J. Parallel Distrib. Comput. 70 (2010) 371–383

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

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

Energy-efficient distributed clustering in wireless sensor networks ☆

N. Dimokas [a], D. Katsaros [b,*], Y. Manolopoulos [a]

[a] Department of Informatics, Aristotle University, Thessaloniki, Greece

[b] Computer & Communication Engineering Department, University of Thessaly, Volos, Greece

ARTICLE INFO

Article history:
Received 9 February 2009
Received in revised form
13 August 2009
Accepted 14 August 2009
Available online 1 September 2009

Keywords:
Sensor networks
Clustering
Network lifetime
Energy conservation
Backbone formation

ABSTRACT

The deployment of wireless sensor networks in many application areas requires self-organization of the network nodes into clusters. Clustering is a network management technique, since it creates a hierarchical structure over a flat network. Quite a lot of node clustering techniques have appeared in the literature, and roughly fall into two families: those based on the construction of a dominating set and those which are based solely on energy considerations. The former family suffers from the fact that only a small subset of the network nodes are responsible for relaying the messages, and thus cause rapid consumption of the energy of these nodes. The latter family uses the residual energy of each node in order to decide about whether it will elect itself as a leader of a cluster or not. This family's methods ignore topological features of the nodes and are used in combination with the methods of the former family. We propose an energy-efficient distributed clustering protocol for wireless sensor networks, based on a metric for characterizing the significance of a node, w.r.t. its contribution in relaying messages. The protocol achieves small communication complexity and linear computation complexity. Experimental results attest that the protocol improves network longevity.

1. Introduction

The rapid technological advances in low-power hardware design have enabled the development of tiny battery-powered sensor nodes which are able to compute, sense physical "parameters" and communicate with each other. A wireless sensor network (WSN) is a network of large numbers of sensors nodes, where each node is equipped with limited on-board processing, storage and radio capabilities [2]. Sensor nodes are quasi-stationary, densely deployed and with limited capabilities. Nodes sense and send their signals towards a data center which is called the "information sink". The design of protocols and applications for such networks has to be energy aware in order to prolong the lifetime of the network because it is quite difficult to recharge node batteries. Additionally, it has to take into account the multi-hop communication nature. Communication in a WSN between any two nodes that are out of one another's transmission range is achieved through intermediate nodes, which relay messages to set up a communication channel between the two nodes.

The WSNs are deployed in a target area in order to facilitate many applications like habitat monitoring [27], disaster relief [28], target tracking [8] and so on. Many of these applications require simply an aggregate value to be reported to the "information sink" (observer, base station, etc.). In these cases, sensors in different regions of the field can collaborate to aggregate the information that they gathered. For instance, in habitat monitoring applications the sink may require the average of temperature; in military applications the existence or not of high levels of radiation may be the target information that is being sought. Grouping nodes into *clusters* has been widely pursued by the research community in order to achieve the network scalability objective. Clustering not only allows aggregation, but also limits data transmission primarily within the cluster [11], thereby reducing both the traffic and the contention for the channel.

An example application of our study is a sensor network that is being deployed in a modern battlefield, with sensor nodes being dispersed in a large area. Each sensor node is equipped with a micro-camera that can take a photograph of a very narrow band around its position. The sensor nodes share (on demand) with each other the new photographs, in order to build a more complete view of the region that is being monitored. The sharing is necessary because every micro-camera can capture a limited view of the whole region, either due to the sensor node's position or because of the obstacles that exist near to the sensor node. Therefore, every

☆ Research supported by the project "Control for Coordination of Distributed Systems", funded by the EU.ICT program, Challenge ICT-2007.3.7.

* Corresponding author.
E-mail addresses: dimokas@delab.csd.auth.gr (N. Dimokas), dkatsar@inf.uth.gr, dimitris@delab.csd.auth.gr (D. Katsaros), manolopo@csd.auth.gr (Y. Manolopoulos).
URLs: http://delab.csd.auth.gr/~dimokas (N. Dimokas), http://www.inf.uth.gr/~dkatsar (D. Katsaros), http://delab.csd.auth.gr/~manolopo (Y. Manolopoulos).

0743-7315/$ – see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jpdc.2009.08.007

sensor may request and receive a large number of photographs taken by some other sensor(s) through multi-hop communication. Afterwards, each sensor is able to respond to queries about "high-level" events, e.g., enemy presence. In this case, if a sensor node uses up its energy the sensor network could be considered not operational. This is because the monitoring of enemy presence will not be valid and accurate.

The clustering formation procedure involves the election of a *cluster head* (CH) node in each cluster, in order to coordinate the cluster nodes. The cluster head is responsible for getting the measured values from its cluster's nodes, aggregating them and sending the aggregates to the sink(s) through other cluster heads. Several studies [19,32] indicate that clustering increases the *network lifetime*. Although the definition of the network lifetime depends on the applications' semantics, a widely accepted definition is the time until the first/last node of the network depletes its energy [33].

The issue of network node clustering first appeared in [5], reconsidered in [18,20], and later improved in [3,6,13,23,25,29] in the context of mobile ad hoc networks. All these efforts recognized the significance of selecting the most "appropriate" nodes as cluster heads and they did this through the use of the notion of *dominating sets* (DS), i.e., a subset of the network nodes such that any node of the network graph either belongs to the DS or is a neighbor of a node of the DS. A survey of such methods can be found in [7]. The major shortcoming of such algorithms is the fact that the nodes belonging to the DS are solely responsible for carrying out all communication, thus running out of energy very soon.

Apart from this family of algorithms, a second family provided mechanisms for addressing the energy consumption problem due to the repetitive communication by the same nodes, i.e., the cluster heads. This family of protocols essentially proposed ways to "rotate" the role of cluster head among nodes of clusters, e.g., the SPAN [14], the LEACH [19], and the HEED [32]. The proposed methods use the residual energy of each node in order to direct its decision about whether it will elect itself as a cluster head node or not. However, this family's methods ignore topological features of the nodes.

All clustering algorithms proposed so far (see [1,7,33] for more complete surveys) present some weaknesses. Some methods rely on node IDs in eliminating potential redundant broadcasting nodes or in defining priorities, e.g., [3,6,13,25,29]. These approaches suffer from the fact that they cannot detect all possible eliminations, because ordering based on node ID (or node weight) prevents this. As a consequence they incur significantly excessive retransmissions. Some methods (e.g., [5,18,20,23,25]) do not *fully* exploit the compiled information; for instance, the use of the node degree as its priority when deciding whether it will become a cluster head might not result in the best local decision. Finally, some methods create a lot of clusters [29], or require excessive communication cost [3].

Here, we propose an energy-efficient distributed clustering protocol for wireless sensor networks, called *GESC* from the initials of the words *GEodesic Sensor Clustering*, that considers energy consumption and topological features of the nodes. The proposed method exploits the localized network structure and the remaining energy of neighboring nodes in order to define a new way for estimating dynamically the cluster heads. *GESC* is compared with LEACH [19], which is a well-known, established clustering protocol for wireless sensor networks, and with the most efficient algorithms reported in [7,33] for mobile ad hoc networks. The *GESC* protocol complies with all requirements described in [33]:

- It is localized, and thus distributed; *it can exploit one-hop, two-hop or k-hop neighborhood information, presenting different tradeoffs in efficiency vs. communication cost, but for the sake of readability we present it here assuming knowledge of the two-hop neighborhood of a sensor node*.
- Network lifetime is prolonged by distributed energy consumption. The cluster heads are estimated dynamically depending on the originator node, which wishes to transmit a message; thus the cluster heads are not static, avoiding fast depletion of their energy.
- It describes a new way for capturing a node's significance/presence w.r.t. the fact that the node resides in network paths that will definitely be traversed by the majority of the transmitted messages.
- It computes a node's significance in time linear in the number of nodes and linear in the number of edges of the network neighborhood of the node, irrespectively of the degree of each node.
- It allows for fast network clustering; thus it is appropriate for reclustering operations.

The rest of the paper is organized as follows. In Section 2 we review the relevant work. Section 3 describes the network model and necessary preliminaries of our work. In Section 4 we describe a new way for measuring sensor nodes' significance and the respective clustering protocol. In Section 5, we evaluate through simulations the proposed protocol and compare it to other clustering protocols [23,25,29] from the area of MANETs. In Section 6, we compare *GESC* with a multi-hop energy-efficient version of the LEACH [19] protocol well-known from the literature and finally in Section 7 we conclude the paper.

