CES: Cluster-based Energy-efficient Scheme for Mobile Wireless Sensor Networks

Mohamed Lehsaini[1,2] , Hervé Guyennet[1], and Mohammed Feham[2]

[1] Laboratory of Computer Engineering of Franche-Comté University (LIFC),
16, Route de Gray, Besançon 25030, France
{lehsaini, guyennet}@lifc.univ-fcomte.fr
[2] Faculty of Engineering Sciences, Tlemcen University,
m_feham@mail.univ-tlemcen.dz

Abstract. Prolonging network lifetime has become a real challenge in Mobile Wireless Sensor Networks (MWSNs) as sensors have limited energy. In this paper, we propose a Cluster-based Energy-efficient Scheme (CES) for electing a cluster-head to evenly distribute energy consumption in the overall network and therefore obtain a longer network lifetime. In CES, each sensor calculates its weight based on k-density, residual energy and mobility and then broadcasts it to its 2-hop neighborhood. The sensor node with the greatest weight in its 2-hop neighborhood will become the cluster-head and its neighboring sensors will then join it. We performed simulations to illustrate the effects of sensor mobility on LEACH and LEACH-C's performance. Unfortunately, our findings showed that sensor mobility had a significant impact on both protocols' performance, but CES provided good results in terms of the amount of data packets received at the sink when compared with LEACH and LEACH-C.

Keywords: Cluster-head, k-density, Network lifetime, Residual energy, MWSNs.

1 Introduction

MWSNs consist of a large number of tiny mobile sensors that are randomly deployed in an interest area to sense phenomena. These mobile sensors collaborate with each other to form a sensor network able to send sensed phenomenon to a data collection point called the sink or base station. MWSNs could become increasingly useful in a variety of potential civil and military applications, such as intrusion detection, habitat and other environmental monitoring, disaster recovery, hazard and structural monitoring, traffic control, inventory management in factory environments and health related applications, etc. [1,2]. However, the deployment of MWSNs still requires solutions to a number of technical challenges that stem primarily from the constraints imposed by simple sensor devices: small storage capacity, low processing power, limited battery lifetime and short radio ranges.

Gathering information in MWSNs while minimizing the overall energy consumption and maximizing the amount of data received at the base station requires an efficient energy-saving scheme. Cluster-based architecture is considered an

efficient approach to achieving this. Hence, we should imply determining parameters enabling to generate a reduced number of stable and balanced clusters.

The above constraints imposed by sensors make the design of an efficient scheme for prolonging MWSNs' lifetime a real challenge. In response to this challenge, we propose a Cluster-based Energy-efficient Scheme (CES) for MWSNs, which consists of grouping sensors into a set of disjoint clusters. In CES, the sensor with the greatest weight in its 2-hop neighborhood becomes the cluster-head. The weight of each sensor is calculated according to the following parameters: 2-density, residual energy and mobility. Furthermore, the cluster size ranges between two thresholds, $Thresh_{Lower}$ and $Thresh_{Upper,}$ which respectively represent the minimal and maximal number of sensors in a cluster. These thresholds are chosen arbitrarily or depend on network topology. Inside a cluster, each sensor is, at most, two hops from its corresponding cluster-head contrary to LEACH [3] and its variant LEACH-C [4], which allow only single-hop clusters to be constructed.

In the cluster-based heuristic methods proposed for WSNs, cluster members do not transmit their gathered data directly to the sink but to their respective cluster-head. Accordingly, cluster-heads are responsible for coordinating the cluster members, aggregating their sensed data, and transmitting the aggregated data to the remote sink, directly or via multi-hop transmission mode. Since cluster-heads receive many packets and consume a lot of power for long-range transmission, they are the ones whose energy is used up most quickly in the cluster if they are elected for a long time. Therefore, a cluster-based scheme should avoid a fixed cluster-head election scheme, because the latter has constrained energy and may rapidly drain its battery power due to heavy utilization. That can cause bottleneck failures in its cluster and trigger the cluster-head election process again. For that, we foresaw in the CES scheme that the cluster-head election process would be periodically carried out after a period of time called "round" to evenly balance the energy load among the sensors during the network lifetime.

In this paper we aim to minimize the energy consumption of the entire network and prolong the network lifetime. For this, we propose the CES scheme, which involves k-density and mobility factors in nodes' weight computation in order to guarantee the stability of clusters, as well as the energy factor to ensure a long cluster-head lifetime.

