On the Monitoring Period for Fault-Tolerant Sensor Networks[*]

Filipe Araújo
Universidade de Lisboa

Luís Rodrigues
Universidade de Lisboa

August 9, 2005

Abstract

Connectivity of a sensor network depends critically on tolerance to node failures. Nodes may fail due to several reasons, including energy exhaustion, material fatigue, environmental hazards or deliberate attacks. Although most routing algorithms for sensor networks have the ability to circumvent zones where nodes have crashed, if too many nodes fail the network may become disconnected.

A sensible strategy for increasing the dependability of a sensor network consists in deploying more nodes than strictly necessary, to replace crashed nodes. Spare nodes that are not fundamental for routing or sensing may go to sleep. To ensure proper operation of the sensor network, sleeping nodes should monitor active nodes frequently. If crashed nodes are not replaced, messages follow sub-optimal routes (which are energy inefficient) and, furthermore, the network may eventually become partitioned due to the effect of accumulated crashes. On the other hand, to save the energy, nodes should remain sleeping as much as possible. In fact, if the energy consumed with the monitoring process is too high, spare nodes may exhaust their batteries (and the batteries of active nodes) before they are needed.

This paper studies the optimal monitoring period in fault-tolerant sensor networks to ensure that: *i)* the network remains connected (i.e., crashed nodes are detected and substituted fast enough to avoid the network partition) and, *ii)* the lifetime of the network is maximized (i.e., inactive nodes save as much battery as possible).

1 Introduction

Sensors have long since been used for monitoring processes where humans are either endangered by hazardous environments, too costly to be an option, or

[*] Selected sections of this report were published in the Proceedings Second Latin-American Symposium on Dependable Computing, Salvador, Bahia, Brazil, October 2005. This work was partially supported by the LaSIGE and by the FCT project P-SON POSC/EIA/60941/2004 via POSI and FEDER funds.

simply not able to effectively perform the sensing task. Recent progresses in miniaturization and networking technologies are empowering the use of sensors in self-organizing wireless networks, where nodes cooperate to more effectively achieve some goal. Wireless sensor networks have a wide range of applications, such as military, commercial, industrial, home or health.

In this paper we study techniques to increase the dependability of sensor networks. Nodes that crash can reduce the accuracy or completeness of the information being collected. Additionally, if too many nodes fail, the network may become disconnected. Therefore, we are particularly concerned with techniques that extend the lifetime of the network by postponing disconnection. A sensible strategy for increasing the dependability of a sensor network consists in deploying more nodes than strictly necessary. In this way, nodes collectively decide which ones remain active and which ones may go to sleep. To ensure proper operation of the sensor network, sleeping nodes should monitor active nodes frequently. Crashed nodes may cause sub-optimal routing (which wastes energy) as well as a network partition. On the other hand, to save the energy, nodes should remain sleeping as much as possible. If the energy consumed with the monitoring process is too high, spare nodes may exhaust their batteries (and the batteries of active nodes) before they are needed.

In this context, we would like to select a value for the monitoring period that maximizes the system availability. This task can be prohibitively complex due to the multiple combinations of factors that affect the system lifetime such as the initial energy available to nodes, power consumption, network topology, etc. The paper addresses this complexity by making the following contributions: in first place, it proposes a methodology of analysis that simplifies the reasoning about the network behavior and, in second place, it proposes two new metrics that capture the importance of the *relative* values of different system parameters. The first metric, called "Failure Weight Factor", $\mathcal{F}$, relates the Mean Time Between Failures, $MTBF$, with the maximum lifetime of the network, in ideal monitoring conditions. The second metric, called "Power On-off Consumption Factor", $\mathcal{P}$, relates the energy spent powering nodes on and off with the energy spent by other sources of energy consumption. Using simulations, we show that these two metrics are useful to reason about the impact of faults in the network lifetime.

The rest of the paper is structured has follows. Section 2 overviews related work. Section 3 presents our reference cell-based algorithm for energy conservation. Section 4 describes our metrics and the analysis methodology. The simulation results are presented and discussed in Section 5. Finally, Section 6 concludes the paper.

2 Background

The benefits gained from having more nodes than necessary have to be balanced against the (energy) costs of managing the nodes. In this section, we overview related work that helps to answer the following questions: How can the lifetime of a sensor network be precisely defined? How is energy consumed in a sensor network? Which are the best techniques to tolerate node failures? Should some redundant nodes be kept idle or, on the contrary, should all redundant nodes be kept sleeping most of the time? How to replace nodes whose energy has been exhausted? Which routing algorithm should be adopted? Previous work on the above topics helps us to define our strategy to build a fault-tolerant sensor network.

Network Lifetime In the literature, there are several definitions of network lifetime [25, 30, 7], like time to first node that dies. In this paper, we adopt the definition from [3], which considers that network life ends when the first partition occurs. For the scenarios considered in this paper, this metric offers a good measure of the availability of the network, because partitions typically occur little after half of the space where the sensor network lies becomes empty of nodes[1].

Energy Consumption In a sensor network, tasks that typically consume more energy are: sending and receiving messages, listening to the channel when idle, and processing. In this paper we do not consider sensing energy, because it depends mainly on the sensing task.

Several papers report that nodes consume a significant amount of energy in idle mode [23, 12, 8]. According to [12], the ratio of power needed in receive (transmit) mode against idle mode can be as low as 1.15 (1.56) [2]. This order of magnitude for idle power consumption paves the way to selectively powering down nodes, to conserve energy, because nodes consume only a small amount of energy while sleeping. One aspect that is often overlooked in literature is the cost of powering on and off a node. We believe that any algorithm that selectively powers down nodes to save energy must address this cost. In fact, there are two issues to consider: the time it takes to wake up and the large spike in energy consumption, due to the wakeup action alone plus a traffic announcement. The exact figures for both of these depend on the communication card and controlling software.

