A distributed energy-efficient clustering protocol for wireless sensor networks

Ali Chamam*, Samuel Pierre

Department of Computer Engineering, Ecole Polytechnique of Montreal, C.P. 6079, succ. Centre-Ville, Montreal (QC), Canada H3C 3A7

Abstract

Minimizing energy dissipation and maximizing network lifetime are among the central concerns when designing applications and protocols for sensor networks. Clustering has been proven to be energy-efficient in sensor networks since data routing and relaying are only operated by cluster heads. Besides, cluster heads can process, filter and aggregate data sent by cluster members, thus reducing network load and alleviating the bandwidth. In this paper, we propose a novel distributed clustering algorithm where cluster heads are elected following a three-way message exchange between each sensor and its neighbors. Sensor's eligibility to be elected cluster head is based on its residual energy and its degree. Our protocol has a message exchange complexity of $O(1)$ and a worst-case convergence time complexity of $O(N)$. Simulations show that our algorithm outperforms EESH, one of the most recently published distributed clustering algorithms, in terms of network lifetime and ratio of elected cluster heads.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

Wireless sensor networks consist of a large number of low-power, limited-processing capability, MEMS (Micro-Electromechanical Systems) capable of measuring and reporting physical variables related to their environment. In surveillance applications, sensors are deployed in a certain field to detect and report events like presence, movement or intrusion in the monitored area [1]. As depicted in Fig. 1, data collected by sensors is transmitted to a special node equipped with higher energy and processing capabilities called "Processing Node" (PN) or "sink" [1]. The PN collects, filters and aggregates data sent by sensors in order to extract useful information. Due to their energy constraint, wireless sensors usually have a limited transmission range making multi-hop data routing toward the PN more energy-efficient than one-hop transmissions. Unlike in wired networks or in regular wireless networks (cellular, WLAN, etc.), energy conservation in wireless sensor networks is a critical issue that has been addressed by substantial research works [2,3]. Generally, energy conservation is dealt with on five different levels [1,2]:

1. efficient scheduling of sensor states to alternate between sleep and active modes;
2. efficient control of transmission power to ensure an optimal tradeoff between energy consumption and connectivity;
3. data compression (source coding) to reduce the amount of uselessly transmitted data;
4. efficient channel access and packet retransmission protocols on the Data Link Layer;
5. energy-efficient routing, clustering and data aggregation.

This paper addresses the fifth issue and more specifically energy-efficient clustering. Clustering has been proven energy-efficient in sensor networks [1,9,10]. In fact, CHs can process, filter and aggregate data sent by sensors belonging to their...
cluster, thus reducing network load and alleviating the bandwidth. Besides, clustering provides for substantial energy saving [4,10] since non-CH sensors are not involved in routing and relaying data and transmissions are only operated by CHs. Yet, since CHs consume more energy in aggregating and routing data, it is important to have an energy-efficient mechanism for CH election and rotation [5,7]. In flat networks, data is routed from sensors to the PN through peer sensors using one of the many routing protocols proposed in the literature [6]. In contrast, in hierarchical (clustered) sensor networks, CHs transmit aggregated data to the PN, either directly (one-hop) [5] or in a multi-hop fashion [8]. For the sake of minimizing energy consumption, both optimal number of CHs and their optimal placement have to be sought.

In this paper, we propose a distributed clustering protocol for sensor networks, called Energy-Efficient Cluster Formation protocol (EECF). We suppose that any sensor can be either a regular sensor or a CH and that every regular sensor has to be connected to a CH, i.e., it has to have at least one CH in its fixed transmission range. We also suppose that only CHs can route data (because they have enough energy to do so and because this limits broadcasting). In EECF, the set of CHs is elected following a three-way message exchange between every sensor and its one-hop neighbors. CH election is mainly based on sensors’ respective residual energies and degrees. Run on a flat network, our protocol ends up with a clustered (admissible) sensor configuration that maximizes the network lifetime, while ensuring, under a certain condition, the connectivity of the overlay network composed of CHs.

