



A Hybrid Weight-Based Clustering Algorithm for Wireless Sensor Networks

Cheikh Sidy Mouhamed Cisse, Cheikh Sarr*

Faculty of Science and Technology, University of Thies, Thies, Senegal

Email: sidymouhamed12@gmail.com, *csarr1979@hotmail.com

Received 3 May 2015; accepted 22 May 2015; published 28 May 2015

Copyright © 2015 by authors and OALib.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Clustering in wireless sensor network (WSN) is an efficient way to structure and organize the network. The cluster head (CH) forms dominant set in the network responsible for the creation of clusters, maintenance of the topology and data aggregation. A cluster head manages the resource allocation to all the nodes belonging to its cluster. In this paper, we propose a novel distributed clustering approach called Hybrid Weight-based Clustering Algorithm (HWCA). HWCA considers the neighborhood, the distance from the base station combined with the consumed energy as a hybrid metric to elect cluster head. The time required to identify the cluster head does not depend on the number of node and can be computed in a finite number of iterations. Our solution also aims to provide better performance such as maximizing the life time, reducing the number of lost frames in order to satisfy application requirements. Simulation results show that HWCA improves the network lifetime and reduces the number of lost frames compared with other similar approaches.

Keywords

Wireless Sensor Network (WSN), Cluster Head, Cluster, Election, Weight, Hybrid Metric

Subject Areas: Network Modeling and Simulation

1. Introduction

A Wireless Sensor Network (WSN) is an ad hoc network with a large number of nodes that are smart micro-sensor able to collect, transmit data (like heat, humidity, vibration...) and convert them into digital quantities autonomously [1]. Each of these micro-sensor is basically equipped with a sensing device to collect data from the environment, a processing unit to do some operations on data, a transceiver to send and receive collected, and an energy source to provide the required energy to operate (usually a battery) [2]. The obtained data are transmitted over multiple hops to a sink also called base station (BS) [3]. The end user can access the data via Internet so

*Corresponding author.

that the data are processed.

The concept of dividing the geographical region to be covered into small zones has been presented implicitly as clustering in the literature [4]. The clustering technique means partitioning network nodes into groups called clusters, giving to the network a hierarchical organization. The grouping of sensor nodes into clusters has been widely pursued by the research community in order to achieve the network scalability objective [5]. Each sensor node in a cluster collects information and transmits them directly to the base station which can quickly deplete their energy that will reduce the network lifetime. To solve this problem, a node is elected as cluster head of the group. Its role is to aggregate data from other nodes in the cluster and then transmits these data to the base station with a long distance communication. Therefore, it is obvious that cluster head should have more resources than other nodes.

The aim of this paper is to propose a distributed algorithm to select these cluster head nodes. The election is based on a hybrid metric taking into account three parameters: the one hop neighborhood, the amount of energy consumed and the distance from nodes to the base station.

The rest of the paper is organized as follows: Section 2 presents relative works for clustering on wireless sensor networks. In Section 3, we propose the Hybrid Weight-based Clustering Algorithm (HWCA). Simulation results are presented in Section 4 while conclusions are offered in Section 5.

2. State of Art

Literature proposes several techniques for cluster formation and cluster head selection. All solutions aim to identify a subset of nodes within the network and bind it a leader. The first solutions are based on a single metric to elect a node as cluster head. The evolution of these algorithms has proven that the combination of several metric is more efficient for better performance.

2.1. Algorithms Based on a Single Metric for Cluster Head Election

The HCC (*High-Connectivity Clustering*) algorithm proposed in [6] use connectivity for the selection of cluster head. The authors of this algorithm consider that a node having a higher degree (number of neighbors) has better connectivity and is therefore preferable to be elected as cluster head. Thus, the node with the highest degree in its one-hop neighborhood becomes cluster head. However, due to the criterion of degree, HCC builds very dense clusters in small numbers of iterations. In fact, clusters are very sensitive to the mobility of nodes. This results lead to frequently reconstructions of cluster structure. HCC operates in synchronous mode and uses TDMA access method to avoid collisions.

The authors of [7] have proposed a distributed clustering algorithm called LEACH for hierarchical routing algorithms in wireless sensor networks. The cluster head election is based on generation of a random number and assigns this role to different nodes according to a Round-Robin policy to ensure fair energy dissipation between nodes. The rounds have approximately the same time interval previously determined [8]. In order to reduce the amount of information transmitted to the base station, the cluster head aggregates data captured by the member nodes that belong to its own cluster, and send an aggregated packet to the base station. If the CH dies, all other nodes that belong to the cluster will not be able to communicate the information they have captured during the current round. Each round consists of two phases. The first named Set-Up Phase during which the network self-organizes into clusters and cluster head are formed. The second phase called Study-State-Phase, during which the data are collected and routed to the base station.

