



Balancing energy consumption in clustered wireless sensor networks

Tony Ducrocq, Michaël Hauspie, Nathalie Mitton

► To cite this version:

Tony Ducrocq, Michaël Hauspie, Nathalie Mitton. Balancing energy consumption in clustered wireless sensor networks. International Scholarly Research Notices, 2013. hal-00863123

HAL Id: hal-00863123

<https://inria.hal.science/hal-00863123>

Submitted on 20 Nov 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Research Article

Balancing Energy Consumption in Clustered Wireless Sensor Networks

Tony Ducrocq,¹ Michaël Hauspie,² and Nathalie Mitton¹

¹ Inria Lille-Nord Europe, 59650 Villeneuve d'Ascq, France

² Université Lille 1, 59655 Villeneuve d'Ascq Cedex, France

Correspondence should be addressed to Michaël Hauspie; michael.hauspie@lifl.fr

Received 29 August 2013; Accepted 17 September 2013

Academic Editors: M. Ekström and B. Tavli

Copyright © 2013 Tony Ducrocq et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Clustering in wireless sensor networks is an efficient way to structure and organize the network. It aims at identifying a subset of nodes within the network and binding it to a leader (i.e., *cluster head*). The leader becomes in charge of specific additional tasks like gathering data from all nodes in its cluster and sending them using a longer range communication to a sink. As a consequence, a *cluster head* exhausts its battery more quickly than regular nodes. In this paper, we present four variants of BLAC, a novel battery level aware clustering family of schemes. BLAC considers the battery level combined with another metric to elect the *cluster-head*. The *cluster-head* role is taken alternately by each node to balance energy consumption. Due to the local nature of the algorithms, keeping the network stable is easier. BLAC aims at maximizing the time with all nodes alive to satisfy the application requirements. Simulation results show that BLAC improves the full network lifetime three times more than the traditional clustering schemes by balancing energy consumption over nodes and still delivers high data ratio.

1. Introduction

Multihop wireless sensor networks (MWNs) consist of sets of mobile wireless nodes without support of any preexisting fixed infrastructure. Such large scale wireless sensor networks offer great application perspectives. Wireless sensors are often tiny devices with hardware constraints (low memory storage, low computational resources) that rely on battery. Sensor networks thus require energy-efficient algorithms to make them work properly in a way that suits their hardware features and application requirements. In this paper, we focus on a given application defined by the ANR *BinThatThinks* (<http://binthatthink.inria.fr>) project. The project aims to ease the collect and recycling of waste and reduce its cost through the use of wireless sensors placed on dustbins. Dustbins are also equipped with GPRS chips for long range communications.

In this paper, our goal is to propose a novel clustering algorithm for wireless sensor networks in which each sensor node sends its data to its *cluster head* (potentially through Multihop paths) based on the context of the *BinThatThinks* project. In this context, *cluster heads* collect data from all

sensors in their cluster and send them through their GPRS link. Since activating the GPRS consumes more energy than peer-to-peer communications (as shown in Table 1, Section 6), each node should take the *cluster head* role in turn in order to allow the network to be operational as long as possible without too much communication overhead and structure modifications. Data aggregation is performed at every hop towards the *cluster head*. The clustering scheme should also provide a good trade-off between the number of clusters (the more clusters, the more GPRS activations) and the size of the cluster radius (the bigger radius, the more peer-to-peer communications to reach the *cluster head*). The network needs to last as long as possible without any dead node. Energy should be equally distributed over nodes allowing all nodes to run out of energy at the same time. In this paper, we address this issue and present a novel battery level aware clustering family of schemes BLAC.

To the best of our knowledge, BLAC is the very first distributed clustering algorithm providing nonoverlapping Multihop clusters with energy concerns. Solutions from the literature mainly propose clustering schemes that either do not balance energy consumption over nodes or provide unsuitable

TABLE 1: Simulation parameter.

Parameter	Value
Propagation	Freospace
mac	802.15.4 CSMA-CA
CC2420 idle	0.77 mW
CC2420 Rx	35.46 mW
CC2420 Tx	31.32 mW
CC2420 data rate	250 kbit/s
GPRS idle	6.4 mW
GPRS Rx/Tx	1.25 W
GPRS Tx data rate	26.8 kbit/s
Battery capacity	32 mWh
Hello length	48 bit + 24 bit/neighbor

clusters for our application (overlapping or one-hop clusters). BLAC considers the battery level of nodes combined with another metric (density and degree) to elect the *cluster head*. Yet, nodes naturally change roles over time based on node energy level but in a limited way in order to provide stability to the structure. BLAC comes with four variants; BLAC-bg combines battery level and node degree; BLAC-bs uses the battery level and node density. BLAC-rg and BLAC-rs variants run in two steps. They first apply a graph reduction before computing the clusters. As we will discuss, each of these variants present specific features that make them more suitable than others under different conditions. Simulation results show that BLAC extends the life of the first dying node up to 300% compared to the literature works, by balancing energy consumption over nodes and allowing a better delivery ratio.

The rest of the paper is organized as follows. Section 4 presents relative work for clustering on wireless sensor networks. The context and the application targeted by BLAC are introduced in Section 2. Section 3 introduces useful notations and algorithms from previous works for self-consistency purposes. BLAC is described in detail in Section 5, and simulation results are given in Section 6. We finally conclude in Section 7.

2. Problem Statement, Context, and Motivation

This paper is conforming the *BinThatThinks* project. The project aim is to ease the collection and recycling of waste and reduce its cost. In order to reduce the distance traveled by the garbage truck and to limit manipulations made by operators, *BinThatThinks* adds intelligence to dustbins. Dustbins are equipped with wireless sensors able to wirelessly communicate with each other and with a base station in order to send information about dustbin location, their filling level (does this garbage need to be emptied?), accuracy of the recycling, adequate content, and so on. All these data are processed by the garbage truck to dynamically optimize on-line the collect path by operators to reduce useless manipulations of empty

or dangerous dustbins. One scenario of the project is to equip dustbins with both:

- (1) a wireless module to allow wireless peer-to-peer communications,
- (2) a GPRS modules for longer range communications with the base station.

Since GPRS consumes much more power than a peer-to-peer communication chip, it is not wise to use it on all nodes all the time. Equipping nodes with GPRS chip can be costly in terms of hardware and subscription plan, but negotiations for machine to machine subscription plan are possible.

Indeed, the idea defended in *BinThatThinks* is as follows. In order to reduce power consumption for each node, a clustering organization is run over the network. 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 or the garbage truck using its GPRS module. By doing so, only the *cluster head* activates its GPRS module and drains more energy (mainly because of the use of GPRS module). BLAC proposes a dynamic energy-efficient trade-off between the size and the number of clusters.

3. Notations and Preliminaries

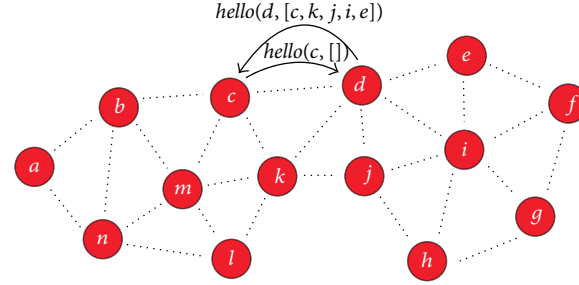
BLAC relies on previous works from the literature. For the sake of self-content and clarity, this section introduces other works on which BLAC relies and related notations.