This paper is organized as follows: in the next section, we present some related work. In Section 3, we outline our problem and enumerate our assumptions. In Section 4, we describe EECF in detail. In Section 5, we provide an analytical study of the complexity of EECF and the connectivity of the solutions. In Section 6, we discuss our simulation results. Finally, Section 7 concludes this paper and points out some future directions.

2. Related work

Energy-efficient clustering algorithms for wireless sensor networks have been widely addressed in literature. The main goal of clustering is to efficiently maintain the energy consumption of sensor nodes by involving them in multi-hop communication within a particular cluster and by performing data aggregation and fusion in order to decrease the number of transmitted messages to the sink. Cluster formation is typically based on the energy reserve of sensors and sensor’s proximity to the CH [9]. For instance, Low-Energy Adaptive Clustering Hierarchy (LEACH) [5], one of the first clustering algorithms proposed for sensor networks, is a distributed, proactive, dynamic algorithm that forms clusters of sensors based on the received signal strength and uses local CHs as routers to the sink. Each node makes its own decision whether to become CH based on how often and the last time it has been CH but also on the optimal percentage of CHs in the network (pre-determined value). Transmissions are operated only by CHs which saves energy. LEACH provides a balance of energy consumption through a random rotation of CHs. However, CHs transmit data directly to the sink, which can be energy-consuming in large-scale sensor networks. Power-efficient Gathering in Sensor Information Systems (PEGASIS) [11] and its variation Hierarchical-PEGASIS are two improvements of LEACH. Rather than forming multiple clusters, PEGASIS forms chains of sensor nodes so that each sensor transmits and receives from a neighbor and only one node is selected from that chain to convey data to the PN. Still, communication between the elected CH and the PN is one-hop, which may waste energy and prove to be unsuitable for large-sized networks. Weighted Clustering Algorithm (WCA) [12] is a reactive clustering algorithm where cluster election is based on the evaluation, for every sensor, of a score function called “combined weight”. This function is a weighted linear combination of the degree, the mobility level, the transmission power and the residual energy of the sensor. Every sensor broadcasts its combined weight to its neighbors and the sensor having the lowest weight is elected CH. Hybrid Energy-Efficient Distributed Clustering (HEED) [13] is a distributed clustering protocol that uses a hybrid combination of the residual energy and the intra-cluster communication cost as attribute for cluster head selection. HEED ensures a uniform distribution of CHs across the network and adjusts the probability of CH-selection to ensure inter-CH connectivity. In its initialization phase, HEED allows sensors to compute a probability of becoming CH, proportional to its residual energy and to a pre-determined percentage of CHs. Then, during a repetition phase, sensors seek the best CH to connect to. If no CH is found, the sensor doubles its probability to become CH and broadcasts it again to its neighbors, and so forth. This phase stops either when this
probability equals 1 (i.e: the sensor elects itself as CH) or when it finds a CH to connect to. Distributed Weight-based Energy-Efficient Hierarchical Clustering (DWEHC) [14] is a distributed clustering algorithm similar to HEED. In DWEHC, every sensor computes its score as a function of its residual energy and its proximity to its neighbors. In a neighborhood, the sensor with the largest score will be elected CH and all the neighborhood members become attached to that CH. Then, every non-CH member tries to find a “cheaper” way to reach the CH through its non-CH neighbors. These neighbors broadcast their cost to reach the CH and the sensor can then decide whether to use a multi-hop or a direct connection to the CH. Energy-efficient Hierarchical Clustering (EEHC) [15] is another distributed clustering algorithm that performs single and multi-level clustering. In the single-level process, every node announces itself to its k-hops neighbors (by forwarding, for k > 1) as a “volunteer” CH, with probability p. Then, every sensor connects to the closest CH. Sensors that do not receive any announcement and are not CHs become “Forced CHs”. The same process is extended to the multi-level clustering with the exception that only designated CHs participate in the process of election of the second-level cluster set. Energy-efficient Strong Head clustering (EESH) [17] is a recently published clustering protocol. In EESH, nodes are promoted CHs according to their respective residual energies, their respective degrees and the distance to and the residual energy of their neighbors. For that, EESH evaluates a cost function for every sensor in the network and iteratively elects the node having the greatest cost as CH. This process terminates when all the sensors in the network are connected to at least one CH. EESH has been shown to outperform HEED and LEACH, so we used it as a comparative base in our performance evaluation.