Fault-Tolerant Wireless Networks Resilience to node failures and energy efficiency must be addressed simultaneously, because an energy-efficient routing algorithm should be fault-tolerant and fault-tolerance can not come at a high energy cost. For this reason, several authors focused on algorithms that are both fault-tolerant *and* energy efficient (e.g, [10] and [11]). Several authors propose heuristics to ensure k-vertex connectivity [13, 5, 17]. Unfortunately, this construction requires nodes to be active when they are not strictly required. The amount of energy consumed this way results in an effective loss of network

[1] More precisely, most partitions occur after half of the network cells become inactive. A precise definition of the concept of cell can be found ahead in this text.

[2] In a Lucent IEEE 802.11 WaveLan PC Card.

lifetime. Therefore, most approaches to extend the network lifetime try to power down redundant nodes.

Powering Down Nodes There are many protocols that explore the idea of powering down redundant nodes, both at network and MAC layers (e.g. [22]). Attacking the problem at the network layer typically enables longer sleeping periods, because decisions are more informed. Instead of powering down a node for a single message or for some predefined number of time slots, knowledge of the routing algorithm can be used to selectively put some (almost) redundant nodes to sleep. For instance, this is the case of Span [8] that allows some nodes to sleep if node density is high enough. In [26], authors address the integration of the connectivity with the coverage problem. Other authors propose to selectively power down nodes in cluster-based routing schemes [30, 29, 28]. In cluster-based algorithms, a good policy to select the cluster-heads is the available energy (e.g., [28, 29]), instead of other criteria, like node id (e.g., [27]), or node degree (e.g., [9]). Due to their very well-structured organization and predictable behavior, division of the space into cells constitutes the ideal scenario to analyze the impact that the monitoring period has on the lifetime. In fact, we will show that this division allows us to evaluate precisely the effect of each input variable on the lifetime and, consequently, on the ideal monitoring period setting. For this reason, in this paper, we will adopt a modified version of Geographical Adaptive Fidelity, GAF [30], which we present in Section 3.

Routing Algorithm Several proposals for energy-aware routing strategies can be found in the literature [21, 6, 25]. While some of these strategies aim to prolong as much as possible the lifetime of the first node to die [6, 25], others try to avoid the exhaustion of the entire network [21]. To reduce power needed to transmit, nodes might adjust their transmission range. Using this technique, two papers [25] and [2] showed that the best strategy to deliver a message over a total distance D is to use equally spaced hops. Although in practice networks do not have nodes ideally located to relay a message, this result allows to derive upper bounds on network lifetime [2] and to build power-aware routing algorithms [25]. In [7, 25] authors simultaneously try to minimize power consumption as a whole and avoid exhaustion of nodes short of energy. Often, avoiding individual node depletion is not an issue in a sensor network, where fairness is less important than maintaining the network functioning.

The use of positional information is also important to conserve energy. As pointed out in [24, 14], positional routing algorithms make a more efficient use of resources than other routing algorithms like AODV [18], DSDV [19] or DSR [15] in large networks, because they use much fewer control messages. Additionally, positional information for the routing algorithm, in a scenario where a cell-based conserving energy algorithm is in use, comes for free, because a GPS receiver or an equivalent mechanism already exists. These facts motivated us to use, for the purpose of this study, a position-based routing algorithm. By avoiding algorithms that require configuration of several parameters, we also avoid the risks of having our results biased by inappropriate settings.

Therefore, we selected the Greedy Perimeter Stateless Routing (GPSR) algorithm [4, 16], because it is localized, and efficient. Furthermore, since GPSR has a very simple configuration and very few dedicated control messages, its operation has very little interference in our results. When possible, GPSR uses the greedy strategy of forwarding messages to the neighbor closest to destination. When it finds a local minimum, GPSR switches to perimeter mode and routes around faces. As soon as it finds a node closest to destination than the previous local minimum, GPSR goes back to greedy mode.

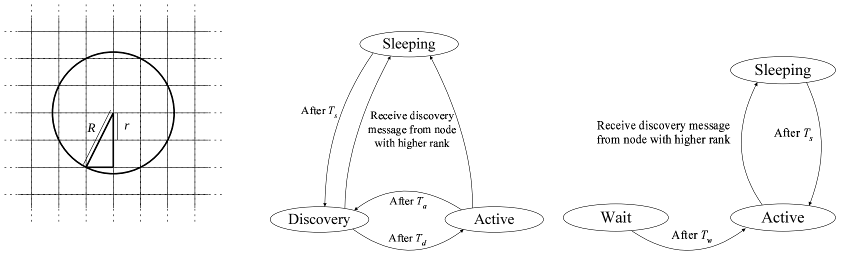
3 An Approach to Build Fault-Tolerant Sensor Networks

To build a fault-tolerant sensor network we include more nodes than strictly required. This allows replacement of failed nodes. To save battery, nodes collectively decide which ones are not fundamental for routing or sensing. These nodes should be sleeping most of the time, and only wake up with the minimum frequency required to replace failed nodes before the network disconnects. There are several issues that have to be defined in order to implement this strategy. In first place, nodes have to agree on some strategy to define which nodes should sleep, and which nodes must remain in idle state to maintain the network connectedness. In second place, one needs to define a strategy to perform the monitoring of idle nodes. Finally, one needs to define how often the monitoring procedure should be performed. This paper tackles the latter two issues, with particular emphasis on the importance of the monitoring period. As motivated in the previous section, we base our architecture in a GAF [30]-like cell based network running GPSR [16].

3.1 Node Monitoring in Geographical Adaptive Fidelity

Geographical Adaptive Fidelity (GAF) [30] is a cell-based energy-conserving algorithm. GAF aims to maintain all but one node sleeping in each cell. It assumes that nodes are aware of their location (for instance, using GPS receivers) and uses this information to divide the two-dimensional space into a grid. The two farthest points in any two adjacent cells must be within communication range, as depicted in Figure 1a. This bounds the cell side, r, to $r \leq R/\sqrt{5}$, where R is the communication range of the nodes. In scenarios where it is worthwhile using GAF, because more than one node exists per cell, resulting graph is very likely to be connected.

