Active time scheduling for rechargeable sensor networks

Volodymyr Pryyma, Damla Turgut and Ladislau Bölöni
School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816–2362
Email: vpryyma,turgut,lboloni@eecs.ucf.edu

Abstract

Recent progress in energy harvesting technologies made it possible to build sensor networks with rechargeable nodes which target an indefinitely long operation. In these networks, the goal of energy management is to allocate the available energy such that the important performance metrics, such as the number of detected threats, are maximized. As the harvested energy is not sufficient for continuous operation, the scheduling of the active and inactive time is one of the main components of energy management. The active time scheduling protocols need to maintain the energy equilibrium of the nodes, while considering the uncertainties of the energy income, which is strongly influenced by the weather, and the energy expenditures, which are dependent on the behavior of the targets. In this paper, we describe and experimentally compare three active time scheduling protocols: (a) static active time, (b) dynamic active time based on a multi-parameter heuristic and (c) utility-based uniform sensing. We show that protocols which take into consideration the probabilistic models of the energy income and expenditure and can dynamically adapt to changes in the environment, can provide a significant performance advantage.

Key words: rechargeable, energy harvesting, sensor network

1. Introduction

The deployment model of sensor networks frequently makes access to external energy resources impossible. One of the earliest proposed application areas of sensor networks was battlefield surveillance: a set of disposable sensor nodes with a finite energy source are deployed randomly over an area. The finite lifetime of the network is a given, but seen as a necessary compromise. Energy management techniques such as energy aware routing and active time scheduling can be used to extend the useful lifetime of the network.

In recent years, research into techniques of energy scavenging (such as Srivastava et al. [10], Paradiso and Starner [19], and Raghunathan and Chou [21]) made the development of rechargeable nodes possible. This, however, does not imply an unlimited energy supply, because the rate of energy consumption is usually greater than the recharging rate. In these settings, the lifetime of the sensor network can be possibly infinite (although in practice it will be limited by the lifetime of components). The goal of the energy management is to maximize the utility of the sensor network under the conditions of a finite energy harvest.

In this paper, we propose several approaches for active time scheduling based energy management for rechargeable sensor networks. We will assume that energy is acquired through a solar cell, thus it is dependent on the weather (more exactly, the available sunlight). We assume that the goal of the sensor network is to observe mobile nodes (threats) traversing the sensor field. The more threats a node detects and the better it observes them, the higher its utility. If a node runs out of its energy hours before dawn, it might miss a number of threats. On the other hand, if the node arrives at dawn with a large energy reserve, most of the time it means that it lost utility by either missing threats during its longer-than-necessary *scheduled* inactive times, or that it did not perform a thorough enough observation of the detected threats as it became inactive while the threat is still in the sensing range. Of course, it might also happen that the energy income was larger than the expenditure, even after the active time was extended to the full time interval and all the threats were fully covered - or there were no threats to cover. In this case, of course, no adjustments are needed.

Nondeterministic factors in the environment add additional challenges to the problem. The energy harvest depends on the weather (through the available sunlight), which shows seasonal and day-to-day variation. The energy consumption might be dependent on the number and temporal distribution of the threats. Threats might be arriving during day-time or night-time, one-by-one or clustered in tight groups. The speed and movement path of the threats affect the amount of time they stay in the sensing range of a given node.

The rest of this paper is organized as follows. We begin by summarizing previous work in Section 2. In Section 3, we define our problem, including the networking and deployment architecture, the energy harvesting model and the threat model. In Section 4, we present three, progressively more complex active time scheduling schemes which will be experimentally compared in Section 5. We conclude in Section 6.

2. Related Work

The idea of rechargeable sensors has been around for some time. Paradiso and Starner [19] discuss several energy scavenging technologies for mobile and wireless electronics. A number of power management issues for energy harvesting embedded systems are addressed by Raghunathan and Chou [21] and Kansal and Srivastava [10]. Jiang et al. [6] describe the hardware aspects of establishing perpetual environmentally powered sensor networks. Performance tasking as well as several power management systems for rechargeable sensors are presented by Kansal et al. [7, 8, 9].

