Consiglio Nazionale delle Ricerche

Silence is Golden with High Probability: Maintaining a Connected Backbone in Wireless Sensor Networks

P. Santi, J. Simon

IIT TR-01/2003

Technical Report

Febbraio 2003

Istituto di Informatica e Telematica

Silence is Golden with High Probability: Maintaining a Connected Backbone in Wireless Sensor Networks

Paolo Santi* János Simon†

Abstract

Reducing node energy consumption to extend network lifetime is a vital requirement in wireless sensor networks. In this paper, we present and analyze the energy consumption of a class of cell-based energy conservation protocols. The goal of our protocols is to alternately turn off/on the transceivers of the nodes, while maintaining a connected backbone of active nodes. The protocols presented in this paper are shown to be optimal, in the sense that they extend the network lifetime by a factor which is proportional to the node density.

Non Student
Regular Presentation

1 Introduction

Wireless sensor networks (WSN for short) are composed of battery-operated microsensors, each of which is integrated in a single package with low-power signal processing, computation, and a wireless transceiver. Sensor nodes collect the data of interest (e.g., temperature, pressure, soil makeup, and so on), and transmit them, possibly compressed and/or aggregated with those of neighboring nodes, to the other nodes. This way, every node in the network acquires a global view of the monitored area, which can be accessed by the external user connected to the WSN. Examples of scenarios where WSN can be used are described in [6, 8, 9, 11].

Perhaps the most important cost in WSN is energy, since it determines battery use, and therefore the lifetime of the network. Since the major source of energy consumption in the sensor node is the wireless interface, considerable energy can be saved if the transceivers are completely shut down for a period of time. Of course, these sleeping times must be carefully scheduled, or network functionality could be compromised. A *cooperative strategy* is a distributed protocol that maintains a (minimal) connected backbone of active nodes, and turns into sleeping state the transceivers of non-backbone nodes. Periodically, the set of active nodes is changed to achieve a more uniform energy consumption in the network.

In this paper, we introduce and analyze the performance of simple cell-based cooperative strategies for WSN. In the cell-based approach, the network deployment area is partitioned into identical non-overlapping cells. Each cell has an active representative. We focus on the process that for each cell updates its active representative.

We found and analyzed two interesting strategies. The first assumes (possibly with unwarranted optimism) that all information about the nodes of the cell is available to the algorithm. The second is a probabilistic algorithm. If the number of nodes in each cell is known to the nodes within the cell, the algorithm is shown to elect a representative in constant expected time. The same behavior can be shown for the very plausible model in which the number is not known,

*Istituto di Informatica e Telematica del CNR, Via G. Moruzzi 1, 56124, Pisa – Italy. e-mail: paolo.santi@iit.cnr.it. Contact Author

†Dept. of Computer Science, University of Chicago. e-mail: simon@cs.uchicago.edu

but the nodes are assigned to cells identically at random. The algorithms are energy-optimal. We also find precise bounds for the running time of these algorithms.

Besides the potential usefulness of our algorithms, we believe that the ideas are interesting. They show how different amounts of 'global' knowledge (about the other nodes in a cell) can be exploited in a distributed algorithm. These techniques could be useful in other settings. One such example is given in Section 9.

The organization of the remainder of this paper is as follows: Sections 2 and 3 provide motivation and review of previous work. Section 4 presents our energy model, section 5 outlines our strategies. The deterministic algorithm is presented in Section 6, and the probabilistic ones in Section 7. The expected running time is derived in Section 8. Further problems, and a final discussion are in Section 9. Many of the details, and proofs can be found in the Appendix.

2 A WSN application scenario

A possible sensor networks application scenario in which the algorithms presented in this paper could be applied is the following.

A very large set of sensors is used to monitor a remote geographical region, for example to promptly detect fires in a vast forest. The (thermal) sensors are dispersed from a moving vehicle (e.g., an airplane or helicopter), hence their positions in the deployment region cannot be known in advance (but we have information about their probability distribution.) Once the sensors are deployed, their positions are mostly fixed.

The network should operate for a long period of time (e.g., the whole summer season), and should guarantee that, wherever the fire occurs, the alarm message generated by the sensor that detects the event quickly reaches the external supervisor, who is connected to some of the nodes in the network. Since the supervisor could be mobile (e.g., the supervisor is a ranger equipped with a PDA who wanders in the forest), the actual sensor nodes to which the supervisor is connected cannot be known in advance. This means that the alarm message should quickly propagate in the whole network, so that the supervisor can get the message independently of his physical location.

Since the batteries that power the sensors cannot be replaced once the unit is deployed, energy is a vital resource. With the current technology (see Section 4), the sensor node component that consumes most energy is the radio interface, even when the radio is in idle mode. Hence, considerable energy savings can be achieved if the radio interface of sensors can be turned off most of the time. In principle, a sensor does not need to have its radio on while it is sensing: it could turn the radio on only when an event (fire) is detected, and the alarm message must be sent. However, some of the sensor nodes must keep their radio on also when they are sensing, so that the alarm message generated somewhere in the network could quickly reach the supervisor. I.e., *a connected backbone of awake nodes (a spanning tree) must exist at any time.*

The class of energy saving protocols presented in this paper aims at extending network lifetime by alternately turning on/off sensor transceivers, in such a way that a connected backbone of awake nodes always exists. The emphasis will be on minimizing the *coordination cost*, i.e., the cost due to the message exchange needed to assign roles (sleep/awake) to sensors.