In our experiments, we conducted extensive simulations to evaluate the performance of both protocols: LEACH and LEACH-C with the same scenario presented in [3,4] but with mobile sensors. We also carried out simulations to evaluate CES's performance and compare the results obtained with LEACH and LEACH-C in terms of the amount of data packets received at the sink during the network lifetime.

The rest of this paper is organized as follows: in Section 2, we provide the necessary preliminary information for describing our scheme; Section 3 reviews several cluster-based algorithms that have been previously proposed; in Section 4, we present our new weighted scheme; and Section 5 presents a performance analysis of the proposed scheme. Finally, we conclude our paper and discuss future research work in Section 6.

2   Notations and hypothesis

Before heading into the technical details of our contribution, we shall start by giving some definitions and notations that will be used later in our paper.

A wireless sensor network is abstracted as an undirected graph G=(V,E), called a connectivity graph, where V represents the set of wireless nodes and $E \subseteq V^2$ is the set of edges that gives the available communications; an edge e=(u,v) belongs to E if and only if the node u is physically able to transmit messages to v and vice versa. Each sensor $u \in V$ is assigned a unique value to be used as an identifier so that the identifier of u is denoted by $Node_{Id}(u)$. The neighborhood set $N_1(u)$ of a node u is in (1). The size of this set is known as the degree of u, denoted by $\delta_1(u)$. The density of the network represents the average of the nodes' degrees.

$$N_1(u) = \{v \in V | v \neq u \wedge (u, v) \in E\}. \tag{1}$$

The 2-hop neighborhood set of a node u, i.e. the nodes which are the neighbors of u's neighbors except those that are u's neighbors, is represented by $N_2(u)$.

$$N_2(u) = \{w \in V | (v, w) \in E \text{ where } w \neq u \wedge w \notin N_1(u) \wedge (u, v) \in E\}. \tag{2}$$

The combined set of one-hop and two-hop neighbors of u is denoted by $N_{12}(u)$.

$$N_{12}(u) = N_1(u) \cup N_2(u). \tag{3}$$

In a general manner, the k-hop neighborhood set of a node u is represented by $N^k(u)$ as shown in (4) and its closet set of k-hop neighbors is denoted by $N^k[u]$ as in (5). Here, d(u,v) represents the minimal distance in the number of hops from u to v. The size of $N^k(u)$ is known as the k-degree of u and denoted by $\delta^k(u)$.

$$N^k(u) = \{v \in V | v \neq u \wedge d(u, v) \leq k\} \tag{4}$$

$$N^k[u] = N^k(u) \cup \{u\} \tag{5}$$

The k-density of a node u represents the ratio between the number of links in its k-hop neighborhood (links between u and its neighbors and links between two k-hop neighbors of u) and the k-degree of u; formally, it is represented by the following formula:

$$k - density(u) = \frac{\left| (v, w) \in E : v, w \in N^k[u] \right|}{\delta^k(u)} \tag{6}$$

However, we are interested only in calculating the 2-density nodes so as not to weaken the CES scheme's performance as presented in (7). Table.1 illustrates the 2-density calculation of the nodes composing the network presented in Fig.1.

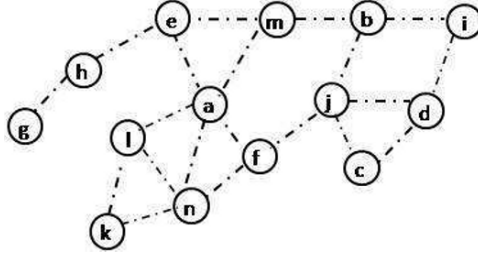$$2 - density(u) = \frac{\left|(v, w) \in E : v, w \in N_{12}[u]\right|}{\delta^2(u)} \tag{7}$$

Fig. 1. Example of an abstracted wireless network

Table 1. Calculation of the nodes' 2-density.

