

# Optimal Period Length for the CGS Sensor Network Scheduling Algorithm

Gábor Bergmann\*, Miklós Molnár†, László Gönczy\* , and Bernard Cousin‡

\*Budapest University of Technology and Economics, Hungary

†IRISA, INSA Rennes, France

‡IRISA, University of Rennes I, France

**Abstract**— In Wireless Sensor Networks (WSN), a tradeoff between the network lifetime (limited by energy of sensors) and redundancy of actively sensing and communicating sensors (implicated by coverage requirements for the measured area) has to be established, typically in an over-deployed environment. This is achieved by scheduling algorithms which periodically alternate the state of sensors between "asleep" and "awake". Obviously, the period length of the periodical synchronized scheduling affects the network performance and lifetime. Controlled Greedy Sleep algorithm is a quasi-optimal synchronized sensor scheduling algorithm which increases network lifetime while maintaining correct functionality, based on local decisions of sensors. This paper investigates the optimization of the period length of this algorithm and highlights best practices with simulations. Studies have been performed within a ring topology and in random square topology.

**Index Terms**—Ad hoc networking, sensor network, energy conservation, sensor scheduling, synchronized alternation, optimal period length, simulation.

## I. INTRODUCTION AND PROBLEM STATEMENT

Wireless sensor networks (WSNs) are deployed to monitor and to measure areas in a reliable manner. WSNs have several application domains: to monitor the environment, to collect information, to register and process environmental parameters for optimization or prediction, and/or to insure security ([1] describes many applications and challenges). Often, the measurement or surveillance task of a WSN requires the complete coverage of a target area or a set of target objects. In a general WSN architecture, several sensor nodes send the observation data to Base Stations (BSs) or sinks. Then data can be processed by the sinks and later send to the potential clients. The sensors are performing sensing and communication tasks and the main problems and challenges of this kind of networks are associated to these two activities [2]. The sensor network should be capable to measure in the target area and to process the measured values and transmit them to sink nodes.

While usual BSs are fix and powered by wire, sensors are generally battery powered. Low cost sensors operate, generally, on limited-power battery. For example, AA type battery equipped sensors can run continuously for some days but the observation duration of the network should last at least some weeks or months [3]. Thus the efficiency of the energy consumption is of the highest importance because it determines the lifetime of the WSN. Moreover WSN are over-dimensioned (*i.e.* sensors are deployed with a high redundancy

rate) to provide alternative sensors for a good coverage of the target area. This over-dimensioning can be exploited to put some sensors in sleep state for a previously determined time interval, thus saving their electrical energy. Consequently all sensors can alternate between sleep and awake states. This alternation permits to keep on the WSN activities: to maintain the coverage over an area and communication while minimizing the network energy consumption (thus maximizing the network lifetime) [4], [5]. A scheduling algorithm (often distributed over all sensors) should be used to decide which sensors (and when) have to move from one state to the other. To assure good space coverage and pervasive services, efficient but simple algorithms should be applied to optimize the utilization of limited network resources and capabilities. Using sleep scheduling in the WSN, both the measurement and the communication tasks should be provided with the remaining awake sensors. Usually, the election and alternation of awake sensor sets is based on periodical synchronization of the sensors. This periodical procedure also consumes energy. Intuitively, if the period length is small, the scheduling procedure has a heavy overhead. Contrarily, if the period is too long, the sensor battery capacities might not be harnessed to their full extent.

In this paper we consider the analysis of the impact of the scheduling period length on the network lifetime. Section II presents the interest of the sensor scheduling and the usual scheduling techniques. Our analysis is based on a quasi-optimal scheduling algorithm proposed in [6], which is briefly described in Section III. The energy consumption model and the used simulation technique are detailed in Section IV Section V respectively. Simulation results on the impact of the period length follow in Section VI. We propose a method for choosing a period length in Section VII. Conclusions and perspectives close our study.

## II. SENSOR SCHEDULING

Sensors have generally limited power capacity but WSNs has to meet relatively strong lifetime requirements, thus the energy conservation is a critical issue in WSN. Mostly, high density of sensors and a *scheduled sleeping* algorithm are employed to preserve energy of the sensors and provide the network services. Sensors periodically alternate between *sleep*

and *awake* states using a given period length  $T$ , thus saving energy.

### A. Our Hypothesis

The performance of a sensor scheduling depends on the network topology (*i.e.* on the sensor locations). In a WSN, sensors can be deployed randomly or in a controlled way. The controlled deployment of sensors may only be feasible in an accessible environment. In inhospitable or inaccessible fields, sensors are randomly placed. In [7] a recent state of the art on optimized node placement identifies the various objectives and enumerates the different models for static and dynamic WSNs. In our problem, we consider both controlled and random sensor topologies.

The measurement and data transmission of sensors in the space can be characterized by their sensing range, communication range respectively. Generally the communication range of a sensor is greater than the twice of the sensing one; so, one can suppose that the sensors sharing the measurement/observation task anywhere can also communicate one with the other. Consequently, if the target area is covered with awake sensors, then the connectivity of the WSN is also ensured [8]. So, scheduling algorithms may focus only at the solution of the coverage problem.

In our study, we suppose that the sensors are placed in such a way that each point of the area is covered by at least  $K$  sensors.