3 Related work

Several cooperative strategies aimed at extending network lifetime have been recently introduced in the literature. All of these strategies are related to some extent to the class of energy saving protocols presented in this paper.

In [12], the GAF protocol is presented. GAF exploits location information to alternately turn on/off node radios while not decreasing the network packet delivery rate (which is closely related to network connectivity). The protocol is based on a subdivision of the deployment region into an appropriate number of non-overlapping square cells, with the property that if at least one node is awake for every cell, then the network is connected. Nodes in the same cell elect a representative, which is left active, while the transceivers of non-representative nodes are shut down. Periodically, the representative election phase is repeated to balance power consumption and to deal with mobility.

The authors in [12] evaluate the energy savings achieved by GAF by means of simulation, since a formal analysis of the protocol energy consumption is difficult. The results of this simulation can be compared to the best possible energy savings achievable by any cell-based cooperative strategy (i.e., the energy savings that are achievable under the assumption that the coordination cost is 0), which have been analyzed in [1]. This comparison shows that GAF achieves good energy savings, but that it leaves some room for further improvement.

Another cooperative strategy is presented in [2]. The strategy, called SPAN, aims at reducing power consumption while preserving both the network capacity and connectivity. SPAN adaptively elects coordinators from all nodes in the network, which are left active, while non-coordinator nodes are shut down. The coordination algorithm is transparent to the routing protocol, and can be integrated into the IEEE 802.11 MAC layer. A major inconvenient displayed by SPAN is that, contrary to what is expected, the energy savings achieved by the protocol do not scale with node density. This is due to the fact that the coordination cost with SPAN tends to "explode" with node density, and counterbalances the potential savings achieved by the increased density.

The CPC strategy presented in [10] is based on the construction of a connected dominating set: the nodes in the dominating set schedule the sleeping times of the other units. The authors show through simulation that CPC achieves energy savings with respect to the case where no cooperative strategy is used. However, the analysis of how CPC performance scales with node density is lacking.

The concern with high node density is due to the fact, showed in [1], that in order to ensure that N cells will be all nonempty, if nodes are assigned to them at random, one needs to start with $O(N \log N)$ nodes. Thus, the expected density of nodes in a cell will be nonconstant in many networks of interest.

4 The energy model

We assume that a sensor can be in the following states:

1. **sleeping**: node can sense and compute, but radio is switched off;
2. **idle**: radio is on, but not used;
3. **receiving**: node is receiving a radio message;
4. **transmitting**: node is transmitting a radio message.

The well known point is that while currents are minuscule in a chip, even a small radio requires energy levels that are orders of magnitude greater than those inside a processor. Measurements on off-the-shelves wireless transceivers have shown sleep:idle:receiving:transmitting ratios of 0.025:1:1.2:1.7 for a AT&T 2Mb/s WaveLAN card [12], and of 0.06:1:1.2:1.8 for a Lucent WaveLAN card operating at 11 Mb/s [4]. It is also known that the energy cost of sensing is very small as compared to the energy consumption of the wireless transceiver, and is comparable to the cost of the radio interface in sleeping state [3].

In this paper, we assume that nodes are equipped with a GPS receiver, which is used to estimate node positions and the synchronize the network via the global clock. The GPS receiver can usually operate in different modes. In our setting, considering that nodes are assumed to be mostly stationary and that GPS is used mainly to (loosely) synchronize the nodes, GPS can be set to operate in the more extreme energy saving mode (8 seconds reporting). In this mode, the GPS receiver consumes about 0.033W, which is comparable to the 0.025W consumed by the wireless interface in sleep mode [12].

Given the discussion above, in this paper we use the following energy model:

- the energy consumption per time unit of a sensor with the radio on is modeled as a constant C, independently of the actual radio activity (idle, receiving, or sending);
- the energy consumption per time unit of a sensor with the radio off is modeled as a constant c, with $c \ll C$ (given the numbers above, we have $c \approx C/100$), independently of the actual sensor activity (sensing, or receiving the GPS clock).

5 Cell-based cooperative strategies

We assume that the some initial information (such as node ID, node position, and so on) is available to the nodes by an initialization protocol. Since nodes are not coordinated in the initialization phase, conflicts to access the wireless channel could arise. These can be solved using traditional backoff-based techniques. The details of the initialization protocol implementation are not discussed in this paper. Rather, we are concerned with minimizing the energy consumed for node coordination *after* the initialization phase.

We assume that all the nodes are equipped with batteries with the same characteristics, i.e., that all the nodes have the same amount of energy E_{init} available at the beginning of the network operational time, which for simplicity, we define as immediately after initialization. (This is likely not to be true: even if nodes are equipped with batteries with the same characteristics, since the energy cost for initialization of a node is expected to be proportional to the number of nodes in its neighborhood. So after initialization different processors in different cells will have not have the same charge. This does not affect our protocol, but would make the presentation more cumbersome.)

Our goal is to turn the radio off as long as possible, while maintaining a connected backbone of active nodes. To further simplify the analysis, we assume here that $c = 0$, i.e., the energy consumed by a node for sensing and receiving the GPS signal with the radio off is negligible with respect to the energy consumed when the radio is on.