2. Related work

The network node clustering technique has been widely investigated in the context of mobile ad hoc networks [3,6,7,13,23,25,29,33]. The proposed protocols are distributed, localized and select the most significant nodes as cluster heads. In order to achieve this they compute a dominating set (DS). In [6], the author assumes quasi-stationary nodes with real-valued weights, while the Weighted Clustering Algorithm (WCA [13]) combines several properties in one parameter that is used for clustering. With Max–Min D-cluster, the authors [3] propose a new distributed cluster head election procedure, where no node is more than d (d is a value selected for the heuristic) hops away from the CH.

Wu & Li [29] proposed a distributed algorithm for finding a connected dominating set (CDS) in order to design efficient routing schemes for a MANET. Every node v exchanges its neighbor list with all its neighbors. A node set itself has a dominating node if it has at least two unconnected neighbors. In order to reduce the size of a CDS, some extension rules are proposed by the authors. According to the first rule, a node deletes itself from the CDS when its close neighbor set (which includes all of its direct neighbors as well as itself) is completely included in the neighbor set of a neighboring dominating node and it has smaller ID than the neighboring dominating node. According to the second rule, a node deletes itself from the CDS when its open neighbor set (which includes all of its direct neighbors) is completely included in the neighbor sets of two connected neighboring dominating nodes and has the smallest ID. Stojmenovic [25] proposed an algorithm for improving the performance of the protocol that has been proposed in [29]. Nodes are classified as follows. A node is called intermediate if there are two neighbors that are not directly connected. An intergateway node is a node that is not deleted from dominating nodes after applying Rule 1 from the Wu & Li protocol, while a gateway node is a node that is not deleted after applying Rule 2. The author replaced node IDs with a record that includes the node's degree and the node's x, y coordinates. The only nodes that are allowed to retransmit a message are intergateway and gateway nodes. Finally, before a node rebroadcasts a message it computes the number of one-hop neighbors that have been covered from the previous rebroadcasting. If there are uncovered neighbors, then broadcasting proceeds.

A high degree of localization is presented by the protocol proposed in [23]. The authors focus on reduction of the duplicate message retransmissions while the messages are being forwarded to the destination nodes, in order to achieve efficient flooding in mobile wireless networks. The relay points of a given source or retransmitting node *u* are defined by the authors of [23] as follows. A node is assumed "covered" if it received a message originated at *u* either directly or through retransmissions by other nodes. Relay points of *u* are one-hop neighbors of *u* that cover all the two-hop neighbors of *u*. The proposed algorithm includes three phases. Initially, each node *u* starts with an empty multipoint relay set. In the second phase, node *u* selects as multipoint relays those one-hop neighbors that are unique neighbors of some nodes in *u*'s two-hop neighborhood and adds them in a multipoint relay set. In the second phase, while there are uncovered nodes from the multipoint relay set in *u*'s two-hop neighborhood, then for each one-hop neighbor not included in a multipoint relay set one computes the number of two-hop neighbors that it covers and are still uncovered. Finally, one adds in a multipoint relay set the node with the biggest number.

Clustering is an effective topology control approach in WSNs which can increase network scalability and lifetime. Sensor node clustering is a very important optimization problem. In order to maintain a certain degree of service quality and a reasonable system lifetime, energy needs to be optimized at every stage of the system operation. A clustering scheme can effectively prolong the lifetime of wireless sensor networks by using the limited energy resources of the deployed sensor nodes efficiently.

LEACH [19] is an energy-efficient protocol designed for sensor networks with continuous data delivery mechanism and no mobility. Sensor nodes elect themselves as cluster heads with some probability and broadcast their decisions. The remaining nodes join a cluster, of which the cluster head is closest in terms of the communication energy cost. Then the role of the cluster head is periodically rotated among the nodes to balance energy consumption, since cluster heads have the extra burden of performing a long-range transmission to a distant sink node. Thus, LEACH counteracts the problem of non-uniform energy drainage by role rotation. HEED [32] introduces a variable known as the cluster radius which defines the transmission power to be used for intracluster broadcast. The initial probability for each node to become a tentative cluster head depends on its residual energy, and final cluster heads are selected according to the intracluster communication cost. HEED relies on the assumption that cluster heads can communicate with each other and form a connected graph; realizing this assumption in practical deployments could be tricky. In [4], the authors use LEACH-like clustering and multi-hop forwarding for both intracluster and intercluster communication. They provide also methods in order to compute the optimal values of the algorithm parameters a priori. Chang and Tassiulas [12] proposed methods for maximizing overall network lifetime by distributing energy consumption fairly. In this protocol, nodes adjust their transmission power levels and select routes to optimize performance. In [15], a multilevel hierarchical structure is proposed where cluster heads are selected according to their residual energy. Buttyan et al. [10] propose a Position-based Aggregator Node ELection (PANEL) in wireless sensor networks. PANEL is an energy-efficient protocol that ensures load balancing in the sense that each node is elected aggregator (CH) nearly equally frequently. However, PANEL uses the geographical position information of the nodes to determine which of them should be the aggregators, which is a restriction in WSNs, since the geographical position is difficult to obtain without the use of GPS-like hardware or central coordination.

In [31], the authors propose a new energy-efficient clustering approach (EECS) for single-hop wireless sensor networks, which is more suitable for the periodical data gathering applications. EECS extends LEACH algorithm by dynamic sizing of clusters based on cluster distance from the base station. In the cluster head election phase, unlike for LEACH, the cluster head is elected by localized competition and its no iteration property makes it differ from HEED. This competition involves candidates broadcasting their residual energy to neighboring candidates. If a given node does not find a node with more residual energy, it becomes a cluster head. However, the EECS protocol does not consider the structural characteristics of network topology and thus cluster heads are elected on the basis of residual energy. Unlike the proposed protocol *GESC* (*GEodesic Sensor Clustering*), which is designed for multi-hop networks, the strategy proposed in [21] is not scalable as it requires all nodes in the WSN to be in the direct transmission range of the base station. The authors proposed a strategy for saving energy in continuous data collection applications in WSN by exploiting the spatiotemporal correlation. Thus, the sink node partitions the sensor nodes with similar measured values into clusters and the sensor nodes within a cluster are scheduled to work alternatively in order to reduce energy dissipation. Youssef et al. [34] proposed MOCA, a randomized, distributed multi-hop overlapping clustering algorithm for organizing the sensors into overlapping clusters. However, the major goal of the clustering process is to ensure that each node is either a cluster head or within *k* hops of at least one cluster head, where *k* is a preset cluster radius.

There are some also works on clustering in mobile sensor networks [35], and also recent works on clustering [36,37] for achieving specialized quality of service constraints, like field coverage, and also research work on clustering in mesh networks [38], though these works are not directly relevant to the present investigation.

Remotely related to the present article are topology control schemes like GAF [30] and SPAN [14], where nodes are classified according to their geographic location into equivalence classes. However, acquiring location information for sensor nodes would require GPS-like hardware or, on the other hand, periodically broadcasting connectivity information. GAF and SPAN use redundancy in the sensor network in order to determine the awake and asleep nodes and finally prolong the network lifetime.

Finally, complementary to our work are the techniques and protocols that proposed in [16]. The authors investigate the design of techniques and protocols that lead to efficient data aggregation without explicit maintenance of a structure.

3. Network model

WSN includes large numbers of sensor nodes, dispersed in a sensor field. We assume that *N* is the total number of sensor nodes. Additionally, no assumptions are made about the network diameter and network density. We consider the following properties of the sensor network:

- The sensor nodes are static; in the majority of applications, sensor nodes have no mobility.
- Initially all sensor nodes are charged with the same amount of energy.
- Links are bidirectional.
- The computation and communication capabilities are the same for all network nodes. Moreover, it is not feasible to recharge nodes' batteries. For example in a battlefield, sensor nodes are dispersed in a large target area where reaching and recharging them is extremely difficult and dangerous. This motivates us to design a protocol that is energy aware in order to prolong the lifetime of the network.
- Sensor nodes do not require GPS-like hardware. So, they are not location aware.
- Sensor nodes are not location aware as regards information sinks. Additionally, they have no knowledge about how many information sinks exist.
- The network "dies" when any of its sensors depletes its energy.

