Abstract—Due to the limited capacity of the nodes’ batteries, energy is a critical resource in wireless mobile networks (e.g., Wireless Sensor Networks: WSNs). Clustering is an effective approach to reduce total energy consumption; indeed, only cluster heads and gateways forward traffic. However, without proper distribution of energy consumption among the nodes, gains in total energy consumption do not necessarily translate to an increase of the network lifespan. In this paper, we propose novel heuristics that allow balancing energy consumption among the network nodes; these heuristics have been developed in combination with passive clustering due to its superior performance. Compared to existing approaches, the proposed heuristics (1) do not have stringent requirements, such as clock synchronization; and (2) do not generate extra control traffic to exchange energy information among the nodes. Furthermore, they can be easily used/adapted with other clustering protocols. The simulation results show that the heuristics considerably increase the network’s lifespan.

Keywords- cluster, energy, lifespan, mobile

I. INTRODUCTION

Flooding is frequently used in wireless ad hoc networks and wireless sensor networks (WSNs) to discover and/or to advertise network topology. It is oftentimes used as a method of information deliveries as well. Flooding generates unnecessary duplicates in dense networks and it speeds up the energy consumption of ad hoc/sensor nodes. Several solutions were proposed to reduce the overhead of the flooding. In general, these solutions improve the efficiency of flooding by selecting a subset of nodes (critical nodes), and by allowing only the subset of nodes participate in flooding. These approaches are also applicable for reducing the power consumption of sensor networks when the sensors’ radio can operate in a sleep mode and the non-critical nodes may shut down their radio when communication is not needed.

Topology based efficient flooding including MPR (multi-point-relay) [1] requires at least 2 hop neighbor information to apply set-cover heuristics. Probabilistic approaches do not require additional overheads such as topology information; unfortunately, these overhead-free approaches do not guarantee flooding coverage. Structure based efficient flooding includes cluster based efficient flooding [2]. In cluster based efficient flooding, the traffic is handled by a limited number of nodes (Cluster heads: CHs and gateways: GWs).

When we organize wireless mobile networks with limited number of critical nodes, the overall energy consumption as well as the number of unnecessary flooding duplicates are reduced; unfortunately, it also greatly increases the energy consumption of those critical nodes. The most important/critical nodes consume their batteries fast and stop functioning early, which may cause disconnections of the network.

There have been extensive efforts to reduce the consumption of energy and to improve the lifespan of wireless mobile networks (e.g., WSNs) in recent years. A common technique is to put some nodes in the sleep mode and put the others in the active mode for the sensing and communication tasks [3]. For example, the authors in [4] present a new energy saving method by dividing the nodes into disjoint sets so that each unit covers each target to be controlled. Energy can be saved with this method by activating only one set of nodes at a time.

The techniques used to reduce the total energy consumption of a network do not necessarily increase the lifespan of the network. Indeed, the lifespan of the network is defined [5] in WSNs according to three criteria: (1) The first node breaks down; (2) A fraction of the nodes breaks down; and (3) Loss of the coverage/connectivity. Figure 1 shows a sensor network where grey nodes run out of energy (e.g., battery capacity is 500mW); even though black nodes have still residual energy, they cannot communicate (i.e., black nodes in the left cannot communicate with black nodes in the right) since no feasible path exists. Thus, the key to increase the lifespan of the network is to balance the consumption of energy among the network nodes while still aiming at reducing the total energy consumption. Indeed, ideally all the network nodes should die at almost the same time.

Figure 1. An example of network disconnection

A number of recent cluster-based systems propose techniques to balance the energy consumption of the network nodes; these techniques consist mainly of rotating the CHs in a randomized fashion [6, 7]. Usually, these cluster-based systems combine active/sleep technique with randomized rotation; further their limitations (e.g., require signaling protocols to build the cluster structure) compared to passive clustering [8], they have stringent requirements to implement.
the active/sleep techniques; examples of these requirements include [6, 7]: (a) nodes have synchronized clocks; (b) nodes have global knowledge (e.g., energy of other nodes); (c) nodes have localization information. A good survey on these systems can be found in [3].

In this paper, we propose novel heuristics, to be used in conjunction with Passive Clustering (PC) [8], which reduce the overall energy consumption and extends the lifespan of the network by balancing the energy consumption among all the network nodes. PC using the proposed heuristics has no specific requirements to operate in terms of signaling, synchronization, global knowledge, etc. as in the existing approaches [3]. The basic idea behind our proposals is to enable the nodes to lock in sleep state (non-critical node state) when reaching a specific energy threshold so that other nodes with bigger amount of residual energy carry out the transmissions (i.e., become CHs or GWs).