Observe that the energy consumption of a node in general is influenced by external factors, such as environmental conditions and, more notably, by "alarm" events. Referring back to the fire detection scenario, sensors in the vicinity of a fire could increase their sensing rate in order to better monitor the event; when a set of thermal measures is above the alarm threshold (this is done to avoid false alarms), the sensor must issue an alarm message to the active node in its neighborhood. During this phase, energy consumption is no longer an issue: sensors must use all the energy that is needed to promptly propagate the alarm (e.g., the radio is turned on for a relatively long period of time). However, external factors are unpredictable by nature, and their influence on the nodes' energy consumption cannot be formally analyzed. For this reason, in the energy analysis of the protocols presented in this paper we will assume that there is no external influence on energy consumption during the network operational time (e.g., no "alarm" event occurs). Thus, we will analyze the intrinsic energy cost of maintaining a connected backbone of active nodes (*the cost of silence*).

With the assumptions above, if no cooperative strategy is used to alternately turn the node transceivers on/off, the network operates for $T_b = E_{init}/C$ time units after initialization. At time T_b, all the nodes die simultaneously, and the network is no longer operative. Time T_b is called the *baseline time*; the objective of our analysis will be to quantify the network lifetime extension yielded by our protocols with respect to this baseline time.

Similarly to the GAF protocol of [12], our protocols are based on a subdivision of the deployment region into non overlapping cells of the same size. More specifically, we make the following assumptions:

- n sensors are deployed (according to some distribution) in a square region of side l; i.e., the deployment region is of the form $R = [0, l]^2$, for some $l > 0$;
- all sensors have the same transmitting range r, with $r < l$; i.e., any two sensors can communicate directly if they are at distance at most r from each other;

– R is divided into $N = 8\frac{l^2}{r^2}$ non-overlapping square cells of equal side $\frac{r}{2\sqrt{2}}$.[1]

Given the subdivision above, the following properties are immediate:

P1. any two sensors in the same cell can communicate directly with each other; in other words, the subgraph induced by all the nodes belonging to the same cell is a clique;

P2. any two sensors in adjacent cells (horizontal, vertical, and diagonal adjacency) can communicate directly with each other.

A straightforward consequence of properties P1 and P2 is that leaving an active node for every cell is sufficient to provide a connected backbone of active nodes (under the assumption that the underlying communication graph is connected). Similarly to GAF, the goal of our protocols will be to elect a representative node for each cell. Once the representative is elected, non-representative nodes turn their radio off, and use energy only for sensing and receiving the GPS signal for synchronization purposes. Given our energy model, this is done at virtually no energy cost. After a given *sleep period*, non-active nodes awake (i.e., turn their radio on) and start a new negotiation phase, with the purpose of (possibly) electing a new representative.

The periodic re-election of a new representative (also called *leader* in the following) is motivated by the fact that energy consumption should be balanced over all nodes. If the leader is not re-elected periodically, it will consume much more energy than non-active nodes. Thus, nodes will die one after the other, at very distant times: when the first leader dies (approximately at time T_b), a new leader is elected; then the second leader dies, and so, until the last node in the cell is dead. With this scheme, network coverage, i.e., the ability of the network to "sense" any point of the deployment region R, could be compromised. Also, sleeping nodes would have to periodically check that their leader is still alive, which costs energy. A preferable solution would be to let the sensors consume energy more homogeneously, in such a way that they will all die approximately at the same time. This way, network coverage is not impaired: as long as the cell is operational (i.e., there is at least one alive node), almost all the nodes in the cell are still alive. This point will be further discussed below.

We will present re-election algorithms that will take advantage of different assumptions on the initial information available to the nodes. Before presenting the algorithms, we need to define the concept of network lifetime more formally. As discussed in [1], the definition of network lifetime depends on the application scenario considered. In this paper, we will use the concept of *cell lifetime*, which is defined as follows:

Definition 1 (Cell lifetime). *The lifetime of a cell is defined as the interval of time between the instant in which the cell becomes operative and the death of the last node in the cell.*

The time needed for network initialization is not considered in the analysis of lifetime, i.e., we assume that the cells are operative only when the initialization phase is over. To simplify the analysis, we further assume that all the cells become operative at the same time.

To conclude this section, we observe that nodes in adjacent cells could cause contention to access the wireless channel also after the initialization phase. To avoid this, we assume that an overall checkerboard pattern is used to allocate transmissions in adjacent cells at different time slots.

6 The FULL protocol

In this section, we assume that the following information is initially available to the nodes:

– its location (in particular, its cell ID);

– the ID of every other node in its cell;

[1]We use here the cell subdivision used in [1], which is slightly different from that used in [12] (see [1] for details). This choice is due to the fact that we will want to compare the energy consumption of our protocols to the best possible energy savings achievable by any cell-based strategy, which have been analyzed in [1].

– access to global time (for synchronization purposes).

To obtain such information, every node sends a message containing its node and cell ID (which can be easily obtained from the GPS signal) during the initialization phase. We recall that at this stage, since intra-cell synchronization is not yet established, conflicts to access the wireless channel must be solved using traditional backoff-based techniques.