3. Problem description and assumptions

As discussed in Section 1, clustering has been proven energy-efficient in sensor networks [5,7,10,13]. During the clustering process, it is necessary to take into account aspects such as: cluster size and form, criteria for CH selection, how to control inter-cluster and intra-cluster collisions and energy saving issues. The design of the clustering process is one of the more important issues for the correct functioning of the network, due to the probed efficiency of using a hierarchical communications scheme. In this paper, we consider a sensor network randomly deployed in a certain area and we propose an Energy-efficient Cluster Formation protocol (EECF) that maximizes network lifetime. Run on a flat topology of sensors, EECF ends up with a hierarchical topology in which sensors are organized into clusters having, each, one sensor promoted as CH and all the other regular sensors connected to the closest CH. CHs also act as data relays since they route data received from peer CHs toward the PN, either directly if the PN is within their range, or through a neighbor CH if not. To control the sensors that perform data relaying, we restrict the routing task to CHs, since they are more eligible than the other nodes in terms of residual energy. Therefore, a route must exist from any CH to the PN. On the other hand, because of the energy-intensive data processing and routing tasks, a CH sensor consumes much more energy than regular sensors. If CHs were chosen a priori and fixed throughout the system’s lifetime, these nodes would quickly exhaust all their limited energy making them no longer operational and therefore, all the nodes that belong to the cluster would lose their communication ability. In this paper, we dynamically designate CHs so that the elected CH sets are adaptive to the change in the residual energies and the network configuration.

In this paper, we make the following assumptions:

1. The ID and the position of each sensor are fixed and known to both the PN and the sensor itself.
2. Active sensors capture events occurring in their range and transmit data associated with these events straightaway, without any buffering, because sensors are usually not equipped with large (and costly) buffers. However, for some channel access protocols, like TDMA, some little buffers are still needed to hold data waiting to be sent.
3. All sensors have the same sensing range $R_s$.
4. All regular (non-CH) sensors have the same transmission range $R_t$ and consume the same power to transmit one bit of data. Likewise, all CHs have the same transmission range $R_t^{CH}$ and consume the same power for transmitting one bit of data. Typically, $R_t^{CH} \gg R_t$.
5. CHs consume more energy during a time period than regular sensors, since they have a higher transmission range and since they perform additional data-related tasks like aggregation, filtering and routing.
6. Each sensor has an initial energy $E_0$. The PN is assumed to have no energy limitation. We assume that regular sensors dissipate the same energy during a unit of time, no matter how the events’ distribution is.
7. Network lifetime is defined as the time separating the instant the network starts operating and the instant the network is no more connected (because of the failure of one or more sensor).
8. Sensors have idealized sensing capabilities. That is, inside the sensor’s range, quality of sensing is not altered with distance from the sensor.
9. We assume ideal MAC layer conditions, that is, perfect transmission of data on a node-to-node wireless link.

4. Description of the EECF protocol

EECF is fully distributed. As shown in Fig. 2, it consists of a number of actions and message exchanges executed by each sensor, which leads to the election of a set of CHs and ensures that each regular sensor (not CH) is connected to a CH.
We will show later that, under a certain condition, any configuration provided by EECF will exhibit a spanning tree connecting all CHs (i.e. a route exists between any pair of CHs). This makes the logical overlay network formed by the set of CHs sufficient to route data from any sensor toward the PN. Here is the description of EECF:

1. Firstly, each sensor broadcasts to all its neighbors a message called Score Advertisement Broadcast (SAB) containing a score function expressing its ability to be promoted CH. In EECF, the score function advertised by each sensor through the SAB is a linear combination of its degree and its residual energy. It is expressed by:
   \[
   \text{Score}(i) = a \frac{\text{degree}(i)}{N} + (1 - a) \frac{\text{Er}(i)}{E_0}.
   \]
   The coefficient \(a\) will weigh the importance of the residual energy with respect to the number of neighbors.
   We will discuss the choice of the value of \(a\) later.