Byers and Nasser [3] propose a utility-based decision-making process to maximize the lifespan of a sensor network. This decision-making process allows the sensor nodes to change their roles over time and dynamically adjust the routing paths to balance the energy consumption in the network. Biand et al. [1] and Padhy et al. [18] show different utility-based mechanisms for managing sensing and communication in large scale multi-agent sensor networks. Nama et al. [14] propose a framework for cross-layer design across transport, network, and link layers to find the optimal set of resource allocation such that network utility and lifetime is maximized.

Kar et al. [11] introduce a dynamic node activation scheme, which is specifically designed for networks with rechargeable sensors. At any given time, a rechargeable sensor node is in one of the following states: active (normal operation), passive (battery recharging), or ready (waiting for job assignment). The dynamic activation scheme is distributed, requiring only local state information, and performs close to the global optimum.

The adaptive duty cycling algorithm, introduced by Hsu et al. [5], allows rechargeable nodes to autonomously adjust their duty cycle based on the energy availability in the environment. Zhu and Ni [25] propose a probabilistic wakeup protocol which reduces the duty cycle of individual sensors, while exploiting the dense deployment of sensor networks. The system ensures that the delay of detecting an event is statistically bound. A scheduling algorithm that relies on the battery capacity of the sensors is presented by Moser et al. [12] and a dynamic reconfiguration scheme for rechargeable sensor networks is proposed by Nahapetian et al. [13]. Chen and Fleury [4] present a unique coloring scheme that integrates duty cycling and collision avoidance into a single schedule of node activities. Zhu et al. [24] focus on energy-efficient event detection in wireless sensor networks and develop a localized algorithm to determine sensor wakeups. Additionally, Premkumar and Kumar [20] propose a scheduling scheme where only a minimal number of nodes are active to minimize energy consumption.

Zafar and Corkill [23] propose a two-phase scheme for estimating a solar energy harvesting model in situated agents. In the pre-deployment phase, the agents learn as much as possible about their environment patterns. This greatly reduces the amount of learning that each agent has to perform during actual deployment. Thus, once in the deployment phase, the agents simply complete their harvesting model.

3. System Architecture and Environmental Models

In this section, we describe the overall architecture of the considered system. This includes (a) the sensor network architecture: the types and roles of the deployed components and the networking protocols used for data transfer, (b) the energy consumption model of the sensor nodes, (c) the energy harvesting model, and (d) the threat model.

3.1. Autonomous network organization

The sensor network considered in our work is based on the autonomous network architecture (ANSWER) proposed by Olariu et al. [15]. ANSWER consists of a large collection of sensor nodes, whose primary actions include continuous environment monitoring. The sensors also have low power data processing and short range wireless communication capabilities. In addition to the sensor nodes with limited energy, computational and data processing capabilities, the ANSWER architecture utilizes stationary or mobile Aggregation and Forwarding Nodes (AFN) to organize the sensors in their vicinity. AFNs have the ability for long range communications, and have an infinite supply of energy.

Each AFN organizes the neighboring sensor nodes into a dynamic coordinate system centered on the AFN. This coordinate system allows for dynamic network reconfiguration and provides a simple and low-cost clustering scheme for organizing the sensor nodes. The dynamic coordinate system divides the surrounding area into *coronas* and *wedges*. Coronas are concentric circles of increasing radii that are centered at the AFN. All coronas have the same width, which is set to be slightly less than the transmission range of the sensor nodes. Wedges are equiangular dividers that originate at the AFN and extend to its full transmission range. The wedges are established using directional transmission. This coordinate system is dynamic in nature because it can be easily re-established in order to accommodate changes in network topology.

The ANSWER architecture provides the means for activating selected subsets of sensor nodes at any given time, so that energy is conserved by the sensors that do not have to be involved in the current sensing process. To accomplish this, the individual sensor nodes are activated based on a coloring scheme. Using the signal strength readings obtained during the establishment of the network, each node is assigned a specific color. Thus, the corona segments are further subdivided into a number of color sets. The color sets are numbered in the same order in each corona, partitioning the entire network into a set of color graphs, such that all the sensors in any one graph are represented as vertices with the same color, and any two vertices within the transmission range of each other are connected by an edge.