### B. The $K$ -coverage Problem

Let us suppose that the relations between targets and sensors is represented by a bi-bipartite graph  $G = (R \cup S, E)$  where  $R$  is the finite set of observed regions and  $S$  is the available sensor set. An edge  $e_{r,s} \in E$  exists in  $G$ , if the sensor  $s \in S$  can observe the region  $r \in R$ .

1) *Static  $K$ -coverage*: The *simple static  $K$ -coverage problem* corresponds to find a sub-graph  $G_a = (R \cup S_a, E_a)$  of  $G$  so that for all regions  $r \in R$  its degree is at least  $K$ . The set  $S_a$  corresponds to an awake subset of available sensors such that it guarantees the  $K$ -coverage of each region. A graph  $G'_a = (R \cup S'_a, E'_a)$  is *non-redundant* if it does not exist  $S_b \subset S'_a$  generating  $G_b = (R \cup S_b, E_b)$  that solves the simple  $K$ -coverage problem.

The *minimal static  $K$ -coverage problem* is to find a non-redundant sub-graph  $G_m = (R \cup S_m, E_m)$  that solves the  $K$ -coverage and  $|E_m|$  is minimal. Similarly to the Minimal Cover Vertex Set problem, the static minimum  $K$ -coverage problem is considered as an NP-difficult problem.

2) *Dynamic  $K$ -coverage*: The goal of the dynamic sensor network scheduling ( $K$ -coverage) is more complex than the computation of static minimal  $K$ -covering sets. The scheduling algorithms aim to prolong the lifetime of the sensor network with the help of the alternation of appropriate (non-redundant but not necessary minimal)  $K$ -covering sets minimizing the power consumption and maximizing the duration of the  $K$ -coverage of the target area. In fact, performing a  $K$ -coverage

using a certain subset of sensors for awhile consumes the power of the participating sensors.

Formally, the *dynamic  $K$ -coverage problem* aims to find a sequence of graphs  $G_k$  each solving the strict static  $K$ -coverage problem for an associated duration time  $t_k$ . The sensors of each subset  $S_k$  are supposed awake during  $t_k$  and have sufficient available power. A solution can be described as:

$$O = \{(S_k, t_k), k = 1, \dots, m\} \quad (1)$$

where the sensors in  $S_k \subseteq S$  are awake during the time interval  $t_k$ . The total duration of the  $K$ -coverage provided by  $O$  is equal to  $t = \sum_{k=1}^m t_k$ .

To take into account topology changes, dynamic scheduling algorithms are needed. The problem of complex dynamic computation is that small size sensors can not perform complex algorithms. Thus simple scheduling techniques are needed.

### C. Scheduling Techniques

Scheduling algorithms can be classified according to different aspects: centralized or distributed, random or deterministic, etc.

Centralized scheduling needs a central node with sufficient computation capacities and induces additional communications between the central node and all sensors. These communications consume energy, may produce network congestion or link collision around the central node and are very susceptible to central node failure but the result of the centralized scheduling is generally efficient. For instance, in [9] a dynamic mixed integer linear programming (MILP) model is presented to solve a coupled coverage and connectivity dynamic problem (CCDP) in flat WSNs. A good example for distributed, random sleeping algorithm can be found in [10] where nodes make local decisions on whether to sleep or to join a forwarding backbone, to ensure measurement and communications. Each node bases its decision on an estimation of the number of other neighbor sensors which will benefit from its awake period and the amount of remaining energy.

In [3] the authors propose a simple randomized scheduling for dense and mostly sleeping sensor networks. They suppose that there are many redundant sensors in the target area and one can compute and ensure the required (identical) duty cycle for individual sensors. In the proposed Randomized Independent Sleeping (RIS) algorithm, time is divided into periods. At the beginning of each period, each sensor decides whether to go to sleep (with probability  $p$  computed from the duty cycle) or not, thus the lifetime of the network is increased by a factor  $1/p$ . This solution is very simple and does not require communication between sensors. The drawback of the random scheduling propositions is that there is no guarantee for coverage nor for network connectivity.

For a more precise scheduling, the authors in [11] propose a Role-Alternating, Coverage-preserving, Coordinated Sleep Algorithm (RACP). Each sensor sends a message periodically to its neighborhood containing its location, residual energy and other control information. An explicit acknowledgment-based

election algorithm permits to decide the sleep/awake status. The coordinated sleep is more robust and reduces the duty cycle of sensors compared to the random sleep algorithm.

Many scheduling algorithms work periodically. The period length of the scheduling impacts the efficiency of the solution: this is the subject of our investigation. We propose to analyze the impact of the period length on the efficiency using a quasi-optimal distributed scheduling algorithm which is presented in the following.

### III. CONTROLLED GREEDY SLEEP ALGORITHM

The analyzed Controlled Greedy Sleep (CGS) algorithm is a synchronized scheduling algorithm which was proposed in [6]. CGS is based on periodic synchronization between sensors in order to determine which of them are the most critical and are eventually allowed to go to sleep for a period of time. In this scheduling nodes have local information on their neighborhood only. Each sensor node  $q$  will use a locally known sub-graph  $G_q(S_q \cup R_q, E_q)$ . This sub-graph contains geographical regions  $R_q$  covered by  $q$ , the set  $S_q$  of sensors which participate the coverage of at least one region of  $R_q$ .  $E_q$  contains the edges between the regions and sensors. The scheduling is based on a particular factor. The *drowsiness factor* of the node with remaining energy  $P_s$  is defined by

$$D_q = \begin{cases} \frac{1}{P_q^\alpha} \sum_{\forall r \in R} \text{if } \phi_r > 0, \forall r \\ -1 \text{ otherwise} \end{cases} \quad (2)$$

where  $\alpha$  is a positive constant (e.g.  $\alpha = 2$ ), and  $\phi_r$  is the coverage ratio of region  $r \in R_q$ .