The main contribution of the paper consists of the proposed heuristics that, when combined with PC, allows extending considerably the lifespan of the network. The key innovations of these heuristics, compared to existing approaches, is that they (1) do not have any stringent requirements (e.g., clock synchronization and global knowledge); and (2) do not generate extra control traffic to exchange energy information among nodes. Furthermore, the proposed heuristics can be easily adapted/used with other clustering protocols; in this paper, we selected PC since it has the best performance of all clustering protocols, especially, in highly mobile dense networks [8].

The remainder of this paper is organized as follows. Section 2 describes the efficient flooding mechanism with passive clustering. Section 3 presents an energy consumption balancing heuristic, called 1 Interval-based Load Balancing Heuristic (1-ILBH). Section 4 demonstrates the effectiveness 1-ILBH via simulations. Section 5 presents a generalization of 1-ILHB, called N-ILHB, and its performance evaluation via simulations. Finally, Section 6 concludes the paper.

II. PASSIVE CLUSTERING

The passive clustering (PC) [8] procedure is simple, easy to implement and most importantly, overhead free. Active clustering requires signalling protocols to exchange information among neighbouring nodes to select CHs and GWs; it generates considerable overhead (especially in the case of high mobility of nodes). In PC, a node that joins the network starts with the Initial state and each node determines its role among 4 states (Initial (Init), Cluster head (CH), Gateway (GW), and Ordinary node (ON)) by bookkeeping neighbors clustering status. The cluster status information (2 bits) of the immediate sender is piggybacked in the packet header of all outgoing packets at the MAC sub-layer. The cluster status information is used to determine the clustering role of the receiver by counting the number of CHs and GWs. The rule “first declaration wins” is used for CH election and the gateway selection heuristic is used to determine GWs.

Figure 2 illustrates an example of clustering. Only CHs and GWs broadcast data packets. ONs receive packets but do not take part in the re-broadcasting, which consequently limits the number of collisions, reduces the overheads, and avoids channel saturation.

A node cannot be a CH if it is already within the transmission range of another CH, i.e. it already received a packet from another CH. The distance between two CHs is at least 2 hops. There are four possible cluster states and 1 internal state, and are as follows: Initial, CH, ON, GW, CH_ready. At the beginning or when there are no network activities for a long time, all nodes are in the ‘initial’ state. This state does not change as long as a node does not receive a packet. When a node receives a packet and if the state of a sender is not CH, the receiver’s state switches to CH_ready. A node in CH_ready state will change to CH if it transmits successively outgoing packets before receiving a packet from a CH.

If the node receives a packet from a CH, it adds this CH’s information to its CH list and the node’s state is determined by gateway selection heuristics. All nodes not in ‘CH’ state maintain the list of CHs and GWs that they can overhear. The list is a soft-state, i.e., each entry of the list has a timer which keeps the most recent reception history and the entry is removed when the timer expires. The clustering state of a non-CH node is determined by the number of CH and the number of GW in its list. A gateway selection heuristics can be as simple as comparing the numbers of CHs and GWs. When the number of CHs is greater or equal to the number of GWs, the node becomes a GW. Otherwise, the node becomes an ON.

III. 1-INTERVAL-BASED LOAD BALANCING HEURISTICS (1-ILBH)

When we use original PC, CHs and GWs work more. These nodes drain their power quickly and die faster. To solve this problem, we propose heuristics which encourage the energy rich nodes become critical nodes in order to achieve even energy consumptions over a wide range of nodes.

To avoid the “early die” problem of critical nodes, CHs and GWs should have a preventive mechanism which avoids critical nodes from consuming all their limited battery power. The primary function of the mechanism is to force critical nodes which served a certain period to change to ‘ordinary’ state and to prevent those nodes from being elected as critical node right away. In addition to the clustering status policing, we switch the nodes from their active state to another "sleep" state where their energy consumption is minimal. The decision
making is completely distributed using the information from local battery; indeed, all the desirable characteristics of PC are preserved. A sensor node has a radio which can be configured according to three parameters: listening time (idle), time of reception (Rx) and time of sending (Tx) [8]. The total energy consumed by the node is the sum of energies: Energy-

\[ \text{Energy} = \text{Energy-idle} \times \text{Time-idle} + \text{Energy-Tx} \times \text{Time-Tx} + \text{Energy-Rx} \times \text{Time-Rx}. \]