3.1. Generic Notations. 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 note that $\mathcal{N}(u)$ is the neighborhood of a node u ; that is, $\mathcal{N}(u) = \{v \mid uv \in E\}$ and $\delta(u) = |\mathcal{N}(u)|$ is the degree of node u . In BLAC, clusters are formed through a tree construction in which each node has a parent node. We denote $P(u)$ as the parent of node u in its cluster tree. We note that $\mathcal{H}(u)$ is the *cluster head* of the cluster of node u .

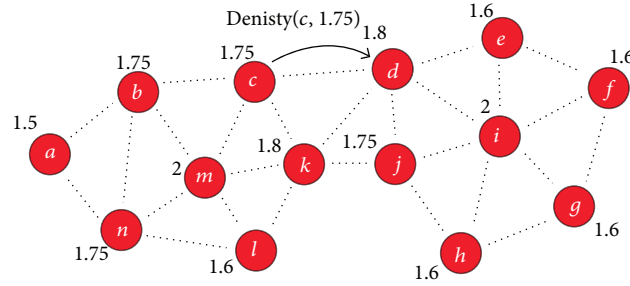
3.2. Density Metric. In the work of Mitton et al. [1], clusters are built as follows. Every node u elects its parent within its neighborhood as the node with the highest density. If u is the node with highest density in its neighborhood, it elects itself as its parent and becomes a *cluster head*. Ties are broken by selecting the node with lowest Id. If a node v is neighbor of two *cluster heads*, it advertises them and the *cluster head* with lowest density elects node v as its parent. This ensures a minimal three-hop distance between *cluster heads*. The density ρ is defined as the ratio of the number of links between and to neighbors of u over the degree of u :

$$\rho(u) = \frac{|\{(v, w) \in E \mid v \in \{u, \mathcal{N}(u)\}, w \in \mathcal{N}(u)\}|}{\delta(u)}. \quad (1)$$

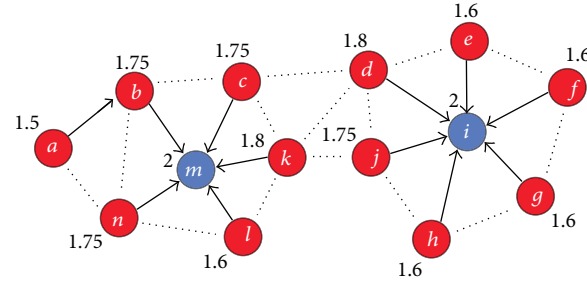
Figure 1(b) shows an example of a network on which we compute the density of nodes. For node a for instance, the degree $\delta(a)$ is equal to 2 and the number of links between neighbors of a is 1. Then, $\rho(a) = (2 + 1)/2 = 1.5$.



(a) Each node sends a *hello* message to its neighbors containing its Id. A second *hello* message is sent with list of neighbors; then each node knows its two-hop neighborhood



(b) Nodes can compute their *density* according to information they gathered; then they send it to their neighbors



(c) Finally, they attach to the node with the highest density in their neighborhood. Nodes with highest density within their neighborhood become *cluster heads*

FIGURE 1: Clusters creation.

3.3. Topology Reduction. Two variants of BLAC, the BLAC- r^* , run in two steps. The first step is the computation of a graph reduction that takes into account the node energy level introduced in [2]. The graph reduction is performed by applying a relative neighborhood graph (RNG) [3]. RNG consists in logically removing an edge in every triangle of the graph, namely, the one with the worst metric value. RNG preserves connectivity. Authors of [2] introduce a new metric called *Power factor*. The *Power factor* of a link is determined based on the energy level of nodes at each end of the link. If both nodes are in normal battery state, *Power factor* is 0, if only one node is in critical battery state, *Power factor* is 1, and if both nodes are in critical battery state, *Power factor* is 2. More common metrics for RNG, that is, the received signal strength indicator (RSSI) and node ids, are used to discriminate nodes.

Algorithm 1 describes how such an RNG is computed on a graph and Figure 2 illustrates it on an example. Note that

most links between two low energy nodes are discarded from the resulting graph. For instance, the link between nodes n and m has been removed, as well as the one between i and j . The link between j and h is kept because there is no link between j and a neighbor of h with a *Better power* factor than that between j and h . Nodes with high level of energy are preferred for communications.

4. Related Work

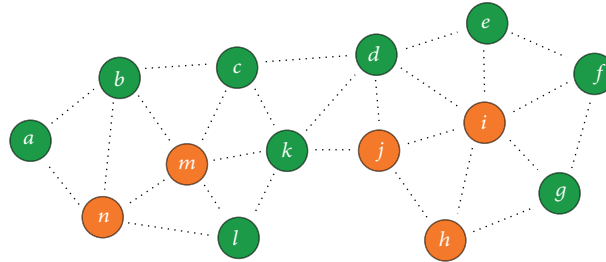
Researchers have proposed several techniques for cluster formation and *cluster head* selection. All solutions aim to identify a subset of nodes within the network and bind it to a leader, that is, *cluster head*, but they do not all aim at the same goal.

First solutions such as LCA [4] proposed by Baker and Ephremides and HCC [5] described by Gerla and Tsai present a similar clustering structure but differ in the metric used.

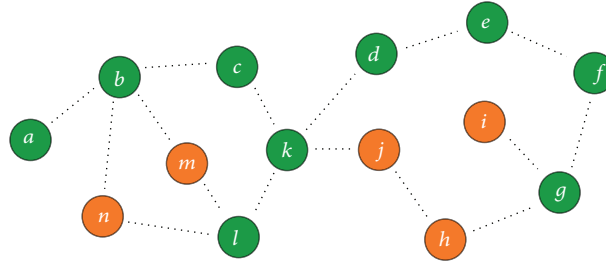
```

(1) begin
(2)  $N_{\text{RNG}}(u) \leftarrow N(u)$ ;
(3) CalculatePowerFactor( $u$ );
(4) {Node  $u$  computes  $PF$  factor of every link within its neighborhood.}
(5) for each  $v, w \in N(u)$  do
(6)   if  $PF(uv) < PF(vw) < PF(uw) \vee PF(vw) < PF(uv) < PF(uw)$  then
(7)      $N_{\text{RNG}}(u) \leftarrow N_{\text{RNG}}(u) \setminus \{w\}$ ;
(8)     Link  $uw$  is removed from RNG.
(9)   else
(10)    if  $PF(uw) < PF(vw) < PF(uv) \vee PF(vw) < PF(uw) < PF(uv)$ 
        then
(11)       $N_{\text{RNG}}(u) \leftarrow N_{\text{RNG}}(u) \setminus \{v\}$ ;
(12)      {Link  $uv$  is removed from RNG.}
(13)    else
(14)      {Link  $vw$  will be removed from RNG.}
(15) Return  $N_{\text{RNG}}(u)$ ;

```

ALGORITHM 1: Calculate RNG at each node u .