The coverage ratio is positive if the region is over-covered, and negative otherwise: in this latter case the operation of all sensors covering  $r$  is essential. Moreover, the smaller the energy of  $q$ , the larger its drowsiness. CGS enforces the sensors in critical positions to go to sleep whenever it is possible, to conserve their energy for times when their participation will become inevitable. A sensor  $q$  can go to sleep when its neighbors with larger drowsiness factor decided their state for the next period and  $q$  has no critical (not over-covered) region to monitor. Consequently, each sensor should know the drowsiness factor of its neighbors and the decision of neighbors with larger factor. To organize the local communication, a communication delay (DTD) is associated with each sensor. This delay is inversely proportional with the drowsiness factor. So the sensors with large factor broadcast their decision earlier. Only the awake state decision should be broadcasted explicitly, in this way the communication overhead can be minimal.

The main steps of the Controlled Greedy Sleep (CGS) Algorithm are the followings:

- 1) At the beginning of the period, wake up all sensors whose remaining energy is enough for spending at least one period awake.
- 2) Alive sensors broadcast local *Hello* messages containing their locations. Based on received *Hello* messages each

sensor  $q$  builds up its local set of alive neighbors  $S_q$  and generates the local bi-partite graph  $G_q(S_q \cup R_q, E_q)$ , and then it calculates its drowsiness factor  $D_q$ .

- 3) Based on  $D_q$  each node  $q$  selects a Decision Time Delay ( $DTD_q$ ). Small drowsiness means large DTD, large drowsiness means small  $DTD$ . These delays provide priorities when nodes announce their Awake Message (*AM*). Each sensor  $q$  broadcasts its  $DTD_q$  and starts collecting  $DTD$  and *AM* messages from the neighborhood. From the received  $DTD$  and *AM* messages it builds a Delay List ( $DL_q$ ) and a List of Awake Neighbors ( $LAN_q$ ) respectively.
- 4) When  $DTD_q$  time elapsed the node  $q$  makes a decision based upon  $LAN_q$  and  $DL_q$ :
  - if all regions in  $R_q$  can be K-covered using only nodes present in  $LAN_q$  and/or nodes  $u$  present in  $DL_q$  for which  $DTD_u > DTD_q$  then go to *sleep*
  - otherwise stay *awake* and broadcast an *AM* to inform the neighbors of this decision.

Figure 1 illustrates a typical period of the CGS algorithm.

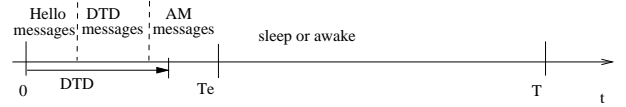


Figure 1. A typical period of CGS

Obviously, the communication overhead of the algorithm depends on the length of periods. At the beginning of the period there are three time intervals: to exchange *Hello*, *DTD* and *AM* messages respectively. A sensor broadcasts at most three messages during  $T_e$  (two if it the node will go to sleep, three otherwise) and must stay awake in order to complete the election process. During this extra  $T_e$  time sensors consume energy. The scheduling communication and awake-time overhead can be low if the length  $T$  of a period is significantly longer than  $T_e$ . But, one can state that this period can not be too long either.

The determination of the optimal length  $T$  is a hard computation problem. In real cases only empirical and estimated solutions can be formulated. The study of this period length with simulation offers significant elements to choose the period length.

### IV. ENERGY CONSUMPTION MODEL

#### A. Periodical energy consumption

For purposes of optimization of the period length  $T$ , the periodical energy consumption of the sensors can be modeled as the sum of two components. The first component contains the communication overhead during the initial phase at the beginning of each period, and any other costs independent from the length of the period, such as clock synchronization or staying awake for the duration  $T_e$ . The second component is the time-proportional energy consumption; it contains the cost of the measurement and communication activities which

are supposedly proportional to  $T$ . It is important to point out that both the fixed communication cost and the consumption rate depend on whether the particular sensor spends the period awake or asleep: sleeping sensors send only 2 CGS messages, while awake ones send 3, and spending a period asleep consumes energy by a much lower rate than staying awake and measuring (which is the whole point of using CGS). Apart from the difference between sleeping and awake sensors, however, we assume that these costs are the same for all sensors. Thus our energy consumption model is described by the following formula:

$$E_X = F_X + C_X * T \quad (3)$$

where  $E_X$  is the periodical energy consumption,  $F_X$  is the fixed overhead of a period,  $C_X$  is the consumption rate of the time-proportional component, and  $X$  is the type of the period (A for awake or S for sleep).