4. The new sensor node clustering protocol

The proposed clustering protocol is distributed, where nodes make autonomous decisions without any centralized control, and energy-efficient, avoiding the fast energy depletion of sensor nodes and the excessive communication cost in terms of retransmitted messages. We name the protocol *GESC*, from the initials of the words *GEodesic Sensor Clustering* protocol. *GESC* exploits the local network characteristic and the residual energy of neighboring nodes to achieve network lifetime prolongation. One of the main parts of the proposed protocol is the estimation of the significance of sensors relative to the network topology. The intuition is that if we discover those energy-efficient nodes which reside in a significant part of the (short) paths connecting other nodes, then these are the cluster coordinators for the clustering protocol.

4.1. Preliminaries

Before proceeding in the presentation of the main ideas of the paper, we will give some necessary definitions. A wireless sensor network is abstracted as a graph $G(V, E)$. An edge $e = (u, v),\ u, v \in E$, exists if and only if u is in the transmission range of v and vice versa. All links in the graph are bidirectional. The set of neighbors of a node v is represented by $N_1(v)$, i.e., $N_1(v) = \{u : (v, u) \in E\}$. The set of two-hop nodes of node v, i.e., the nodes which are the neighbors of node v's neighbors except for the nodes that are the neighbors of node v, is represented by $N_2(v)$, i.e., $N_2(v) = \{w : (u, w) \in E$, where $w \neq v$ and $w \notin N_1$ and $(v, u) \in E\}$. The combined set of one-hop and two-hop neighbors of v is denoted as $N_{12}(v)$.

Definition 1 (*Local Network View w.r.t. Node v*). The Local Network view, denoted as LN_v, of a graph $G(V, E)$ w.r.t. a node $v \in V$ is the *induced subgraph* of G associated with the set of vertices in $N_{12}(v)$.

We define a *path* from $u \in V$ to $w \in V$ as an alternating sequence of vertices and edges, beginning with u and ending with w, such that each edge connects its preceding vertex with its succeeding vertex. The *length* of a path is the number of intervening edges. We denote by $d_G(u, w)$ the *distance* between u and w, i.e., the minimum length of any path connecting u and w in G, where by definition $d_G(v, v) = 0,\ \forall v \in V$ and $d_G(u, w) = d_G(w, u),\ \forall u, w \in V$. Note that the distance is not related to network link costs (e.g., latency): it is a purely abstract metric measuring the number of hops.

4.2. Measuring node significance

We mentioned in the introduction that all methods to date use the node ID or the node's degree in prioritizing the node for inclusion in the dominating set; e.g., [23,25]. Some methods first consider the node(s) which serve(s) as the only neighbor of a node in $N_{12}(\cdot)$ and then examine the node(s) with the maximum degree w.r.t. nodes not covered yet, whereas other methods simply consider the node(s) with the highest degree. None of these approaches is appropriate because: (a) the former methods treat nodes in a heterogeneous way, and (b) the latter methods, even though they are aware of the two-hop neighborhood, do not make full usage of the available information. In the sequel, we will present a new definition of a node's significance that avoids both drawbacks.

Let $\sigma_{uw} = \sigma_{wu}$ denote the number of shortest paths from $u \in V$ to $w \in V$ (by definition, $\sigma_{uu} = 0$). Let $\sigma_{uw}(v)$ denote the number of shortest paths from u to w that some vertex $v \in V$ lies on. Then, we define the *node importance* index $NI(v)$ of a vertex v as follows:

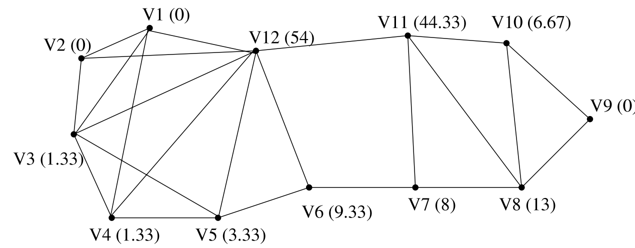

Fig. 1. Calculation of *NI* for a sample graph. Each node is characterized by a pair of IDs (*NI*).

Table 1
NI indexes of the nodes belonging to $LN_{V_{11}}$.

Node	$NI_{V_{11}}(n)$	Node	$NI_{V_{11}}(n)$	Node	$NI_{V_{11}}(n)$
V_1	0	V_5	0	V_9	0
V_2	0	V_6	6	V_{10}	7
V_3	0	V_7	6.33	V_{11}	44.33
V_4	0	V_8	12.67	V_{12}	81.67

Definition 2. The node importance index $NI(v)$ of a vertex v is equal to

$$NI(v) = \sum_{u \neq v \neq w \in V} \frac{\sigma_{uw}(v)}{\sigma_{uw}}. \quad (1)$$

Large values for the *NI* index of a node v indicate that this node v can reach others on relatively short paths, or that the node v lies on considerable fractions of shortest paths connecting others. Let us look at the *NI* indexes for the nodes of the graph presented in Fig. 1.

We observe that the *NI* index calculated over the whole graph captures structural features of the graph better than the node degree does. Moreover, it induces a *ranking* of the nodes according to their contribution in covering the whole network. Actually, the *NI* value identifies what we would call the **geodesic** nodes of the network, i.e., nodes that act as *articulation points*, or nodes with large degree relative to their neighbors. In this sense, it can be proved that *NI* generalizes the concept of node degree, as this has been used for clustering purposes so far.

The *NI* index would be useful in designing clustering protocols in sensor networks only if it captures structural features of small graphs, e.g., of the two-hop neighborhood of a node, and only if it can be computed really fast. If these conditions hold, then it can be used in designing localized algorithms. Fortunately, they both hold. The reader can easily verify that, for any node v, the *NI* indexes of the nodes in $N_{12}(v)$ calculated only for the subgraph LN_v reveal the relative importance of the nodes in covering the subgraph N_{12} (from v's point of view). For instance, the *NI* indexes for the nodes belonging to $LN_{V_{11}}$ (see Fig. 1) are indicated in Table 1. For a node u, which belongs to the two-hop neighborhood of a node v (or if $u \equiv v$), the *NI* index of u (calculated over LN_v) will be denoted as $NI_v(u)$.

The pseudo-code for the algorithm *ComputeNI* for the calculation of the *NI* index of a node can be found in [9]. The algorithm is capable of handling multiple shortest paths between two nodes; that is why some nodes have fractional *NI*.

Theorem 4.1 (*[9]*). *The complexity of the algorithm ComputeNI is* $O(n * m)$ *for a graph with n vertices and m edges.*

At first glance, the computation of the *NI* seems expensive, i.e., $O(m * n^2)$ operations in total for a two-hop neighborhood, which consists of n nodes and m links. Fortunately, we can do better than this by making some smart observations. We will not present the details here, but direct the readers to the work [22], which computes an index analogous to *NI* for the edges of a graph.

4.3. The clustering protocol

The clustering protocol is divided into two major procedures: the clustering formation procedure (CFP) and the network operation procedure (NOP). We assume that the duration of the clustering formation procedure T_{CFP} is the time interval needed by the clustering protocol in order to cluster the network, while the duration of the network operation procedure T_{NOP} is the time interval between two subsequent T_{CFP} intervals. Thus, the clustering protocol is divided into *rounds* where at the beginning of each round CFP is triggered, in order to select the optimum cluster heads for each individual node. The clustering formation procedure is followed by the network operation procedure when data are transferred from the nodes to cluster heads and through multi-hop paths to the information sink.

We assume a sensor network in which the nodes exchange with their neighbors "Hello" messages (beacon messages), which contain the list of their neighbors and their residual energy (E_{residual}). (Recall that the protocol works unchanged also in the case where the nodes simply notify of their existence, without informing about their neighbors.) We also consider that we are able to determine an assignment of time slots to the sensor nodes such that no interference occurs, i.e., no two nodes transmit in the same time slot. Such a scheme can be found using the D2-coloring algorithm from [17]. Thus, each node is able to form a graph that corresponds to its two-hop neighborhood (or its one-hop neighborhood). Additionally, each node, when it receives a packet, is able to figure out from which one-hop neighbor this packet was sent. The exchange of beacon messages with the list of one-hop neighbors that is being carried out among nodes is being performed only during the CFP of the first round of the sensor network operation. This is because the sensor nodes have no mobility and thus it is efficient to form the local graph once and transform it in the case of node failures. A node failure is considered when battery depletion is occurs. During the following CFPs in subsequent rounds the nodes exchange with their neighbors only E_{residual}.

4.3.1. Clustering formation procedure

As discussed above, clustering is triggered every T_{CFP} + T_{NOP} seconds to select new cluster heads. The CFP takes time T_C, which should be long enough for receiving messages from any neighboring node. The CFP combines the structural features of the local graph with E_{residual} for neighboring nodes in order to achieve the optimum selection of cluster heads for each individual node. The CFP includes the following phases.