Node	a	B	c	d	e	F	g	h	i	j	K	l	M	n
1-density	1,60	1	1,66	1,33	1,33	1,33	1	1	1	1,25	1,66	1,66	1,33	1,75
2-density	1,55	1,50	1,40	1,40	1,37	1,60	1	1,25	1,40	1,50	1,75	1,60	1,44	1,57

We propose to generate balanced clusters whose size ranges between two thresholds: $Thresh_{Upper}$ and $Thresh_{Lower}$. These thresholds are chosen arbitrarily or depend on network topology. If their values depend on network topology, they will be calculated as follows:

- u: the node that has the maximum number of 2-hop neighbors,

$$|N_{12}(u)| = Max(|N_{12}(u_i)| : u_i \in V) \tag{8}$$

- v: the node that has the minimum number of 2-hop neighbors,

$$|N_{12}(v)| = Min(|N_{12}(v_i)| : v_i \in V) \tag{9}$$

- Avg: the average number of 2-hop neighbors of all nodes in the network,

$$Avg = \frac{\sum_{i=1}^{n} N_{12}(u_i)}{n} \quad \text{where n : number of nodes} \tag{10}$$

$$Thresh_{Upper} = \frac{1}{2}(|N_{12}(u)| + Avg) \tag{11}$$

$$\text{Thresh}_{\text{Lower}} = \frac{1}{2}(|N_{12}(v)| + \text{Avg}) \tag{12}$$

In this paper, we assume that all sensors are given in a two dimensional space and we measure the distance between the two nodes u and v in terms of the number of hops. Each sensor has an omni-directional antenna which means that a single transmission from a sensor can be received by all sensors within its vicinity, and we consider that the sensors are almost stable in a reasonable period of time during the clustering process. We also assume that each sensor has a generic weight and that it is able to evaluate it. Weight represents the fitness of each node to be a cluster-head, and a greater weight means higher priority.

3 Related Work

Recently, many cluster-based techniques [3-12] have been proposed to deal with the main challenges in WSNs. However, most of these contributions focus on lifetime maximization in WSNs with stationary sensors. To the best of our knowledge, this paper is the first to tackle lifetime extension in WSNs with mobile sensors. In this section, we will review some of the most relevant papers related to cluster-based network architecture, which have been carried out to prolong lifetime in WSNs.

In [3], the authors propose LEACH, which is a distributed, single hop clustering algorithm for homogeneous WSNs. In LEACH, the cluster-head role is periodically rotated among the sensors to evenly distribute energy dissipation. After each round, each sensor elects itself as cluster-head with a probability which is equal to:

$$p_{\text{CH}} = k\frac{E(u)}{E_{\text{Total}}} \tag{13}$$

where $E(u)$ represents remaining energy of node u, E_{Total} is the total energy in the whole network and k is the optimal number of clusters. However, the evaluation of E_{Total} presents a certain difficulty since LEACH operates without other routing schemes and any central control.

In [5], the authors compared homogeneous and heterogeneous networks in terms of energy dissipation in the whole network and analyzed both single-hop and multi-hop networks' performance. They chose LEACH as a representative of a homogeneous network and compared it with a heterogeneous single-hop network. The authors noticed that using single-hop communication between cluster members and their corresponding cluster-head may not be the best choice when the propagation loss index k (k>2) for intra-cluster communication is large, because LEACH might generate clusters whose size is important in dense networks and clusters whose size is limited in small networks. In both cases, cluster-heads could rapidly exhaust their battery power either when they coordinate among their cluster members or when they are placed away from the base station. Therefore, the authors proposed an improved version of LEACH called M-LEACH [5] (Muti-hop LEACH), in which cluster

members can be more than one hop from their corresponding cluster-head and communicate with it in multi-hop mode. They also illustrate the cases where LEACH-M outperforms LEACH protocol. However, this proposed version requires each sensor to be capable of aggregating data, which increases the overhead for all sensors. To improve the performance of this strategy, in [6], the authors focus on heterogeneous sensor networks, in which two types of sensors are deployed: super and basic sensors. Super sensors have more communication and processing capabilities and act as cluster-heads, while basic sensors are simple (with limited power) and are affiliated to a nearby cluster-head and communicate with it directly or via multi-hop mode.

Furthermore, another variant of LEACH called LEACH-C [4] has been conceived to improve LEACH performance. This variant utilizes a centralized architecture to select cluster-heads while using a base station and location information from sensors. However, it increases network overhead since all sensors send their location information to the base station at the same time during every set-up phase. Several works have proven that a centralized architecture is particularly suitable for small networks, whereas it lacks scalability to handle the load when the network's size increases.