Once the network is established, an ANSWER network can be used as an environmental monitoring system by collecting the data acquired by the AFN nodes at a sink. In a more interesting application pattern, a trusted mobile node moving in the sensor instrumented area can improve its environmental awareness by directly communicating with the AFNs.

This can be used in several potential applications:

Avoiding threats. In this application, we assume a set of mobile threat nodes which are observed by the network. A trusted mobile node is trying to move from a source to a destination position while trying to avoid the threat nodes. In our case, a threat is considered to be an enemy vehicle capable of destroying the mobile node. The AFNs, in turn, perform task scheduling for the sensors under their control. If a threat is detected by a sensor, the threat's approximate position is first reported to an appropriate AFN, and then relayed to the mobile node. Based on the information reported by the AFNs, the mobile node can adjust its path in order to avoid the threats.

Intercepting targets. In this scenario, which corresponds, for instance, to a border patrol application, the moving trusted node is asking the AFNs for information which allows it to intercept a moving target. While the networking architecture remains the same, the flow of information and the

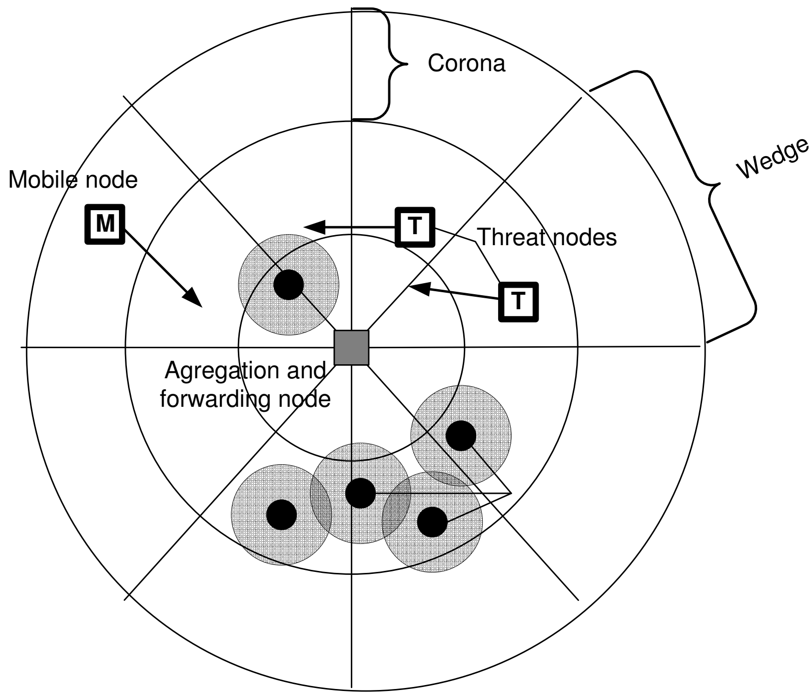

Figure 1: The ANSWER architecture.

associated active time scheduling challenges are different. In the previous case, the focus of the network is on the trusted node and its predicted path. Nodes which are far from the trusted node will not be activated. In the new setting, the network needs to perform an overall search for possible targets. Once the trusted node chooses a target, the network needs to follow the selected target and forward information to the trusted node.

This architecture is illustrated in Figure 1. More details on the ANSWER architecture and its functionality can be found in [15, 16, 17, 22].

3.2. Energy consumption model

Let $E_a(t)$ represent the available energy of a rechargeable sensor node at time t (measured in submultiples of joules). Let $d(t)$ represent the energy consumption (power) at time t, $r(t)$ represent the harvested energy, and $e_{leak}(t)$ represent the constant energy leak. These values are measured in (submultiples of) watts (joules/second). We have:

$$E_a(t) = \int_0^T (r(t) - d(t))\, dt - \int_0^T e_{leak}(t)\ dt \quad (1)$$

The energy consumption of a node depends on its current state.

