Praise Worthy Prize

International Review on Computers and Software (I.RE.CO.S.), Vol. 3, N. 2
March 2008

Cluster-based Self-Organization Scheme for Mobile Wireless Sensor Networks

M. Lehsaini[1], H. Guyennet[2], M. Feham[3]

Abstract – *One of the main challenges in Mobile Wireless Sensor Networks (MWSNs) is to maximize the amount of data received at the base station during the network lifetime. In this paper, we propose a Cluster-based Self-Organization Scheme (CSOS) for electing cluster-head nodes to evenly balance energy consumption in the overall network. In CSOS, each sensor calculates its weight based on k-density, residual energy and mobility and then broadcasts it to its 2-hop neighborhood with a fixed transmission range. The sensor with the greatest weight in its 2-hop neighborhood will become the cluster-head during a fixed period and its neighboring nodes will then join it. In our experiments, firstly we performed simulations to illustrate the impact of sensor mobility on LEACH and LEACH-C's performance using the same simulated model presented in [1]-[2]. Unfortunately, the obtained results showed that sensor mobility had a significant impact on both protocols performance. Secondly, to prove substantial performance gains of CSOS scheme, we evaluated it with the same simulated model but with mobile sensors. Findings demonstrated that CSOS provided good results in terms of the amount of data received at the sink during the network lifetime when compared with LEACH and LEACH-C.*

Keywords*: Cluster-based, k-density, MWSNs, Residual energy, Self-organization*

I. Introduction

MWSNs consist of a large number of small mobile sensors with a limited battery power, which may be deployed more densely in an interest area for sensing certain phenomena and reporting data through a short range and low radio transmission range to a data collection point called sink or base station. Mobile sensors collaborate with each other to form a sensor network capable to accomplish sensing tasks during the whole system lifetime.

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 environment and health related applications etc. [3]-[4].

However, sensors that compose them, present some constraints such as low storage and processing power, limited battery lifetime, and short radio transmission ranges.

In a flat network, all sensor nodes are identical and there is no predetermined architecture.

Although its installation is simple and efficient for small networks, it lacks scalability and increases energy dissipation for reporting collected data to a remote sink in dense MWSNs.

Thus, to relay streams of data while minimizing the overall energy consumption and the broadcast overhead in the network, the design of an efficient scheme proves to be necessary to allow MWSNs to accomplish their missions with a great effeteness. One promising approach is to utilize a clustering network architecture, which is considered as an efficient approach to mimic the operation of a centralised infrastructure and therefore benefit of its substantial performance gains for small networks. Hence, we should involve adequate criterion for cluster-head election to generate steady and balanced clusters.

All of the above constraints imposed by sensors make the design of an efficient scheme for maximizing the amount of data spent to the remote base station during MWSNs lifetime a real challenge. In response to this challenge, we propose a Cluster-based Self-Organization Scheme for MWSNs (CSOS), which consists of grouping sensors into a set of disjoint clusters. In CSOS scheme, the sensor with the highest weight in its 2-hop neighborhood not affiliated becomes the cluster-head. The weight of each node 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 membership is, at most, two hops from its corresponding cluster-head contrary

Manuscript received and revised February 2008, accepted March 2008

to LEACH [1] (Low Energy Adaptive Clustering Hierarchy) and its variant LEACH-C [2] (LEACH Centralized), which allow only single-hop clusters to be constructed.

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

Considering a stable clustering scheme considerably alleviates communication and broadcast overhead among the sensor nodes. In our proposed scheme, we aimed to generate steady and balanced clusters in the purpose to maximize the amount of data spent to the remote base station during the network lifetime while minimizing energy consumption in the entire network. For this, we involved k-density and mobility factors in nodes' weight computation in order to guarantee less clusters structure changes, as well as the energy factor to ensure a long cluster-head lifetime.

In our experiments, first we conducted extensive simulations to illustrate the impact of sensor mobility on LEACH and LEACH-C's performance. For that, we used the same scenario presented in [1]-[2] in both cases: model with stationary sensors and model with mobile sensors. Then, we also carried out simulations with the same scenario in order to demonstrate the substantial performance gains of our proposed scheme. On the other hand, we compared the results obtained with both protocols LEACH and LEACH-C in terms of the amount of data received at the base station per the same amount of energy dissipation, as well as we estimated the number of nodes alive per the amount of data received at the base station.

