pp. 1033–1037, doi:10.1109/VETECS.2006.1682991.

[14] K. Joongheon, C. Jihoon, L. Wonjun, Energy-Aware distributed topology control for coverage-time optimization in clustering-based heterogeneous sensor networks, in: Proceedings of the 2006 IEEE 63rd Vehicular Technology Conference, Vol. 3, IEEE, 2006, pp. 1033–1037, doi:10.1109/VETECS.2006.1682991.

[15] F. Liberati, G. Oddi, A. Lanna, A. Pietrabissa, A lightweight sensor scheduling algorithm for clustered wireless sensor networks, in: Proceedings of the 2015 23rd Mediterranean Conference on Control and Automation (MED), IEEE, Torremolinos, ES, 2015, pp. 953–959, doi:10.1109/MED.2015.7158881.

[16] D. Macone, G. Oddi, A. Pietrabissa, MQ-Routing: mobility-, GPS- and energy-aware routing protocol in MANETs for disaster relief scenarios, Ad Hoc Netw. 11 (3) (2013) 861–878, doi:10.1016/j.adhoc.2012.09.008.

[17] K.S. Nithyakalyani, S. Suresh, An approach to data aggregation in wireless sensor network using Voronoi fuzzy clustering algorithm, J. Sci. Ind. Res. 72 (5) (2013) 287–293.

[18] G. Oddi, A. Pietrabissa, F. Liberati, Energy balancing in multi-hop wireless sensor networks: an approach based on reinforcement learning, in: Proceedings of the 2014 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), IEEE, 2014, pp. 262–269, doi:10.1109/AHS.2014.6880186.

[19] M.F. Othman, K. Shazali, Wireless sensor network applications: a study in environment monitoring system, Proc. Eng. 41 (2012) 1204–1210 Retrieved from http://www.sciencedirect.com/science/article/pii/S1877705812027026 .

[20] N.A. Pantazis, S.A. Nikolidakis, D.D. Vergados, Energy-efficient routing protocols in wireless sensor networks: a survey, IEEE Commun. Surv. Tutor. 15 (2) (2013) 551–591, doi:10.1109/SURV.2012.062612.00084.

[21] A. Pietrabissa, C. Poli, D.G. Ferriero, M. Grigioni, Optimal planning of sensor networks for asset tracking in hospital environments, Decis. Support Syst. 55 (1) (2013) 304–313, doi:10.1016/j.dss.2013.01.031.

[22] U. Prathap, P.D. Shenoy, K.R. Venugopal, L.M. Patnaik, Wireless sensor networks applications and routing protocols: survey and research challenges, in: Proceedings of the 2012 International Symposium on Cloud and Services Computing, 2012, pp. 49–56, doi:10.1109/ISCOS.2012.21.

[23] P. Rodrigues, A. Oliveira, R. Mendes, F. Alvarez, M. Crosnier, T. Vladimorova, A. Pietrabissa, Wireless sensor networks for moon exploration, in: Proceedings of the 64th International Astronautical Congress 2013, IAC 2013, Beijing, China, 2013, pp. 1–10.

[24] P. Suri, R.K. Bedi, S.K. Gupta, Review paper on various clustering protocols used in wireless sensor network (WSN), in: Proceedings of the 2015 International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO), IEEE, 2015, pp. 1–4, doi:10.1109/EESCO.2015.7253700.

[25] G.S. Tewolde, W. Sheng, Energy aware adaptive clustering in wireless sensor networks, in: Proceedings of the 2011 IEEE International Conference on Electro/Information Technology, IEEE, 2011, pp. 1–6, doi:10.1109/EIT.2011.5978632.

[26] A. Vidács, R. Vida, Wireless sensor network based technologies for critical infrastructure systems, in: E. Kyriakides, M. Polycarpou (Eds.), Intelligent Monitoring, Control, and Security of Critical Infrastructure Systems Studies in Computational Intelligence, Springer, Berlin Heidelberg, 2015, pp. 201–316, doi:10.1007/978-3-662-44160-2_11.

[27] Z. Xiaojun, J. Hongyuan, T. Vladimirova, Multi-sensor data fusion in wireless sensor networks for planetary exploration, in: Proceedings of the 2014 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), IEEE, 2014, pp. 188–195, doi:10.1109/AHS.2014.6880176.

[28] M. Ye, C. Li, G. Chen, J. Wu, EECS: an energy efficient clustering scheme in wireless sensor networks, in: Proceedings of the 24th IEEE International Performance, Computing, and Communications Conference, 2005 PCCC 2005, 2005, doi:10.1109/PCCC.2005.1460630.

[29] O. Younis, S. Fahmy, HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks, IEEE Trans. Mobile Comput. 3 (2004) 366–379, doi:10.1109/TMC.2004.41.

[30] T.-C. Yu, C.-C. Lin, C.-C. Chen, W.-L. Lee, R.-G. Lee, C.-H. Tseng, S.-P. Liu, Wireless sensor networks for indoor air quality monitoring, Med. Eng. Phys. 35 (2) (2013) 231–235 Retrieved from http://www.sciencedirect.com/science/article/pii/S1350453311002761 .

[31] G. Zhao, Wireless sensor networks for industrial process monitoring and control: a survey, Netw. Protoc. Algorith 3 (1) (2011) 46–63, doi:10.5296/npa.v3i1.580.

Antonio Pietrabissa (MS'00–PhD'04) is Assistant Professor of Automatic Control with the Department of Computer, System and Management Engineering of the University of Rome "La Sapienza" since 2010. His main research focus is the application of system and control theory methodologies to telecommunication networks. Since 2000, he has been participating in 15 European Union projects (3 as scientific responsible), 2 ESA projects and 5 national projects (one as scientific responsible) on telecommunications. He is author or coauthor of about 100 papers on journals, conference proceedings and book chapters on these topics.

Francesco Liberati received the master degree in Control Engineering in 2011 and the Ph.D. in Systems Engineering in 2015 from the University of Rome "La Sapienza", Italy. Since 2015, he is Assistant Professor at the eCampus University, Novedrate, Italy. He has published papers mainly in the area of smart grids and model predictive control. He is involved in several European research projects dealing with smart grids and wireless sensor networks management, critical infrastructure protection and control of virtualized network infrastructures.

Guido Oddi was born on June 17, 1984, in Milan, Italy. He received the Master Degree (cum laude) in Computer Engineering in March 2009 and the PhD in Systems Engineering in March 2013 at the University of Rome "La Sapienza". He worked in several EU-financed research projects (e.g., OMEGA, MONET, SWIPE, MOBINCITY) and Italian projects (e.g., PLATINO) in the field of ICT. His main research interests include the application of control-theory and game-theory methodologies to network resource management, in particular routing and load balancing problems in heterogeneous and resource-constrained scenarios (e.g., wireless ad-hoc and sensor networks). He is co-author of more than 20 technical papers on international conferences and journals in the field of telecommunications and control.