Similarly to LEACH-C, BCDCP (Base-Station Controlled Dynamic Clustering Protocol) [7] uses energy information sent by all sensors to the base station to build balanced clusters during the set-up phase. In BCDCP, the base station randomly changes cluster-heads while guaranteeing a uniform distribution of their locations in the interest field and carries out an iterative cluster splitting algorithm to find the optimal number of clusters. After that, it constructs multiple cluster-to-cluster (CH-to-CH) routing paths to use for data transfer, creates a schedule for each cluster and broadcasts it to the sensor network. In the second phase, which relates to data transfer, cluster-heads transmit collected data to the base station through the CH-to-CH routing paths [8]. However, BCDCP presents the same limitations as LEACH-C since it utilizes a centralized architecture to elect cluster-heads.

4   Our Contribution

In our proposed scheme, each sensor uses weight criteria to elect a cluster-head in its 2-hop neighborhood. The CES scheme assumes that sensors have 2-hop knowledge and operate asynchronously without a centralized controller. In CES, each sensor calculates its weight based on its k-density, its residual energy, and its mobility and broadcasts it to its 2-hop neighborhood. The sensor with the greatest weight in its 2-hop neighborhood is chosen as the cluster-head for the current round.

4.1   Cluster formation

The cluster formation process consists of grouping sensors into disjoint clusters, thus giving the network a hierarchical organization. Each cluster has a cluster-head which is chosen from its 2-hop neighborhood based on nodes' weight. The weight of each sensor is a combination of k-density, residual energy and mobility as presented in

(14), wherein the coefficient of each parameter can be chosen depending on the application.

$$\text{Weight(u)} = \alpha * 2 - \text{density(u)} + \beta * \text{Res} - \text{Energie(u)} + \gamma * \text{Mobility(u)} \quad (\mathbf{14})$$

$$\text{where } \alpha + \beta + \gamma = 1$$

Since the cluster head is responsible for carrying out several tasks - such as coordinating the cluster members, transmitting gathered data to the remote base station, and managing its own cluster - we propose to set up periodical cluster-head election processes after each round so that cluster-heads do not rapidly exhaust their battery power. We also propose that each cluster has a size ranging between two thresholds, $Thresh_{Lower}$ and $Thresh_{Upper}$, and that cluster members are, at most, 2-hops from their respective cluster-head.

In the CES scheme, each sensor is identified by a state vector as follows: *(Node$_{Id}$, Node$_{CH}$, Weight, Hop, Size, Thresh$_{Lower}$,Thresh$_{Upper}$)* where *Node$_{Id}$* is the sensor identifier, *Node$_{CH}$* represents the identifier of its cluster-head, *Hop* indicates the number of hops separating it from its respective cluster-head, and *Size* represents the size of the cluster to which it belongs. Each sensor is responsible for maintaining a table called '*Table$_{Cluster}$*', in which information from the local cluster members is stored. The format of this table is defined as *Table$_{Cluster}$(Node$_{Id}$, Node$_{CH}$, Weight)*. The sensors could coordinate and collaborate between each other to construct and update the above stated table by using Hello messages**.** We used Hello messages to achieve these operations in order to alleviate the broadcast overhead and not degrade the CES scheme's performance.   Moreover, each cluster-head has another table called '*Table$_{CH}$*', in which information from cluster-heads is stored. The format of this table is defined as *Table$_{CH}$(Node$_{CH}$, Weight)*.

Cluster formation is performed in two consecutive phases: set-up and re-affiliation.

4.1.1 The set-up phase

At the beginning of each round, each sensor calculates its weight and generates a 'Hello' message with two extra fields in addition to other regular contents: *Weight* and *Node$_{CH}$*, where *Node$_{CH}$* is set to zero. Then, it broadcasts it to its 2-hop neighborhood and eavesdrops on its neighbors' 'Hello' messages. The sensor with the greatest weight among its 2-hop neighborhood is chosen as the cluster-head (CH) for the current round. The latter updates its state vector by assigning the value of its identifier *Node$_{Id}$* to *Node$_{CH}$* , and sets, respectively, *Hop* and *Size* to 0 and 1. Then, it broadcasts an advertisement message *(ADV_CH)* including its state vector to its 2-hop neighborhood requesting them to join it, as illustrated by Fig. 2. Each sensor in the 1-hop neighborhood that receives the message and does not belong to any cluster and that has a lower weight than CH's weight, transmits a *REQ_JOIN* message to *CH* to join it. The corresponding cluster-head checks and, if its own cluster size does not reach *Thresh$_{Upper}$*, it will transmit an *ACCEPT_CH* message to this sensor; if not, it will simply drop the affiliation request message. Thereafter, *CH* increments its *Size* value, and the affiliated sensor node sets *Hop* value to 1 and *Node$_{CH}$* with *Node$_{CH}$* as