2. After receiving a SAB from all its neighbors, each node \(i\) partitions the set of its neighbors into two subsets:
   \[N^+(i) = \{ v \in \text{Neighbors}(i) \text{ such that Score}(v) \geq \text{Score}(i) \}\]
   \[N^-(i) = \{ v \in \text{Neighbors}(i) \text{ such that Score}(v) < \text{Score}(i) \}\].
   The idea behind this partitioning is that a regular sensor tends to be connected to a CH that has a higher score (more eligible to be a CH) and thus belonging to \(N^+(i)\). Subsequently, node \(i\) computes, for each node \(v \in N^+(i)\), a function \(\text{RCR}_i(v)\) representing the relative preference that sensor \(i\) has for sensor \(v\) as its potential CH. In our implementation, \(\text{RCR}_i(v)\) takes the form of a probability function for sensor \(v\) to be CH for sensor \(i\), that is proportional to the rank of sensor \(v\) within \(N^+(i)\). It is expressed as follows:
   \[
   \text{RCR}_i(v) = \frac{|N^+(i)| + 1 - \text{Rank}(v)}{|N^+(i) + (|N^+(i)| + 1)|/2}
   \]
   where \(\text{Rank}(v)\) is the rank of sensor \(v\) within \(N^+(i)\) and \((|N^+(i)| + (|N^+(i)| + 1)|/2\) is the sum of all possible ranks.
3. Next, sensor $i$ broadcasts a message called Relative Cluster head Rank Advertisement (RCRA) containing the values $RCR_v(i)$ of all neighbors $v \in N^+(i)$. In the same manner, each sensor $i$ receives a distinct RCRA from all its $N^+(i)$ neighbors expressing its relative “importance” (as a potential CH) to each of them.

4. After receiving all RCRA messages from its $N^-(i)$ neighbors, every sensor $i$ computes a function $RCN(i)$ representing its Relative Contribution To the Network (RCN). $RCN(i)$ represents the probability for a node $i$ to be elected CH, based on the information received from its $N^-(i)$ neighbors. Since every sensor $i$ receives from each of its $N^-(i)$ neighbors an RCRA representing sensor $i$’s probability to be elected CH and since all $N^-(i)$ neighbors have the same weight, we express $RCN(i)$ as follows:

$$RCN(i) = 1 - \prod_{v \in N^-(i)} (1 - RCR_v(i))$$

Let us note that $RCN(i)$ represents the probability for sensor $i$ to be elected CH. Indeed, in our protocol, a sensor promotes itself CH if, at least, one of its $N^-(i)$ neighbors elects it as CH. Therefore, if we assume that the CH election probabilities are independent, the probability for sensor $i$ not to be elected CH equals: $\prod_{v \in N^-(i)} (1 - RCR_v(i))$ and thus $RCN(i)$ represents the probability for sensor $i$ to be elected CH.

5. Afterwards, each sensor $i$ broadcasts a message, called Contribution to the Network Advertisement (RCNA). The RCNA sent by sensor $i$ will be received by all sensors in $N^-(i)$. In the same manner, each node $i$ receives a distinct RCNA from all its $N^+(i)$ neighbors.