In GAF, nodes can be in one of three states: *active*, *discovery* or *sleeping*. Changes from one state to another are controlled by *discovery* messages and by timers. A node uses *discovery* messages to inform other nodes of its presence and of its application-dependent rank. In [30], authors propose as a ranking criterion, first, the state of the node (active > discovery) and then the expected lifetime, *enat* (higher ranks correspond to longer expected lifetimes). Hence,


R
r
Sleeping
After T_s
Receive discovery message from node with higher rank
After T_a
Discovery
Active
After T_d
Sleeping
Receive discovery message from node with higher rank
After T_s
Wait
After T_w
Active

(a) Division in cells (b) GAF (c) SQA

Figure 1: GAF and SQA algorithms

discovery messages consist of the following tuple: {node id, grid id, estimated node active time (*enat*), node state}. As depicted in Figure 1b, timers can change state of a node from sleeping to discovery (after T_s), from discovery to active (after T_d) and from active to discovery (after T_a). Nodes send discovery messages in any of the following situations: *i*) when they enter the discovery state, *ii*) when they enter active state after timeout T_d takes them from discovery to active; *iii*) periodically, after each period of T_d seconds in active state; *iv*) in active state when they receive a *discovery* message from a node with lower rank. Whenever a node in discovery or active states receives a *discovery* message from a node with higher rank it immediately resets its ongoing timers, sets up a timer to wake up and changes to sleeping state.

If nodes are put to sleep for too long, it may happen that the node occupying the cell either exhausts its battery or abandons the cell (if it is mobile) leaving it unattended. On the other hand, if sleeping nodes wake up too early, they will consume everybody's resources without further improving routing fidelity, thus defeating the goal of maximizing network lifetime. To achieve a good trade-off, GAF dynamically sets the sleeping period of a node, T_s, to depend on the estimated lifetime of the cell leader. In GAF, T_s is set to a fraction (50%) of the estimated lifetime of the leader. Hence after the T_a timer of the leader expires, it switches from active to discovery state, thus having an opportunity to be replaced in the cell. This is important for load balancing purposes (see [30] for further details).

3.2 Sleep-Query-Active Algorithm

Unlike [30], in this paper we consider some additional characteristics that make a more realistic scenario: *i*) nodes can fail and *ii*) waking up and putting nodes to sleep has fixed non-negligible cost. Furthermore, since we only consider sensor networks of fixed nodes, load balancing is not an issue. These differences motivated us to develop a variation of the GAF algorithm, which we call Sleep-Query-Active Algorithm (SQA), specifically suited to our setting. The states of SQA are depicted in Figure 1c. SQA is a pretty simple algorithm where nodes can only be in one of two states: either sleeping or active. The purpose of the *wait* state is only to desynchronize nodes that start at the same time. In our experiments, T_w was randomly set between 0 and 1 with uniform probability.

SQA nodes send discovery messages in the following situation: *i*) when they enter active state, *ii*) periodically when they are in the active state (to overcome the loss of messages) and *iii*) in active state when they receive a *discovery* message from a node with lower rank. Differences to GAF in the exchange of *discovery* messages, mainly reflect the way the rank is determined. In SQA the rank of the node is determined by the *enat* alone. Despite not providing any additional protection against node failures, nodes with larger supplies of energy will give an additional degree of protection against unexpected energy consumption caused by some peak of traffic. Perhaps the most important difference between GAF and SQA is that in SQA the sleeping timeout, T_s, which we deem as the *monitoring period*, is randomly chosen from an interval that is fixed beforehand. When we say that $T_s = c$, we really mean that T_s is selected from the interval $[0.5 \times c, 1.5 \times c]$. Then, each time a node goes to sleep, it picks the value for T_s from that interval with uniform probability. Our experimental evaluation shows that this choice is appropriate, because more often that not, the sensor networks will tend to behave in a very predictable way and using an optimal fixed value for T_s will yield longer lifetimes than the dynamic approach of GAF. The reader should notice that tuning SQA resumes to determining T_s. Selecting the most appropriate T_s is a challenging task that we address in the next sections. In fact, as we show in Section 5, for an appropriate choice of the monitoring period, SQA can successfully replace GAF in sensor networks.

4 Proposed Metrics and Analysis Methodology

When using SQA, we would like to determine the monitoring period T_s that maximizes network lifetime. Unfortunately, following a theoretical approach to determine T_s is a task of great difficulty. An example of such an attempt can be found in [3], where a theoretical bound for the network lifetime in a scenario where dead nodes are replaced at once without spending energy (we will call this the “ideal scenario” or “ideal network”) is derived. However, that work does not account for all the parameters we consider in this paper (e.g. faults) and, as noted in [3], it cannot be easily extended to capture practical scenarios. Hence, in this paper we have opted to use simulations to evaluate the effect of different parameters on the T_s. Unfortunately, without a correct methodology, the process of determining the effect of T_s on a network using simulations is also a daunting task. In fact, there are many factors that can influence network lifetime and consequently, T_s, including initial energy of nodes, idle energy consumption, transmission power, consumption power, sleep energy consumption, not to mention power on consumption and faults. Furthermore, these factors can be combined in multiple ways and often can not be completely isolated in order to analyze their impact on network lifetime. Finally, but not the least, a single ns-2 [1] simulation of a given configuration (i.e., for a single monitoring period), even when in executed on a Pentium IV 2.8 GHz with 2Gb of RAM, takes more than 100 seconds to complete.

To handle this complexity, the paper makes two contributions. In first place we propose a new set of metrics to reason about the influence of faults in the network lifetime. An interesting feature of these metrics is that they capture the *relative* weight of different factors, and highlight that networks with different absolute values of some parameters may exhibit a comparable behaviour. In second place, we propose a methodology of analysis that allows to reason about the impact of these metrics before assessing the impact of network topology in the final system availability. We will address these two contributions in the following paragraphs.