(a) Nodes with low energy level are displayed in orange and nodes with high energy level are displayed in green



(b) A battery level-based RNG is applied removing links between low energy nodes

FIGURE 2: RNG construction.

Each node chooses its parent in its neighborhood such as its parent metric is higher than its own one. If a node has the highest metric value in its neighborhood, it becomes *cluster head*. LCA uses as metric the unique Id of a node and HCC as the degree then the node identifier (*Id*) to break ties. LCA and HCC create one-hop overlapping clusters which are quite small. A cluster consists of a *cluster head* and all its one-hop neighbors. Two *cluster heads* can not be neighbors and a *noncluster head* node belongs to at least one *cluster*. Since clusters overlap (a node may belong to several clusters), they are subject to chain reaction in case of changes and are not energy aware since energy is not considered in the clustering construction. Such algorithms are thus not suitable to large

dynamic networks. In addition, they do not have energy concerns.

WCA [6] considers multiple parameters like positions, mobility, and energy. The algorithm is centralized, which makes it hardly scalable. It computes the best weighted sum for a given application and uses this metric in order to elect *cluster heads*. WCA is one of the first clustering algorithms taking into account energy in its construction. Nevertheless, it produces one-hop clusters, which generates more *cluster heads* and then, because of the increase of active GPRS, increases the global energy consumption of the network.

Low et al. [7] offer a clustering solution that minimizes risks of bandwidth saturation of gateways (*cluster heads*).

Dali and Chan [8] describe a solution to balance the energy consumption of the network clusters and to minimize energy consumption in clusters by positioning *cluster head* near the center of the clusters. This proposal gives interesting energy performances, but it is centralized and then is not very scalable.

Some works focus on analyses before deployment [9, 10] and are then not scalable. Indeed, these proposals assume manual or, at least, controlled deployment of nodes, limiting scalability and expansion of the network.

When solutions have energy concerns [2, 3, 9, 11–16], they require not straightforward nodes features (like GPS). They can also require a lot of control messages or, in the case of [12], a declared sink.

The LEACH [17] protocol considers energy at the foundation of the clustering design. Each node becomes *cluster head* based on a predefined probability p defined by the number of expected clusters and whether it has already been *cluster head* in previous rounds or not. A node can be elected as *cluster head* even if its remaining energy is very low. This node can then die rapidly, reducing the network lifetime and triggering a new *cluster head* selection and related traffic. The probability p needs to be tuned off-line; then the number of *cluster heads* does not dynamically adapt the underlying topologies. Once *cluster heads* are advertised, nodes attach to the *cluster head* which requires the least energy to be reached, creating one-hop clusters. Then, every *cluster head* synchronizes the nodes belonging to its cluster and assign them a time slot in which they will be allowed to send their data. Also because of the way of *cluster heads* election, there is a probability for nodes to have no *cluster head* in their neighborhood. This can lead to loss of data if there are too many nodes in a cluster or if they have too much data to send.

In this paper, we focus on distributed Multihop nonoverlapping energy-efficient clusters with no predefined size in order to match the underlying network topology and to be reliable to small topology changes. DDR [18] and density-based algorithms [1] are distributed algorithms proposing Multihop nonoverlapping clusters. They run a similar clustering algorithm and mainly differ in the metric used. DDR [18] uses the degree as metric while density based [1] introduces a new metric called *density*. More details on algorithm and density computation are given in Section 3.2. Despite being efficient in terms of reliability, these two propositions do not consider energy constraints. BLAC is thus based on similar algorithm but differs in the metric used, making the cluster organization dynamic regarding to evolution of energy consumption. This maintains the network available as a whole as long as possible.

Unlike solutions from the literature, BLAC builds dynamic energy-efficient Multihop clusters in a distributed way. Its main goal is to extend network lifetime (where lifetime is considered until first node is running out of its energy) for application contexts as defined in Section 2.

5. Contribution

In order to maximize the lifetime of the network, we introduce BLAC. BLAC comes in 4 distributed and local variants

that do not need large scale modifications when local changes appear. BLAC aims to keep as many nodes alive as long as possible. The role of the *cluster head* is played by every node in turns in order to balance the energy consumption over nodes. All variants use a similar algorithm to that of density based [1] with different metrics.

To be as scalable as possible and to mutualize the GPRS use of *cluster heads*, BLAC has to be distributed.

BLAC combines the *remaining energy* $B(u)$ with another metric. We define the *remaining energy* of node u as

$$B(u) = \left\lfloor \frac{\text{batt}(u) \cdot 10}{\text{battcap}} \right\rfloor, \quad (2)$$

where battcap is the initial capacity of the node battery (similar for every node) and $\text{batt}(u)$ is the current battery level of node u . Then, the remaining power $B(u)$ is an integer between 0 and 10 to limit frequent changes in the metric which would result in a nonstable cluster hierarchy.

BLAC is declined in four variants: BLAC-bg and BLAC-bs from one hand (detailed in Section 5.1), BLAC-rg and BLAC-rs from the other hand (detailed in Section 5.2).

5.1. BLAC- b^* Algorithms. BLAC-bg and BLAC-bs apply the same algorithm but differ in the metric they use. We first detail the metrics they use, respectively.

5.1.1. Degree and Energy. BLAC-bg for battery level aware clustering-battery degree is based on node degree. Algorithm 2 is run over all nodes with $h(u) = \delta(u)$ (Section 3.1). This variant uses a one-hop neighborhood to build the network, so it stabilizes quickly. However, the clustering structure built is sensitive to node apparition and failure as shown in [1] since any single change has a direct impact on neighbors and so on degree.

5.1.2. Density and Energy. BLAC-bs for battery-level aware clustering-battery density uses the density $\rho(u)$ as described in Section 3.2. Algorithm 2 is run with $h(u) = \rho(u)$. This variant computes the clustering structure with two-hop information, but the stability is improved because a single node has less impact on its neighbors as proved in [19].