2.2. Algorithms Based on a Multiple Metrics for Cluster Head Election

The use of a single metric to elect cluster head is not wise to generate stability on cluster head formation. Thus, clustering algorithms that combine multiple metrics for the cluster head election have been proposed in the literature.

The Hybrid Energy Efficient Distributed clustering protocol (HEED) [9] has been proposed. It extends the basic scheme of LEACH by using residual energy as primary parameter and network topology features (e.g. node degree, distances to neighbors) as secondary parameters to break tie between candidate cluster head. For cluster selection HEED achieves to power balancing. The clustering process is divided into a number of iterations, and in each iteration, nodes that are not covered by any cluster head double their probability of becoming

a cluster head. Since these energy-efficient clustering protocols enable every node to independently and probabilistically decide on its role in the clustered network, they cannot guarantee optimal set of cluster head.

The MWBCA algorithm (Multi-Weight Based Clustering Algorithm) [10] combines three metric in calculating the weight of a node: the residual energy, the length of time the node remained *cluster head* and the number of neighbor. In this algorithm, nodes with the higher remaining power have the opportunity to be CH. MWBCA uses probability parameter, information power remaining on a node and the average residual energy of neighboring nodes to select the CH.

WCA (Weighted Clustering Algorithm) [11] is a clustering algorithm for Mobile Ad Hoc Networks based on the combination of four metrics to calculate the weight of a node: the difference of degree D_u , the sum of the distances between node u and each of its neighbors P_u , average relative mobility M_u and the time a node remains cluster head Δ_u . Each metric is weighted by a coefficient w_i and the weight of a node is obtained by the following formula: $W_u = w_1\Delta_u + w_2D_u + w_3M_u + w_4P_u$ with $w_1 + w_2 + w_3 + w_4 = 1$. A cluster head can handle ideally M sensor nodes. M is already fixed and represents the size of a cluster threshold, otherwise the optimal number of nodes around a cluster head. Indeed, the authors of this algorithm claim that a cluster with more than M nodes becomes overloaded and rapidly depletes its resources. With this algorithm, the node with the smallest weight is selected as cluster head. Note also that node with less mobility is always best placed to be elected as cluster head.

BLAC (Battery-Level Aware Clustering) proposed in [12] is a novel Battery-Level Aware clustering family of schemes. BLAC considers the battery-level combined with another metric to elect the cluster head. BLAC comes in four distributed and local variants. BLAC aims to keep as many nodes alive as long as possible. The role of cluster head is played by every node in turns in order to balance the energy consumption. All variants are similar to Density-based [13] but the metrics used differ. BLAC combines the remaining energy $B(u)$ with another metric. The remaining energy of node u is defined as followed: $B(u) = \frac{batt(u) \cdot 10}{battcap}$ where *battcap* is

the initial capacity of the node battery (initially the same for every node), $batt(u)$ is the current battery level of node u . This algorithm limits the remaining power between 0 and 10 to limit the frequent changes in the metric to avoid a non-stable cluster hierarchy. The four variants of BLAC are BLAC-bg, BLAC-bs, BLAC-rg and BLAC-rs. BLAC-bg and BLAC-bs apply the same algorithm but differ in the metric they use.

- BLAC-bg for Battery-Level Aware Clustering—Battery deGree is based on node degree. This variant uses a one-hop neighborhood to build the network, so it stabilizes quickly. Any single change has a direct impact on neighbors and so on degree.
- BLAC-bs for Battery Level Aware Clustering—Battery denSity uses the density $\rho(u)$. This variant computes the clustering structure with 2-hop information but the stability is improved because a single node has less impact on its neighbors.
- Battery-Level Aware Clustering—RNG deGree (BLAC-rg) and Battery-Level Aware Clustering—RNG density (BLAC-rs) variants are variations of the first and the second ones. The biggest difference is that the algorithm runs in two steps. Before computing its metric (degree or density), a relative neighborhood graph is computed in order to keep only an interesting subset of nodes. This allows memory storage saving and the use of less computing capacity for the clustering computation.