The remainder of this paper is organized as follows. In Section 2, we provide the necessary notations and hypothesis for describing our scheme. Section 3 reviews several cluster-based algorithms that have been previously proposed. In Section 4, we present our scheme, and Section 5 presents a performance analysis of proposed scheme and compares it to other protocols' performance. Finally, Section 6 concludes our paper by pointing out some possible future research directions.

II. Notations and Hypothesis

Before heading into the technical details of our contribution, we first give some definitions and notations that will be used later in our paper.

A mobile wireless sensor network is abstracted as an undirected graph $G=(V,E)$, where V represents the set of wireless sensors 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 u is 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 represented by (1) and the size of this set is known as the degree of u, denoted by $\delta_1(u)$.

$$N_1(u) = \{ v \in V \mid v \neq u \wedge (u,v) \in E \} \quad (1)$$

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 $N_2(u)$:

$$N_2(u) = \left\{ w \in V \;\middle|\; \begin{array}{c} (v,w) \in E \text{ where} \\ w \neq u \wedge (u,v) \in E \wedge w \notin N_1(u) \end{array} \right\} \quad (2)$$

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

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

In a general manner, the set of k-hop neighborhood 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 $\delta^k(u)$.

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

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

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:

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

TABLE I
COMPUTATION OF 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

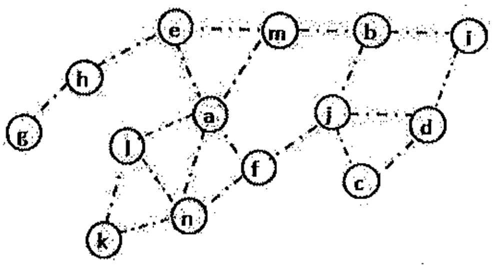

Fig. 1. Example of an abstracted wireless network

However, we are interested only in calculating the 2-density nodes so as not to weaken the *CSOS* scheme's performance as presented in (7):

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

Table I illustrates the 2-density calculation of the nodes composing the network presented in Fig. 1.

In this paper, we assume that all sensors are given in a two dimensional space and each sensor has an omni-directional antenna what allows for a single transmission of it can be received by all sensors within its vicinity. 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 sensor node to be a cluster-head, and a greater weight means higher priority.

III. Related Work

Recently, many cluster-based schemes [1], [2], [5]-[11] have been proposed to tackle the main challenges in WSNs. However, these contributions utilize models with stationary sensors. To the best of our knowledge, this paper is the first to tackle maximization of the amount of data sent to the base station during the network lifetime in MWSNs. 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 and maximize the amount of data sent to the base station in WSNs.

In [1], the authors propose LEACH protocol, which is a distributed, single hop cluster-based scheme without any central control or dependence on other routing schemes. In LEACH, after each round, each sensor node elects itself as cluster-head with a probability which is equal to:

$$p_{CH}=k\frac{E(u)}{E_{Total}} \quad (8)$$

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 the 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, single-hop 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, 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 quickly exhaust their power battery either when they coordinate among their cluster members or when they are placed away from the base station. Therefore, the authors proposed an extended 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 enhanced version requires each sensor to be capable of aggregating data, which increases the overhead for all sensors. Hence, to improve the performance of this strategy, in [6], the authors focus on heterogeneous sensor networks instead of using homogeneous sensors, in which two types of sensors are deployed: super and basic sensors. Super sensors have more processing and communication 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 via multi-hop mode.

Furthermore, in [2], another variant of LEACH called LEACH-C has been conceived to improve LEACH performance. This variant utilizes a centralized architecture to select cluster-heads while using base station and location knowledge of sensors. However, it enormously deploys energy to achieve this task and it consequently increases network overhead since all sensors send their location information to the base station at the same time during every set-up phase. On the other hand, several works have proven that a centralized architecture is particularly suitable for small networks, whereas it lacks scalability to handle the load when the number of nodes increases in a network.