its corresponding cluster-head. Then, the affiliated sensors whose *Hop* value is equal to 1, broadcast received message again with the same transmission power to its neighbors. Similarly, each sensor belonging to $N_2(Node_{CH})$ that is not affiliated to any cluster and whose weight is lower than that of *CH*, transmits a *REQ_JOIN* message to the corresponding *CH*. In the same way, *CH* checks if its *Size* value remains under *Thresh_{Upper}*, and if so transmits ACCEPT_CH and updates its state vector. If not, it will drop the message of affiliation request. In the end, each sensor will know which cluster it belongs to and which sensor is its cluster-head.
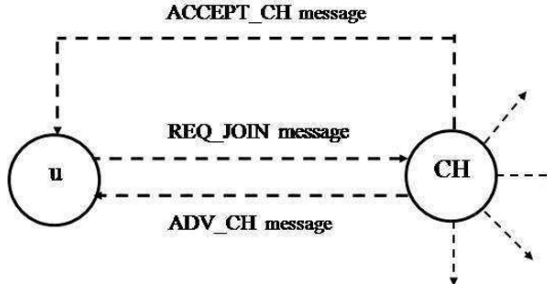

Fig. 2. Affiliation procedure of a node to a cluster

4.1.2 The re-affiliation phase

During the set-up phase, it may not be possible for all clusters to reach the *Thresh_{Upper}* threshold. Moreover, it is possible that clusters whose size is lower than *Thresh_{Lower}* may be created, since there is no constraint relating to the generation of these types of clusters. In this phase, we propose to re-affiliate the sensors belonging to clusters that have not attained the cluster size *Thresh_{Lower}* to those that did not reach *Thresh_{Upper}* in order to reduce the number of clusters formed and obtain balanced clusters.

The execution of this phase proceeds in the following way: cluster-heads that belong to clusters whose size is strictly lower than *Thresh_{Upper}* and higher than *Thresh_{Lower}* broadcast a new message called *RE-AFF_CH* to re-affiliate nodes belonging to the smaller clusters. Each sensor that receives this message and that belongs to a small cluster should be re-affiliated to the nearest cluster-head based on the received signal strength. Finally, each cluster-head creates a time schedule in which time slots are allocated for intra-cluster communication, data aggregation, inter-cluster communication and maintenance processes. This allows the sensors to remain in sleep state as long as possible and prevents intra-cluster collisions.

4.2   Cluster maintenance

In our contribution, the cluster maintenance process should be triggered in the event of a cluster losing its cluster-head either when the latter exhausts its battery power or migrates towards another cluster. Moreover, the cluster-head's re-election process

only concerns clusters that have lost their cluster-head and the future cluster-head would be chosen among the members of the cluster. We adopted this solution so as not to weaken our scheme's performance and to avoid chain reactions which can occur during the launching of the clustering process. Furthermore, the cluster maintenance process is performed in a similar way to the set-up phase, where a random node among the members cluster initiates the clustering process.

5   Evaluation and simulation results

In our experiments, we conducted simulations to evaluate the CES scheme and compare it with LEACH and LEACH-C in terms of the number of nodes alive and the data packets received at the base station during the network lifetime. Simulations have been performed in NS-2 [13] using the MIT_uAMPS ns code extensions [14] to implement the CES scheme. We carried out these simulations with the same scenario presented in [3,4] but with mobile nodes. We considered a network topology with 100 mobile sensors with a sensing range of 25 meters. Sensors are randomly placed in a 100m×100m square area by using a uniform distribution function, and the remote base station is located at position $x = 50$, $y = 175$. At the beginning of the simulation, all the sensors had an equal amount of energy, i.e. the sensors started with 2 Joules of energy. Simulations were carried out until all the sensors exhausted their battery power and the average values were calculated after each round (duration of 20 seconds). After this time, the CES scheme triggered the cluster-head's election process again. Moreover, we performed simulations using two distinct values for threshold $Thresh_{Upper}$: 30, 50, i.e. CES_30 and CES_50, and a fixed value for threshold $Thresh_{Lower}$=15. These values were attributed arbitrarily.