HWCA is the a distributed clustering algorithm which combines a hybrid metric composed by one hop neighborhood, consumed energy and distance from base station. Yet, nodes naturally change roles over time based on value of the hybrid metric to provide better performance. HWCA also provides efficient cluster head with no predefined size in order to match the underlying network topology and to be reliable to small topology changes. Unlike solutions from literature, HWCA builds dynamic metric efficient clusters in a distributed way. Its main goal is to provide better performance (like for instance network lifetime, delivery ratio, etc.).

3. HWCA: A Hybrid Weight Based Clustering Algorithm

In order to minimize the energy consumption of each micro-sensor and to generally increase the lifespan of the network, we propose the HWCA algorithm whose originality is based on a hybrid metric combining the neighborhood, distance from nodes to the BS and the energy consumed by each node. The HWCA algorithm also provides an energy balance through the network nodes. Indeed, the nodes then change naturally cluster head according to the value of the metric. In our algorithm a clustering organization is run over the network to carry

better performance. Each node sends its data to its cluster head. Once all data are gathered, the cluster head aggregates them and sends them to a base station. In this section, we present first our new hybrid metric and secondly the cluster head election algorithm based on this metric.

3.1. A New Hybrid Metric

We use for cluster head election of a node u a combined weight metric $m(u)$ that takes into account three system parameters:

- $|N(u)|$ the number of nodes in the one-hop neighborhood of node u .
- $dist(u, b)$ the euclidian distance from node u to the base station b .
- $E_c(u)$ the energy consumed by node u .

We also use the following notations:

- N the maximum number of nodes that a cluster head can handle ideally. This is to ensure that cluster head are not over-loaded and the efficiency of the system is maintained at the expected level.
- L_{max} the maximum distance between two nodes that represents the diagonal of the square.
- $E(u)$ the initial energy of node u .

Therefore we defined the combined metric for each node u as:

$$m(u) = \alpha_1 \frac{|N(u)|}{N} + \alpha_2 \frac{dist(u, b)}{L_{max}} + \alpha_3 \frac{E_c(u)}{E(u)}. \quad (1)$$

For easily identification we use:

$$\Delta_1 = \frac{|N(u)|}{N}, \quad \Delta_2 = \frac{dist(u, b)}{L_{max}} \quad \text{and} \quad \Delta_3 = \frac{E_c(u)}{E(u)}.$$

Equation (1) becomes:

$$m(u) = \alpha_1 \Delta_1 + \alpha_2 \Delta_2 + \alpha_3 \Delta_3. \quad (2)$$

The three system parameters Δ_1, Δ_2 and Δ_3 are affected with a certain weighting factors chosen according to the system needs. For example, for improving network lifetime, energy control is very important in WSN networks, thus the weight of the corresponding parameter can be made larger. The flexibility of changing the weight factors helps us applying our algorithm to various networks.

A node has better chance to become cluster head if it's metric $m(u)$ is the smallest in the cluster meaning that we must minimize Δ_1, Δ_2 and Δ_3 . The first component Δ_1 contributing towards $m(u)$ in efficient neighborhood functioning. In fact when node u has fewer neighbor nodes, it carries less information to the base station and Δ_1 becomes smaller. The second component Δ_2 contributing towards $m(u)$ in energy consumption during transmission process. It is known that more power is required to communicate to a larger distance. Therefore when node u is nearer from the base station, it needs less power to transmit and Δ_2 is smaller. The third and last component Δ_3 contributing towards $m(u)$ in efficient energy consumption. Indeed, when node u has not consumed too much energy, the value of Δ_3 becomes smaller.

After values of all the components are identified, we compute the weighted factors α_1, α_2 and α_3 . The contribution of each individual weighing factor represents its importance relatively to the others one. We consider that in wireless sensor network the most important factor relies to the energy consumption, the second one relies to the power transmission and the last one to the neighborhood. Therefore we choose $\alpha_1 = 0.1, \alpha_2 = 0.3$ and $\alpha_3 = 0.6$. Note that these weighing factors are chosen such that $\alpha_1 + \alpha_2 + \alpha_3 = 1$. These values are arbitrary at this time and should be adjusted according to the system requirements.

3.2. Cluster Head Election Algorithm

Based on the preceding calculation of metric $m(u)$ of a node u , we propose a new distributed algorithm that elects a set of node as cluster head. We suppose no mobility in our algorithm even if the analysis can be extended to mobility scheme. However, node mobility would make clustering very challenging since the node membership will dynamically change, forcing clusters to evolve over time. The cluster head election procedure is invoked during system activation and for a service time and also when a cluster head dies or the existing do-

minant set can no longer cover the entire network.