Given the assumptions on the initial information available to each node, the following coordination protocol for leader re-election can be easily implemented.

ALGORITHM FULL (Algorithm for cell i)
Assumptions:
- assume there are n_i nodes in the cell (this is known by hypothesis);
- assume also that each node p knows its ordering in the set of all IDs of nodes belonging to i. For simplicity, we can equivalently assume that nodes in cell i have IDs ranging from 0 to $n_i - 1$, with $0 \leq p \leq n_i - 1$ denoting the ID of node p (the p-th in the ordering);
- assume the re-election process starts at time T_r. Each step below will take time T_s, assumed sufficient to turn the radio on, send a message to a node in the same cell, and to perform the simple calculation needed by the algorithm.

Protocol for generic Node p ($p \neq 0, n_i - 1$):
- at time $T_r + (p-1) \cdot T_s$:
  - turn radio on and receive message $M = (E_{max}, m)$ from node $p-1$
  - let E_p be the estimated energy remaining in the battery of node p at the end of the protocol execution
  - $E_{max} = maximum(E_{max}, E_p)$
  - if $E_{max} \geq E_p$, then $m \leftarrow p$
- at time $T_r + p \cdot T_s$:
  - send message $(Emax, m)$
  - turn radio off
- at time $T_r + n_i \cdot T_s$:
  - turn the radio on and receive message $M = (m, E_{max})$; m is the leader for the next sleep period, and E_{max} it the remaining energy of the leader
  - if $p \neq m$ then turn the radio off
  - END ; the negotiation phase ends: a new leader is elected

The protocol for nodes 0 and $n_i - 1$ is slightly different: node 0 simply sends message $(E_0, 0)$ at time T_r, and wakes up at time $T_r + n_i \cdot T_s$ to know the leader identity; node $n_i - 1$ ends the protocol after sending the message $M = (m, E_{max})$ at time $T_r + n_i \cdot T_s$ (in case it is not the leader, it turns its radio off before ending the protocol).

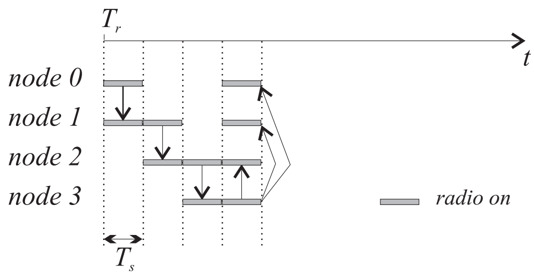

Figure 1: Time diagram of the FULL protocol execution when $n_i = 4$.

The time diagram of the protocol execution is depicted in Figure 1. It is immediate that each node other than the first and the last has its radio on for exactly $3T_s$ time, while node 0 and node $n_i - 1$ have their radio on for only $2T_s$ time. Assuming w.l.o.g. that T_s corresponds to one time unit, we have that the overall energy consumption of the FULL protocol is $(3(n_i - 2) + 4)C \in \Theta(n_i)$, i.e., the per node energy consumption is $\Theta(1)$. Thus, the energy spent by nodes to coordinate themselves is asymptotically optimal (note that every node must at least turn the radio on for one time unit to know the identity of the next leader, which require $\Omega(1)$ energy cost per node).

In contrast, a naive re-election algorithm would have all the nodes with the radio turned on in the whole negotiation phase, with a $\Theta(n_i)$ energy cost per node ($\Theta(n_i^2)$ overall).

When the protocol ends, all the nodes in the cell (except the leader) have their radio off. At some time, the nodes must turn their radio on again and start a new negotiation phase, possibly electing a new leader. The duration of the sleep period should be chosen in a such a way that the energy consumption is balanced in the cell, and nodes die approximately at the same time. In what follows, we will assume that the duration of the sleep period depends on the energy E_{max} still available in the battery of the leader. More specifically, we set $T_{sleep} = \lfloor \frac{E_{max}}{2C} \rfloor$; i.e., we set T_{sleep} to half of the expected leader lifetime[2].

A fuller discussion on the choice of the sleep period can be found in the Appendix.

Theorem 1. *Assume that cell i contains n_i nodes. If nodes coordinate their sleep periods using the* FULL *protocol, the cell lifetime is at least $\Theta(n_i T_b)$, where T_b is the baseline cell lifetime that corresponds to the situation where no cooperative strategy is used.*

Proof. The proof of the Theorem is reported in the Appendix. □

A straightforward consequence of Theorem 1 is that the FULL protocol (when it is used in every cell) consumes the minimal energy required to maintain a connected backbone of active nodes, i.e., it is optimal from the point of view of energy consumption (see also [1]). Another consequence of Theorem 1 is that a sensor network that uses FULL to coordinate node activity takes full advantage of a high node density: the lifetime of a cell is extended of a factor asymptotically equal to the number of nodes in the cell.