6. Finally, after receiving all RCNAs from its $N^+(i)$ neighbors, each node $i$ takes its decision about the CH it will connect to, unless it has been elected, itself, CH by one of its $N^-(i)$ neighbors. If $N^-(i) \neq \emptyset$, sensor $i$ has to wait until all $N^+(i)$ neighbors are received from its $N^-(i)$ neighbors or until a predefined timeout has elapsed to make its decision. This decision is then broadcast to all neighbors through an Election Message holding the ID of the sensor designated by sensor $i$ as its CH (either an existing CH or a newly elected one). The decision process for every node $i$ is as follows:

(a) If sensor $i$ has no neighbor with lower score ($N^+(i) = \emptyset$), it proceeds with step (b). Otherwise, it waits until all the decisions of its $N^-(i)$ neighbors are received or until a predefined timeout has elapsed and then proceeds with step (b).

(b) If, at least, one sensor in $i$’s one-hop neighborhood has already been elected CH, $i$ can connect to it and thus does not need to designate a CH among its $N^+(i)$. In this case, sensor $i$ will broadcast its Election Message with the ID of its CH and a flag CONNECTED set to 1, which tells the sensors in $N^+(i)$ that sensor $i$ does not need to elect a new CH.

(c) If sensor $i$ does not have any already-elected CH in its range, it has to elect one among its $N^+(i)$ neighbors. It will then select the node $v_{\max} = \arg\max_{v \in N^+(i)} \{CTN(i)\}$ as its CH and broadcasts its Election Message with the ID of $v_{\max}$ so that $v_{\max}$ knows that it has been designated CH.

(d) Finally, if a sensor $i$ has no neighbors with higher score ($N^+(i) = \emptyset$) and has no elected CH within its range, it promotes itself CH.

7. When a sensor is elected CH, it broadcasts a message holding its ID and its new state to its neighbors.

Let us note here that a sensor is elected CH if, at least, one of its $N^-$ neighbors elects it as CH. Furthermore, to avoid having multiple CHs within the same cluster, every sensor waits for the decisions of its $N^-$ neighbors before making its own decision. In this way, CH election will be made in chain: sensors with the lowest Score (having an empty $N^-$ set) will be the first ones to elect their CH and broadcast their decision. When a sensor is elected CH, it will inform its neighbors so that they can connect to it instead of electing, uselessly, another CH within their respective $N^+$ neighbors. For the election chain to function properly, every sensor $i$ has to send its Election Message to its $N^+(i)$ neighbors even if it chooses to connect to an already-elected CH neighbor, because sensors in $N^+(i)$ will wait for $i$’s decision to send their own decisions to their respective $N^+$ neighbors.

5. Protocol analysis

As described in the previous section, CH election is performed after a three-way message exchange between each sensor and its neighbors. After receiving the score of each of its neighbors through the SAB, every sensor processes and broadcasts the $RCR$ function of its neighbors that have higher SAB. Some of these neighbors can be two-hops away from each other, so the $RCR$ received by a sensor may encompass information about some neighbors of its neighbors that are two-hops away, which provides that sensor with aggregated information about its extended (two-hops) neighborhood. Then, every sensor assesses its capacity to become CH by computing its $RCN$ function, based on the $RCR$s received from its neighbors having a lower score. Therefore, the $RCN$ sent by a node to its one-hop neighbors to express its “eligibility” to become CH takes indirectly into account information about its two-hops neighbors, through the received $RCR$s. In this three-way message exchange, sensors play the role of “information relay” for their potential CH (neighbors having higher scores) by informing them through the $RCRA$ about their relative “rank” among the other potential CH in their range. Similarly, every sensor $i$ plays the same role of “information relay” for sensors in $N^-(i)$, by informing them (through the $RCNA$) of its “importance” (or eligibility) as a potential CH, so that they can decide which one of their neighbors will be chosen as their CH. This “importance”, expressed by the $RCN$, cannot be computed by $i$’s neighbors since they do not necessarily reach each other and thus cannot
necessarily hear each other’s RCRA. Hence, the final CH election will be based not only on local information (one-hop) but also on aggregated information from the two-hop neighbors and this makes the election protocol more efficient than traditional localized clustering algorithms \[5, 13\] where each node’s decision is based solely on its one-hop neighbors information. This is the main originality of EECF and we will show later that this novel clustering technique performs very well in terms of energy consumption and network lifetime. But let us discuss first the complexity of EECF and the connectivity of the set of elected CHs.