We model a wireless sensor network as a graph $G=(V,E)$ where V is the set of sensors and E is the set of wireless links uv between each pair of sensors u and v which are in radio range of each other. We use the following notations:

- $N(u)$ the neighborhood of a node u .
- $m(u)$ the metric of node u as computed in Equation (1).
- $CH(u)$ the identity of the cluster head of node u .
- $mCH(u)$ the metric of the cluster head of node u .
- $L(u)$ the list of neighbors nodes v that chose u as cluster head (for each $v \in N(u)$, $v \in L(u)$ if $CH(v) = u$).

The goal of **Algorithm 1** is to determine for each node u its cluster head. Once the metric of each node is computed, HWCA runs algorithm 1 at each node as follow: initially, each node u is its own cluster head and there are no other nodes that have chosen u as cluster head (*instruction 1 to 3*). Then, for each node v in the neighborhood of u , if both metric of u and metric of cluster head of u is higher than metric of v (*instruction 4 to 12*) and node u is not cluster head of another node (*instruction 13*), node u chooses v as cluster head (*instruction 14*).

In our algorithm we can notice that if a node doesn't have any other node in its neighborhood, it stays as cluster head since the initialization phase and send directly his data to the base station. When a cluster head receives data from a regular node, it stores them until it needs to send its own data and then sends the aggregated data to the base station.

3.3. An Illustrative Example

We demonstrate our weighted clustering algorithm with the example shown in **Figure 1**. A node can hear broadcast beacons from nodes which are within its transmission range. An edge between two nodes signifies that the nodes are neighbors of each other. Initially, all nodes are cluster head with a specific metric. For example node u has a metric of $3(u, 3)$ and is its own cluster head with metric initialized to 3 ($CH(u) = u$ and $mCH(u) = 3$). **Figure 2** shows the achieved cluster head selection algorithm. Note that two cluster head can be immediate neighbors. **Table 1** explains the final situation of each node on the network.

We suppose that after a certain time, node y and v die. The cluster head reelection procedure is depicted in **Figure 3** and **Table 2** explain the final situation of residual node on the network.

4. Simulations

To evaluate the performances of HWCA, we perform some simulations under the NS-2 simulator (NS-2.35¹). We compare HWCA to LEACH because of its energy efficiency concern. In order to observe different behaviors for LEACH, two values of the p parameter are used 5% and 10% (p is the average number of cluster head in the network).

```

1 CH(u) ← u
2 mCH(u) ← m(u)
3 L(u) = {∅}
4 for v ∈ N(u) do
5   if m(u) > m(v) and mCH(u) > m(v) then
6     for t ∈ (N(u) - {v}) do
7       if CH(t) == u then
8         | add(t,L(u))
9       end
10    end
11  end
12 end
13 if L(u) == {∅} then
14 | CH(u) ← v
15 end

```

Algorithm 1. HWCA algorithm runs at each node u .

¹<http://www.isi.edu/nsnam/ns/>

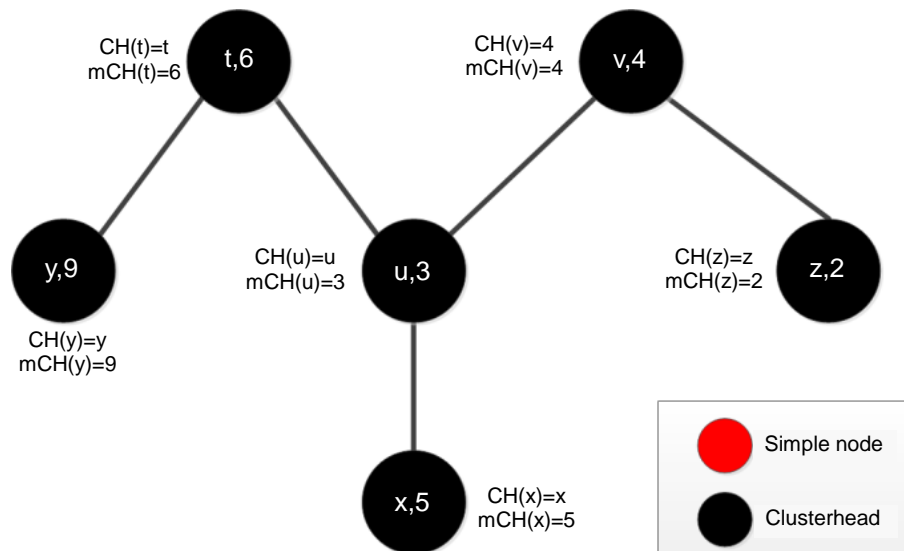


Figure 1. Initial state.