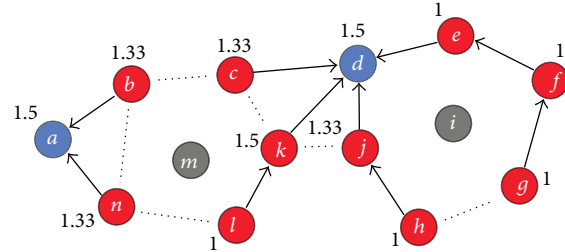
5.1.3. BLAC- b^* . Once the metric is computed, BLAC runs Algorithm 2 with $h(u)$ being either the degree ($\delta(u)$) for BLAC-bg variant or the density ($\rho(u)$) for BLAC-bs, variant and by considering the battery level jointly with that metric, ($g(u) = B(u)$). Figure 1 illustrates Algorithm 2 which runs at each node as follows: each node sends to its neighbors a *hello* message with its *Id* (Figure 1(a)). Knowing its neighborhood, a node sends a new *hello* message with its *Id* and its neighbors *Ids* (Figure 1(a)). With this information, nodes can compute their *density* and send it to their neighbors (Figure 1(b)). Finally nodes can elect their parent (node a elects node b as its parent; node h elects node i , etc.) or decide to elect themselves as a *cluster head* if they have the highest metric (Figure 1(c): nodes m and i become *cluster heads*). Note that clusters are thus built through a tree construction: a elects b that elects m

```

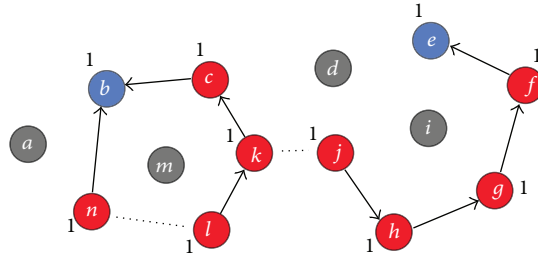
(1) begin
(2)  $C_h \leftarrow 0$ ;
(3)  $Id_{\min} \leftarrow +\infty$ ;
(4) if BLAC- $r^*$  then
(5)    $\mathcal{R}(u) \leftarrow \text{RNG}(\mathcal{N}(u))$ ;
(6)    $g(u) = 1$ ;
(7) else
(8)    $\mathcal{R}(u) \leftarrow \mathcal{N}(u)$ ;
(9)    $g(u) = B(u)$ ;
(10) for  $i \in \mathcal{R}(u)$  do
(11)   if  $C_h < h(i) \times g(i) \vee (C_h = h(i) \times g(i) \wedge Id(i) < Id_{\min})$  then
(12)      $P(u) \leftarrow i$ ;
(13)      $C_h \leftarrow h(i) \times g(i)$ ;
(14)      $Id_{\min} \leftarrow Id(i)$ ;
(15) if  $h(u) \times g(u) > C_h \vee (h(u) \times g(u) = C_h \wedge Id(u) > Id_{\min})$  then
(16)   /*  $u$  becomes cluster-head */
(17)    $\mathcal{H}(u) \leftarrow u$ ;

```

ALGORITHM 2: BLAC algorithm runs at each node u .



(a) Cluster heads die faster than other nodes, creating holes in the network



(b) After a while, more nodes are dead.

FIGURE 3: Battery drain: case of density.

that elects itself as a *cluster head*. This same tree is then used to route data towards *cluster heads*.

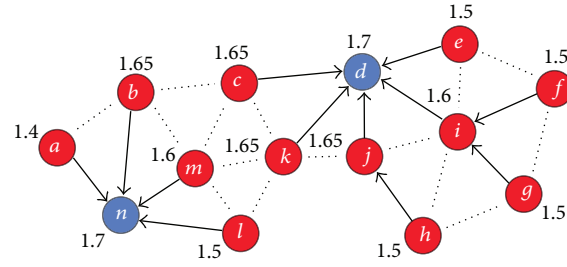
At the beginning, all nodes have the same battery level so the algorithms act like density based or DDR, respectively. Battery starts to drain differently on each node regarding the neighborhood of the node, its activity, and whether it is a *cluster head* or not. For instance, in Figure 1(c), $B(m)$ will decrease slower than $B(n)$, and then m will leave its *cluster head* role to n (Figure 4(a)).

Figure 3 shows that *density-based* clustering scheme will create holes in the network. First, nodes m and l will die because *cluster heads* consume more energy than regular nodes (Figure 3(a)). After some reorganization, a and d are the new *cluster heads* and they will finally die (Figure 3(b)),

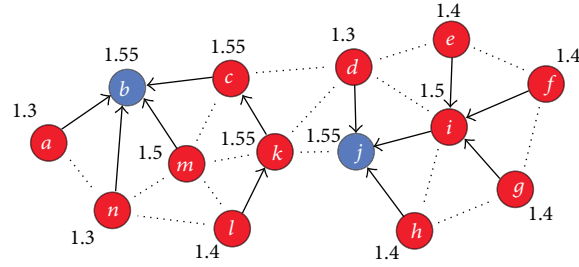
leaving holes in the network, and nodes, which have never been *cluster heads*, with a lot of energy.

Nevertheless, with BLAC algorithm (Figure 4) instead of dying, nodes m and l will leave their *cluster heads* roles before their energy level run to low. Energy consumption is thus balanced over nodes. In their turn, the new *cluster heads* n and d will leave their roles to b and l , balancing the energy consumption of the whole network.

5.2. BLAC- r^* . 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 main difference is that Algorithm 2 runs in two steps. Before computing its metric (degree or density),

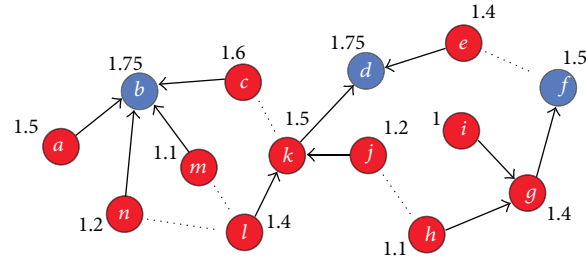


(a) After cluster heads drain their batteries faster than other nodes, cluster heads roles change

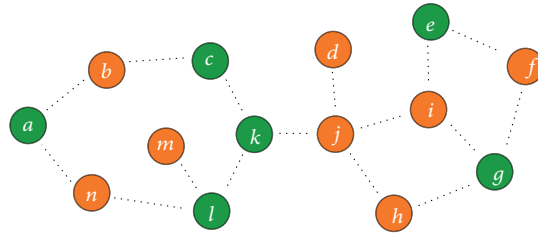


(b) Cluster heads change again after some time, and energy is balanced

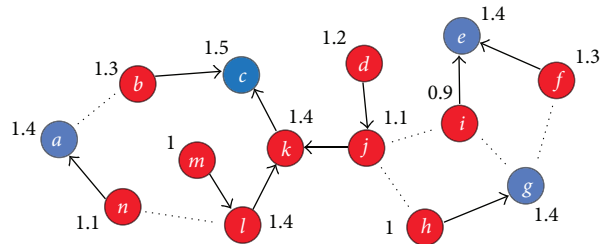
FIGURE 4: Battery drain: case of BLAC-b*.



(a) After cluster heads drain their batteries faster than other nodes, cluster heads roles change



(b) Cluster heads change again after some time, and energy is balanced



(c) Cluster heads change again after some time, and energy is balanced

FIGURE 5: Battery drain: case of BLAC-r*.



FIGURE 6: Network topologies, each red dot represent a node.

a relative neighborhood graph [20] is computed in order to keep only an interesting subset of nodes (lines 4-5 in Algorithm 2).

This allows memory storage saving and the use of less computing capacity for the clustering computation. The clustering algorithm then runs over the reduced graph without considering the battery level anymore since it has already been taken into account in the reduction step ($g(u) = 1$).

Figures 4 and 5 show the evolution of a clustering structure over RNG for both families of BLAC variants. We can see that first round elect nodes b , d and f as *cluster heads* (Figure 5(a)). After a while *cluster heads* nodes have consumed more energy than the others leading to a new construction of the graph. In Figures 5(b) and 5(c) we can see that links between a and b and between b and n for examples will now be avoided.