The FULL protocol described above works only if all the n_i nodes that were in the cell at the beginning of the cell operational time are still alive. Otherwise, the message containing the current value of E_{max} is lost when it is supposed to be received by a node which is actually dead. However, a simple modification of the protocol is sufficient to deal with this situation. A n_i-bits mask is included in the message propagated during the negotiation phase. The purpose of this mask is to keep trace of the nodes that are still alive. Initially, all the bits in the mask are set to 1 (the number of alive nodes is n_i). When node p receives the message, it compares the energy E_p remaining in its battery with a "minimum threshold" value E_{min}. If $E_p \leq E_{min}$, the node sets the p-th bit of the mask to 0, and propagate the message to the next node. When the last node in the ordering sends the message containing the leader for the next step, it includes in the message also the bit mask. This way, all the nodes in the cells know the number of nodes still alive, and they can turn their radio on at the appropriate instant of time (which depends on the position of the node ID in the ordering of the nodes which are still alive).

The value of E_{min} must be chosen carefully. In principle, E_{min} could be set to 3, which is the minimum value that guarantees that the node can remain alive for the entire next negotiation phase[3]. However, a more conservative value for E_{min} that accounts for the possible occurrence of external factors (that induce an increased energy consumption in the node) is preferable in practice.

7 The RANDOM protocol

In this section, we present and analyze a simple randomized coordination protocol.

[2]We recall that we are assuming that the energy needed to sense and to receive the GPS signal is negligible with respect to that needed to power the radio. Further, we are assuming that no external factor is influencing energy consumption. Recalling that the leader has its radio on during the sleep period, with these hypotheses the expected leader lifetime is $\frac{E_{max}}{C}$.

[3]Observe that a node with $E_p = E_{min}$ is elected as the leader for the next sleep period only if the energy remaining in the other nodes still alive is at most E_{min}. If this is the case, the cell is at the end of its operational time, and no further negotiation phase will take place.

Algorithm Random (Algorithm for cell i)
Assumptions:
- each node knows its cell ID (this information can be easily obtained via GPS);
- each node has access to global time;
- nodes can detect conflicts on the wireless channel;
- the re-election process starts at time T_r. Step 2 below will take time T_s, assumed sufficient to flip a coin, turn the radio on, possibly send a message to the nodes in the same cell, and detect conflicts on the wireless channel.

Protocol for generic Node j:
- at time T_r:
  1. END = **False**
  2. repeat until END=**True**
     2.1 flip a coin, with probability of success p
     2.2 if SUCCESS, send message (E_j, j) (E_j is the estimated energy remaining in the battery); otherwise, set the radio in receiving mode;
     2.3 if nobody sent a message or COLLISION, goto step 2
     2.4 END=**True**
     2.5 if not SUCCESS, turn the radio off (if node j is not the leader, it turns its radio off; since exactly one node $k \neq j$ sent a message, node j knows the current leader k and its remaining energy E_k)
     2.6 goto step 2

Let $\sharp S$ be the random variable denoting the number of times that step 2 must be repeated before a unique leader is elected. Clearly, the energy consumption of the Random protocol depends on $\sharp S$, which, in turn, depends on the probability p of obtaining a success in the coin flip. It is easy to see that $\sharp S$ has geometric distribution of parameter $(1-q)$, where $q = 1 - n_i p(1-p)^{(n_i-1)}$ and n_i is the number of nodes in cell i. In fact, step 2 is repeated if the number of successes in the n_i independent coin flip experiments is 0 or at least 2, which occurs with probability q. The value of $Prob(\sharp S \leq k) = 1 - q^k$ converges to 1 as k increases for any $q < 1$, and the velocity of convergence to 1 is higher for smaller value of q. Thus, the minimum value for q is desirable. This value is obtained setting $p = \frac{1}{n_i}$, for which we get $q = 1 - (1 - \frac{1}{n_i})^{(n_i-1)}$. For large enough values of n_i, we have $q \approx 1 - \frac{1}{e}$. Thus, with the best possible setting for p, RANDOM is expected to execute $E[\sharp S] = \frac{1}{1-q} \approx e$ times step 2 before a unique leader is elected. If we assume w.l.o.g. that T_s equals the time unit, and observing that all the nodes in the cell i have their radio on during RANDOM execution, we have that the average energy cost of the protocol is $en_iC \in \Theta(n_i)$, with a per node average energy consumption of $eC \in \Theta(1)$, which is optimal. Note that the deterministic Full protocol is energy optimal in the worst case, while Random is energy optimal on the average.

The analysis of the average case behavior of Random in the entire network (under the hypothesis that the value of p in every cell is set to the actual number of nodes in the cell) is quite straightforward. Recalling that N denotes the total number of cells, we have that, on the average, a fraction of $\frac{1}{e}$ cells select the leader in the first step, a fraction of $\frac{1}{e}$ of the remaining $N\left(1 - \left(\frac{1}{e}\right)\right)$ cells select the leader in the second step, and so on. Thus, the average number of cells that have selected the leader in at most k steps is $N_k = N\left(1 - \left(1 - \frac{1}{e}\right)^k\right)$. Note that N_k converges to N quite quickly; for instance, when $k = 10$, we have $N_k = 0.9898N$. We can conclude that, on the average, a number of steps in the order of ten is sufficient to elect the leader in every cell.

Note that setting p to the optimal value $\frac{1}{n_i}$ implies that the number of nodes in every cell must be known, which is the same assumption as in the deterministic Full protocol. Assume now that we only have *probabilistic* information about node placement. In particular, assume that the n nodes are distributed uniformly and independently at random in the deployment region R. In this case, we know that the expected number of nodes in a cell is $\alpha = \frac{n}{N} = \frac{nr^2}{8l^2}$. If we set $p = \frac{1}{\alpha}$, we have the following result.