Phase 1. Assuming that node v has just gathered the collection of its neighbors and their neighbors through beacon messages, it executes *CalculateNodeImportanceIndex* over its two-hop neighborhood graph LN_v. If node v does not have links to all the other nodes of the local network, then there exists at least one node u such that $u \in N_{12}(v)$, but $u \notin N_1(v)$. Therefore, the transmission range of v does not cover its two-hop neighborhood.

Phase 2. Then it runs a sorting algorithm to obtain a list of its neighbors, sorted in descending value of their $NI_v(\cdot)$. Note that the execution of any efficient sorting algorithm, e.g., *quicksort*, does not harm the computation complexity of the broadcasting scheme, since the sorting complexity is $O(n\log n)$, where n is the cardinality of the set $N_1(v)$.

Phase 3. While its two-hop neighborhood is not covered, examine one-by-one the members of the list obtained in Phase 2. If the currently examined one-hop neighbor u covers at least one two-hop neighbor, then designate the one-hop neighbor as a candidate cluster head node.

Phase 4. Then it runs a sorting algorithm to obtain a list of its candidate cluster head nodes, sorted in descending value of their E_{residual}. After examining one-by-one the candidate cluster head nodes, it selects those with maximum residual energy that cover the two-hop neighborhood as cluster heads. If two or more nodes have the same E_{residual}, then it selects the minimum set of one-hop neighbors that cover the two-hop neighborhood. This could be achieved if the nodes with equal E_{residual} are examined according to their *NI* values, starting from the node with maximum *NI* value towards the node with minimum *NI* value. If the currently examined one-hop neighbor u covers at least one (not covered yet) two-hop neighbor, then designate the one-hop neighbor as a cluster head node.

During the first round, all four phases of the CFP are being executed. All sensor nodes have the same amount of remaining energy. Thus, in *Phase* 4 only the *NI* values of neighboring nodes are considered in selection of the cluster heads. In the subsequent CFP executions (during following rounds) and until a neighboring node dies, only *Phase* 4 is being executed in order to elect the new cluster heads.

4.3.2. Network operation procedure

After the network is clustered, each node can communicate with the information sink. In order to achieve this, each node transmits the sensed data to the elected cluster heads. Once the cluster heads receive the data from neighboring nodes, it performs data aggregation to enhance the common signal and reduce the uncorrelated noise among the signals. The cluster heads aggregate the received messages and send the new data item to their cluster heads. This happens as follows. When a message arrives in a sensor node, say u, then u computes firstly the sensor nodes that received the message from the previous transmission. After that, u selects its cluster heads according to clustering formation procedure and retransmits the message to its cluster heads. Therefore, through multi-hop communication the source data from each network node reach the information sink.

Each time a cluster head is going to transmit a message, it checks whether there are any one-hop neighbors which have already broadcast the message. In the case of previous transmissions of the same message, the cluster head computes the part of the two-hop neighborhood that has not been covered yet. The current cluster head selects as next cluster heads only those significant nodes that cover the two-hop neighborhood that is uncovered. In order to avoid rooting loops, each node is equipped with a local cache. The new messages that arrive at the node are cached in the local cache. Thus, the duplicated messages are discarded by the nodes, since a copy of them has already been transmitted. The cache size is assumed large enough and so the replacement of messages is performed rarely, based on the *FIFO* technique.

Each sensor node is aware of its two-hop neighborhood through the beacon messages that it exchanges with its one-hop neighbors. Thus, each node can recognize whether the information sink is in the two-hop distance. Therefore, each cluster head that is a two-hop neighbor of the information sink selects the one-hop neighbor with the maximum remaining energy that is also a one-hop neighbor of the information sink to retransmit the message. This operation achieves the load balancing of the energy dissipation in one-hop neighbors of the information sink. A sensor node is considered "dead" if it has lost all of its initial energy. Each node that has lost 99.99 per cent of its initial energy considers itself a "dead" node and transmits to one-hop neighbors a "DEAD" message. The nodes that receive the message delete from the neighbor list the "dead" node and those two-hop neighbors that are covered only by the "dead" node. Finally, they execute the CFP in order to elect the new cluster heads.

The proposed clustering protocol is *dynamic* or *source dependent* and energy-efficient. The elected cluster heads depend on the *location of the source, the residual energy and the progress of the*

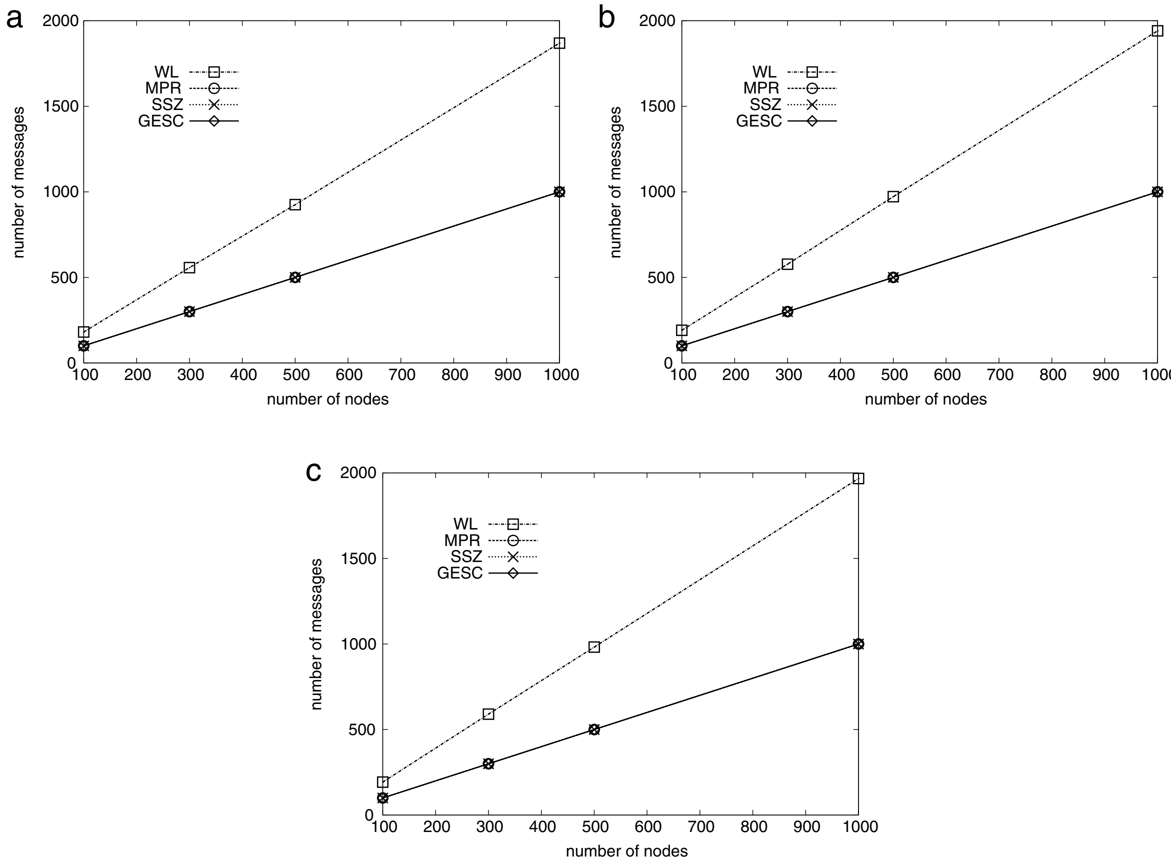

Fig. 2. Impact of the nodes' number on cluster formation for: (a) degree equal to 4, (b) degree equal to 7, (c) degree equal to 10.

network operation, avoiding thus the effect of "hot-spots". *GESC* is dynamic, since every intermediate sensor node that has been selected to retransmit a message by a neighboring node will transmit the message to its cluster heads that may differ from the elected cluster heads of neighboring nodes. This is because every node has a different local network view and, according to it, decides which nodes are more important. The protocol can operate in a centralized way assuming that each node is aware of the whole network graph and computing the significant node in the whole network. However, this approach has two major drawbacks for being proposed in a WSN. The communication cost will be extremely high for achieving each node being aware of all the other nodes of the network. Thus, the energy dissipation in sensor nodes will be increased and the reclustering operation will not be efficient. The second major drawback is that the cluster heads will be the same for all sensor nodes and therefore they run out of energy very soon.