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 clusters during the set-up phase. Moreover, it aims to generate balanced clusters to avoid overload. 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 a multiple cluster-to-cluster (CH-to-CH) routing paths to use them 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 phase, cluster-heads transmit collected data from sensors to the remote base station through the CH-to-CH routing paths [8]. However, BCDCP presents the same drawbacks as LEACH-C since it uses a centralized architecture to elect cluster-heads.

In the proposed scheme, we tackled the self-organization in MWSNs to maximize the amount of data received at the remote base station. To achieve this goal, we aimed to generate steady and balanced clusters.

IV. Contribution

In this section, we present our proposed scheme that enables to generate steady and balanced clusters. To carry out our scheme, we assume that:

- all sensors are homogeneous with constrained energy and the same transmission range,
- the network topology changes, and sensors move with a speed ranging between 0 and 10 (m/s),
- each sensor operates asynchronously without a centralized controller and does not require that the location of sensors be known.
- the sensors make all decisions without reference to a centralized controller,
- each sensor is able to calculate its weight according to its k-density, residual energy and mobility,
- the sensors have 2-hop neighborhood knowledge.

IV.1. Cluster Formation

Cluster formation process consists of grouping sensors into a set of disjoint clusters, thus giving the network a hierarchical organization. Each cluster has a cluster-head, which is elected among its 2-hop neighborhood based on nodes' weight. The weight of each sensor is a combination of the following parameters k-density, residual energy and mobility as illustrated by the formula (9). The coefficient of each parameter represents its implication degree, it can be chosen depending on the application. Therefore, we attribute adequate values to the different coefficients in the purpose to generate steady clusters:

$$Weight(u) = \alpha * P_{2-density} + \beta * P_{Energy} + \gamma * P_{Mobility} \quad where \quad \alpha + \beta + \gamma = 1 \tag{9}$$

As mentioned above, each sensor uses weight criteria to decide whether to be a cluster-head in its 2-hop neighborhood during a round.

At the beginning of each round, each sensor calculates its weight and broadcasts it to its 2-hop neighborhood via a 'Hello' message as well as it eavesdrops its neighbor's 'Hello' message. Then, the sensor node with the greatest weight among its 2-hop neighborhood is chosen as the cluster-head (CH) for the current round. We involved k-density factor in weight computation of each node in the purpose to generate steady clusters. On the other hand, we also implied remaining energy factor in order to choose the sensor having more energy in its 2-hop neighborhood, which would be capable to handle the cluster load, ensure a long cluster lifetime, and avoid the launch of re-election process before the round ends.

In our context, cluster-head is responsible to coordinate among the cluster members, aggregate their data and transmit them to the remote base station. Accordingly, cluster members do not transmit their gathered data directly to the sink, but only to their corresponding cluster-head. Furthermore, cluster-head enables to manage its own cluster, to accept or refuse adhesion of new arrivals based on its capacity without perturbing the functionality of the other cluster members. In spite of this heavy load supported by the cluster-head, we find several cluster-based schemes, which keep to them status as cluster-heads for a long time, what can rapidly exhaust their battery power. Hence, we propose to set up periodically cluster-head election process after each round so that cluster-head relinquish its role as cluster-head node either when the round ends or when it migrates towards another cluster. On the other hand, each cluster has a size ranging between two thresholds $Thresh_{Lower}$ and $Thresh_{Upper}$ for better management except in certain cases wherein its value can be lower than $Thresh_{Lower}$, and that cluster members are, at most, two hops from their respective cluster-head. Furthermore, if during the set-up phase, there is formation of clusters whose size is lower than $Thresh_{Lower}$, then re-affiliation process will be triggered in order to reorganize the clusters.

In *CSOS* scheme, each sensor is identified by a state vector as follow: *(*$Node_{Id}$*,* $Node_{CH}$*, Weight, Hop, Size,* $Thresh_{Lower}$*,* $Thresh_{Upper}$*)* where $Node_{Id}$ is the sensor identifier, $Node_{CH}$ represents the identifier of its cluster-head, in particular if this sensor is a cluster-head then its identifier will be assigned to $Node_{CH}$, *Hop* indicates the number of hops separating it from its respective cluster-head, and *Size* represents the size of cluster to which it belongs. Moreover, each sensor is responsible for maintaining a table called '$Table_{Cluster}$' in which the information of the local members cluster 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 *CSOS* scheme's performance. At the beginning of each round, each sensor calculates its weight and generates a Hello message, which includes two extra fields addition to other regular contents: *Weight* and $Node_{CH}$, where $Node_{CH}$ is set to zero.