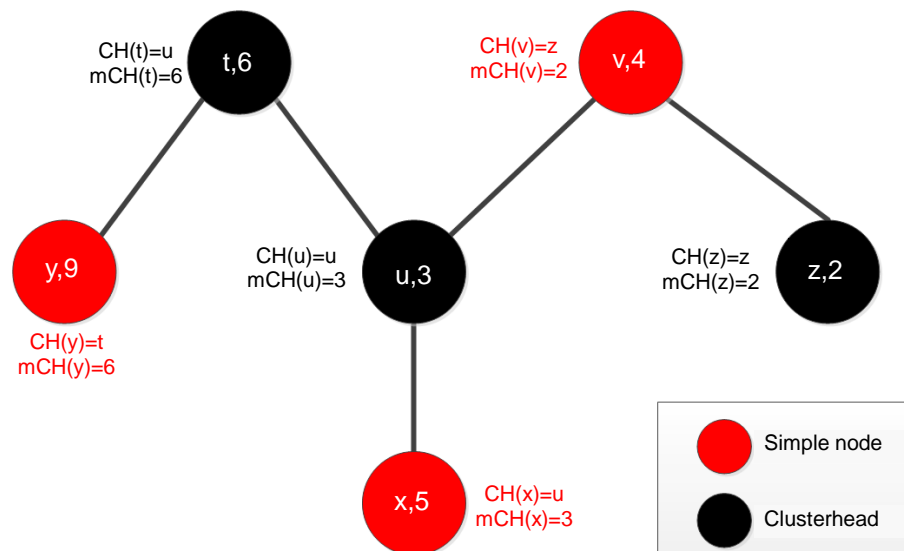


Figure 2. Cluster head creation.

Table 1. Situation of nodes in the network.

Node	CH	Explication
v	z	Node v selects z as CH because it has the lowest metric from its neighborhood. Therefore v becomes a simple node.
y	t	Node y selects t as CH because it has the lowest metric from its neighborhood. Therefore y becomes a simple node.
x	u	Node x selects u as CH because it has the lowest metric from its neighborhood. Therefore x becomes a simple node.
z	z	Node z becomes CH because it has the lowest metric from its neighborhood.
u	u	Node u becomes CH because it has the lowest metric from its neighborhood.
t	t	Even if node t does not have the lowest metric from its neighborhood ($m(u) < m(t)$), it becomes a CH because another node for instance y has already chose t as CH.

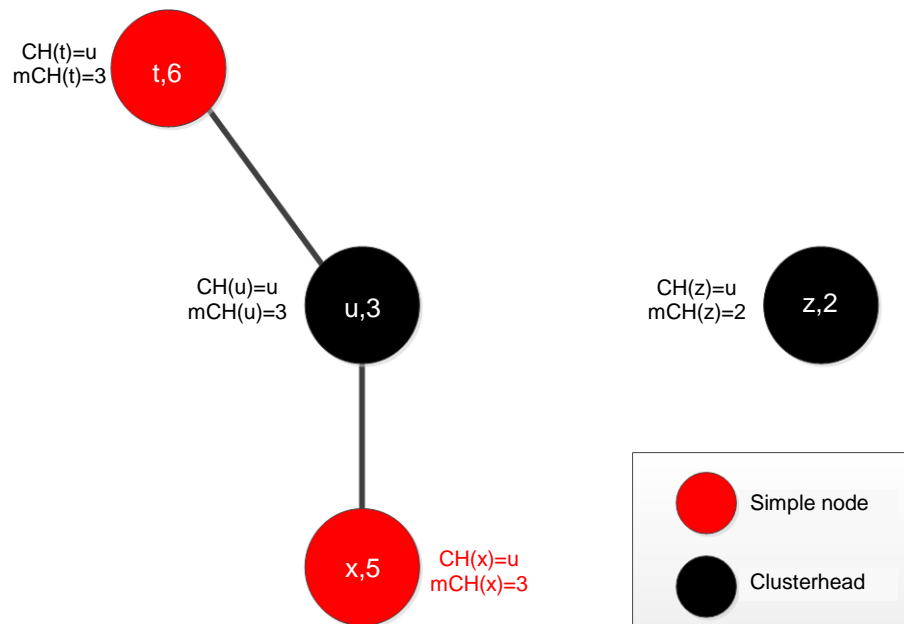


Figure 3. Cluster reelection after the death of node y and v.