6. Performance Evaluation

To evaluate the performances of the different versions of BLAC, we perform some simulations under the WSNET simulator (<http://wsnet.gforge.inria.fr/>). We compare our

variants of BLAC to three close approaches taken from the literature. DDR [18] and density-based clustering [1] have been chosen because of their algorithmic proximity to BLAC. LEACH [17] has been chosen for its energy efficiency concern in clustering. In order to observe different behaviors for LEACH, we chose three representative values for the parameter P of the algorithm: $P = 5\%$, $P = 10\%$, and $P = 20\%$ (P is the average percentage of *cluster heads* in the network). Table 1 sums up the simulation parameters.

Nodes are placed using realistic maps from districts of Lille (urban environment) and in country regions (rural environment) in north of France to observe the behavior of algorithms at different densities. Nodes are placed on streets and next to building areas using OpenStreetMap data (<http://www.openstreetmap.org>) in order to get realistic topologies. Figure 6 shows the topologies we used to run the simulations; red dots represent nodes.

In order to use a realistic model for transmitting and receiving costs in both peer-to-peer and GPRS communications, we consider the *Texas Instruments CC2420* Zig-Bee chip (<http://focus.ti.com/lit/ds/symlink/cc2420.pdf>) and

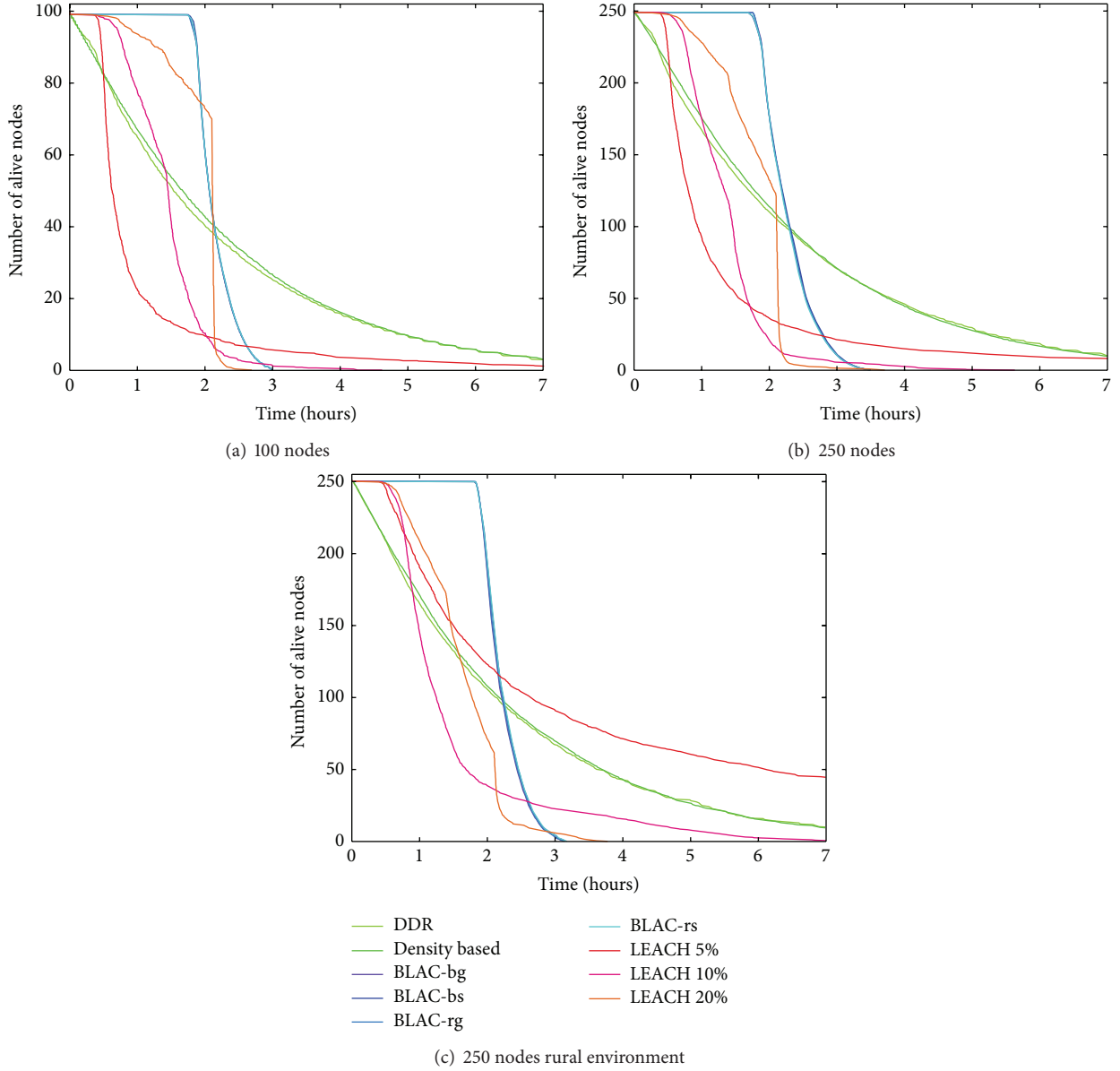


FIGURE 7: Network lifetime.

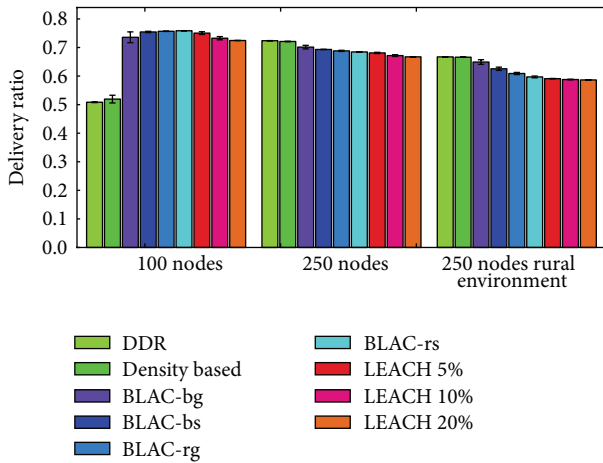


FIGURE 8: Data delivery ratio.

the LEON-G100 GSM/GPRS module from *u-blox* (<http://www.u-blox.com/en/wireless-modules/gsm-gprs-modules/leon-gsm-module-family.html>), respectively. The former one consumes 0.77 mW when idle, 35.46 mW for receiving (Rx), and 31.32 mW for transmitting (Tx). The latter one consumes 6.4 mW when idle and 1.25 W for receiving (Rx) or transmitting (Tx). A data traffic is also simulated. Each node generates 16 kbit of data periodically (every 5 seconds) and sends them to its parent. When a node has data coming from a child, it stores them until it needs to send its own data and then sends the aggregated data to its own parent. For instance, in Figure 1(c), after node a has generated its own 16 kbit of data, it sends them to b which will store them until it generates on its turn 16 kbit of data. Then, b sends 32 kbit to m who wait for its own data to send the aggregated 48 kbit to a base station using the GPRS radio chip.