**Proposition 1.** EECF has an algorithmic complexity of \(O(N)\) at each node.

**Proof.** The execution of EECF at each node \(i\) consists of a sequence of four steps: (1) calculating the score and broadcasting it through the \(SAB\) message (2) calculating the \(RCR(v)\) function for each neighbor \(v\) in \(N^+(i)\) (after all \(SAB\) messages are received from neighbors) and broadcasting them through the \(RCRA\) (3) calculating \(RCN(i)\) (using the \(RCRs\) of \(N+\) \(i\) neighbors) and broadcasting it through the \(RCNA\) message (4) choosing the neighbor having the highest \(RCN\) among \(N^+(i)\) as CH and broadcasting the decision. Processing time at Steps (2) and (3) is proportional to the number of \(i\)’s neighbors which, in the worst case, is equal to \(N\). Hence, the overall complexity is \(O(N)\). \(\square\)

**Proposition 2.** EECF has a worst-case convergence time complexity of \(O(N)\) and an average convergence time complexity of \(O(\log(N))\).

**Proof.** We denote by “convergence time” the time it takes for a flat network to become fully clustered using EECF. The worst-case sensor configuration for the convergence time is the one in which sensors are sorted according to their Scores and placed in a chain of increasing scores, from right to left, for instance. In this case, the rightmost sensor will start the election process since it has the lowest Score. It will elect its left neighbor as CH. Then, every sensor will wait for the Election Message from its right neighbor \((N^+)\) and send its own decision to its left neighbor \((N^-)\). The process terminates at the leftmost sensor which will either be CH or connect to its right neighbor if this latter has been elected CH. With this worst-case configuration, the convergence time is proportional to the number of sensors and thus the complexity is \(O(N)\). However, in a dense network where sensors are randomly scattered, we have, in average, \(|N^-(i)| = |N^+(i)| = \frac{\delta_i}{2}\), where \(\delta_i\) denotes the average degree of node \(i\). In such a case, every sensor has to wait for the Election Messages of its \(N^+(i)\) neighbors, which average number is \((\delta_i/2)\) and then sends its own decision to the set of its \(N^+(i)\) neighbors, having the same average cardinality. The set of sensors having the lowest score (i.e: an empty \(N^-\) set) start the process. Each of these sensors will send it decision to its \((\delta_i/2)\) neighbors. Then, each of these latter neighbors will send, in turn, its own decision to its \((\delta_i/2)\) neighbors, and so forth. Hence, in average, the whole set of sensors can be seen as a layered structure of sensors in which sensors on a certain layer have their \(N^-\) neighbors on the lower layer and their \(N^+\) neighbors on the upper one. Therefore, if we start with \(p\) sensors having the lowest score (and thus forming the first layer), the number of sensors on layer \(k\) \((k > 1)\) is \(p (\frac{\delta_i}{2})^k\). And since the average convergence time is proportional to the number of layers, it has thus a complexity of \(O(\log(n))\). \(\square\)

**Proposition 3.** When the inter-cluster transmission range \(R_{CH}^c\) is greater or equal to three times the intra-cluster transmission range \(R_c\), the EECF protocol ends up with a clustered configuration where the overlay graph composed of cluster heads is connected.