Theorem 4.2. *The GESC algorithm is reliable, in the sense that the broadcast packet can be disseminated to every node in the network (if it is connected).*

Proof. We prove the theorem by contradiction. Consider a node u in the random network that receives the message using a flooding protocol but not using *GESC*. We call the message originator node the source node and the node u under consideration the destination node. Since the destination node has received the message in flooding, there exists a path from the source to the destination. That the destination node has not received the message under *GESC* implies that none of the neighboring nodes has been elected as a cluster head node. Thus, even if a node received the message, it has not retransmitted it because it is not a cluster head node. However, according to the *GESC* protocol a node v elects as cluster heads its one-hop neighbors that cover the two-hop neighborhood (*Phase 3 and Phase 4 of the clustering formation procedure*). This implies that the destination node does not belong to the two-hop neighborhood of any other node of the network, or none of the one-hop neighbors of the destination node have received the message. In the first case the destination node will be either a one-hop neighbor of the source node or the local network that constitutes the destination node and its one-hop neighbors will be disconnected from the entire network. Both of these are contradictions. According to the second case, none of the one-hop neighbors of the destination node received the message. This implies that none of its two-hop neighbors has received the message since some of them will have been elected as cluster heads and they retransmit the message. By continuing in a similar fashion, we can show that the cluster heads of the source node also did not receive the message. This implies that no message has been flooded in the network, which is a contradiction. □

5. Performance evaluation

We conducted several experiments to evaluate the performance of our algorithm and compare it to other protocols. All the protocols have been implemented using the J-Sim simulation library [24] with the ADOV routing protocol. The experiments were carried out in two major phases. In the first phase we compare *GESC* with protocols from the mobile ad hoc networks (MANETs) family in order to show that the *GESC* protocol overcomes the major shortcoming of this type of algorithm. We examined the most efficient algorithms reported in [7,33], and thus we compare *GESC* to MPR [23], WL [29] and with SSZ ([25], which was selected as a Fast Breaking Paper in Computer Science for October 2003). In the second phase (Section 6), we compare *GESC* with

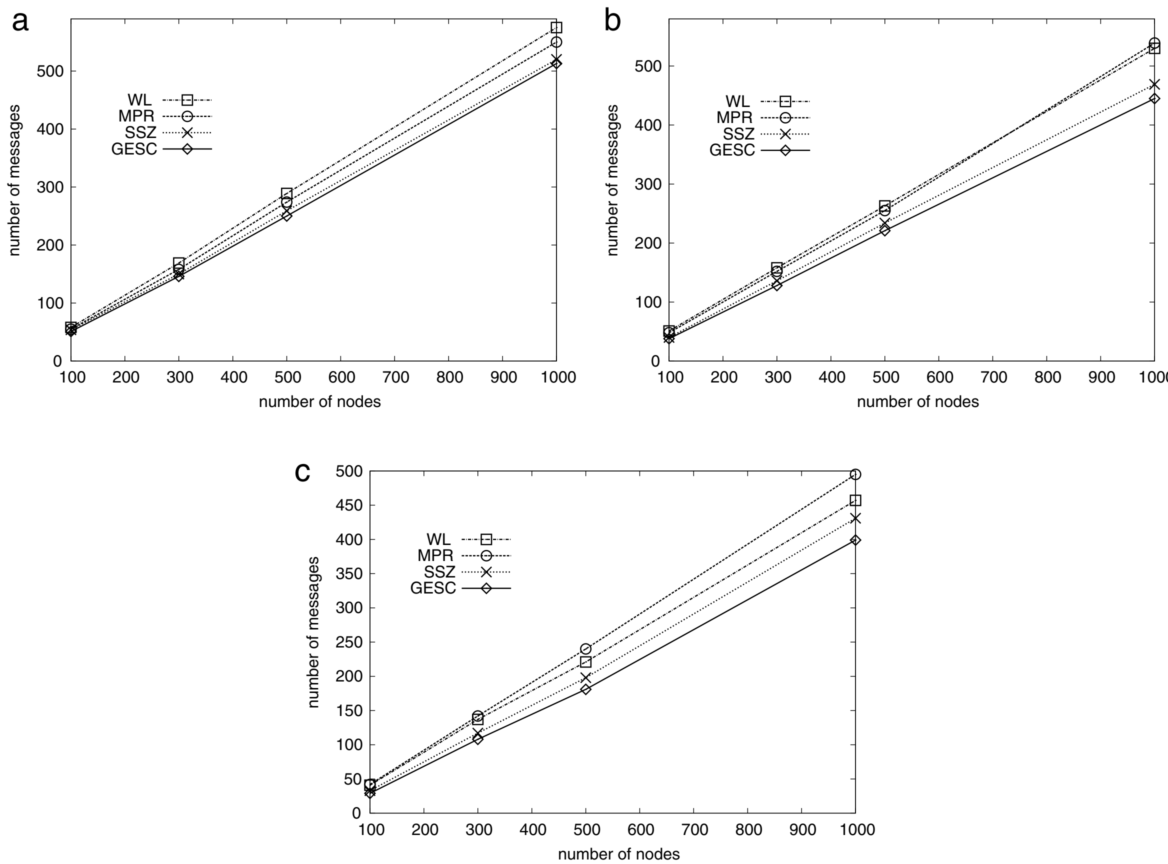

Fig. 3. Impact of the nodes' number on transmission for: (a) degree equal to 4, (b) degree equal to 7, (c) degree equal to 10.

a multi-hop energy-efficient version of the LEACH [19] protocol, well-known from the literature, from the wireless sensor network family. LEACH was proposed as an application specific clustering protocol. Thus, we compare *GESC* with LEACH in the following section using an experimental setup similar to that described in [19]. The experimental results attest that the *GESC* protocol, which takes into consideration topological features of the nodes and the remaining energy, achieves a better performance than LEACH in terms of network lifetime and number of messages that reach the base station.

This section contains the comparison of the *GESC* protocol with MANET protocols reported in [23,25,29]. We measure, as performance metrics, the messages broadcast during the protocols' network operation, the number of messages exchanged during the protocols' clustering formation and finally the energy dissipated. The number of broadcast messages indicates the performance of clustering protocols for achieving a full coverage of the sensor network. This is because a protocol may select more effectively the cluster head nodes that are responsible for forwarding the messages, and thus the number of message retransmissions is reduced. The energy dissipated metric indicates the residual energy of each node when the network operation has finished. The cluster head nodes belonging to the DS are solely responsible for carrying out all communication, thus running out of energy very soon. Therefore according to the energy dissipated metric, we can work out whether the cluster head selection policy of each protocol is energy-efficient. Finally, the clustering formation metric indicates the number of messages that have to be exchanged among nodes in order for the network to be clustered and for cluster heads to be selected. The lesser the number of messages required during the clustering formation the greater the efficiency of the protocol. The protocols' network operation is defined as the message dissemination from an originator node to all the other nodes of the network until a full coverage of the network is achieved. This network operation is desirable in applications where many information sinks are dispersed in the whole network and their locations are unknown to the sensor networks.

We tested the protocols for a variety of sensor network topologies with 100, 300, 500 and 1000 nodes, to simulate sensor networks with varying levels of node degree, from 4 to 10. In order to achieve the protocols' network operation, we run each protocol at least 100 times for each different node degree, before computing the averages of the number of broadcast messages diffused into the network and the average consumed energy. In each run, a different node is selected to start the broadcasting process until all network nodes have received the transmitted data item. Each data packet has size 100 bytes, while the header has size 25 bytes.

The network topology consists of many square grid units where one or more nodes are placed. The number of square grid units depends on the number of nodes and the node degree. The topologies are generated as follows: the location of each of the n sensor nodes is uniformly distributed between the point ($x = 0, y = 0$) and the point ($x = 500, y = 500$). The average degree d is computed by sorting all $n * (n-1)/2$ edges in the network by length, in increasing order. The grid unit size corresponding to the value of d is equal to $\sqrt{2}$ times the length of the edge at position $n * d/2$ in the sorted sequence. Two sensor nodes are neighbors if they placed in the same grid or in adjacent grids. The simulation area is assumed of size 500 m $\times$ 500 m and is divided into equal sized square grid units. Beginning with the lower grid unit, the units are named 1, 2, . . ., in a columnwise fashion.