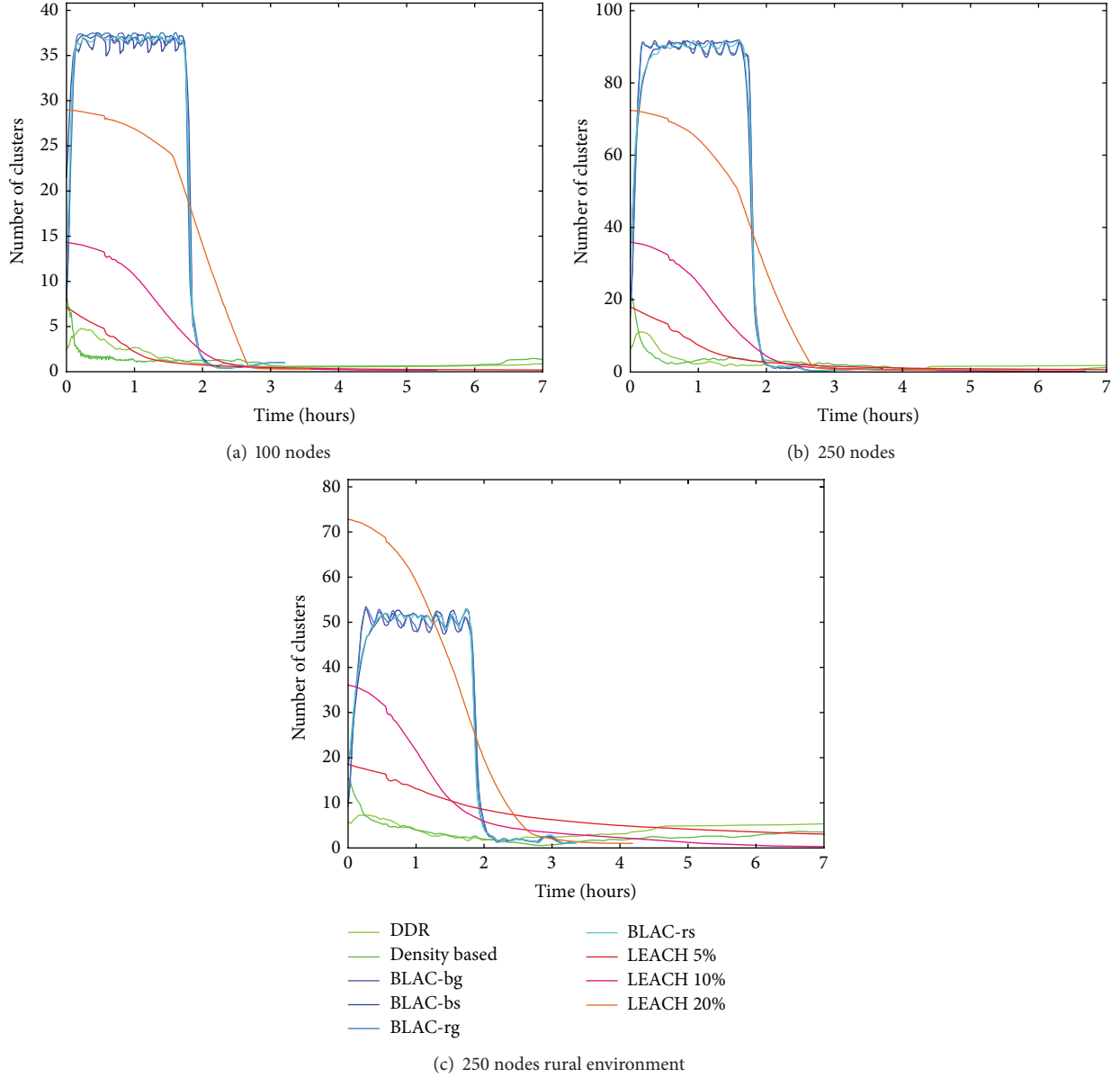


FIGURE 9: Number of clusters.

At bootstrap, each node earns the same energy level set to 32 mWh of energy, and when a node sends or receives a packet, the correct amount of energy (Table 1) is removed from the battery depending on the size of the message and data-rate of the chip set used.

6.1. Network Lifetime. Figure 7 illustrates the benefits of our algorithms regarding the network lifetime. It shows the number of nodes alive regarding time.

We can notice that in solutions that do not consider battery level such as density based and DDR, *cluster heads* die quickly. When *cluster heads* are dead, other nodes take their role till they die as well, and so on till there is no remaining alive node. LEACH performs much better, improving the lifetime with no dead node greatly compared to DDR

and density-based algorithms. Nevertheless, its performances greatly depend on the number of *cluster heads* that have been set up. The more *cluster heads*, the more nodes forwarding data (either as *cluster head* or to *cluster head*) and thus the shorter lifetime. BLAC-* maintains the maximum number of alive nodes up to 300% longer than other selected algorithms as expected (see Section 2). BLAC-bg and BLAC-rg present a slightly improvement compared to BLAC-bs and BLAC-rs, respectively, regarding the lifetime as they keep more nodes alive during the decreasing phase. This is due to the overall number of clusters built by every scheme. Indeed, BLAC-bg and BLAC-rg build less clusters than their corresponding variant and thus propose a different balance between the number and the size of *cluster heads* that perform a little bit better.