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

a) The Set-Up Phase.

Cluster-head election process proceeds in the following way as illustrated by Fig. 2. Initially, a random node initiates clustering process while broadcasting a 'Hello' message to its $N_{12}(u)$ neighbors. The sensor with the greatest weight among its $N_{12}[u]$ neighbors is chosen as the cluster-head *(CH)* during 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. Each sensor belonging to $N_1(Node_{CH})$ whose $Node_{CH}$ value is equal to zero i.e. does not belong to any cluster and its weight is lower 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. Thereafter, it increments its *Size* value and the affiliated node sets *Hop* value to 1 and $Node_{CH}$ with $Node_{Id}$ of its corresponding cluster-head, and broadcasts received message again with the same transmission power to its neighbors, otherwise CH simply drop the message of affiliation request. 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 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. Finally, at the end of the set-up phase, each sensor will know which cluster it belongs to and which sensor is its cluster-head.

b) 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 clusters.

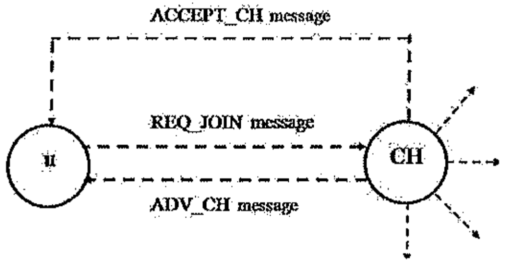

Fig. 2. Affiliation procedure of a node to a cluster

Hence, in the second 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 the purpose to reduce the number of clusters formed and obtain balanced clusters.

The execution of the second 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 to them. 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 and whose weight is greater than its weight.

After the unfolding of both phases, we obtain balanced and steady clusters considering that we involved k-density, residual energy and mobility to structure network in clusters. The first phase generates clusters whose size does not reach the $Thresh_{Upper}$ threshold while the second arranges clusters by re-affiliating nodes that belong to clusters whose size is lower than $Thresh_{Lower}$ towards the nearest cluster-heads, what permits to reduce the number of clusters formed. The both phases: set-up and re-affiliation would end after a fixed interval of time, which should be long enough to guarantee that every sensor can affiliate to a cluster. Furthermore, after the end of both phases, clusters are formed and each cluster-head creates the time schedule in which time slots are allocated for intra-cluster communication, data aggregation, inter-cluster communication, maintenance process as mentioned in Fig. 3. Then, the generated clustered sensor network starts the steady phase of round to transfer collected data to the remote base station.

CSOS scheme can greatly alleviate the broadcasting traffic overhead for intra-cluster since it utilizes TDMA (Time Division Multiple Access) schedule with ACA (Adaptive Channel Assignment), in which a division of channels is assigned to current cluster members in a fixed manner while the rest of channels are reserved for prospective new arrivals.

Moreover, since each cluster has its own spread spectrum code, *CSOS* enables to minimize the impact of the interference between clusters during intra-cluster communications.

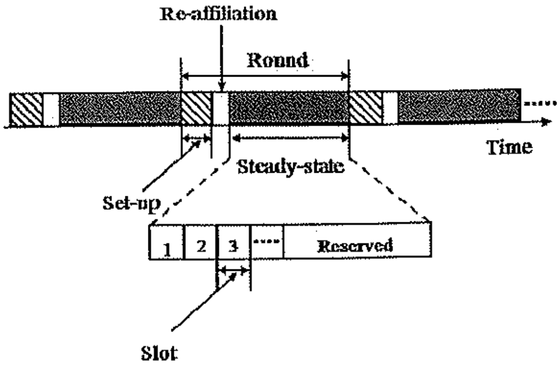

Fig. 3. Time schedule for cluster management

IV.2. Cluster Maintenance

The cluster maintenance process should be triggered if a cluster lost its cluster-head either when the latter exhausts its battery power or migrates towards other 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 as a similar way to the set-up phase where a random node among the members cluster initiates the clustering process. On the other hand, the cluster-heads manage easily the procedures of adhesion and departure of nodes during data transfer phase since the adopted mechanism reserved time slots to achieve these prospective operations.