We define two thresholds \( \alpha \times \text{battery} \) (battery represents the capacity of the node battery) and \( \beta \times \text{battery} \) according to the local information of node batteries (\( 0 < \alpha < \beta < 1 \)). When a node has not reached the first threshold (i.e., energy consumed so far is smaller than \( \alpha \times \text{battery} \)), it operates normally as in original PC. When a node reaches the first threshold, its state is forced to ‘ordinary’; furthermore, its radio is reconfigured by decreasing its listening time and time of reception to reduce energy consumption (in this case, we say that the node is in ‘sleep’ state). A node in ‘sleep’ state cannot change its state (i.e., cannot become CH or GW) until it reaches the second threshold (\( \beta \times \text{battery} \)); in this case, it returns back to the radio’s normal configuration and its state can change following the PC normal operation. Note that if we force the nodes which reach the \( \alpha \times \text{battery} \) threshold to go into the ‘ordinary’ state without reconfiguring the radio, the energy consumption will be balanced but it would trigger much more energy consumption.

```
/* change state and reconfigure radio card */
Begin
Variables
Packet; /* packet received by the node */

Input
    Packet; /* packet received by the node */

       

    battery=500; /* maximum capacity of the battery */

\[ \alpha, \beta; \] /* values between 0 et 1 */

K=10; /* reduction factor of listen and reception time */

Node; /* node receiving the packet ; it has two attributes : (1)

energyLevel : consumed energy ; and (2) state : the

node state (CH, GW, ordinary) */

listen_time_default_value;

Reception_time_default_value;

Inside PC;

if (battery-Node.energyLevel > \alpha \times \text{battery} &&
(battery-Node.energyLevel<\beta \times \text{battery}))
/* change state and reconfigure radio card */

Node.state= sleep;

Listen_time= listen_time_default_time/K;

Reception_time= reception_time_default_value /K;

Else

Execute PC;

Endif

Else

if (battery-Node.energyLevel >= \beta \times \text{battery})
    Node.state= initial;

/* reconfigure radio card */

Listen_time= Listen_time_default_value;

Endwhile
```

IV. 1-ILBH: Simulations

Simulations were carried out with GloMoSim [9]. The simulation parameters are shown in Table 1. Nodes are distributed randomly in the roaming area. Each node sends 100 packets with inter-arrival time of 0.5 second. The node’s maximum speed is 4m/s. AODV [10] is the routing protocol of our choice for this study; PC (original/with 1-ILBH) is used in the flooding phase of AODV (route-request flooding).

<table>
<thead>
<tr>
<th>Simulation parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roaming space</td>
<td>600X600 square meters</td>
</tr>
<tr>
<td>Radio propagation range</td>
<td>100 meters</td>
</tr>
<tr>
<td>Channel capacity</td>
<td>2Mbs/s</td>
</tr>
<tr>
<td>Duration of a single simulation</td>
<td>30 minutes</td>
</tr>
<tr>
<td>MAC protocol</td>
<td>802.11b</td>
</tr>
<tr>
<td>Traffic model</td>
<td>CBR (constant bit rate)</td>
</tr>
<tr>
<td>Radio power consumption (transmission/reception)</td>
<td>18 mW/second</td>
</tr>
<tr>
<td>Radio power consumption (idle)</td>
<td>5mW/second</td>
</tr>
<tr>
<td>Packet size</td>
<td>512 bytes</td>
</tr>
<tr>
<td>Node mobility model</td>
<td>Waypoint model</td>
</tr>
</tbody>
</table>

Table 2. Simulation Parameters

When a node is forced to go to ‘sleep’ state, its radio reception and listening time is divided by k=10; this value is determined after running extensive simulations with different values; the objective is to use a value that minimizes energy consumption.
consumption but at the same time avoid complete idleness (in this case, we will end up with a situation where all the network nodes are idle and no data delivery is possible).

PC superiority over existing clustering protocols has been clearly shown in [8]. Thus, our simulations will compare PC combined with 1-ILHB using the proposed heuristics to balance energy consumption against only original PC. This being said, the proposed heuristics can be easily adapted/used with other clustering protocols to improve balancing energy consumption among network nodes.

The objective of the simulations is twofold: (a) determine adequate values for $\alpha$ and $\beta$; and (b) show that the proposed heuristic allows balanced energy consumption among the network nodes and quantify its benefits in increasing the lifespan of the network. The values of $\alpha$ and $\beta$ should be carefully selected to avoid situations where all or a large subset of the network nodes are forced in prolonged ‘sleep’ state. In such situations, no data can be delivered because there are not enough (or none) CHs and GWs to forward data between sources and destinations.