Theorem 2. *Assume that n nodes are distributed uniformly and independently at random in $R = [0,l]^2$, and set $p = \frac{1}{\alpha}$ in the* Random *protocol. If n, r and l are chosen in such a way that $r^2 n = kl^2 \ln l$ for some constant $k \geq 16$, then:*

a. the communication graph obtained when all the nodes are awake is connected w.h.p.

b. $\lim_{n,l\to\infty} E[\sharp S] = e$.

Proof. The proof of the Theorem is reported in the Appendix. □

Theorem 2 is quite interesting, since it states that, also in the case in which only probabilistic information about the number of nodes in a cell is known, the average energy consumption of Random is optimal.

8 Time Requirements.

While our main goal is to analyze battery life, it is interesting to analyze more precisely the time spent in the coordination routines. Assume that the number n of deployed sensors is sufficient to ensure that the communication graph obtained when all the nodes are awake is connected w.h.p.. It is clear that the time spent by algorithm FULL is $\Theta(\log N)$, since in cell i we have n_i rounds of $O(1)$ messages, and with overwhelming probability $n_i = O(\log N)$.

The analysis for algorithm Random is more involved. If we assume that in each cell n_i is known, and the probability p is set to $\frac{1}{n_i}$, then it is not hard to see that with high probability there will be a cell where the protocol will finish in $\Theta(\log N)$ steps (see Appendix.) This may be surprising, as the energy consumption is $O(1)$: the difference is due to bounding the expected value of the time (which is a constant) and bounds on the tail of the distribution, needed to analyze the time spent (which is $\Theta(\log N)$).

The same is true even if n_i is not known. More precisely:

Theorem 3. *Under the hypothesis of Theorem 2 there are constants a and b such that*

1. $\lim_{n,l\to\infty} Pr[$*all cells have a leader by time* $a \log n] = 0$.
2. $\lim_{n,l\to\infty} Pr[$*all cells have a leader by time* $b \log n] = 1$.

The proof is deferred to the full version. A sketch is found in the Appendix.

9 Discussion and further work

The protocols presented in this paper are designed to work in WSN where the node density is quite high, and sufficient to ensure the connectivity of the underlying communication graph. We have demonstrated that our protocols take full advantage of the node density, increasing the cell lifetime of a factor which is proportional to the number of nodes in the cell. However, when the node density is particularly high, the energy spent during the initialization phase (which is needed to know the number of nodes in every cell) could become an issue. In this setting, the adoption of the Random protocol might be preferable. We remark that, when the node density is very high, the performance of Random with $p = \frac{1}{\alpha}$ is very likely to be close to that of the average case.

As we claimed, these techniques have wider applicability. For example, in the case of a sensor network with high density, it is likely that an event will be reported by a large number of sensors. This can potentially yield communication delays due to contention, paradoxically making the system less reliable. Our techniques may be used to regulate access. For example, if processors have a good estimate of the number of other sensors s that are observing the event, it may choose, in a first phase, to communicate it with probability $\Theta(\frac{1}{s})$. The analysis of our protocol (both energy consumption and time) applies to this scenario.

Another issue for our election algorithm is the choice of the time for the next negotiation. If the only criterion is to maximize battery life – and this paper is an attempt to quantify how much one can gain by these techniques – then our algorithms are optimal. In practice, one might not have good estimates of actual battery use, so one might want to run the protocols much more frequently than $\frac{E_{max}}{C}$. It makes sense to make each of these as efficient as possible, so our efficient algorithms are useful, but a quantitative estimate depends on the specifics of the protocol.

The choice of the value of p in the RANDOM protocol leaves space for several optimizations. For example, the value of p could account for the amount of energy still available at the node; this way, nodes with more energy are more likely to be elected as the next leader, and energy consumption is likely to be well balanced. Another possibility would be to change the value of p depending on the duration of the previous negotiation phase. We make this argument clearer. If p would be set to the optimal value, the expected duration of the negotiation phase is e time units. If the negotiation phase lasts much more than this, it is likely that p is far from the optimal value. For instance, if the previous negotiation phase required many steps because nobody transmitted, this could indicate the the value of p is too low. Conversely, if the excessive duration of the negotiation phase was due to conflicts on the wireless channel, it is likely that the value of p is too high. Since the information about the duration and the collision/absence of transmission during negotiation is available to all the nodes in the cell, a common policy for modifying the value of p can be easily implemented. The definition of such a policy is matter of ongoing work.

Another major direction of further research is the generalization of our class of protocols to the asynchronous setting.

References

[1] D.M. Blough, P. Santi, "Investigating Upper Bounds on Network Lifetime Extension for Cell-Based Energy Conservation Techniques in Stationary Ad Hoc Networks", *Proc. ACM Mobicom 02*, pp. 183–192, 2002.

[2] B. Chen, K. Jamieson, H. Balakrishnan, R. Morris, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintainance in Ad Hoc Wireless Networks", *Proc. ACM Mobicom 01*, pp. 85–96, 2001.