**Proof.** Since the whole network is assumed to be connected and clustered, every node is either a CH or attached to a CH. Fig. 3 depicts a configuration exhibiting the longest distance between two adjacent CHs. In fact, when sensors are lined-up, each regular sensor has to be connected to a CH and the two regular sensors has to be directly connected since the network is assumed to be connected. In this worst-case configuration, it is clear that when the transmission range of a CH is equal to or greater than three times \(R_c\), the overlay graph composed of CHs will be connected. \(\square\)

![Fig. 3. Necessary condition to have a connected overlay CH graph: \(R_{CH}^c \geq 3 R_c\).](image-url)
6. Protocol implementation and performance evaluation

6.1. Experiment setup

In our EECF protocol, CHs are elected based on their respective degrees (which are static) and their residual energies at the instant the clustering protocol is executed. Hence, different residual-energy maps can lead to different sets of CHs. When run once, the EECF protocol elects the set of CHs that tends to maximize the network lifetime as if this configuration will be kept for the whole network lifetime. However, since sensors’ residual energies are gradually depleted during network operation, our protocol will be run periodically to adapt the dynamic residual energies: every pre-determined period of time \( T \), a new set of CHs will be elected, based on the actual residual energies at the instant EECF is run. This adaptivity will balance the CH role among sensors and thus balance energy dissipation. The period \( T \) could be chosen arbitrarily but it has to provide a tradeoff between network dynamic and stability. In fact, when \( T \) is too small, the set of CHs will be re-computed too frequently leading to an instability in the network because of the frequent changes of sensor states. On the other hand, when \( T \) is too big, the adaptivity aspect of the protocol will fade out due to the relatively quick change in residual energies. Without loss of generality, we chose \( T = 1 \) unit. To measure the sensors’ energy dissipation during a time period \( T \), we assumed a dense network and we used a simplified form of the log-distance path loss channel model described in [16]: we used the free-space fading model for both regular sensors and CHs. Thus, the energy dissipated by a sensor to transmit \( l \) bits over a range \( R \) is: \( E_{s} = l.E_{s} + l.\epsilon.R^{2} \) and the energy spend to receive an \( l \)-bits message is: \( E_{r} = l.E_{r} \) where \( E_{s} \) is the energy dissipated by the radio hardware to transmit or receive the signal and \( \epsilon.R^{2} \) is the amplifier’s energy necessary to transmit 1 bit of data over a relatively short distance (Actually, \( \epsilon = 1/\lambda^{2} \) and \( \lambda \) is the wavelength). Thus, the energy dissipated by a CH to transmit \( l \) bits of data over its range \( R_{CH} \) and to receive and aggregate an \( l \)-bit length message from each of the regular sensors connected to it, is:

\[
E_{CH} = l.E_{s} + l.\epsilon.(R_{CH}^{2}) + N_{CH}.l.E_{s} + N_{CH}.l.E_{r}
\]

where \( E_{s} \) is the energy dissipated in data aggregation and \( N_{CH} \) is the number of regular sensors connected to the CH.

For our simulations, we considered the same parameter values as [16], that are: \( \epsilon = 10 \text{ pJ/bit/m}^{2}, E_{s} = 50 \text{ nJ/bit} \) and \( E_{r} = 50 \text{ nJ/bit/signal} \). We also considered a unique length of 100 bytes for all data and protocol messages.

To evaluate the performance of EECF, we implemented it in C++ and we first studied the effect of the score coefficients \( \alpha \) and \( \beta \) on the network lifetime and the cluster head ratio (number of elected CHs with respect to the total number of sensors). Then, we compared EECF to a recently published clustering protocol: EESH [17]. As performance metrics, we chose the network lifetime and the ratio of CHs with respect to the total number of sensors. The second metric assesses the efficiency of our protocol in terms of the minimum number of CHs required to fulfill a fully clustered network. For performance evaluation, we considered 4 grid networks of increasing sizes, composed, respectively, of 100, 400, 900 and 1600 sensors.