V. Simulation Results

In our experiments, firstly we performed extensive simulations to illustrate the impact of sensor mobility on LEACH and LEACH-C's performance. For that, we used the same simulated model presented in [1], [2] and we carried out simulations with the both scenarios: model with stationary sensors and model with mobile sensors. Secondly, to illustrate the substantial performance gains of our proposed scheme, we carried out simulations with the same model and compared the results obtained with those obtained with both protocols LEACH and LEACH-C. We performed these simulations with NS-2 [12] using the MIT_uAMPS ns code extensions [13]. 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. Furthermore, simulations were carried out until all the sensors exhausted their battery power and the average values were calculated after each round. We performed simulations using two distinct values for threshold $Thresh_{Upper}$: 30, 50, and a fixed value for threshold $Thresh_{Lower}$=15.

As mentioned above, we used the same energy parameters and radio model as discussed in [1], [2] wherein energy consumption is mainly divided into two parts: receiving and transmitting message. The transmission energy consumption needs additional energy to amplify the signal depending on the distance to the destination. Thus, to transmit a k-bit message to a distance d, the radio expends energy as described by the formula (10), where ε_{elec} is the energy consumed for radio electronics, $\varepsilon_{friss\text{-}amp}$ and $\varepsilon_{two\text{-}ray\text{-}amp}$ for a amplifier:

$$E_{Tx} = \begin{cases} \varepsilon_{elec} \times k + \varepsilon_{friss-amp} \times k \times d^2 & if\ d < d_{Crossover} \\ \varepsilon_{elec} \times k + \varepsilon_{two-ray-amp} \times k \times d^4 & if\ d \geq d_{Crossover} \end{cases} \quad (10)$$

The reception energy consumption is represented by the formula (11):

$$E_{Rx} = \varepsilon_{elec} \times k \quad (11)$$

Simulated model parameters are set as shown in Table II.

TABLE II
PARAMETERS FOR SIMULATION

Parameter	Value
Network Grid	(0,0) × (100,100)
Position of 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
$Thresh_{Upper}$	30, 50
$Thresh_{Lower}$	15

To illustrate the impact of sensor mobility on LEACH and LEACH-C's performance, we evaluated the both protocols with the same simulated model described in [1] with both scenarios: model with stationary sensors and model with mobile sensors. Figs. 4a-b demonstrate that the sensors mobility had an important impact on both protocols' performance. Indeed, LEACH and LEACH-C performance respectively degrade of 14% and 21% in terms of the amount of data received at the base station during the network lifetime.

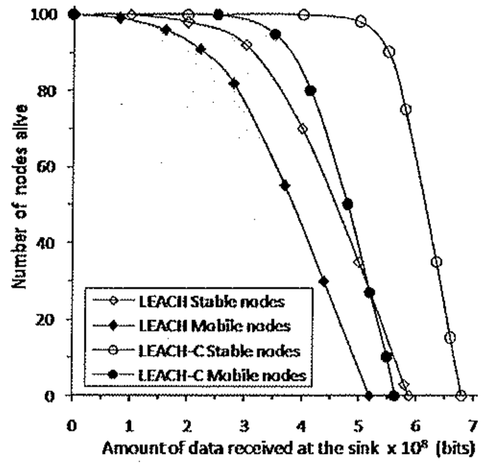

Fig. 4a. Comparison of LEACH and LEACH-C performance with stable and mobile model in terms of the number of nodes alive according to the amount of data received at the sink