Figure 3 shows the distribution of energy consumption of 100 nodes using original PC while supporting 10 sessions (10 sources and 10 destinations). The energy consumption of the network nodes varies between 350mW and 124mW. The maximum difference of energy levels is 226mW.

After extensive simulations using different values for $\alpha$ and $\beta$, we found that to avoid blocking situations $\beta$-$\alpha$ should be around 0.1; in this case, the delivery rate is similar to original PC which means that there is no session interruption. However, this requirement does not guarantee a balanced energy consumption among the network nodes; for example, with $\alpha =$0.6 and $\beta$=0.7, we found that the maximum difference of energy levels is around 150 mW. The best results, in terms of balancing the energy consumption, were obtained with $\alpha =$0.5 and $\beta$=0.6; Figure 6 shows that the energy consumption of each of the network nodes is about 250 mW with the maximum difference of energy levels equal to 34mW.

Similar blocking situation occurs for ($\alpha$=0.2, $\beta$=0.8), ($\alpha$=0.3 $\beta$=0.5), and ($\alpha$=0.4 $\beta$=0.7). The simulations show for (a) ($\alpha$=0.2, $\beta$=0.8): an average energy consumption of 100mW, a variation of 24mW, and a delivery ratio of 40%; (b) ($\alpha$=0.3 $\beta$=0.5): an average energy consumption of 150mW, a variation of 35mW, and a delivery ratio of 55%; and (c) ($\alpha$=0.4 $\beta$=0.7): an average energy consumption of 200mW, a variation of 31mW, and a delivery ratio of 84%. For lack of space, we do not show the energy consumption graphs for these values.

After extensive simulations using different values for $\alpha$ and $\beta$, we found that to avoid blocking situations $\beta$-$\alpha$ should be around 0.1; in this case, the delivery rate is similar to original PC which means that there is no session interruption. However, this requirement does not guarantee a balanced energy consumption among the network nodes; for example, with $\alpha =$0.6 and $\beta$=0.7, we found that the maximum difference of energy levels is around 150 mW. The best results, in terms of balancing the energy consumption, were obtained with $\alpha =$0.5 and $\beta$=0.6; Figure 6 shows that the energy consumption of each of the network nodes is about 250 mW with the maximum difference of energy levels equal to 34mW.

Figure 7 shows energy consumption of the network (i.e., sum of energy consumption of all the nodes) using different values for $\alpha$ and $\beta$ while varying the size of the network. In the cases where sessions were interrupted, the energy consumption is lower, as expected, than other cases. When using $\alpha =$ 0.5 and $\beta = 0.6$, Figure 10 shows a slightly higher energy consumption than original PC. This increase is due to the control packets generated when nodes are forced to ‘sleep’ state (and thus giving up their roles as CHs or GWs) in order to re-cluster the network using nodes with higher residual energy.
Now that we determined the ‘best’ values of $\alpha$ (0.5) and $\beta$ (0.6), let us compare PC using 1-ILHB with original PC. We run the simulations by increasing the traffic between sources and destinations (700 packets per source) to cause the nodes to run out of energy (i.e., become dead). The battery capacity is equal to 500 mW; thus, when a node consumes 500 mW it is considered dead.

Figure 7. Energy consumption Vs. Number of nodes

Figure 8 shows the number of dead nodes, over time, in the network using original PC and PC with 1-ILHB. At time $t = 715$ s, we record 20 dead nodes for PC; this number grows to 40 at $t = 815$ s, then to 50 at $t = 850$ s. During this time period, no node is dead when using PC with 1-ILHB; the first 20 deaths occur at $t = 920$ s and this number reaches 40 at $t = 945$ s. The delay between the first deaths of original PC and these of PC with 1-ILHB is equal to 205 s; therefore, the network with balanced PC functions normally and longer than in the case of original PC (see Figure 9); all the deaths using 1-ILHB occur within 80 seconds (585 seconds with original PC) which minimizes the probability of network disconnections. More specifically, in these simulations, the lifespan of the network is increased by about 28% when using 1-ILHB ($205/715=0.286$).

V. N-ILBH

In this section, we aim at generalizing the basic idea of 1-ILHB; we investigate the use of multiple intervals $[\alpha_1, \beta_1], \ldots, [\alpha_n, \beta_n]$ where $\alpha_1 < \beta_1 < \ldots < \alpha_n < \beta_n$, instead of a single interval, in balancing energy consumption among the network nodes. With N-ILHB, when a node’s consumption reaches $\alpha_i \cdot \text{battery}$ ($\forall 1 \leq i \leq n$), its state is forced to “sleep” and when its energy consumption reaches $\beta_i \cdot \text{battery}$, its state can change (to CH or GW) following the PC normal operation. If $n$ intervals are used, then a node will be forced to “sleep” state $n$ times.