6.2. Effect of the score coefficient \( \alpha \) on performance

As we can see from the protocol description, the weighting coefficient \( \alpha \) is not analytically linked to the network lifetime or the cluster head ratio. In fact, \( \alpha \) will only “push” nodes having a good tradeoff between the degree and the residual energy to be elected CHs but we cannot derive any exact mathematical relationship between \( \alpha \) and the network lifetime, or the cluster head ratio. In fact, these two metrics are average values that also depend on the energy map (residual energy of each sensor with respect to its neighbors’ residual energies). Hence, we empirically studied the effect of the weighting coefficient \( \alpha \) on the network lifetime and the cluster head ratio. Fig. 4 depicts the variation of the network lifetime with respect to \( \alpha \) for two values of the transmission range: \( R_{t} = 1 \) and \( R_{t} = 3 \). We can see that for \( \alpha = 0.1 \), an optimal value of the network lifetime can be obtained. This small value of \( \alpha \) means that the residual energy is a more important factor in CH election than the degree. However, since the optimal value of \( \alpha \) in not zero, the degree still has a small significance. This result was pretty expectable in our case because we considered a grid network where all nodes have almost the same degree (except edge nodes). The optimal value of \( \alpha \) is roughly the same for \( R_{t} = 1 \) and \( R_{t} = 3 \) because, even if the sensors have a greater degree for \( R_{t} = 3 \), sensors still have the same degree as their neighbors. As far as the CH ratio is concerned, Fig. 5 shows that the bigger the value of \( \alpha \), the lower (the better) this ratio. This is also expectable because higher values of \( \alpha \) favor the election of nodes having the highest degree as CHs, regardless of their residual energies, thus reducing the overall number of CHs. In the next paragraph, we will use the average value of \( \alpha = 0.5 \) to compare our protocol to EESH.

6.3. Comparison of EECF and EESH in terms of network lifetime

To measure the network lifetime generated by EECF and EESH, we run these two protocols periodically, at instants \( kT (k \in \mathbb{N}) \) and we used \( T \) as a measure unit of the network lifetime. Fig. 6 depicts the network lifetime generated by EECF and EESH, for 2 different values of the transmission range, \( R_{t} = 1 \) and \( R_{t} = 3 \). We can see that EECF outperforms EESH by approximatively 60% for \( R_{t} = 1 \) and more than 70% for \( R_{t} = 3 \). This could be explained by the two-hops information (resulting from the 3-way message exchange) every sensor uses to make its decision, and which generates better-cost clusters. We also notice that the better performance of EECF compared to EESH is more emphasized for \( R_{t} = 3 \) than for \( R_{t} = 1 \). This is because,
the bigger the transmission range, the more potential CHS are reached by each sensor and thus EECF has more clustering possibilities and is more likely to find a better configuration compared to EESH.
6.4. Comparison of EECF and EESH in terms of the CH ratio (Number of CHs/Total number of sensors)

The ratio “Number of CHs/Total number of sensors” expresses the efficiency of the protocol in terms of the number of CHs required to cluster the network. Fig. 7 depicts the ratio of CHs for EECF vs. EESH, for \( R_t = 1 \) and for \( R_t = 3 \). We can see that, for large-scale configurations (400 sensors and more), EECF outperforms EESH by approximately 20% for \( R_t = 1 \) and \( R_t = 3 \). This means that our protocol is able to find admissible configurations with less CHs, which is very consistent with the results regarding the network lifetime, because using less CHs will, in average, increase the network lifetime. Here again, the better
CH ratio of EECF compared to EESH can be explained by the two-hops information that every sensor uses to make its clustering-related decision.

7. Conclusion

In this paper, we proposed EECF, a novel dynamic distributed cluster formation protocol for sensor networks. Our protocol is based on a three-way message exchange between every sensor and its one-hop neighbors and ends up with a clustered network where the overlay graph composed of CHs is connected when the CH transmission range is equal or greater than three times the transmission range of non-CH sensors. We showed that EECF has a message exchange complexity of $O(1)$ and a worst-case convergence time complexity of $O(N)$. Performance evaluation showed that EECF provides a better network lifetime and a better ratio "Number of CHs/Total number of sensors" than EESH, a recently published clustering protocol for wireless sensor networks. As for future research, we intend to extend our protocol to sensor networks with multiple transmission ranges, where sensors can reach more neighbors using higher transmission power (at higher cost). Furthermore, we plan on extending our protocol to handle multi-level clustering.

References