### B. Limiting the lifetime of K-Coverage

Let  $E_T$  be the combined average energy consumption rate of a single sensor, including both fixed and time-proportional costs, during the lifetime of K-coverage. If  $N_K$  is the number of periods while K-coverage is maintained, and  $N_A$  is the average number of periods spent awake by a sensor during K-coverage, then

$$E_T = \frac{N_A * E_A + (N_K - N_A) * E_S}{N_K * T} \quad (4)$$

Let's define  $p_A$  as the average portion of periods spent awake by a sensor:

$$p_A = \frac{N_A}{N_K} \quad (5)$$

The value of  $p_A$  is dependent on the overdosing of the area to be observed, and also the actual topology of the sensors. Now we can combine the above equations to get:

$$\begin{aligned} E_T &= \frac{p_A * E_A + (1 - p_A) * E_S}{T} \\ &= \frac{p_A * (F_A + C_A * T) + (1 - p_A) * (F_S + C_S * T)}{T} \end{aligned}$$

and thus

$$E_T = p_A * C_A + (1 - p_A) * C_S + \frac{p_A * F_A + (1 - p_A) * F_S}{T} \quad (6)$$

The goal to be maximized would be the performance  $P$  of the network, which is the total lifetime of K-coverage:

$$P = N_K * T \quad (7)$$

Let  $E$  be the initial energy level of the sensor nodes; once again, we assume that sensors are symmetric. Obviously

$$E \geq P * E_T \quad (8)$$

Combining the above two and expanding the energy consumption model yields the following:

$$P \leq \frac{E}{E_T} = \frac{E}{p_A * C_A + (1 - p_A) * C_S + \frac{p_A * F_A + (1 - p_A) * F_S}{T}} \quad (9)$$

The above formula suggests that by increasing  $T$ , the communication overhead diminishes and performance of the network increases. This is misleading: one could even assume that the best choice would be  $T_{max}$ , the largest possible period length permitted by the battery capacity, for which a single awake period consumes all energy:

$$T_{max} = \frac{E - F_A}{C_A} \quad (10)$$

However, (9) merely gives an upper bound for the performance. As our simulations reveal in Section VI, there are also other factors to consider that may be more relevant with realistic parameters, making the choice of  $T$  a much more difficult decision.

### C. Estimation of realistic values

It is quite common to employ standard AA batteries as the energy source of wireless sensor nodes. The energy capacity of these batteries typically range between  $5kJ$  and  $16kJ$ . In accordance with [12], we assume that sleeping sensors with a disabled radio consume energy by a rate in the order of magnitude of  $10^{-5}$  Watts, while an active sensor drains power in the order of magnitude of  $10^{-2}$  Watts.

To assess communication costs in accordance with [13], we assume that sending messages (with potential multipath fading) has a cost of  $lE_{elec} + l\epsilon_{mp}d^4$  where  $l$  is the number of transmitted bits,  $d$  is the range of acceptable reception,  $E_{elec}$  is  $5 * 10^{-8} J/bit$  and  $\epsilon_{mp}$  is approximately  $10^{-15} J/bit/m^4$ . Furthermore, we take 125 bytes including overhead ( $l = 1000bits$ ) as a safe over-approximation of the length of CGS messages, and they are broadcast to a range of  $d = 100m$ . This puts the cost of one CGS message at around  $1.5 * 10^{-4} J$ ; sleeping nodes have to send two, while awake nodes have to send three such messages.

While a single set of parameters cannot predict the behaviour of a broad range of sensor networks, running simulations with realistic data samples improves the credibility of the conclusions. Table I summarizes our estimates of energy consumption parameters.

Table I  
ESTIMATED ENERGY PARAMETERS

$E$	$F_A$	$F_S$	$C_A$	$C_S$
$10^4 J$	$4.5 * 10^{-4} J$	$3.0 * 10^{-4} J$	$10^{-2} W$	$10^{-5} W$

## V. SIMULATION TECHNIQUE

To observe the behaviour of a WSN without actually building it, a simulator is required. Network simulators (e.g., NS2, J-Sim, OPNET) usually require a long development process to adapt to a protocol, and further efforts each time the same protocol is examined in a different way. Therefore we opted to implement the simulator over a graph transformation system instead, which allowed us to specify the behaviour of CGS in a formal, declarative way. Our simulator was implemented in VIATRA2, a model transformation framework primarily used for software engineering purposes. The motive

of this decision was twofold. First, the graph transformation formalism provides an easy way to capture complex graph-like relationships and transitions. Additionally, the formal and declarative nature of the behaviour specification can be capitalized upon: most of it will be reusable in a stochastic simulation, like the one in [14], or a more thorough analysis based on state space exploration; also a high level service/fault model can be introduced in a heterogeneous sensor environment where different measurement objectives arise (a planned future research step).

The VIATRA2 framework, an Eclipse subproject [15] developed at department MIT of the Budapest University of Technology and Economics (BME), aims primarily at providing efficient model transformation, which is a key step in model based software engineering. The descriptive power of the mathematical formalisms and techniques (graph transformations, abstract state machines) utilized by the transformation language [16], however, makes the tool suitable for a broader range of research areas, for example simulation of domain-specific models. Also the underlying pattern matching mechanism makes VIATRA2 an efficient tool for simulating problems with many local state changes and complex pattern matching inbetween [17]. Simulating sensor network models is merely one possible utilization of VIATRA2.