Table 2. Situation of nodes in the network after death of nodes y and v.

Node	CH	Explication
x	u	Node x selects u as CH because it has the lowest metric from its neighborhood. Therefore x becomes a simple node.
z	z	Node z has no more neighbors therefore it stays CH.
u	u	Node u becomes CH because it has the lowest metric from its neighborhood.
t	u	Node t changes its state from CH to a simple node because no node chooses t as cluster head and node u has the lowest metric from its neighborhood

In order to use a realistic model for transmitting and receiving costs we consider the Texas Instruments CC2420 ZigBee² wireless module as sensor network. These sensor nodes consume 0.77 mW when idle, 35.46 mW for receiving (Rx), and 31.32 mW for transmitting (Tx). Data traffic is also simulated and each node generates 256 kb/s of data and sends them to its cluster head. When a cluster head receives data from a child, it stores them until it needs to send its own data and then sends the aggregated data to its base station. The number of nodes is 20 either 50 while the initial energy of each one is set to 115 J. Nodes are placed on a 100 × 100 grid.

4.1. Network Lifetime

We define the network lifetime as the time until all nodes die [6]. **Figure 4** and **Figure 5** illustrate the benefits of HWCA while regarding the number of alive nodes over the time. Results show clearly that with HWCA, the network lifetime is extended by respectively 35% and 40% compared to LEACH $p = 5\%$ and $p = 10\%$ for 20 nodes as show on **Figure 4**. On **Figure 5**, with 50 sensor nodes in the network, this lifetime extension remains at respectively 12% and 35% compared to LEACH $p = 5\%$ and $p = 20\%$. These positive results could be explained by the fact that in HWCA, when cluster head are dead or their hybrid metric has increased, other nodes take their role till dying as well, and so on till there is no remaining alive node. This dynamic changing role results in an efficient energy balance over sensor nodes that extend globally the network lifetime. Nevertheless, LEACH performances greatly depend on the number of cluster head that have been set up (through the parameter p). The more cluster head, the more nodes forwarding data and thus the shorter lifetime.

²<http://focus.ti.com/lit/ds/symlink/cc2420.pdf>

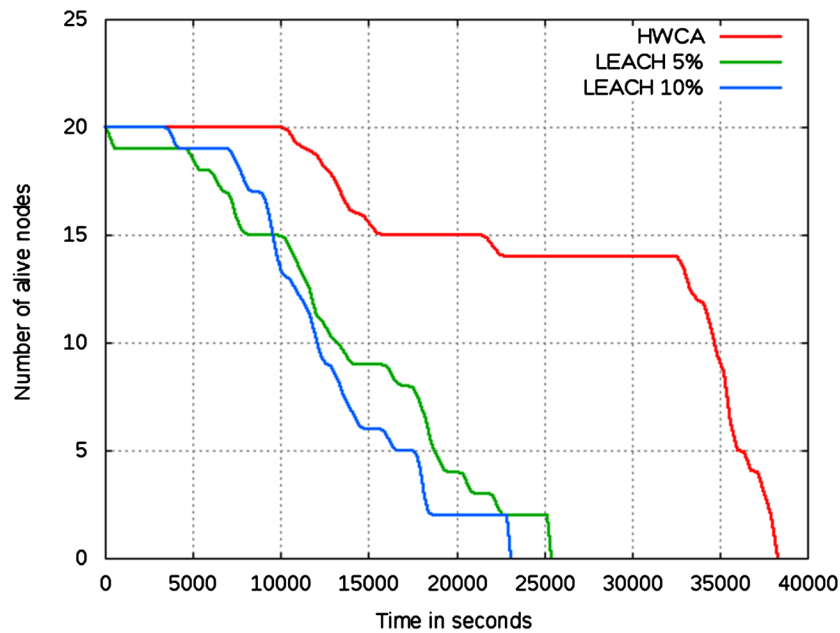


Figure 4. Network lifetime for 20 nodes.