The network is generated as above with the precondition that it is connected and the sensor nodes are static. Each node is assigned a unique ID and x, y coordinates within the simulation area. The radio characteristics used in our simulations are summarized in

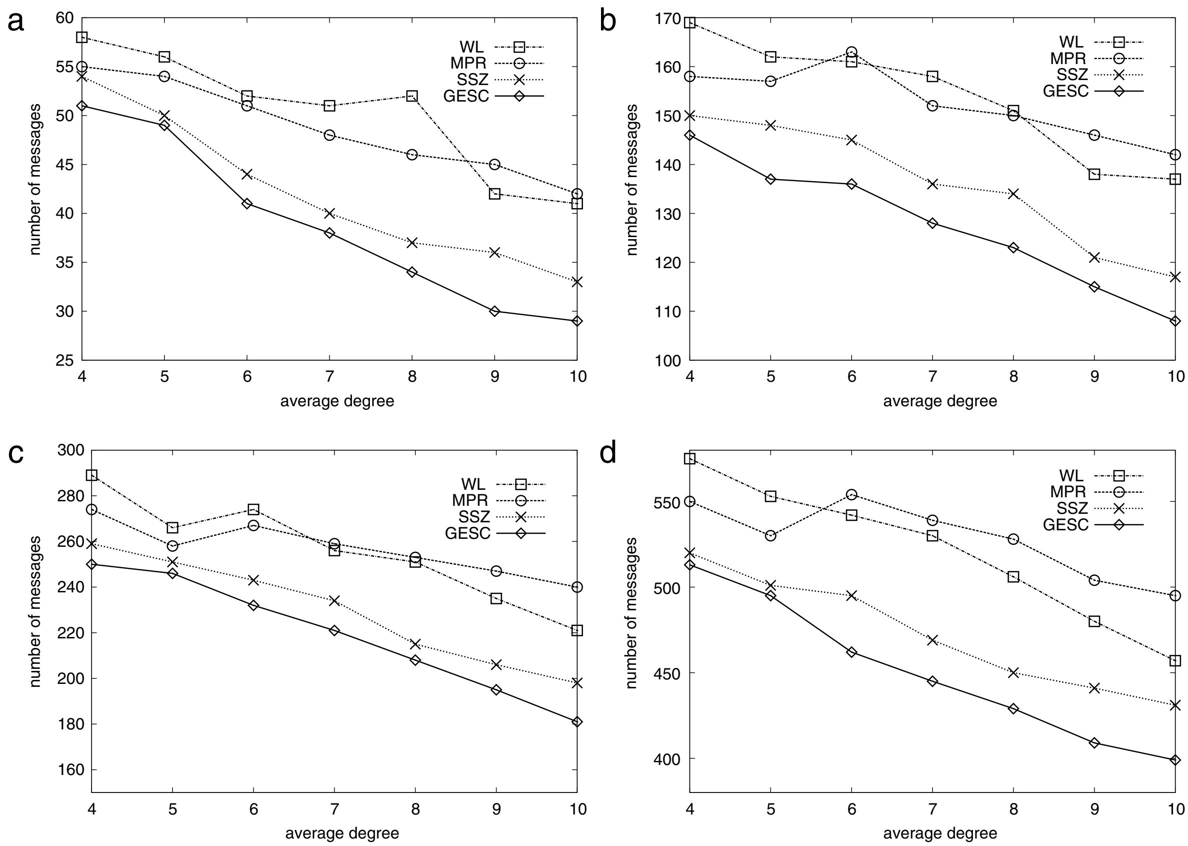

Fig. 4. Impact of the average node degree on the number of transmissions for: (a) a network of 100 sensor nodes, (b) a network of 300 sensor nodes, (c) a network of 500 sensor nodes, (d) a network of 1000 sensor nodes.

Table 2
Radio characteristics used in our simulations.

Operation	Energy dissipated
Transmitter/receiver electronics	$E_{elec} = 50\ nJ/bit$
Transmit amplifier if $d_{toBS} \le d_0$	$e_{fs} = 10\ pJ/bit/m^2$
Transmit amplifier if $d_{toBS} \ge d_0$	$e_{mp} = 0.0013\ pJ/bit/m^4$
Data aggregation	$E_{DA} = 5\ nJ/bit/signal$

Table 3
Number of cluster heads for a WSN with 100 nodes.

Method/degree	4	5	6	7	8	9	10
WL	58	56	52	51	52	42	41
MPR	55	54	51	48	46	45	42
SSZ	54	50	44	40	37	36	33
GESC	51	49	41	38	34	30	29

Table 2. Finally, the initial energy of each sensor node is set to $E_0 = 2\,J$.

5.1. Evaluation

We performed a large number of experiments, varying the size of sensor networks (in terms of the number of its sensor nodes). In particular, we performed experiments for 100, 300, 500 and 1000 sensors, for average sensor node degree from 4 (sparse sensor network) to 10 (very dense sensor network).

5.1.1. Impact of the number of nodes

The first experiment evaluated the impact of the number of nodes of the sensor network on the number of messages generated during the cluster procedure (Fig. 2) and during the network operation (broadcast) procedure (Fig. 3). We can easily work out the linear dependence of the number of transmitted messages on the network size and the efficiency of the *GESC* protocol for the broadcasting procedure, since it generated few clusters, and thus cluster heads, which are responsible for forwarding any messages. *GESC* always performs from 4% to 15% better than the second-best performing algorithm (SSZ) no matter what the scale of the network is (in terms of number of nodes). Fig. 3a illustrates the number of transmitted messages when the average node degree is equal to 4. The *GESC* protocol performs slightly better than the second-best performing algorithm SSZ. However, for dense network topologies (Fig. 3(b)) the performance gains of the *GESC* protocol over SSZ protocol increased. In Fig. 3(c) finally, we notice that for very dense network topologies the *GESC* protocol performs 15% better than the SSZ protocol. We can observe that the difference between the two protocols becomes larger as the numbers of nodes participating in the network topology and node degree increase (see Tables 3 and 4). This is because as the node degree increases, the number of shortest paths that may be recognized in a localized two-hop neighborhood is also increased. Thus for dense and very dense sensor network topologies, *GESC* identifies more effectively the significant nodes, because the number of nodes that participate in local neighborhoods increases and therefore the number of shortest paths that participate in computation of the *NI* also increases. The increased number of shortest paths offers more efficient results in computation of the node significance and thus better decision support in the election of cluster heads and message retransmissions.

It is also worthy of note that all methods generate the same number of messages during cluster formation except the WL method (Fig. 2(a), Fig. 2(b), Fig. 2(c)), since they all have to learn their immediate (one-hop or two-hop) neighbors, and apparently this number does not depend on the connectivity (average degree)