We have implemented a VIATRA2-based discrete time simulator to model, demonstrate and measure the dynamic behaviour of CGS. The development of the simulator consisted of the following steps:

- A metamodel was defined over the elements of the simulation: sensors, energy, states, regions, covering, messages, etc. The state of the sensor net was captured as a graph formed by elements of this metamodel.
- Graph patterns were specified to capture conditions and relations of simulation elements. For example, one graph pattern identifies neighbors of a sensor sharing a certain common region and having lower drowsiness. Relying on efficient pattern matching algorithms, this pattern can be used to decide whether a particular sensor can afford to go to sleep.
- Graph transformation rules were defined to describe elementary actions that change the model, for instance broadcasting messages to neighbor sensors.
- The control flow of the entire simulation, most notably the sequence of CGS steps, was captured in the transformation language of VIATRA2, based on the mathematical notion of abstract state machines.

## VI. SIMULATION RESULTS

### A. Ring Topology

The first measurement is carried out on a synthetic ring topology. 10 sensors are arranged into a ring, each neighboring 4 sharing a common region; the goal is 2-coverage. Because of the regularity of the problem, the optimal solution is easily obtained: every other sensor should stay awake for the first period (thereby providing exact 2-coverage for all

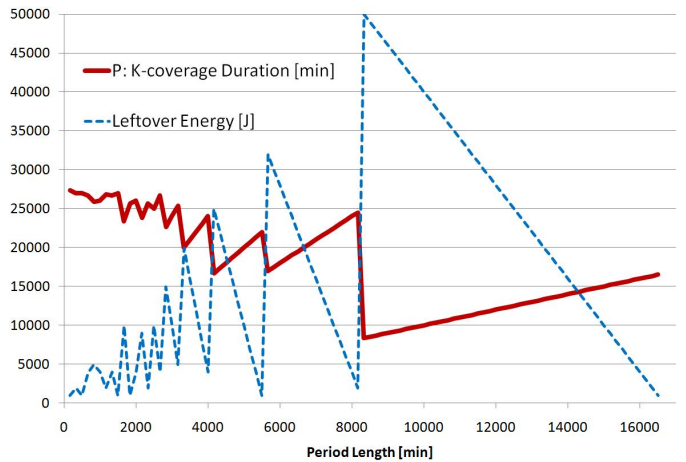


Figure 2. Ring Topology Measurement Results (10 sensors)

regions), while the rest are asleep, and then switch roles at the second period; thus the optimal lifetime of the network is approximately twice the awake lifetime of an individual sensor. We used the realistic energy parameters introduced in IV-C; this puts the lifetime of one sensor ( $E/C_A$ ) at one million seconds or cca. 16700 minutes. The actual CGS simulation was carried out with period lengths ranging from 1% to 99% of this lifetime. The performance metric  $P$  (the total duration of providing K-coverage for the entire network) is plotted against the chosen period length in Figure 2 (continuous line).

It is apparent from the figure that contrary to the superficial estimation in Section IV-B, the performance of the CGS sensor network is not monotonous with respect to  $T$ ; it admits a much more complex behaviour, and remains far from the optimum (cca. 33300 minutes, see above) except for small values of  $T$ . To explain the sharp discontinuities, we have to recall that according to the CGS algorithm (see Section III) only sensors with at least  $E_A$  energy left can contribute to the sensing. Sensors with less energy become permanently inactive and their leftover energy can be considered lost. Therefore if the period length is too large for sensors to remain awake for two periods (and asleep for two periods while others are awake), then they can only stay awake for one period, and the rest of the energy (up to half of the total) is lost. On the other hand, if the period length is just small enough so that two awake periods fit in (cca. 500 000 seconds), then most of the energy can be utilized, and the lifetime of the network is significantly longer. A similar singularity is present between two or three awake periods, between three or four, etc., and also one between zero or one awake periods. For shorter period lengths (enough for several thousand periods), the number of asleep periods can also factor in, making the performance even less predictable. The total amount of leftover energy of sensors at the end of the simulation is plotted on the graph with a dashed line; it is easy to see a connection between leftover energy and performance. As the potential amount of leftover energy is upper bounded by  $E_A$ , it diminishes by choosing smaller period lengths, therefore even the local minima are

less disadvantageous at smaller ranges of  $T$ .

On the other hand, one would expect that the performance at local maxima should increase by the period length, as larger  $T$  values mean a smaller number of periods, fewer messages, thus  $F_A$  and  $F_S$  cause less overhead. However, this turns out to be false: the measurements show that the performance is significantly worse at larger values of  $T$ . This is a consequence of CGS being a distributed algorithm, that can only take into account local information. In this particular example, the global optimum would be a regular awake-asleep-asleep-asleep-awake-... configuration of the sensors, alternating every other round. This symmetry can be easily broken by local decisions; while most of these decisions are still locally optimal, they just do not connect properly to form a regular pattern on the whole ring. For example, awake-awake-asleep-asleep-awake-awake-... provides strict 2-coverage for some regions, but the ring size is not divisible by 4, therefore it has to contain four adjacent awake sensors at some point. Such symmetry breaking ultimately results in overcovering and thus wasted energy and the premature depletion of a few sensors. In this experiment, the majority of the network is actually still 2-covered for the projected lifetime, but our goal is *complete* 2-coverage, which fails much earlier due to some sensors having spent disproportionately more energy than others. This explanation of the energy deficit is supported by evidence: if CGS is artificially hinted to choose the optimal awake partition, the deficit disappears and optimal performance is obtained.