[3] S. Coleri, M. Ergen, J. Koo, "Lifetime Analysis of a Sensor Network with Hybrid Automata Modelling", *Proc. ACM WSNA 02*, Atlanta, pp. 98–104, 2002.

[4] L.M. Feeney, M. Nilson, "Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment", *Proc. IEEE INFOCOM 2001*, pp. 1548–1557, 2001.

[5] V.F. Kolchin, B.A. Sevast'yanov, V.P. Chistyakov, *Random Allocations*, V.H. Winston and Sons, Washington D.C., 1978.

[6] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, J. Anderson, "Wireless Sensor Networks for Habitat Monitoring", *Proc. ACM WSNA 02*, pp. 88–97, 2002.

[7] P. Santi, D.M. Blough, "The Critical Transmitting Range for Connectivity in Sparse Wireless Ad Hoc Networks", *to appear on IEEE Trans. on Mobile Computing*.

[8] L. Schwiebert, S.K.S. Gupta, J. Weinmann, "Research Challenges in Wireless Networks of Biomedical Sensors", *Proc. ACM Mobicom 01*, pp. 151–165, 2001.

[9] M.B. Srivastava, R. Muntz, M. Potkonjak, "Smart Kindergarten: Sensor-based Wireless Networks for Smart Developmental Problem-solving Environments", *Proc. ACM Mobicom 01*, pp. 132–138, 2001.

[10] C. Srisathapornphat, C. Shen, "Coordinated Power Conservation for Ad Hoc Networks", *Proc. IEEE ICC 2002*, pp. 3330–3335, 2002.

[11] D.C. Steere, A. Baptista, D. McNamee, C. Pu, J. Walpole, "Research Challenges in Environmental Observation and Forecasting Systems", *Proc. ACM Mobicom 00*, pp. 292–299, 2000.

[12] Y. Xu, J. Heidemann, D. Estrin, "Geography-Informed Energy Conservation for Ad Hoc Routing", *Proc. ACM Mobicom 01*, pp. 70–84, 2001.

[13] F. Xue, P.R. Kumar, "The Number of Neighbors Needed for Connectivity of Wireless Networks", *internet draft*, available at `http://decision.csl.uiuc.edu/J~prkumar/postscript_files.html#Wireless%20Networks`.

A Appendix

A.1 Sleep periods in the FULL protocol

The example shown in Figure 2 motivates our choice.

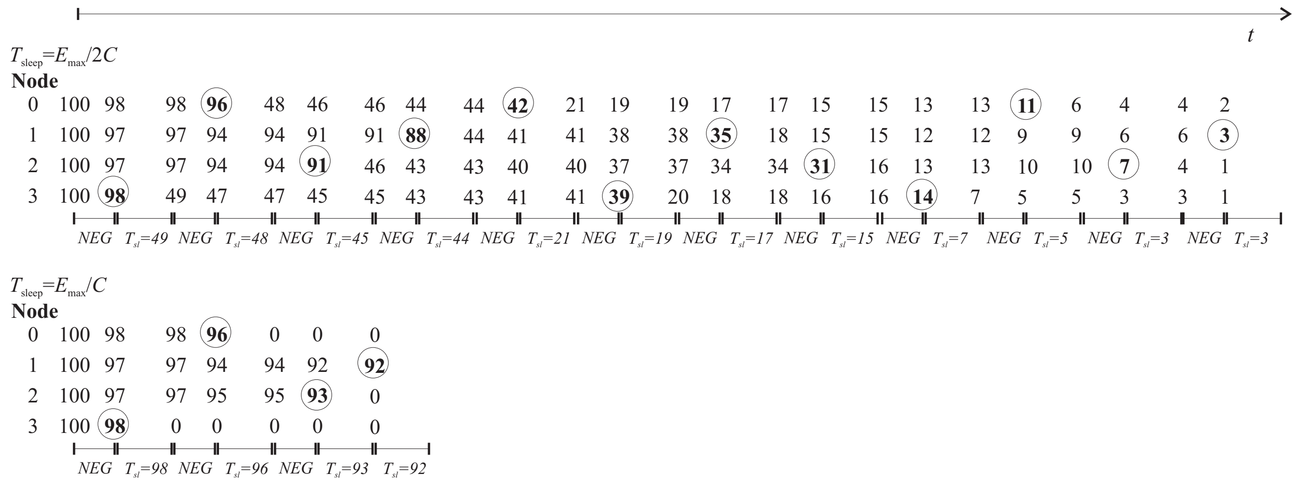

Figure 2: Time diagram of the Full protocol execution for different choices of the sleep period. Every node has 100 units of energy available after initialization. The leader is highlighted with a circle.

In the figure, we report the time diagram of the Full protocol execution for different choices of the sleep period. If the sleep period is set to E_{max}/C, only three negotiation phases (each during 4 units of time) are needed. Node 3 dies at time 102, node 0 at time 202, node 2 at time 299, and node 1 at time 391, causing the cell death. Thus, the cell lifetime is extended from the baseline time of 100 to 391, with an increment of 391%. This is the 97.75% of the best possible cell lifetime achievable, which is 400. However, the energy consumption is very unbalanced, and this is detrimental for coverage. In fact, the average number of alive nodes during the cell operational time is 2.54, which is only 63.5% of the best possible coverage (obtained when all the nodes in the cell are alive). Further, the average lifetime per node is only 248.5 units of time.