As mentioned above, we used the same energy parameters and radio model as discussed in [3,4], wherein energy consumption is mainly divided into two parts: receiving and transmitting messages. The transmission energy consumption requires additional energy to amplify the signal according to its distance from the destination. Thus, to transmit a k-bit message to a distance d, the radio expends energy as described by the formula (15), where ε_{elec} is the energy consumed for radio electronics, $\varepsilon_{friss-amp}$ and $\varepsilon_{two-ray-amp}$ for an amplifier.   The reception energy consumption is $E_{Rx}=\varepsilon_{elec}\times k$.

$$E_{Tx} = \begin{cases} \varepsilon_{elec} * k + \varepsilon_{friss-amp} * k * d^2 & \text{if } d < d_{Crossover} \\ \varepsilon_{elec} * k + \varepsilon_{two-ray-amp} * k * d^4 & \text{if } d \geq d_{Crossover} \end{cases} \tag{15}$$

Simulated model parameters are set as shown in Table 2. The data size was 500 bytes/message plus a header of 25 bytes. The message size to be transmitted was: k=(500 bytes + 25 bytes)×8= 4 200 bits.

Table 2. Parameters for simulation

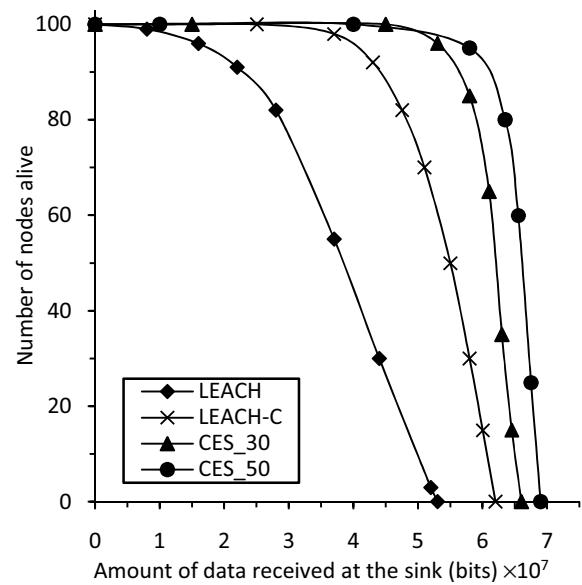| Parameter	Value
Network Grid	$(0,0) \times (100,100)$
Base Station	$(50,125)$
ε_{elec}	50 nJ/bit
$\varepsilon_{friss\text{-}amp}$	10 pJ/bit/m^2
$\varepsilon_{two\text{-}ray\text{-}amp}$	0.0013 pJ/bit/m^4
$d_{Crossover}$	87 m
Data packet size	500 bytes
Packet header size	25 bytes
Initial energy per node	2J
Number of nodes (N)	100
Round	20 seconds
Thresh$_{Upper}$	30, 50
Thresh$_{Lower}$	15

Fig. 3. Number of nodes alive per amount of data received at the sink
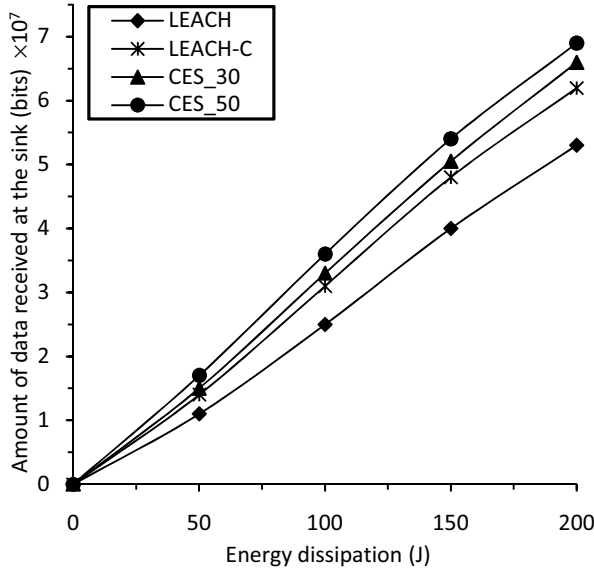
Fig. 4. Amount of data received at the sink according to energy dissipation

Fig.3 shows that CES_30 considerably outperforms LEACH and slightly outperforms LEACH-C in terms of the amount of data sent to the base station during network lifetime, whereas CES_50 largely outperforms them. Moreover, in Fig. 3, the shape of the curves of CES_30 and CES_50 shows that the number of nodes alive degrades rapidly at the end of simulation. That means that the time difference between the demise of the first and last sensor is too small compared to LEACH, where sensors gradually wear out during the network lifetime. On the other hand, Fig.4 illustrates that CES_30 and CES_50 outperform LEACH and LEACH-C in terms of the number of data packets received by the base station with the same total amount of energy.