Our observation is that for shorter period lengths, the network has more CGS periods at its disposal, therefore more opportunities to adaptively reorganize the set of awake sensors and compensate for the inequal distribution of energy. Even though some energy is wasted each round due to the overcovering (inevitable for a distributed protocol), this energy deficit is evenly distributed among sensors, which helps to maintain K-coverage as long as possible. This energy redistribution effect (actually caused by the the increased drowsiness of sensors with energy deficit) is one of the main benefits of CGS. We have also conducted measurements with larger ring topologies (e.g., 50 sensors) and different parameter settings; all experiments confirmed that the local maxima of the performance generally increase towards small values of  $T$ , except for extremely short periods (not depicted on the figure) when the communication overhead becomes significant due to the very large number of periods.

### B. Ring Topology, Short Periods

We have also conducted a second experiment with the same topology to inspect small values of  $T$ . Figure 3 shows how CGS behaves for  $T$  between 0.1s and 10s. It is easily seen that for extremely short period lengths, the fixed periodical communication cost become a dominant waste of energy, seriously impacting the lifetime of K-coverage. However, if  $T$  is sufficiently large (at least a couple of seconds with our data), then this effect can be neglected, and CGS has consistently good performance.

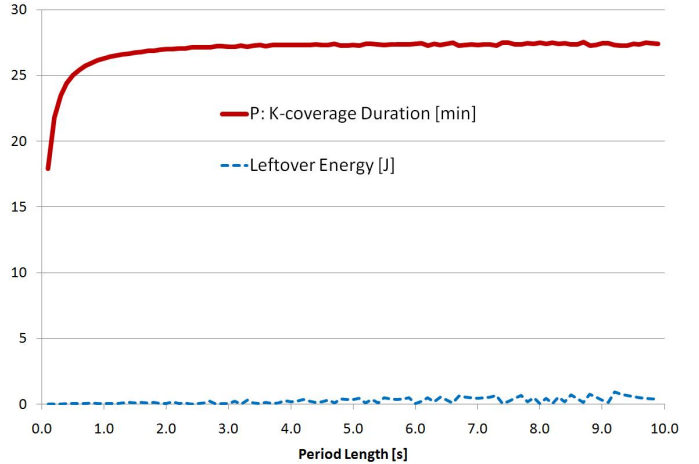


Figure 3. Ring Topology, Reduced Energy, Short Periods

Note that for such short periods, a simulation run could last several million periods for every single value of  $T$ . In order to plot this graph in a reasonable time, we have downscaled the battery capacities (and thus the resulting performance) by a factor of 1000. We believe this change has little effect on the overall shape of the performance graph, as the simulation still lasts for several hundred or thousand periods. But it should be noted that relative effects of leftover energy (the slight fluctuation) are consequently upscaled by a factor of 1000 on this graph; they are actually even less noticeable, and the graph should appear even smoother.

### C. Mesh Topology, Fair Random Placement

The behaviour of sensors placed randomly in a 2D area is much less predictable than the ring topology, therefore the importance of simulation over analytical planning is increased.

This simulation involved scattering 144 sensors with 50m sensing radius each in a 300m by 300m square territory, of which the central 200 by 200 square meters are the observed area that should be 3-covered. Thus the scattering area dominates the observation area by a frame as wide as the observation radius, as depicted on Figure 4. This ensures that the corners of the observation area are covered by the same expected density as the central areas; on the other hand, a small number of sensors (e.g., 8 or 10) will go to waste by being deployed entirely outside. After the placement of sensors, regions are formed with a granularity of 10m.

If sensors were placed with a uniform distribution, some areas would likely be disproportionately underdosed (while other could end up significantly overdosed), making K-coverability impossible. Placing sensors on a regular grid, however, would be unrealistic to deploy. Therefore we have opted to strike a balance and model the deployment in a sufficiently random but spatially fair way. The scattering area is divided into square sectors (6 by 6 sectors in our simulation) and a fixed amount of sensors (4 in our case) are deployed in each sector, with uniform random placement within the sector.

Figure 5 shows the result of a mesh topology simulation run.



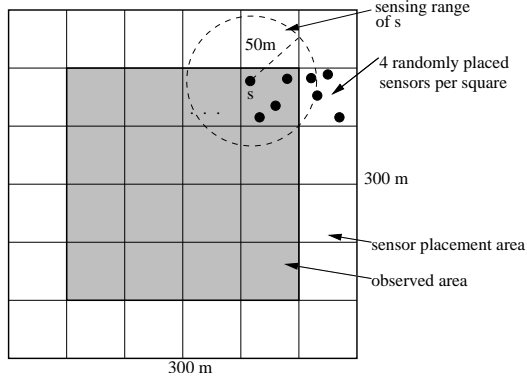


Figure 4. Illustration of Mesh Topology

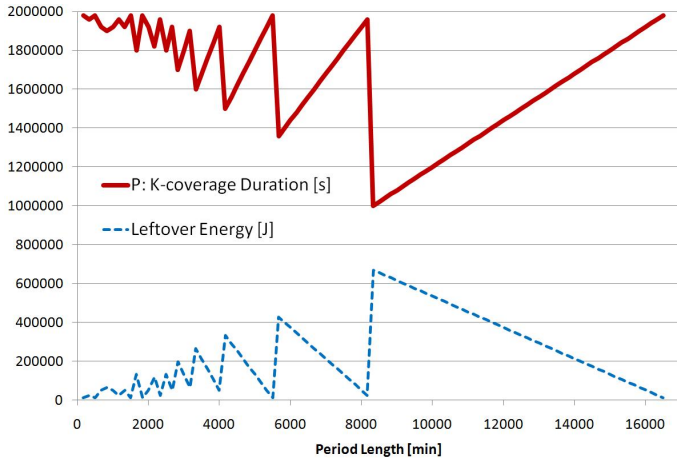


Figure 5. Mesh Topology Measurement Results (134 sensors, 373 regions)