4.1 The $\mathcal{P}$ and $\mathcal{F}$ Metrics

Our metrics are motivated by the insight that, in the context of assessing the network availability, time intervals – in particular the monitoring period – should be analyzed in a relative sense: a monitoring period of 1 second has a different impact on a network whose lifetime is just 10 seconds than on a network whose lifetime is 1000 seconds. In a similar manner, the magnitude of values like power needed to transmit or to receive should also be measured in a relative way.

To reason in a generic manner about the fault-tolerance and power-on consumption of sensor networks, we start by defining the notion of *ideal lifetime*, LT_I. LT_I is the network lifetime in a scenario where *i*) there are no faults, *ii*) switching nodes on and off has no cost and *iii*) nodes in the cells are omnisciently replaced at once (if replacement is available). LT_I is determined by simulation and measures the available initial energy versus average consumption of the network. Using LT_I we propose the following metrics to assess the network behavior:

- The *power on-off consumption factor*, $\mathcal{P}$, measures the impact of the energy spent powering nodes on and off. We define it as the ratio between the energy needed for one power on-off operation versus remaining energy spent in 1 time unit. This is determined as $\mathcal{P} = POE/(TE_0/LT_I)$, where POE is the power on-off energy and TE_0 is the total energy available in the beginning of the network life (if we assume that all N nodes have the same energy, E_0, in the beginning, $TE_0 = N \times E_0$). This makes $\mathcal{P}$ a function of all remaining energies of the system *but not* of node failure rate.

- The *failure weight factor*, $\mathcal{F}$, measures the impact of faults in the network. We define it as the lifetime of the ideal network, LT_I, relative to $MTBF$, i.e., $\mathcal{F} = LT_I/MTBF$. This makes $\mathcal{F}$ a function of all energies *except* power on-off energy. Large $\mathcal{F}$ means many node failures (possibly due to

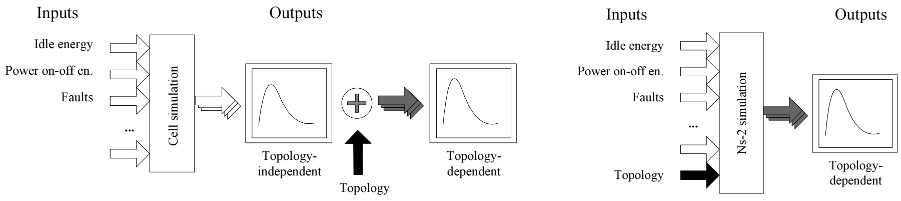


Figure 2: Cell Based Methodology *vs* Network Simulation

a long network lifetime), while large $\mathcal{P}$ means a lot of energy needed to power a node on and off (at least compared with remaining energies, like idle and traffic energies).

4.2 From Cell Level to Network Level Simulation

We propose and use the following methodology to evaluate the lifetime of the wireless network. Instead of always running simulation on a complete network, we first perform a careful study of the behavior of each network cell. Then, by estimating how many cells are required to maintain the connectivity of a given topology, we extrapolate the impact of the parameters in the entire network. We illustrate this methodology in Figure 2. The approach has both conceptual and practical advantages. From the conceptual point of view, it allows to separate the analysis of the influence of topology from other factors. From the practical point of view, cell level simulations *i*) allow to isolate factors that influence network lifetime and *ii*) run much faster. Therefore, cell simulation allows a much richer analysis of different combinations of factors in practical time. We validate our methodology by comparing the results obtained using this method with the results obtained by simulating the entire network. An additional advantage of the cell simulations is that its results can be used to assess other system properties. For instance, although outside the scope of this paper (where we focus on network lifetime) the analysis of cell simulations could be easily extended to study the problem of assessing the coverage of the sensor network in presence of faults.

5 Experimental Results

In this section we present our simulation results. We start by describing the settings used to perform cell level simulations and network level simulations. We then validate our methodology by comparing results derived from it (based on cell level simulations) with results obtained by directly simulating the entire network. Later, we show the relevance of the $\mathcal{P}$ and $\mathcal{F}$ metrics and their impact on the network lifetime. Finally, we illustrate the importance of appropriately selecting the correct monitoring period.

Node	Rx (W)	Tx (W)	Idle (W)	Sleep (W)	Init. Energy (J)
IEEE 802.11	0.974072	1.3410736	0.843	0.066303	15
MEDUSA-II	0.01248	0.01565	0.01234	0.00002	1
Rockwell's WINS	0.7516	1.0805	0.7275	0.064	20

Table 1: Consumption of energy for the nodes tested

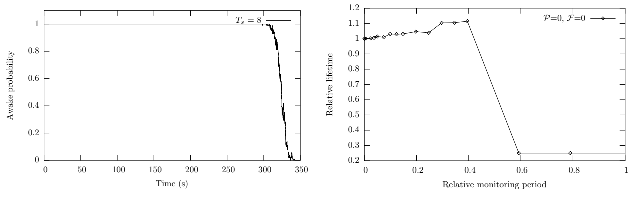
5.1 Simulation Settings

In our experiments we have used three different types of nodes: a node equipped with a Lucent IEEE 802.11 2 Mbps WaveLAN PC Card, a Rockwell's WINS node and a MEDUSA-II node. Table 1 resumes the consumption of the three different nodes in the situations considered in our simulations. Figures for the first node were taken from [12], while values for the other two types of nodes were inferred from [20].

We assume that failures of nodes follow an exponential distribution. However, for simulation purposes, we have modeled this as a geometric distribution. After constant time intervals P, all nodes may fail with a given random probability p (we set $P = 0.5$ seconds in our simulator). Hence parameter r of the exponential distribution is $r \approxeq -\frac{1}{P} \ln(1-p)$, while $MTBF = 1/r$.

To plot a graphic that represents lifetime relative to LT_I against the monitoring period relative to LT_I (e.g., Figure 4), we select a number of monitoring periods, T_s, not exceeding the ideal lifetime. Then, we fix all the parameters, like power on-off consumption, idle power, initial energy, etc. and we experimentally analyze the lifetime achieved for each T_s. We used a square size of 800×800 meters with 256 nodes, which we divided into 8×8 squares (giving an average of 4 nodes per cell). Communication range was 250 meters. The main difference between the cell and the ns-2 experiments is the way in which lifetime is determined. In ns-2 we run a simulation of the entire network to determine this value, while in the cell simulations we use a method that we describe next. We have performed additional simulations that show that these results also apply when other topologies are used (this aspect is discussed in Section 5.5).