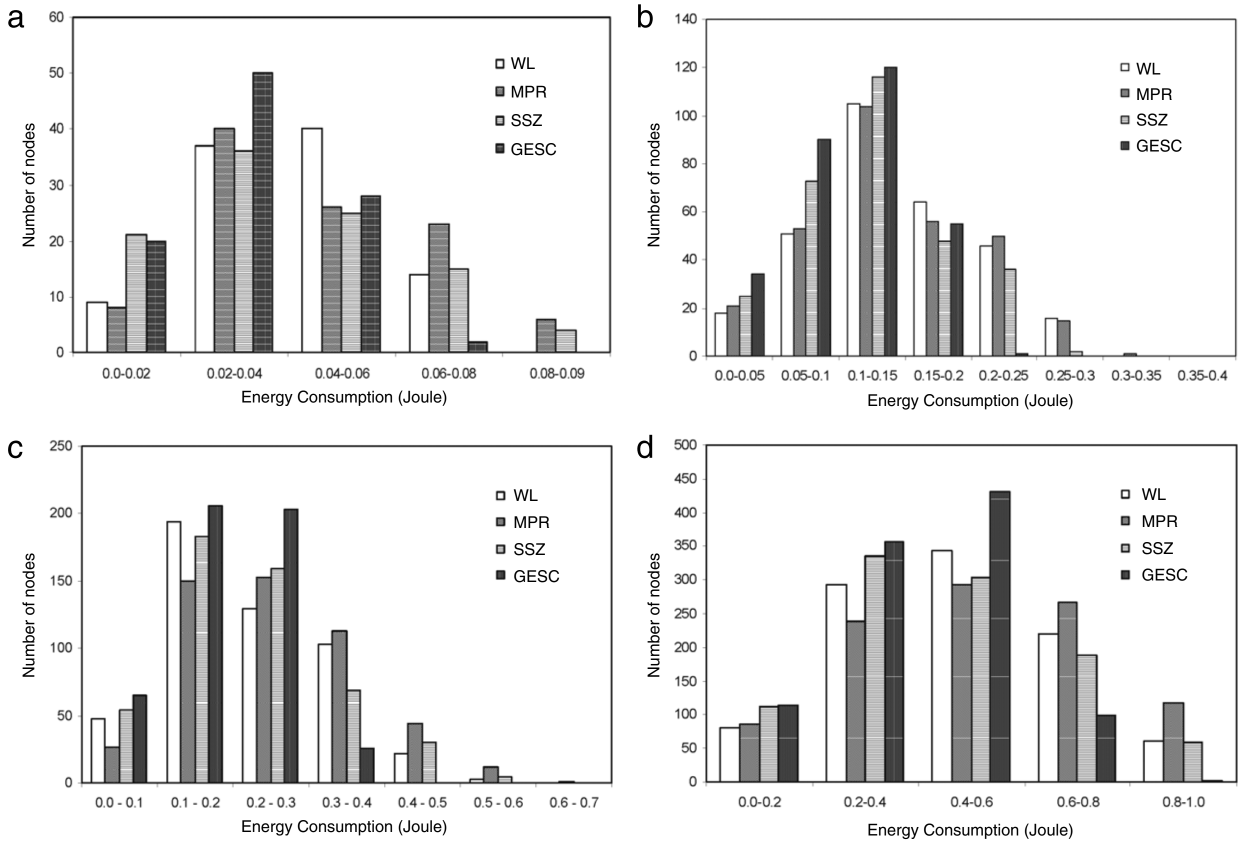

Fig. 5. Residual energy for degree equal to 10 for: (a) a network of 100 sensor nodes, (b) a network of 300 sensor nodes, (c) a network of 500 sensor nodes, (d) a network of 1000 sensor nodes.

Table 4
Number of cluster heads for a WSN with 500 nodes.

Method/degree	4	5	6	7	8	9	10
WL	289	266	274	256	251	235	221
MPR	274	258	267	259	253	247	240
SSZ	259	251	243	234	215	206	196
GESC	250	246	232	221	208	195	179

Table 5
Average number of nodes in the clusters for a WSN with 100 nodes.

Method/degree	4	5	6	7	8	9	10
WL	1.74	1.81	1.96	1.99	1.95	2.41	2.47
MPR	1.84	1.87	1.98	2.10	2.20	2.25	2.41
SSZ	1.87	2.02	2.30	2.53	2.74	2.81	3.06
GESC	1.99	2.08	2.51	2.66	2.97	3.37	3.49

Table 6
Average number of nodes in the clusters for a WSN with 500 nodes.

Method/degree	4	5	6	7	8	9	10
WL	1.87	2.05	1.95	2.12	2.21	2.42	2.71
MPR	1.95	2.12	2.07	2.14	2.19	2.25	2.33
SSZ	2.14	2.21	2.29	2.42	2.81	2.96	3.11
GESC	2.21	2.27	2.43	2.71	2.92	3.18	3.36

of the network. Finally, in Figs. 5–7 we present the impact of the number of nodes on the energy dissipation. As the number of nodes increases the energy dissipation is also increased (see Tables 5 and 6). According to our simulation setup, for 100 sensor nodes in the network topology there will be 100 messages diffused in the network, while for 1000 sensor nodes in the network topology the diffused messages increase to 1000. The second-best performing protocol (SSZ) maintains the same elected cluster heads during the whole simulation time. This has the result that the same nodes carried out all the communication load in terms of message retransmissions. However, the *GESC* protocol, due to the dynamic and independent election of cluster heads by each node, performs better in terms of energy dissipation. Each node retransmits a message to its cluster heads. The fact that neighboring nodes could have different cluster heads, since the cluster heads are elected by each node according to the localized network topology and the remaining energy of neighboring nodes, results in smaller energy consumption.

5.1.2. Impact of the average degree of the nodes

The second experiment evaluated the impact of the average node degree on the number of messages during the broadcasting procedure (Fig. 4). We present the results concerning a sensor network with 100 (Fig. 4(a)), 300 (Fig. 4(b)), 500 (Fig. 4(c)), and 1000 nodes (Fig. 4(d)). Again, *GESC* is the best performing algorithm, with gains up to 15% relative to the second-best performing method. In all figures, we observe that *GESC* and SSZ have a significant difference when the degree is larger than 7. For small node degrees, their performance is almost equivalent, but with larger degrees the performance of *GESC* gets steadily better than SSZ's. In the *GESC* protocol for small values of the node degree (sparse networks), the local network view that a sensor node has is limited. This results in the election of many cluster heads. Therefore, the broadcast procedure includes a greater number of retransmissions than those in dense and very dense networks. Although the number of retransmissions of the *GESC* protocol is increased for sparse networks, it continues to perform better than the competitive protocols. However, for dense and very dense networks *GESC* performs significantly better than SSZ, since the significance of nodes can be revealed better. The figures also

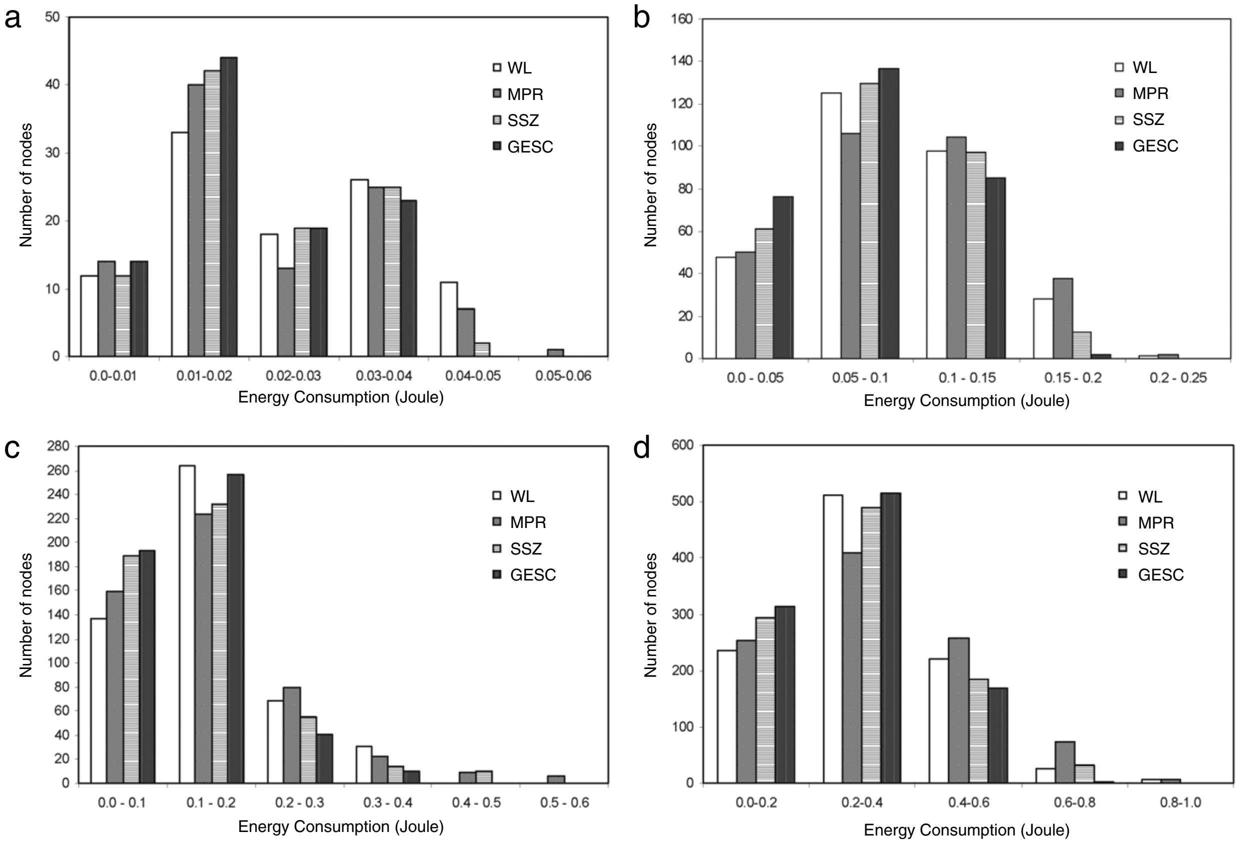

Fig. 6. Residual energy for degree equal to 7 for: (a) a network of 100 sensor nodes, (b) a network of 300 sensor nodes, (c) a network of 500 sensor nodes, (d) a network of 1000 sensor nodes.

present a better performance of the WL protocol than the MPR protocol for large degrees.