Due to the randomness of the topology generation, subsequent runs yielded different results, but they all shared the same basic shape and characteristics. This more realistic topology confirms the observation that choosing small period lengths is beneficial due to greater CGS adaptivity and a smaller risk of leftover energy. Furthermore, a priori analytical derivation of the local optima is completely unfeasible in a random scenario, and optimization after the placement would break the distributed nature of the system; thus a general observation (choose small period lengths!) is more useful than formulae here.

## VII. CHOICE OF PERIOD LENGTH

Section VI-A and VI-C suggested choosing a small period length. Smaller values of  $T$  are also preferred in order to be able to quickly adapt after sensor failures or other events. However, minimizing  $T$  should not be taken to the extreme. Section VI-B shows that for very short values of  $T$ , fixed periodical costs ( $F_A$  and  $F_S$ ) dominate time-proportional costs ( $C_A$  and  $C_S$ ), and a large portion of energy will be wasted for communication. Additionally,  $T$  cannot be shorter than  $T_e$ , the length of the election process at the beginning of each period. Therefore our proposition is to choose the smallest suitable

period length for which fixed cost are already negligible. Our task is now to find this ideal  $T$ .

To find a  $T$  where the fixed costs are negligible, we have to recall the combined average energy consumption rate  $E_T$  and (6). We want to limit the portion of energy wasted for periodical fixed costs. Specifically, we choose a small value  $\lambda$  (e.g.,  $\lambda = 0.01$ ) with the goal that

$$\lambda * (p_A * C_A + (1 - p_A) * C_S) \geq \frac{p_A * F_A + (1 - p_A) * F_S}{T} \quad (11)$$

This goal can be satisfied by the following subgoals:

$$\frac{\lambda}{2} * (p_A * C_A) \geq \frac{p_A * F_A}{T} \quad (12)$$

$$\frac{\lambda}{2} * (p_A * C_A) \geq \frac{(1 - p_A) * F_S}{T} \quad (13)$$

It is important to point out that we used  $C_A$  for the upper bound in both cases, as it can be several orders of magnitude larger than  $C_S$ , and also the latter has coefficient  $1 - p_A$ , which would be harder to find a lower bound for. The two subgoals impose the following constraints on  $T$ :

$$T \geq \frac{2 * F_A}{\lambda * C_A} \quad (14)$$

$$T \geq \frac{2 * F_S * \frac{1 - p_A}{p_A}}{\lambda * C_A} \quad (15)$$

The problem is that the value of  $p_A$  is not known a priori; this will be addressed by the following argument. Let  $n$  be the number of sensors and  $n_A$  the average number of awake sensors during the periods of K-coverage. Clearly

$$\frac{n_A}{n} = \frac{N_A}{N} = p_A \quad (16)$$

If  $A_{obs}$  is the area under observation, and  $R$  is the measurement radius of the sensors, then K-coverage requires that the area is covered K times, implying

$$n_A * R^2 * \pi \geq K * A_{obs} \quad (17)$$

Thus

$$p_A = n_A/n \geq \frac{K * A_{obs}}{n * R^2 * \pi} = p_A^0 \quad (18)$$

where  $p_A^0$  (basically the reciprocal of the area overdosing factor) can be a priori known.

We have established an a priori lower bound  $p_A^0$  for  $p_A$ , which gives an upper bound for  $\frac{1 - p_A}{p_A}$ , thereby allowing us to specify a stricter condition than (15) to ensure proper choice of  $T$ :

$$T \geq \frac{2 * F_S * \frac{1 - p_A^0}{p_A^0}}{\lambda * C_A} \quad (19)$$

$$\geq \frac{2 * F_S * \frac{1 - p_A}{p_A}}{\lambda * C_A} \quad (20)$$

Finally, this gives us the following a priori decision method to ensure that fixed cost can be practically neglected, also

observing the lower and upper limits:

$$T = \min(T_{max}, \max(\frac{2 * F_A}{\lambda * C_A}; \frac{2 * F_S * \frac{1-p_A^0}{p_A^0}}{\lambda * C_A}; T_e)) \quad (21)$$

Of course, if the resulting  $T$  is larger than  $\lambda * T_{max}$ , then the result could not be considered a short period after all, and leftover energy could dominate energy wasted on communication, making this argument pointless. In these cases,  $\lambda$  is too strict, a larger value should be chosen.

For example, with  $\lambda = 0.01$  and the mesh topology of Section VI-C:  $T_{max}$  would be approximately  $10^6s$ ,  $p_A^0 = 0.114$ , and the first two lower limits of  $T$  turn out to be  $9s$  and  $23.31s$ .  $T = 24s$  is therefore likely a good choice of period length, and enough to include  $T_e$ . For the ring topology of Section VI-A:  $p_A^0 = 0.5$ , and the lower limits are  $9s$  and  $3s$ , respectively; therefore  $9s$  is predicted to be a safe choice (justified by Figure 3).