Our proposed scheme allows the even distribution of consumption among the sensors in the network. Therefore, it maximizes sensor lifetime and minimizes the time difference between the demise of the network's first and last sensor.

6   Conclusion and future work

In this paper, we have proposed a Cluster-based Energy-efficient Scheme (CES) for Mobile Wireless Sensor Networks (MWSNs) which relies on weighing k-density, residual energy and mobility parameters for cluster-head election. The CES scheme carries out a periodical cluster-head election process after each round. Moreover, CES enables the creation of balanced 2-hop clusters whose size ranges between two thresholds: $Thresh_{Upper}$ and $Thresh_{Lower}$.

Simulation results demonstrate that the CES scheme provides better performance than LEACH and LEACH-C in terms of the amount of data received at the base station during the network lifetime, as well as considerably outperforming LEACH in terms of the amount of data sent to the base station with the same amount of energy dissipation.

With these results obtained, the CES scheme can provide good performance for coverage and broadcasting in MWSNs. Therefore, its evaluation could be the subject of future work.

References

1. Chong, C. Y., Kumar, S. P.: Sensor networks: Evolution, opportunities, and challenges. In: Proceedings of the IEEE, vol. 91 N° 8, pp. 1247--1256 (2003).
2. Estrin, D., Govindan, R., Heidemann, J., Kumar, S.: Next century challenges: Scalable coordination in sensor networks. In: *IEEE/ACM* MobiCom'99, pp. 263--270 (1999).
3. Heizelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-Efficient Communication Protocol for Wireless Micro Sensor Networks. In: IEEE Proceedings of the Hawaii International Conference on System Sciences (HICSS '00) (2000).
4. Heinzelman, W. R., Chandrakasan, A. P., Balakrishnan, H.: An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions Wireless Communications, vol. 1 N°. 4, pp. 660--670 (2002).
5. Mhatre, V., Rosenberg, C.: Design guidelines for wireless sensor networks: communication, clustering and aggregation. Ad Hoc Network Journal, vol. 2 N°. 1, pp. 45--63 (2004).
6. Ye, M., Li, C., Chen, G., Wu, J.: EECS: an energy efficient cluster scheme in wireless sensor networks. In: IEEE International Workshop on Strategies for Energy Efficiency in Ad Hoc and Sensor Networks (IEEE IWSEEASN2005), Phoenix, Arizona, (2005).
7. Lee, G., Kong, J., Lee, M., Byeon, O.: A Cluster-based Energy Aware Routing Protocol for Sensor Networks. In: Proceedings (466) Parallel and Distributed Computing and Systems, Phoenix, AZ, USA (2005)**.**
8. Muruganathan, S. D., Ma, D.C.F., Bhasin, R.I., Fapojuwo, A. O.: A Centralized Energy-Efficient Routing Protocol for Wireless Sensor Networks. IEEE Communication Magazine, Vol. 43, No. 3, pp. S8--S13 (2005).
9. Lindsey, S., Raghavendra, C.S.: PEGASIS: power efficient gathering in sensor information systems. In: Proceeding of the IEEE Aerospace Conference, Vol. 3, March, pp.1125-1130. Big Sky, Montana (2002).
10. Manjeshwar, A., Agrawal, D. P.: TEEN: A Protocol for Enhanced Efficiency in Wireless Sensor Networks. In: Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco, CA (2001).
11. Younis, O., Fahmy, S.: Distributed clustering in ad-hoc sensor networks: A hybrid, energy efficient solution. In: Proceedings of IEEE INFOCOM, Hong Kong (2004).
12. Qing, L., Qingxin, Zhu, Q., Wang, M.: Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks. Elsevier Computer Communications 29, pp. 2230--2237 (2006).
13. NS-2, Network Simulator, http://www.isi.edu/nsnam/ns/ns-build.html.
14. MIT_uAMPS LEACH NS2, www.ece.rochester.edu/research/wcng/code/index.html.

Wireless Sensor and Actor Networks II
Proceedings of the 2008 IFIP Conference on Wireless Sensor
and Actor Networks (WSAN 08), Ottawa, Ontario, Canada,
July 14-15, 2008