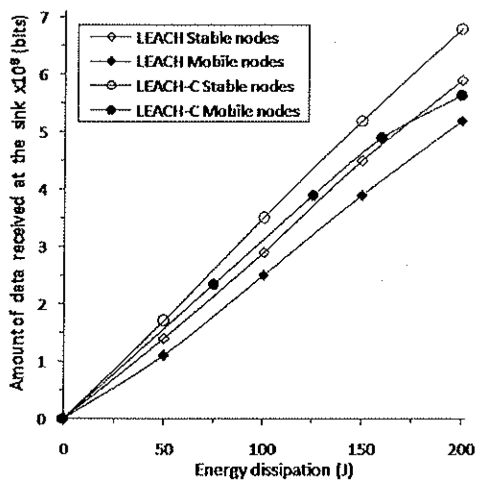

Fig. 4b. Comparison of LEACH and LEACH-C performance with stable and mobile model in terms of the amount of data received at the sink with the same amount of energy dissipation

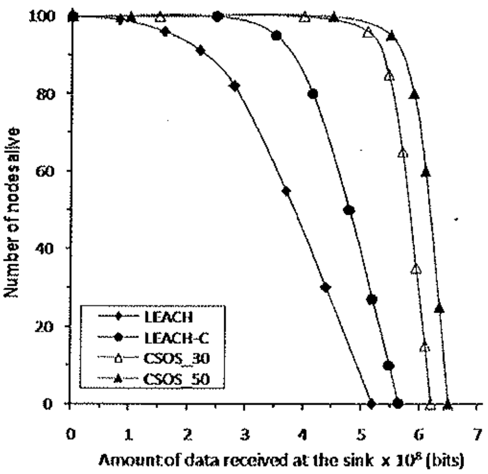

Fig. 5a. Number of nodes alive according to the amount of data received at the sink with mobile simulated model

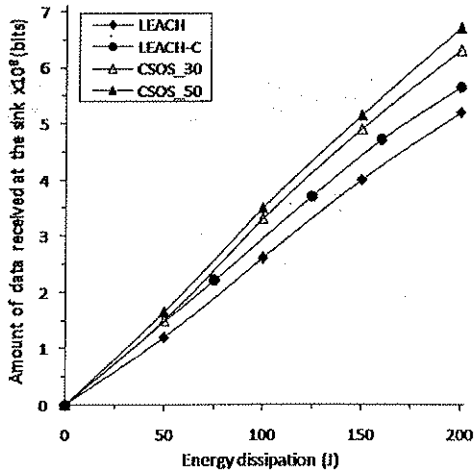

Fig. 5b. Amount of data received at the sink according to the same amount of energy dissipation with mobile simulated model

Fig. 5a shows that *CSOS_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 *CSOS_50* largely outperforms them. Moreover, in Fig. 5a, the shapes of the curves of *CSOS_30* and *CSOS_50* show 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 contrary to LEACH wherein the demise of sensors is done gradually during network lifetime. On the other hand, Fig. 5b illustrates that *CSOS_30* and *CSOS_50* outperform LEACH and LEACH-C in terms of the amount of data received at the base station with the same amount of energy.

VI. Conclusion

In this paper, we have proposed a Cluster-based Self-Organisation Scheme (*CSOS*) for Mobile Wireless Sensor Networks (MWSNs) based on weighted criteria for cluster-heads' election. The *CSOS* scheme carries out periodically cluster-head election process after each round what permits to evenly distribute energy load among the sensors.

Simulations results demonstrate that the *CSOS* scheme provides better performance than LEACH and LEACH-C in terms of the amount of data received at the base station during network lifetime. Furthermore, it considerably diminishes the time difference between the demise of the first and last sensor relatively to LEACH.

With these results obtained, the *CSOS* scheme can be regarded as a promising scheme to deal with the main challenges in MWSNs. Therefore, its evaluation could be the subject of future work.

Acknowledgements

The authors would like reviewers for their valuable feedback.

References

[1] Heinzelman, 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) , Vol.8, Page: 8020, 2000, ISBN:0-7695-0493-0.

[2] W. R., Heinzelman, A. P., Chandrakasan, H., Balakrishnan, An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions Wireless Communications*, Vol. 1, n. 4, pp. 660-670, 2002.

[3] Chong, C. Y., Kumar, S. P., *Sensor networks: Evolution, opportunities, and challenges*, Proceedings of the IEEE, Vol. 91, n. 8, pp. 1247-1256, 2003, ISSN: 0018-9219.