### VIII. CONCLUSIONS AND PERSPECTIVES

We have conducted theoretical and experimental analysis of the choice of period length for the CGS scheduling algorithm.

Analytical a priori (*i.e.* before deployment) computation of the optimum of network lifetime, even by neglecting communication costs and power consumption of asleep sensors, is only easy for synthetic topologies like the ring arrangement, and completely infeasible for realistic, randomly scattered deployment. But even if a global, non-distributed decision of period length could be made after the scattering of sensors in an arbitrary topology, it would likely be not worth the risk. As the performance function is non-continuous, the computed period length is easily tipped over into a local minimum due to small measurement errors. Therefore we recommend that period length is not chosen based on exact analytical calculation of local extrema, but based on our proposal that takes into account the potential risk of landing in a local minimum.

We propose that small  $T$  values should be chosen, just long enough so that the communication overhead can be neglected. Both the minima and the maxima have larger performance at smaller values of  $T$ , so this is a safe choice if we cannot predict whether a given  $T$  is a local maximum or minimum, which is the case for most topologies. Short periods also result in better adaptability with respect to energy imbalance or external events. Section VII elaborated the proposed method of choosing  $T$ , and yielded the formula (21). This reasoning is supported by measurements that show the same basic characteristics over different topologies.

The problem addressed in this paper can be relevant for other distributed periodic WSN schedulers as well. [18] has similar constraints and observations about the period length of a different protocol. We believe that analogous arguments can be constructed to optimize the period length of other scheduling protocols mentioned in Section II-C.

### ACKNOWLEDGEMENT

This work was supported by the Hungarian–French Inter-governmental S&T Cooperation Program under the "Methods

and algorithms to enhance the dependability of services in sensor networks" (F-Egide-PHC-19476SH, FR 6/2008)

### REFERENCES

- [1] C.-Y. Chong and S. P. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003. [Online]. Available: <http://dx.doi.org/10.1109/JPROC.2003.814918>
- [2] N. Ahmed, S. S. Kanhere, and S. Jha, "The holes problem in wireless sensor networks: a survey," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 9, no. 2, pp. 4–18, 2005.
- [3] S. Kumar, T. H. Lai, and J. Balogh, "On k-coverage in a mostly sleeping sensor network," in *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, 2004, pp. 144–158.
- [4] M. Cardei, M. T. Thai, Y. Li, and J. Wu, "Energy-efficient target coverage in wireless sensor networks, handbook of sensor networks," in *IEEE INFOCOM*, 2005, pp. 1974–1984.
- [5] C.-F. Huang and Y.-C. Tseng, "A survey of solutions to the coverage problems in wireless networks," *Journal of Internet Technology*, vol. 6, no. 1, pp. 1–8, 2005.
- [6] G. Simon, M. Molnár, L. Gönczy, and B. Cousin, "Robust k-coverage algorithms for sensor networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, August 2008.
- [7] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," *Ad Hoc Netw.*, vol. 6, no. 4, pp. 621–655, 2008.
- [8] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration for energy conservation in sensor networks," *ACM Trans. Sen. Netw.*, vol. 1, no. 1, pp. 36–72, 2005.
- [9] F. G. Nakamura, F. P. Quintão, G. C. Menezes, and G. R. Mateus, "An Optimal Node Scheduling for Flat Wireless Sensor Networks," in *ICN : International conference on networking*, 2005, pp. 475–482.
- [10] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *Wirel. Netw.*, vol. 8, no. 5, pp. 481–494, 2002.
- [11] M. Liu and C. Hsin, "Network coverage using low duty-cycled sensors: Random and coordinated sleep algorithms," 2004. [Online]. Available: [citeseer.ist.psu.edu/liu04network.html](http://citeseer.ist.psu.edu/liu04network.html)
- [12] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Optimizing sensor networks in the energy-latency-density design space," *IEEE Transactions on Mobile Computing*, vol. 1, no. 1, pp. 70–80, 2002.
- [13] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for," 2002. [Online]. Available: [citeseer.ist.psu.edu/heinzelman02applicationspecific.html](http://citeseer.ist.psu.edu/heinzelman02applicationspecific.html)
- [14] P. Torrini, R. Heckel, and I. Ráth, "Stochastic simulation of graph transformation systems," in *Fundamental Approaches to Software Engineering (FASE)*, 2010, submitted.
- [15] *VIATRA2 Project Page*, <http://www.eclipse.org/gmt/VIATRA2/>.
- [16] D. Varró and A. Balogh, "The model transformation language of the VIATRA2 framework," *Science of Computer Programming*, vol. 68, no. 3, pp. 214–234, October 2007.
- [17] G. Bergmann, Á. Horváth, I. Ráth, and D. Varró, "A benchmark evaluation of incremental pattern matching in graph transformation," in *ICGT*, 2008.
- [18] T. Yan, Y. Gu, T. He, and J. A. Stankovic, "Design and optimization of distributed sensing coverage in wireless sensor networks," *ACM Trans. Embed. Comput. Syst.*, vol. 7, no. 3, pp. 1–40, 2008.