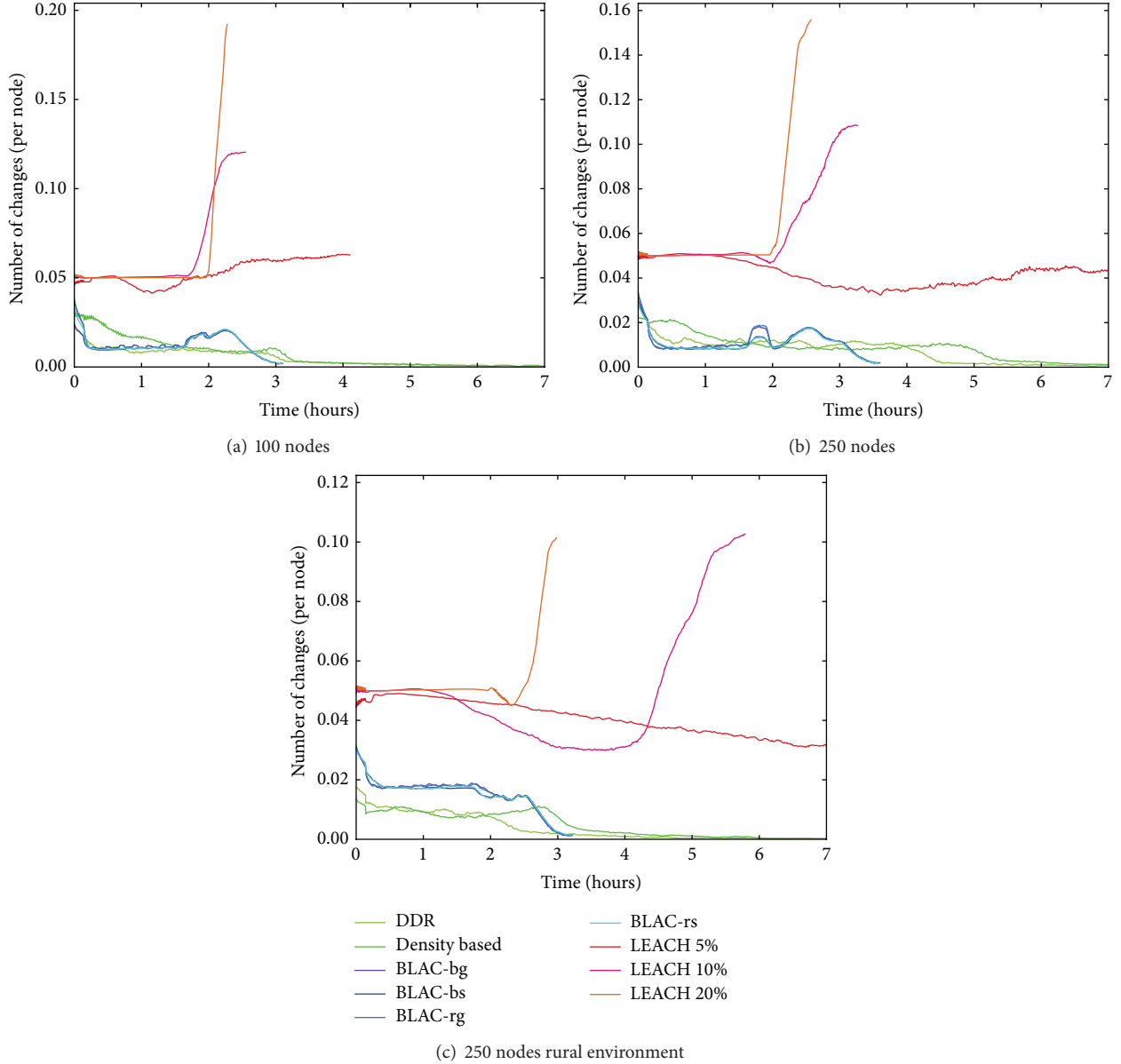


FIGURE 10: Network stability.

In addition, note that we did not consider in LEACH power consumption the costs relative to node synchronization.

6.2. Delivery Ratio. Figure 8 displays the delivery ratio of every algorithm with regards to time for 100 nodes and 250 nodes in urban environment and 250 nodes in rural environment. Delivery ratio is computed as the amount of data received divided by the amount of data generated on all nodes. We observe that the different variants of LEACH lose slightly more data than BLAC algorithms regardless of the number of *cluster heads*. Indeed, in LEACH, nodes without *cluster head* can not 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 time to transmit all data on time. In addition, since there are more *cluster heads*, nodes die more quickly (see Figure 7) since they send more data, and thus, more data are finally lost.

When the number of nodes increases in the network, every scheme loses more data except DDR and density based (more on this later). For BLAC variants, density based, and DDR, when some intermediary nodes die, the data they were holding are lost. For LEACH, there are more and more nodes in a cluster that need to share the medium, and thus, some of the data can not be sent on time.

We can observe that DDR and density based loose much more data than other schemes simulations with 100 nodes. As depicted in Figure 9, the number of *cluster heads* is very small and at the same time, the nodes die constantly (see Figure 7).

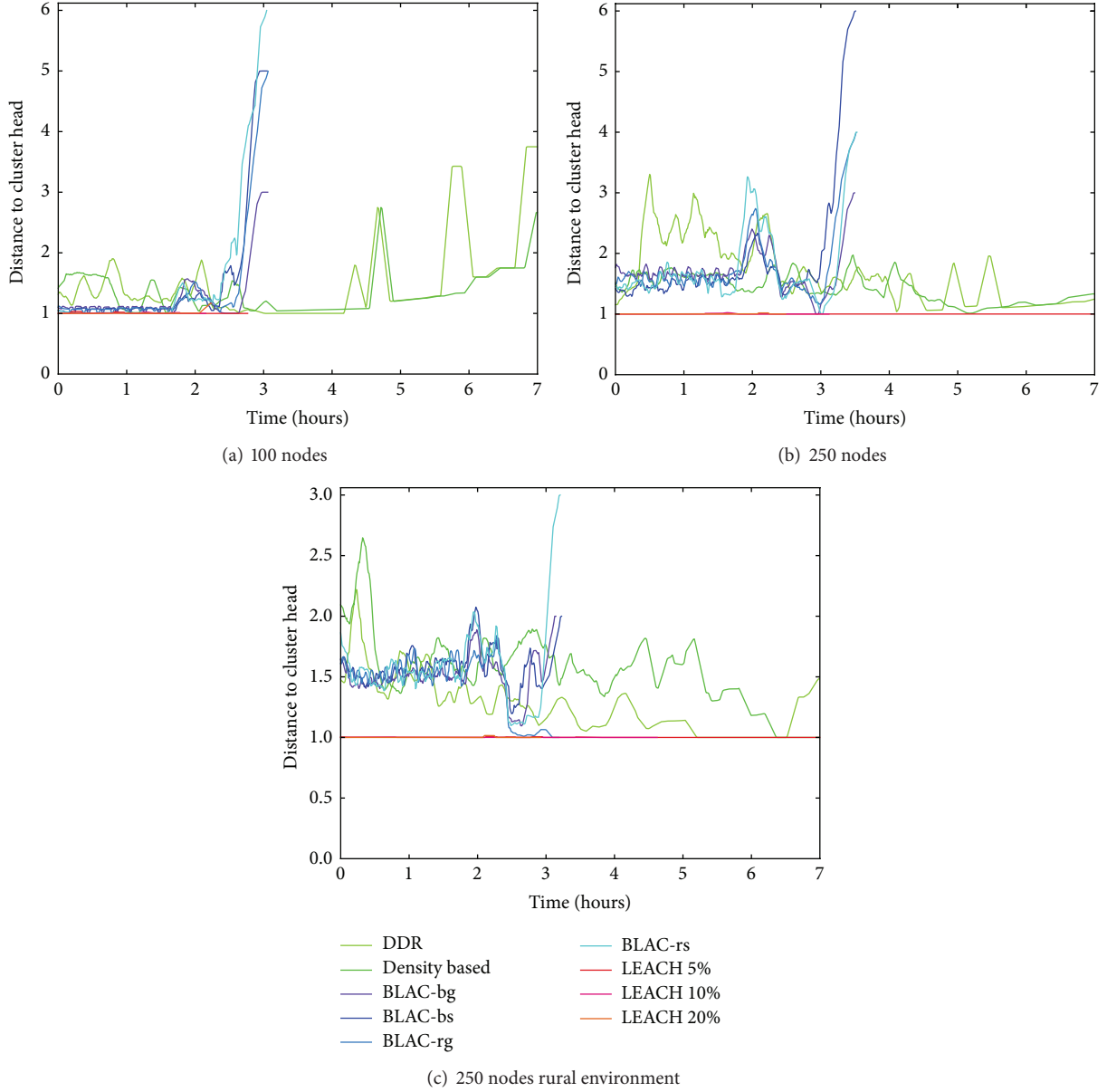


FIGURE 11: Average eccentricity.

As the number of *cluster heads* is low, the path to *cluster heads* is longer, combined to nodes disappearance; this increases the probability for data to be lost on the way as explained previously.