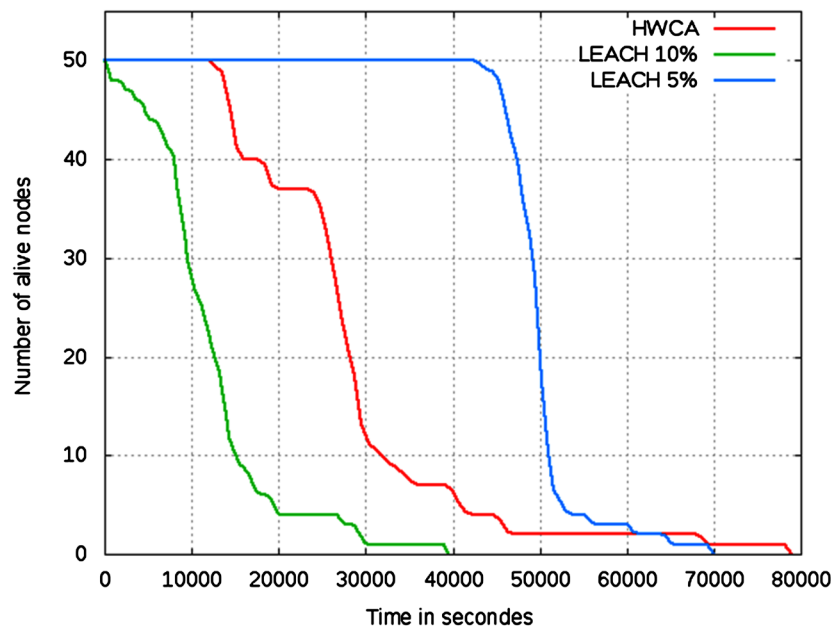


Figure 5. Network lifetime for 50 nodes.

4.2. Delivery Ratio

Figure 6 and Figure 7 display the delivery ratio of every algorithm with regards to time for 20 and 50 nodes. Delivery ratio is computed as the amount of data received by the base station divided by the amount of data sent through the whole network. We observe that the different variants of LEACH ($p = 5\%$ and $p = 10\%$) lose more data than HWCA algorithm regardless of the number of cluster-heads when the network becomes denser (Figure 7) even if this value is better for LEACH $p = 10\%$ when the network is not too load (Figure 6). Indeed, in LEACH, nodes without cluster-head cannot send data so their data are lost. Even when the number of cluster-heads increases, some data are still lost because nodes need to share the medium with other nodes in the same cluster and thus may not have enough time to transmit all data.

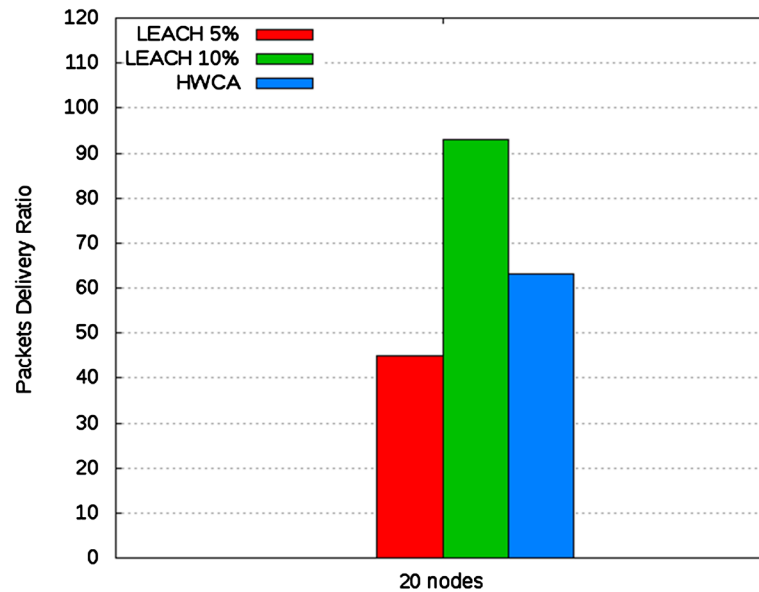


Figure 6. Delivery ratio for 20 nodes.

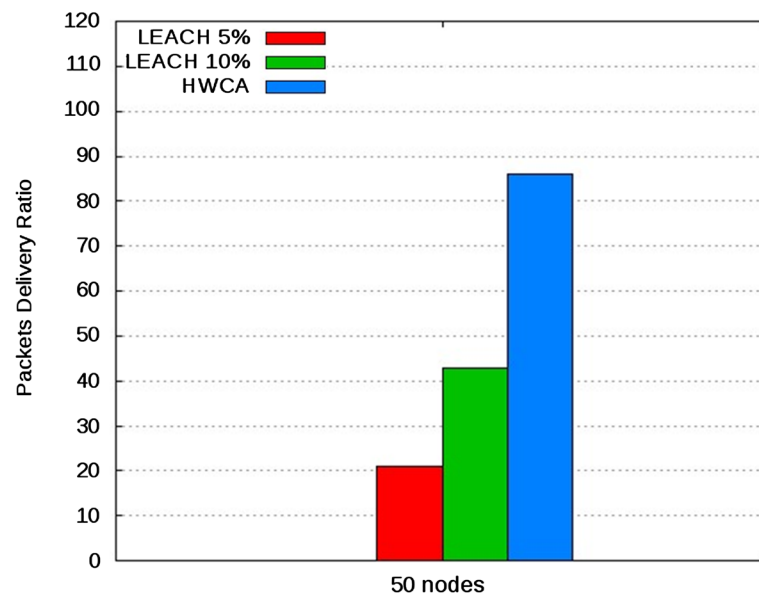


Figure 7. Delivery ratio for 50 nodes.