Cell Level Simulation Settings To determine the lifetime for a given monitoring period, we fix this monitoring period and use time as the independent variable. Then, as time goes by we assume a constant consumption of energy and observe whether the cell is awake or sleeping (it is awake if there is any node awake, otherwise it is sleeping). We used an average of 100 of these trials to approximate a continuous random variable, function of time t, that represents the probability that the cell is awake. An example of a random variable like this is depicted in Figure 3a, for a specific value of T_s. To infer network behavior from this, we need to know the topology of the network. If disconnection occurs when an average number of D out of N cells are sleeping, we use a rough approximation and assume that when the awake probability of a cell



(a) Probability of a cell being awake

(b) Relative network lifetime

Figure 3: Derivation of network lifetime in cell simulation

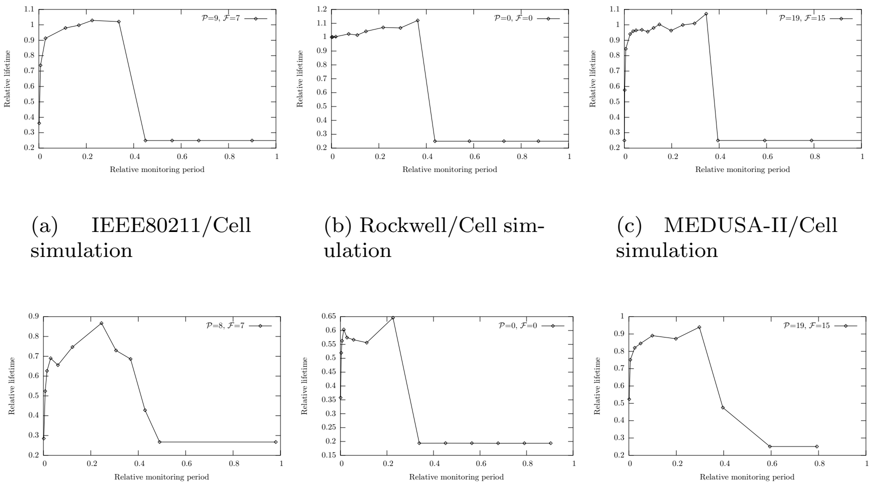
drops below $(N - D)/N$, the network gets disconnected. Taking our grid for example, we used a simple simulation to derive the probability density function of the number of sleeping cells that cause network disconnection. This looks like a Gaussian curve centered at 40 and truncated at the 64 cells. Therefore, in such a topology, the threshold $(64 - 40)/64 = 0.375$ corresponds to a point where, more often that not, network will be disconnected [3]. Figure 3b shows the relative lifetime graph as a function of the monitoring period for these settings.

Lifetime and monitoring periods represented in this plot are relative to the ideal lifetime LT_I, to abstract away the absolute magnitudes that govern the network behavior. Note that an entire data series needed to create a graphic like the one represented in Figure 3a produces a single point in Figure 3b. In this case, this point should occur around $t = 327$ seconds (where the line $y = 0.375$ intersects the probability curve). In the cell simulations LT_I is estimated as the number of nodes of the cell × the time it takes to consume all the energy of a node[4]. For the settings of these figures, this is around 324. Since $T_s = 8$ and $LT = 327$, this gives a relative monitoring period of $8/324 \approxeq 0.025$ and a relative lifetime of $327/324 \approxeq 1.009$. It is not really counterintuitive to have a lifetime greater than the ideal, due to the large idle power. In fact, this makes it advantageous to let some cells sleeping from time to time, to prolong their lives. On the contrary the ideal lifetime assumes that all the cells should be constantly awoken, which is not always the best strategy.

Network Level Simulation Settings We used the ns-2 simulator [1], version 2.27, to perform the network level simulations presented in this paper. This required us to implement the SQA algorithm as well as port the GPRS routing algorithm to the same version of ns-2. We used a simulation environment similar to the one described in [30]. Nodes were divided in traffic and transit nodes. Traffic nodes serve as sources and sinks of traffic, while transit nodes

[3] In this case, disconnection occurs when a significant proportion of the network is, in fact, unusable. We also observed this for other grid configurations.

[4] The reader should keep in mind that this only refers to the cell simulations.



(a) IEEE80211/Cell simulation

(b) Rockwell/Cell simulation

(c) MEDUSA-II/Cell simulation

(d) IEEE80211/Ns-2 simulation

(e) Rockwell/Ns-2 simulation

(f) MEDUSA-II/Ns-2 simulation

Figure 4: Lifetime estimated using cell level and network level simulations

are only used as intermediate hops for that traffic. Only transit nodes run the SQA/GPSR protocol. Traffic was generated by constant bit rate (CBR) traffic sources. In all our experiments we fixed the number of traffic nodes to 10. To prevent traffic nodes to stop generating traffic, their supply of energy was infinite.

5.2 Validation of the Methodology

To validate our methodology, we compare the results obtained from the application of the methodology, with the results obtained from complete network level simulations, using ns-2. Samples of several simulation we have performed for three different concrete node characteristics are depicted in Figure 4. Although the shape of the lines is slightly different, the peak in the relative lifetime plots is comparable, despite huge differences in power figures of nodes. This is very important, because in this peak lies the answer to the main question of this paper: what is the optimal selection of T_s? The fact that its width is similar in both types of simulations, allows us to use the simpler cell simulations to reason about the impact of the $\mathcal{P}$ and $\mathcal{F}$ metrics.