6.3. Network Properties. The number of clusters in the whole network along time is illustrated in Figure 9. Each curve stops when every node is dead. It shows that our clustering algorithm creates a relatively close number of clusters to DDR and density based. LEACH algorithms produce a number of clusters close to the parameter set. The number of clusters decreases with the number of alive nodes as expected. This highlights the need of an off-line tuning of the number of clusters, a complex task considering an application aimed

to be deployed on heterogeneous and dynamic network topologies.

Figure 10 compares the network stability between algorithms by depicting the number of parent changes for each round. Precisely, we measure the number of times a single node changes its parent regarding the previous round.

We can see that algorithms with a metric not based on remaining battery (DDR, density based) have more activity periodically. With such algorithms, a parent change occurs each time a parent node dies. When it happens, *degree* or *density* in the neighborhood of the dead node changes and restructuring appears.

For BLAC-* variants that integrate battery level (Section 5.1.1), since the battery level is changing over the time, restructuring can appear even if there are no dead

nodes. When the battery level of *cluster head* decreases, one of its neighbors can get a more interesting value of its metric then the neighbor becomes *cluster head* even if the original *cluster head* is not dead. This is also the way to have a more efficient task balancing between nodes. Again it shows that RNG variants are close to non-RNG variants of BLAC allowing the ability to choose the best one regarding application needs.

Figure 11 shows how far the nodes are from their *cluster head* in number of hops. We can observe a correlation between eccentricity and the number of clusters. Regarding Figure 9, we can see that the deeper the clusters are, the bigger they are. Deeper clusters result in more ZigBee communications to reach the *cluster head*, but they offer a better trade-off in terms of node lifetime.

7. Discussion

We have seen through the results analysis that the four versions of BLAC are close even if BLAC-rs seems to offer the best performances in terms of quality of service (delivery ratio versus network lifetime). The choice of the BLAC version must be made regarding the needs of the application concerning cluster size, stability, and quality of service. For instance, if nodes are mobile, BLAC-bs is the best choice as it offers a better stability against mobility as proven in [17]. If the size of clusters is not the main issue of the application, like in the case of static nodes, BLAC-rg is the best choice for data delivery ratio.

8. Conclusions

In this paper, we have introduced a new family of clustering techniques. Two variants combine the battery level with the degree or the density as a metric for cluster creation. The two others apply a battery level-based RNG construction. By completely integrating the battery level in the metric used to elect *cluster heads*, BLAC balances energy consumption over nodes and maximizes the network lifetime. The algorithm is distributed and modifications due to network dynamics are handled locally, allowing scalability. Results show that our proposition improves network lifetime with no dead nodes up to 300% which is useful for applications like ours. For future work, we will extend the comparison of BLAC with other algorithms like WCA [6]. Other energy models should also be used.

We will also run experimentations in real world using for instance SensLAB (<http://www.senslab.info/>) or Wisebed (<http://www.wisebed.eu/>) platforms. Another field of investigation is range adjustment.

Acknowledgments

This work is partially supported by CPER Nord-Pas-de-Calais/FEDER Campus Intelligence Ambiante and ANR ECOTECH BinThatThinks.

References

- [1] N. Mitton, B. Sericola, S. Tixeuil, E. Fleury, and I. Guérin Lassous, "Self-stabilization in self-organized wireless multihop networks," *Ad-Hoc and Sensor Wireless Networks*, vol. 11, no. 1-2, pp. 1-34, 2011.
- [2] Y. Yao and G. B. Giannakis, "Energy-efficient scheduling for wireless sensor networks," *IEEE Transactions on Communications*, vol. 53, no. 8, pp. 1333-1342, 2005.
- [3] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2826-2841, 2007.
- [4] D. J. Baker and A. Ephremides, "The architectural organization of a mobile radio network via a distributed algorithm," *IEEE Transactions on Communications*, vol. 29, no. 11, pp. 1694-1701, 1981.
- [5] M. Gerla and J. Tzu-Chieh Tsai, "Multicluster, mobile, multimedia radio network," *Wireless Networks*, vol. 1, no. 3, pp. 255-265, 1995.
- [6] M. Chatterjee, S. K. Das, and D. Turgut, "WCA: a weighted clustering algorithm for mobile ad hoc networks," *Cluster Computing*, vol. 5, 2002.
- [7] C. P. Low, C. Fang, J. M. Ng, and Y. H. Ang, "Efficient Load-Balanced Clustering Algorithms for wireless sensor networks," *Computer Communications*, vol. 31, no. 4, pp. 750-759, 2008.
- [8] W. Dali and H. A. Chan, "Clustering algorithm to balance and to reduce power consumptions for homogeneous sensor networks," in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '07)*, pp. 2723-2726, September 2007.
- [9] T. Shu, M. Krunz, and S. Vrudhula, "Power balanced coverage-time optimization for clustered wireless sensor networks," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '05)*, pp. 111-120, May 2005.
- [10] T. Kaur and J. Baek, "A strategic deployment and cluster-header selection for wireless sensor networks," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 4, pp. 1890-1897, 2009.
- [11] X. Min, S. Wei-ren, J. Chang-jiang, and Z. Ying, "Energy efficient clustering algorithm for maximizing lifetime of wireless sensor networks," *International Journal of Electronics and Communications*, vol. 64, no. 4, pp. 289-298, 2010.
- [12] N. Dimokas, D. Katsaros, and Y. Manolopoulos, "Energy-efficient distributed clustering in wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 70, no. 4, pp. 371-383, 2010.
- [13] O. Younis, M. Krunz, and S. Ramasubramanian, "Node clustering in wireless sensor networks: recent developments and deployment challenges," *IEEE Network*, vol. 20, no. 3, pp. 20-25, 2006.
- [14] T. Anker, D. Bickson, D. Dolev, and B. Hod, "Efficient clustering for improving network performance in wireless sensor networks," in *Wireless Sensor Networks*, R. Verdones, Ed., vol. 4913 of *Lecture Notes in Computer Science*, pp. 221-236, Springer, Berlin, Germany, 2008.
- [15] O. Buyanjargal and Y. Kwon, "AEEC-Adaptive and Energy Efficient Clustering algorithm for content based wireless sensor networks," in *Proceedings of the 2nd International Conference on Computer Science and Its Applications (CSA '09)*, December 2009.
- [16] K. Donghyun, W. Yiwei, L. Yingshu, Z. Feng, and D. Ding-Zhu, "Constructing minimum connected dominating sets with

- bounded diameters in wireless networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 2, pp. 147–157, 2009.
- [17] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless micro-sensor networks,” in *Hawaii International Conference on System Sciences (HICSS ’00)*, 2000.
- [18] N. Nikaein, H. Labiod, and C. Bonnet, “DDR-distributed dynamic routing algorithm for mobile ad hoc networks,” in *Workshop on Mobile and Ad Hoc Networking and Computing*, 2000.
- [19] N. Mitton, *Auto-organisation des réseaux sans fil multi-sauts à grande échelle [Ph.D. thesis]*, INSA, Lyon, France, 2006.
- [20] G. T. Toussaint, “The relative neighbourhood graph of a finite planar set,” *Pattern Recognition*, vol. 12, no. 4, pp. 261–268, 1980.