We run several simulations with different numbers of intervals and with different values of $\alpha$ and $\beta$ (using the findings of the simulations in Section IV that $\beta - \alpha$ should be around 0.1) for each interval. We found that using more than 3 intervals does not help increasing the lifespan of the network; in the following, we present the simulation results for the following intervals (most interesting/useful after extensive simulations):

- **1-ILHB**: $\alpha = 0.5$, $\beta = 0.6$.
- **2-ILHB**: $\alpha_1 = 0.26$, $\beta_1 = 0.36$; $\alpha_2 = 0.54$, $\beta_2 = 0.64$.
- **3-ILHB**: $\alpha_1 = 0.2$, $\beta_1 = 0.3$; $\alpha_2 = 0.4$, $\beta_2 = 0.5$; $\alpha_3 = 0.6$, $\beta_3 = 0.7$.
- **4-ILHB**: $\alpha_1 = 0.17$, $\beta_1 = 0.27$; $\alpha_2 = 0.34$, $\beta_2 = 0.44$; $\alpha_3 = 0.5$, $\beta_3 = 0.6$; $\alpha_4 = 0.66$, $\beta_4 = 0.76$.

Figure 10 shows the variation of the number of dead nodes over time for original PC and PC with N-ILHB ($1 \leq N \leq 4$) while Figure 11 shows the corresponding delivery ratio over time.
With original PC, we record the first deaths at t=720. With PC using 1-ILBH, the first deaths occur at t=920; this represents a network lifespan increase of 28%. With PC using 2-ILBH, the first deaths occur at t=985; this represents a network lifespan increase of 36%. With PC using 3-ILBH, the first deaths occur at t=1400; this represents a network lifespan increase of 94% (almost doubling the network lifespan). With PC using 4-ILBH, the first deaths occur at t=1180; this represents a network lifespan increase of 63%. Thus, 3-ILHB provides the best increase. Intuitively, using more intervals allows better balancing of energy consumption and thus increases the lifespan of the network. Indeed, in the case of 1-ILHB, when the energy consumption of all the nodes exceeds $\alpha \cdot \text{battery}$ (let us assume this happens at time T1), the network behaves, to its death at T1d$\geq$T1, in the same way as using original PC causing a non-balanced energy consumption. Thus, if we use a second interval after T1, energy consumption will be balanced and the network lifespan will increase (dies at T2d$\geq$T1d); indeed, using the second interval is similar to using 1-ILHB after T1. However, using several intervals may cause the opposite since it increases the number control packets required to re-cluster and increases the probability of the occurrence of “blocking” situations. In our simulations, we found that 3 intervals are sufficient for best performance.

Figure 11 shows the variation of delivery rate over time; for the different schemes, as soon as the first deaths occur, the delivery rate starts decreasing; this decrease is the result of disconnections (causing interruptions of sessions) in the network caused by the dead nodes.

Figure 12 shows the increase in terms of control packets (RREQ of AODV), incurred when using the proposed heuristics. As expected, the number of control packets increases with N (N-ILHB); the increase is almost linear; in the case of 3-ILHB, 70% more control packets are generated compared with original PC. These results are expected since at the beginning of each interval $[\alpha, \beta]$, extra control packets are generated to re-cluster the network since a number of nodes are forced to give up their role as CHs or GWs; more specifically, new RREQ packets are generated to discover new paths (to recover/replace the failed paths). However, even with this increase, the lifetime of the network increases using N-ILHB since it enables balanced energy consumption among the nodes.

VI. CONCLUSION

In this paper we proposed a set of heuristics (N-ILHB) that increase the network lifespan, via energy consumption balancing among the network nodes, when clustering is used for data transmission; we developed N-ILHB in combination with passive clustering. The heuristics do not have stringent requirements in terms, for example, of synchronization and do not generate extra traffic to exchange energy information among nodes. Furthermore, the proposed heuristics can be easily adapted/used with other clustering protocols. We are currently investigating the development of an approach to determine analytically, if possible, the impact of the environment parameters (e.g., node distribution, roaming area) on the values of $\alpha$ and $\beta$. In other words, we would like, for a given configuration, to determine the values of $\alpha$ and $\beta$ and the number of intervals (N in N-ILHB) to use.

REFERENCES