5.3 Relevance of $\mathcal{P}$ and $\mathcal{F}$ Metrics

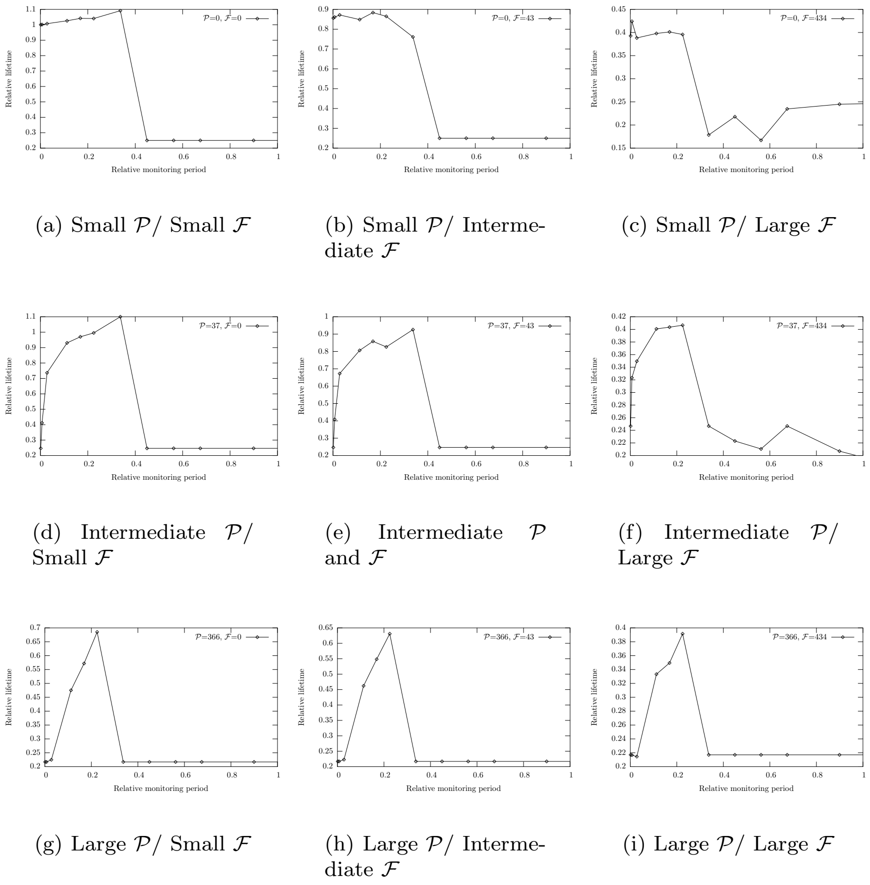
Impact of the Power Parameters on the Lifetime We observed that the impact of the power parameters, like idle, transmission or reception, can be hidden by plotting curves relative to the ideal lifetime. This was a surprising result of our simulations. Experiments made both with cell level simulations and in ns-2 confirmed this observation. Figure 4 allows to confirm this, because the three types of nodes have similar curves despite the differences in their power ratings (note for instance that MEDUSA-II consumptions are orders of magnitude away from the other types of nodes). Hence, the effect of the absolute values of the power consumptions are almost entirely ruled out, by using the simple technique of plotting lifetime curves relative to the ideal lifetime. This considerably simplifies the analysis of the metrics $\mathcal{P}$ and $\mathcal{F}$ to be done ahead.

The parameters that have larger impact on relative network lifetime curves are the power on consumption (assessed by $\mathcal{P}$) and the faults (assessed by $\mathcal{F}$). Impact of the node density is discussed in Section 5.4.

Impact of the Metrics $\mathcal{P}$ and $\mathcal{F}$ on the Lifetime We now use cell level simulations to discuss the impact of faults (represented by $\mathcal{F}$) on the network lifetime considering a non-negligible replacement cost (represented by $\mathcal{P}$). For most values of $\mathcal{P}$ and $\mathcal{F}$, the stability of the lifetime peak still holds. Since several combination of input parameters are captured by the two metrics, a precise determination of these metrics should be enough to qualitatively determine the behavior of the network. Figure 5 shows extreme as well as typical values for $\mathcal{P}$ and $\mathcal{F}$. We can see that results confirm the initial intuition: large values of $\mathcal{F}$ tend to require smaller monitoring periods (thus shrinking the curve at the right and making the peak start slightly earlier). On the other hand, larger values of $\mathcal{P}$ will penalize small monitoring periods (thus shrinking the curve at the left). Hence, as these two metrics grow, the curve tends to become thinner. Moreover, the growth of these metrics also makes the curve shorter as they impact network lifetime. To conserve space we only depict results for the IEEE 802.11 adapter. However, results for the other types of nodes show similar behaviors. Together with other simulations that we have done, this shows that very different operational conditions have similar behaviors, as long as the metrics $\mathcal{P}$ and $\mathcal{F}$ are similar (this effect also occurs in Figure 4).

Table 2, which summarizes the results obtained, offers a qualitative analysis of this issue. Outside the parenthesis we describe the system parameter that dominates network lifetime (other energies refers to idle and traffic energies), while inside we describe the shape of the peak that exists in the monitoring period (earlier, normal or later, respectively means that peak starts closer, in the normal place or farther away from the origin). Given the values of Table 1 and the huge idle mode power, we expect current technology to operate in the first line of the table ("Small $\mathcal{P}$"). If with technological improvements idle energy decreases, $\mathcal{P}$ will depend mainly on data traffic generated on the network. In this case, the network will operate in a zone captured by the bottom line of the table ("Large $\mathcal{P}$"), whenever average traffic becomes low. In such scenarios, the appropriate choice of T_s will make an even more significant impact on the network lifetime.