[4] Estrin, D., Govindan, R., Heidemann, J., Kumar, S., *Next century challenges: Scalable coordination in sensor networks*, Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking *MobiCom'99*, pp. 263-270, 1999, ISBN:1-58113-142-9.

[5] V., Mhatre, C., Rosenberg, 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 clustering scheme in wireless sensor networks*, Proceedings of the 24th IEEE International Workshop Performance, Computing, and Communications Conference, IPCCC2005, pp. 535-540, 2005, ISSN:1097-2641, ISBN: 0-7803-8991-3.

[7] Lee, G., Kong, J., Lee, M., Byeon, O., *A Cluster-based Energy Aware Routing Protocol for Sensor Networks*, Proceedings (466) Parallel and Distributed Computing and Systems, Page: 204, 2005, ISSN: 1027-2658, ISBN: 0-88986-525-6.

[8] S. D., Muruganathan, D.C.F., Ma, R.I., Bhasin, A. O., Fapojuwo, A Centralized Energy-Efficient Routing Protocol for Wireless Sensor Networks, *IEEE Communication Magazine*, Vol. 43, n. 3, pp. S8-S13, 2005.

[9] Lindsey, S., Raghavendra, C.S., *PEGASIS: power efficient gathering in sensor information systems*, Proceeding of the IEEE Aerospace Conference, Vol. 3, pp.1125-1130, 2002, ISBN: 0-7803-7231-X.

[10] Manjeshwar, A., Agrawal, D. P., *TEEN: A Protocol for Enhanced Efficiency in Wireless Sensor Networks*, Proceedings of 15th International Parallel and Distributed Processing Symposium (IPDPS'01) Workshops, Page: 189, 2001, ISBN:0-7695-0990-8.

[11] L., Qing, Q., Zhu, Q., M., Wang, Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks, *Elsevier Computer Communications*, Vol. 29, n. 12, pp. 2230-2237, 2006.

[12] NS-2, Network Simulator, http://www.isi.edu/nsnam/ns/ns-build.html.

[13] MIT_uAMPS LEACH NS2 Extensions, www.ece.rochester.edu/research/wcng/code/index.html.

Authors' information

[1]Laboratory of Computer Engineering of Franche-Comte (LIFC), 16, Route de Gray, Besançon 25030, France.
Tel./Fax: +33-3-81-66-20-75 / +33-3-81-66-64-50
E-mail: lehsaini@lifc.univ-fcomte.fr

[2]Laboratory of Computer Engineering of Franche-Comte (LIFC), 16, Route de Gray, Besançon 25030, France.
Tel./Fax: +33-3-81-66-64-57 / +33-3-81-66-64-50
E-mail: guyennet@lifc.univ-fcomte.fr

[3]Faculty of Engineering Sciences, Tlemcen University, 13000, Algeria.
E-mail: m_feham@mail.univ-tlemcen.dz

M. Lehsaini received his B.S and M.S degrees in Computer Engineering from Oran University of Sciences and Technology (USTO), Algeria in 1992 and 2000, respectively. In 2000, he joined Tlemcen University where he worked as an assistant professor in Computer Science and Engineering. Dr. Mohamed is a member of SDR "Networks and Distributed Systems" group at LIFC " Laboratory of Computer Engineering of Franche-Comté University" and also a member of STIC Laboratory at Tlemcen University. His research interests include wireless sensor networks and discovery of services.

H. Guyennet is a professor of Computer Science at the University of France-Comté (Besançon – France). He received his PhD in 1985 and became a full professor at the same university in 1995. Prof. Hervé is working on distributed systems : load balancing, cooperative work, distributed platform, quality of service in networks and he is the head of the "Networks and Distributed Systems" group at LIFC "Laboratory of Computer Engineering of Franche-Comté University". He has been the scientific advisor for 15 theses and has published more than 80 papers.

M. Feham received his Dr. Eng. degree in optical and microwave communication from the University of Limoges (France) in 1987, and his PhD in Science from the University of Tlemcen (Algeria) in 1996. Since 1987, Prof. Mohammed FEHAM has been an Assistant Professor and Professor of microwave and communication engineering. He has served on the Scientific Council and other committees of the Electronics and Telecommunication Departments of Tlemcen University. His research interests are wireless networks and discovery of services.