5.1.3. Impact on the energy consumption

Our last experiment investigated the issue of energy consumption. We examined the *residual energy* of each sensor node after all experimental tests. We present the results for sensor networks of 100 sensor nodes, 300 sensor nodes, 500 sensor nodes and 1000 sensor nodes with degree equal to 10 (Fig. 5), with degree equal to 7 (Fig. 6) and with degree equal to 4 (Fig. 7). In each figure, the first column corresponds to the WL protocol, the second column corresponds to the MPR protocol, the third column corresponds to the SSZ protocol and the fourth column corresponds to our *GESC* protocol. For degree equal to 10 (Fig. 5), we can readily observe the "energy starvation" that SSZ, WL and MPR cause in some nodes, whereas *GESC* achieves maintaining more balance in the energy of the nodes. In the *GESC* protocol there are many nodes that have consumed a small amount of energy while only a few of them have consumed a significant amount of energy. In contrast, the other three protocols and especially WL and MPR cause in many nodes energy dissipation. This is because each node running the *GESC* protocol can dynamically and independently elect the cluster heads using the residual energy of neighboring nodes and the structural characteristics of the local network graph. Therefore, the elected nodes that retransmit the messages are not always the same for neighboring nodes. When the sensor network topology is not very dense (Fig. 6) and especially in the case of sparse network deployment (Fig. 7), the *GESC* protocol also performs better than the SSZ protocol. However the difference in energy consumption that is caused for the nodes is not very big. Finally, we can notice that *GESC* is scalable in the number of nodes and the node's degree.

6. Clustering applications

LEACH [19] is one of the most popular clustering algorithms for WSNs that are proposed for prolonging network lifetime. LEACH was proposed for a specific application in which sensor nodes are randomly distributed in a sensor field and are continuously sensing the target area in order to send messages to a base station (BS). In this section we compare our *GESC* protocol with LEACH and HEED in order to attest firstly that *GESC* can be used effectively in specific sensor applications like that described in [19], and to exhibit secondly the superiority of the *GESC* protocol over the algorithms that use the remaining energy of each node in order to direct its decision about whether it will elect itself as a cluster head node or not.

LEACH forms clusters on the basis of the received signal strength and uses the cluster head nodes as relays to the base station. All the data processing, such as data fusion and aggregation, is local to the cluster. LEACH forms clusters by using a distributed algorithm, where nodes make autonomous decisions without any centralized control. Initially a node decides to be a cluster head with a probability p and broadcasts its decision. Each non-cluster head node determines its cluster by choosing the cluster head that can be reached using the least communication energy. The role of being a cluster head is rotated periodically among the nodes of the cluster in order to balance the load. The rotation is performed by getting each node to choose a random number "T" between 0 and 1. The authors considered that the cluster head node has a long communication range so that the data can reach the BS directly. However, this is not a realistic assumption since the cluster heads are regular sensors and the BS is often not directly reachable for all nodes. Additionally, the authors assumed a single-hop communication for both intracluster and intercluster communication. Thus,

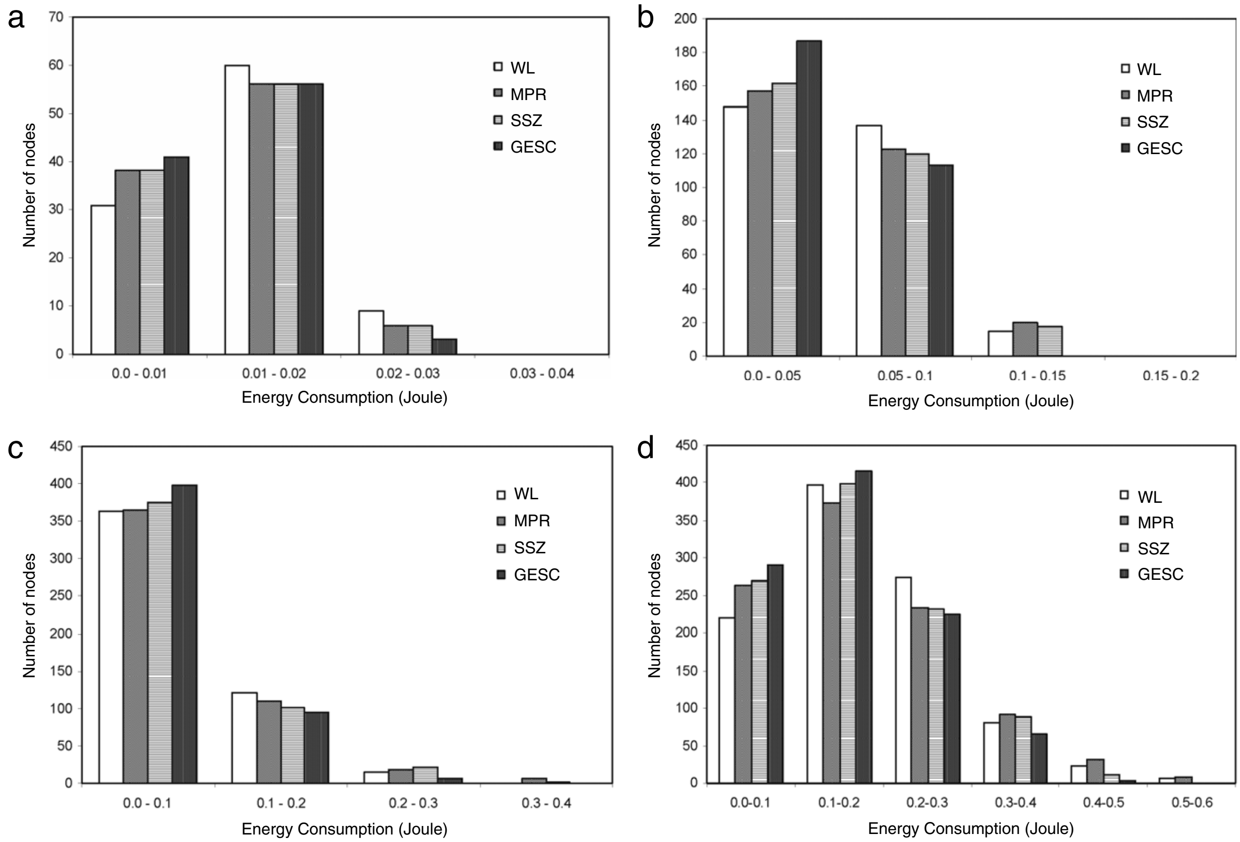

Fig. 7. Residual energy for degree equal to 4 for: (a) a network of 100 sensor nodes, (b) a network of 300 sensor nodes, (c) a network of 500 sensor nodes, (d) a network of 1000 sensor nodes.

Fig. 8. Performance of *GESC* on network applications. (a) Network lifetime (first node death). (b) Network lifetime (last node death).

a node can reach the BS directly. Therefore, LEACH is inapplicable for WSNs deployed in large regions. In our simulations we compare our *GESC* clustering protocol to a multi-hop version of LEACH described in [19]. The feature that has been added to the application-specific LEACH protocol is that each node selects a cluster head in its cluster range proximity which is not assumed to span the entire network area. The HEED [32] protocol introduces a variable known as the cluster radius which defines the transmission power to be used for intracluster broadcasting. The initial probability for each node to become a tentative cluster head depends on its residual energy, and final cluster heads are selected according to the intracluster communication cost. HEED relies on the assumption that cluster heads can communicate with each other and form a connected graph.

We tested the protocols for a variety of sensor network topologies with 300, 500 and 1000 nodes, to simulate sensor networks with varying levels of node degree, from 4 to 10. The sensor network field is assumed of size 500 m $\times$ 500 m and the network topologies are generated as in Section 5. Additionally, the BS is in the center of each topology ($x = 250, y = 250$) and the initial energy of each sensor node is set to $E_0 = 2\,J$. The radio characteristics used in our simulations are similar to those in [19] and are listed in Table 2. Each data packet has size 100 bytes, while the header has size 25 bytes. Finally, the duration of the cluster construction phase for both protocols is set to 5 s and the duration of the network operation phase is set to 20 *seconds*. For multi-hop LEACH k_{opt} is selected to be 11 for network topologies with 300–1000 nodes according to k_{opt} computed in [19].