In our simulations, including results depicted in Figures 4 and 5, longest



(a) Small $\mathcal{P}$/ Small $\mathcal{F}$

(b) Small $\mathcal{P}$/ Intermediate $\mathcal{F}$

(c) Small $\mathcal{P}$/ Large $\mathcal{F}$

(d) Intermediate $\mathcal{P}$/ Small $\mathcal{F}$

(e) Intermediate $\mathcal{P}$ and $\mathcal{F}$

(f) Intermediate $\mathcal{P}$/ Large $\mathcal{F}$

(g) Large $\mathcal{P}$/ Small $\mathcal{F}$

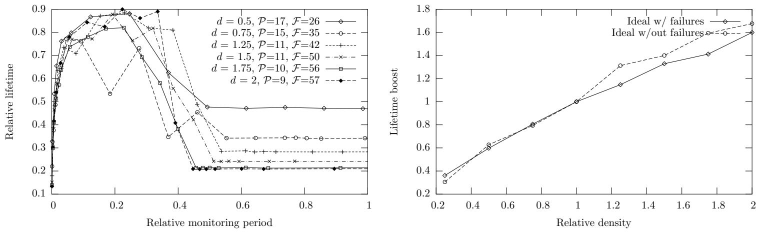
(h) Large $\mathcal{P}$/ Intermediate $\mathcal{F}$

(i) Large $\mathcal{P}$/ Large $\mathcal{F}$

Figure 5: Impact of $\mathcal{P}$ and $\mathcal{F}$

	Small $\mathcal{F}$	Intermediate $\mathcal{F}$	Large $\mathcal{F}$
Small $\mathcal{P}$	Other en. (earlier)	Other en. & Failures (earlier)	Failures (earlier)
Intermediate $\mathcal{P}$	All en. (normal)	None (normal)	Failures (slightly earlier)
Large $\mathcal{P}$	On-off (later)	On-off (later)	Depends rel. magnitude (later)

Table 2: Dominating parameter (and peak shape) for variations of $\mathcal{F}$ and $\mathcal{P}$



(a) Varying node densities

(b) Lifetime boost

Figure 6: Relative lifetime and lifetime boost with varying node densities

lifetimes are almost always achieved when monitoring period is in the range of 10 to 20% of the ideal lifetime, for most values of $\mathcal{P}$ and $\mathcal{F}$. This stability has to do with the fact that a perfect monitoring algorithm should ensure that network has as few active nodes as possible (fewer than the number of cells, in practice), but preserving the minimum required to prevent disconnection from occurring. Hence, substitution of nodes depends on the rate nodes die, which on its turn will determine lifetime. This explains why better strategies for (potentially) longer lifetimes, should use longer monitoring periods. Nevertheless, if this period goes over some threshold (30 to 50%), the relative lifetime sharply decreases, because nodes that die are not replaced and many cells become empty. This reveals a thin line between optimal and disastrous configuration.

5.4 Impact of Node Density on the Lifetime

One aspect of our results that is difficult to understand with the ns-2 simulations, but evident in cell simulations is the impact of node density. Cell experiments (that we omit to conserve space), have shown that the peak of the lifetime curve shrinks when the number of nodes per cell increases. This is consistent with results obtained in ns-2 (IEEE 802.11) and depicted in Figure 6a, where this effect is quite subtle. In this experiment we fixed all parameters and varied the number of nodes from 64 to 512 (density $d = 1$ represents 256 nodes). The gain in lifetime (relative to the lifetime of density 1) is depicted in Figure 6b for different network densities. We have studied two scenarios of independent interest: ideal replacement policy with and without node failures. The approximately linear growth of lifetime when there are no failures is consistent with [3]. However, when we consider failures of nodes, as absolute lifetime increases, failures become more important ($\mathcal{F}$ grows). This makes lifetime (relatively) shorter as density increases.

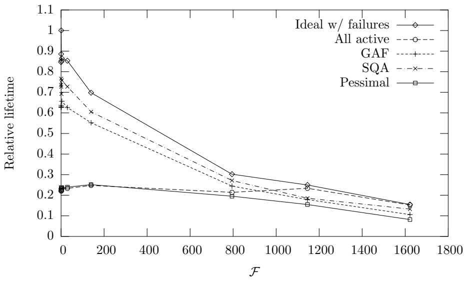


Figure 7: Lifetime for Different Replacement Methods

5.5 Impact of Topology on the Lifetime

Other experiments that we have made with ns-2 for the topology settings originally described in [30] did not show significant changes to the results presented here. This scenario is of particular interest, because nodes are scattered in a rectangle of 1500 × 300 meters with 100 × 100 meters for each cell, which gives only 3 cells in one of the directions. Cell simulations with thresholds different from 0.375 also have also produced similar results. Nevertheless, we believe that it is still an open problem to know if there are configurations that considerably impact the lifetime of the network and how can that impact be predicted.

5.6 Practical Relevance

We finally show in Figure 7 the benefit from adequately selecting the monitoring period T_s. We illustrate this by using several different replacement policies in scenarios with increasing node failure rates, simulated for 256 nodes in ns-2 (IEEE 802.11 adapter). $\mathcal{F} = 0$ means that there are no failures, i.e., $MTBF = \infty$. First, we determine an upper bound for the lifetime using an ideal scenario with node failures ("Ideal w/ failures"). Next, we use a worst-case setting where T_s is so long that no actual substitution ever occurs ("Pessimal"). The third intermediate scenario consists of keeping all nodes awake. In this case, no idle energy is conserved ("All active"). The y-axis of the graphic is normalized to the ideal lifetime, LT_I (which does not vary along the x-axis, as it does not have node failures).

Then we plot two additional curves in the graphic: lifetime obtained by the GAF algorithm and lifetime obtained by SQA. For SQA we select the monitoring period using the results from the analysis presented in Subsection 5.3: we selected smaller monitoring periods for larger values of $\mathcal{F}$, starting at 20% of LT_I, for small values of $\mathcal{F}$ and decreasing for 15%, 10% and finally 5% as $\mathcal{F}$ grew larger. From the figure we can reach the following conclusions:

- Not adjusting the monitoring period (for instance, using the pessimal or the all active approaches) offers a network lifetime that is much worse than the ideal.

- Using the analysis presented in this paper, SQA can be tuned to achieve a lifetime that is frequently between 80 and 90% of the ideal.
- SQA offers, for most values of $\mathcal{F}$, a much longer network lifetime than GAF, that can be as high as 25%.

As a promising future research topic, we envision to combine the advantages of SQA and GAF. The resulting algorithm could have the ability to dynamically set the monitoring period, according to the importance of faults existing on the network or to the power on-off consumption.