5. Conclusion and Future Works

In this paper, we have introduced a new clustering algorithm. HWCA combines the neighborhood, the distance from nodes to the BS and the energy consumed by each node as a hybrid metric. HWCA also balances energy consumption through the network as nodes dynamically change their role (from simple node to cluster head and *vice-versa*) depending on the value of the metric and the cluster head selection algorithm. The algorithm is distributed and modifications due to network dynamics are handled locally, allowing scalability. Results show that our proposition improves network lifetime with up to 40% and delivery ratio. For future work we will extend the comparison of HWCA with other algorithms and plan to run experimentations in real world.

References

- [1] Al-Karaki, J.N., Ul-Mustafa, R. and Kamal, A.E. (2004) Data Aggregation in Wireless Sensor Networks-Exact and

Approximate Algorithms. 2004 *Workshop on High Performance Switching and Routing*, 241-245.

<http://dx.doi.org/10.1109/HPSR.2004.1303478>

- [2] Chen, F., Chandrakasan, A.P. and Stojanovic, V.M. (2012) Design and Analysis of a Hardware-Efficient Compressed Sensing Architecture for Data Compression in Wireless Sensors. *IEEE Journal of Solid-State Circuits*, **47**, 744-756. <http://dx.doi.org/10.1109/JSSC.2011.2179451>
- [3] Kour, H. and Sharma, A.K. (2010) Hybrid Energy Efficient Distributed Protocol for Heterogeneous Wireless Sensor Network. *International Journal of Computer Applications*, **4**, 37-41. <http://dx.doi.org/10.5120/828-1173>
- [4] Chandrakasan, A., Heinzelman, W. and Balakrishnan, H. (2000) Energy Efficient Communication Protocol for Wireless Microsensor Networks. *Hawaii ICSS*.
- [5] Joa-Ng, M. and Lu, I.-T. (1999) A Peer-to-Peer Zone-Based Two-Level Link State Routing for Mobile Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications*, **17**, 1415-1425. <http://dx.doi.org/10.1109/49.779923>
- [6] Gerla, M. and Tsai, J.T.C. (1995) Multicluster Mobile Multimedia Radio Network. *Wireless Networks*, **1**, 255-265. <http://dx.doi.org/10.1007/BF01200845>
- [7] Heinzelman, W.R., Chandrakasan, A. and Balakrishnan, H. (2000) Energy-Efficient Communication Protocol for Wireless Microsensor Networks. *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, 10 p. <http://dx.doi.org/10.1109/HICSS.2000.926982>
- [8] Heinzelman, W.B. (2000) Application-Specific Protocol Architectures for Wireless Networks. Doctoral Dissertation, Massachusetts Institute of Technology, Cambridge.
- [9] Younis, O. and Fahmy, S. (2004) HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks. *IEEE Transactions on Mobile Computing*, **3**, 366-379. <http://dx.doi.org/10.1109/TMC.2004.41>
- [10] Fan, Z. and Jin, Z. (2012) A Multi-Weight Based Clustering Algorithm for Wireless Sensor Networks. College of Computer Science & Educational Software Guangzhou University.
- [11] Chatterjee, M., Das, S.K. and Turgut, D. (2002) WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks. *Journal of Cluster Computing (Special Issue on Mobile Ad Hoc Networks)*, **5**, 193-204. <http://dx.doi.org/10.1023/A:1013941929408>
- [12] Ducrocq, T., Mitton, N. and Hauspie, M. (2013) Energybased Clustering for Wireless Sensor Network Lifetime Optimization. *IEEE Wireless Communications and Networking Conference*, 968-973.
- [13] Mitton, N., Sericola, B., Tixeuil, S., Fleury, E. and Guerin Lassous, I. (2011) Self-Stabilization in Self-Organized Wireless Multihop Networks. *Ad Hoc and Sensor Wireless Networks*.