The situation is very different when we set T_{sleep} to $\lfloor \frac{E_{max}}{2C} \rfloor$. In this case, the energy spent for cooperation is higher (12 negotiation phases are needed), and the overall cell lifetime is decreased with respect to the previous case (324 instead of 391, corresponding to 81% of the best possible lifetime). However, energy consumption is very well balanced among nodes: the average number of alive nodes is 3.97 (99.25% of the best possible coverage), and the average lifetime per node is 321.75. Thus, we have traded a relatively small decrease in the cell lifetime with a significantly larger increase in the average lifetime per node, which induce a much better cell coverage. We remark that also in this scenario the energy spent during the 12 negotiation phases is limited (only 30% of the overall energy).

A.2 Proof of Theorem 1.

Let us consider a given node p in the cell. We recall that a node spends $\Theta(1)$ energy during each negotiation phase, while during the sleep period it consumes $E_{max}/2$ energy if it is the leader, and 0 energy otherwise. Under the hypothesis that $E_{init} - 3n_i > \frac{E_{init}}{2}$ (which holds in many realistic scenarios, where $E_{init} \gg n_i$), p will be elected as leader before any other node is elected leader for the second time (see Figure 2). In fact, when a node is first elected as leader it consumes at least $E_{init}/2$ energy, and the condition above ensures that, at most during the

n_i-th negotiation phase, the energy remaining in node p (which is at least $E_{init} - 3n_i$) is greater than the energy remaining in every other node (which is at most $E_{init}/2$). Assuming w.l.o.g. of generality that p is the last node to be elected as leader for the first time, and observing that each negotiation phase lasts exactly n_i time units, we have that the cell lifetime is at least $n_i\left(n_i + \frac{T_b - 3n_i}{2}\right) \in \Theta(n_i T_b)$.

A.3 Proof of Theorem 2.

Part $a.$ of the Theorem is proved in [7] (Th. 8). To prove part $b.$, let $Max(n, N)$ and $Min(n, N)$ be the random variables denoting the maximum and minimum number of nodes in a cell when n nodes are distributed into N cells, respectively. Setting $n = \frac{kl^2 \ln l}{r^2}$ for some constant $k \geq 16$, and $N = 8\frac{l^2}{r^2}$, we have that $Max(n, N) \in \Theta(\alpha)$ and $Min(n, N) \in \Theta(\alpha)$ [1, 5].[4] Assume w.l.o.g. that RANDOM is executed in cell i with minimum occupancy (i.e., $n_i = Min(n, N)$), and set $p = \frac{1}{\alpha}$. The probability $q(n, N)$ of repeating step 2 of the protocol when n nodes are distributed uniformly into N cells is:

$$q(n, N) = 1 - \frac{Min(n, N)}{\alpha}\left(1 - \frac{1}{\alpha}\right)^{Min(n,N)-1}$$

We have:

$$\lim_{n,N\to\infty} q(n, N) = 1 - \lim_{n,N\to\infty} \frac{Min(n, N)}{\alpha}\left(1 - \frac{1}{\alpha}\right)^{Min(n,N)-1}$$

Taking the logarithm, and using the first term of the Taylor expansion of $\ln(1 - x)$ for $x \to 0$, we obtain:

$$\lim_{n,N\to\infty} \ln \frac{Min(n, N)}{\alpha} - \frac{1}{\alpha}(Min(n, N) - 1) \tag{1}$$

Rewriting the first term of (1) as $\ln\left(1 - \frac{\alpha - Min(n,N)}{\alpha}\right)$, and observing that $Min(n, N) \in \Theta(\alpha)$ implies that $\lim_{n,N\to\infty} \frac{\alpha - Min(n,N)}{\alpha} = 0$, we can use again the Taylor expansion, obtaining:

$$(1) = \lim_{n,N\to\infty} -\frac{\alpha - Min(n, N)}{\alpha} - \frac{Min(n, N) - 1}{\alpha} = -1\ .$$

It follows that $\lim_{n,N\to\infty} q(n, N) = \lim_{n,l\to\infty} q(n, N) = 1 - \frac{1}{e}$, and the theorem follows by observing that $E[\sharp S] = \frac{1}{1-q}$.

A.4 Time bounds

We sketch the proof of our time estimate for the case n_i is known. We have N cells. In each we run a process that has probability $\frac{1}{e}$ of success (a representative for the cell is chosen) The probability that a leader has not been elected in t steps is $q_t = (1 - \frac{1}{e})^t$. The probability that all N cells have leaders is $(1 - q_t)^N$. Fix some parameter ϵ (we want to succeed with probability $1 - \epsilon$). So we need $(1 - q_t)^N = (1 - (1 - \frac{1}{e})^t)^N > 1 - \epsilon$ which holds if t is $O \log N$)

The proof of Theorem 3 is similar. We use the fact that if we distribute $N \log N$ balls into N bins at random, there are constants a and b such that the probability that there exists a bin with a number of balls outside the interval $[a \log N, b \log N]$ tends to 0 as N goes to infinity [13]. Furthermore, the expected time bound for algorithm RANDOM holds also for non-optimal choice of the parameter α.

[4] A similar and somewhat stronger property has been proved in [13].