6 Conclusions

In this paper we studied the dependability of sensor networks, considering energy constraints and fault-tolerance requirements. We aimed at determining the ideal monitoring period for cell-based energy conserving techniques, to maximize network lifetime, here defined as time to the first network partition. To simplify this task, this paper made two contributions: a methodology of analysis, which consisted of inferring network behavior from inspection of individual cells; and two metrics, $\mathcal{P}$ and $\mathcal{F}$ that are able to capture the operational conditions of the sensor network.

Experimental results demonstrated the appropriateness of using these metrics to assess network behavior, by showing that, often, $\mathcal{P}$ and $\mathcal{F}$ strongly determine network operation. Furthermore, results have shown that it is possible to achieve a lifetime close to the ideal by selecting the monitoring period adequately and according to $\mathcal{P}$ and $\mathcal{F}$. More precisely, we have shown that network lifetime can be within 80 and 90% of that provided by an (non-implementable) ideal replacement policy, even for very large failure rates.

References

[1] *The* ns *Manual.* http://www.isi.edu/nsnam/ns/ns-documentation.

[2] M. Bhardwaj, A. Chandrakasan, and T. Garnett. Upper bounds on the lifetime of sensor networks. In *IEEE International Conference on Communications*, pages 785–790, 2001.

[3] D. Blough and P. Santi. Investigating upper bounds on network lifetime extension for cell-based energy conservation techniques in stationary ad hoc networks. In *ACM Mobicom*, 2002.

[4] Prosenjit Bose, Pat Morin, Ivan Stojmenović, and Jorge Urrutia. Routing with guaranteed delivery in *ad hoc* wireless networks. In *International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM)*, pages 48–55, 1999.

[5] G. Calinescu, I.I. Mandoiu, and A. Zelikovsky. Symmetric connectivity with minimum power consumption in radio networks. In *17th IFIP World Computer Congress*, pages 119–130, 2002.

[6] J. Chang and L. Tassiulas. Routing for maximum system lifetime in wireless ad-hoc networks. In *37-th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, September 1999.

[7] Jae-Hwan Chang and Leandros Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *INFOCOM (1)*, pages 22–31, 2000.

[8] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Wireless Networks*, 8(5):481–494, 2002.

[9] G. Chen and I. Stojmenovic. Clustering and routing in wireless ad hoc networks. Technical Report TR-99-05, Department of Computer Science, SITE, University of Ottawa, Ottawa, Ontario K1N 6N5, Canada, June 1999.

[10] S. Chessa and P. Santi. Crash faults identification in wireless sensor networks. *Computer Communications*, 45(2):126–143, November 2002.

[11] Amitava Datta. Fault-tolerant and energy-efficient permutation routing protocol for wireless networks. In *International Parallel and Distributed Processing Symposium (IPDPS'03)*, 2003.

[12] Laura Marie Feeney and Martin Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *IEEE INFOCOM*, 2001.

[13] MohammadTaghi Hajiaghayi, Nicole Immorlica, and Vahab S. Mirrokni. Power optimization in fault-tolerant topology control algorithms for wireless multi-hop networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 300–312. ACM Press, 2003.

[14] R. Jain, A. Puri, and R. Sengupta. Geographical routing using partial information for wireless ad hoc networks. *IEEE Personal Communication*, pages 48–57, February 2001.

[15] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[16] Brad Karp and H. T. Kung. GPRS: Greedy perimeter stateless routing for wireless networks. In *ACM/IEEE International Conference on Mobile Computing and Networking*, 2000.

[17] Xiang-Yang Li, Peng-Jun Wan, Yu Wang, and Chih-Wei Yi. Fault tolerant deployment and topology control in wireless networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 117–128. ACM Press, 2003.

[18] C. Perkins. Ad-hoc on-demand distance vector routing, 1997.

[19] Charles Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, pages 234–244, 1994.

[20] Vijay Raghunathan, Curt Schurgers, Sung Park, and Mani B. Srivastava. Energy-aware wireless microsensor networks. *IEEE Signal Processing Magazine*, pages 40–50, March 2002.

[21] V. Rodoplu and T. Meng. Minimum energy mobile wireless networks. In *1998 IEEE International Conference on Communications, ICC'98*, volume 3, pages 1633–1639, Atlanta, GA, June 1998.

[22] S. Singh and C. Raghavendra. Pamas: Power aware multi-access protocol with signalling for ad hoc networks. *ACM Computer Communication Review*, July 1998.

[23] M. Stemm and R. H. Katz. Measuring and reducing energy consumption of network interfaces in hand-held devices. *IEICE Transactions on Communications*, E80-B(8):1125–31, 1997.

[24] Ivan Stojmenovic. Position-based routing in ad hoc networks. *IEEE Communications Magazine*, July 2002.

[25] Ivan Stojmenovic and Xu Lin. Power-aware localized routing in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(11):1122–1133, 2001.

[26] Xiaorui Wang, Guoliang Xing, Yuanfang Zhang, Chenyang Lu, Robert Pless, and Christopher Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 28–39, New York, NY, USA, 2003. ACM Press.

[27] Yu Wang and Xiang-Yang Li. Geometric spanners for wireless ad hoc networks. In *The 22nd IEEE International Conference on Distributed Computing Systems*, 2002.

[28] J. Wu, B. Wu, and I. Stojmenovic. Power-aware broadcasting and activity scheduling in ad hoc wireless networks using connected dominating sets. *Wireless Communications and Mobile Computing*, 4(1):425–438, June 2003.

[29] Ya Xu, Solomon Bien, Yutaka Mori, John Heidemann, and Deborah Estrin. Topology control protocols to conserve energy inwireless ad hoc networks. Technical Report 6, University of California, Los Angeles, Center for Embedded Networked Computing, January 2003. submitted for publication.

[30] Ya Xu, John S. Heidemann, and Deborah Estrin. Geography-informed energy conservation for ad hoc routing. In *Mobile Computing and Networking*, pages 70–84, 2001.