Active state: when the node is actively communicating and sensing it consumes energy at the rate given by the active power level, P_{active}.

Obviously, the assumption that the energy consumption during the active phase is constant is a simplification. The node might have various energy levels depending on which sensors are turned on, and how frequently they are sampled. Moreover, the cost of transmission is typically larger than the one of sensing. The cost of transmission might also depend on the transmission power, which, in turn, depends on the physical location of the node and the distance to the neighbors.

Nevertheless, these variations will be amortized over time, unless the node radically changes its active mode, such as renouncing the use of one of the sensors or changing its physical location. Thus, P_{active} needs to be understood as an average value.

Inactive state: when the node is in the inactive (standby) mode, it consumes energy at the rate given by the inactive power level, $P_{inactive}$.

Wake-up: the movement from the inactive to the active state requires a transition which requires a relatively high energy consumption for a short period of time. The power consumption of the wake-up phase is very high, but it sums up to a fixed amount of consumed energy, E_{wakeup}. Thus, the energy consumption of the active phase has a constant and a phase-variable component.

3.2.1. Energy harvesting model

As our model assumes nodes which use solar energy for energy harvesting, the amount of energy collected is proportional to the solar radiation energy falling on the solar panels. This energy has both seasonal and day to day variations. This information is relatively easy to acquire even before deployment. The National Solar Radiation Data Base [26] maintains hourly statistics for all the major airports in the US. Figure 2 shows three representative examples of the measured global diffuse solar radiation $r(t)$ at the Orlando International Airport. Note that this information was obtained at a horizontal, unobstructed surface. In a practical deployment, the existence of various obstructions can change the shape of these diagrams. For instance,

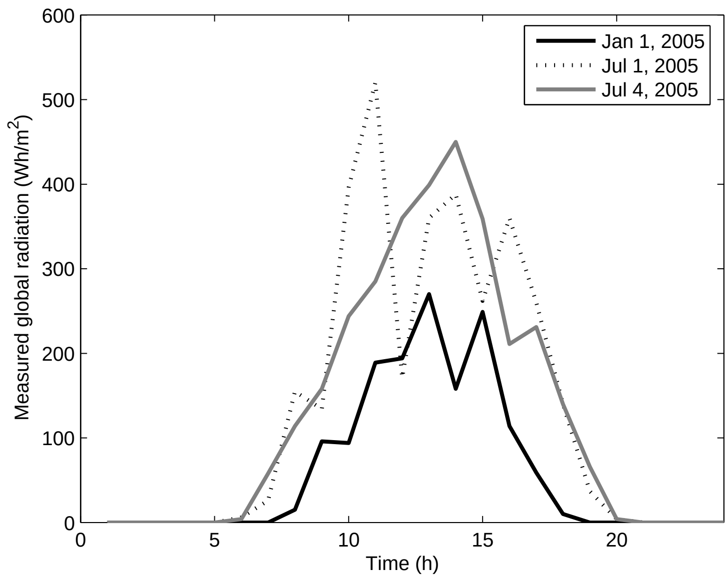

Figure 2: Solar radiation values at the Orlando International Airport for a winter day (January 1st, 2005), and two summer days (July 1, 2005 and July 4 2005) based on the National Solar Radiation Database.

a sensor with a solar panel mounted on the east side of a tree would experience significantly lower afternoon radiation values. On the other hand, a device which can orient its panel to follow the sun can obtain higher values than in the specified graph. Naturally, due to the limited energy efficiency of the solar panels, the actual harvested energy is only percentage of the solar radiation.

Predicting the shape of the energy harvest curve can be an important factor in the decision making process of the sensor nodes. Over the long run, the node needs to be in a dynamic energy equilibrium. The energy level will inevitably decrease during night time. It may even be possible for a node to consume more energy than it produces over a 24 hour period. In the long run, however, the node needs to equalize its energy consumption and energy harvest, and it needs to guarantee that the energy never becomes zero (even temporarily). Note that even for a node which has an overall positive energy budget, it is possible to run out of energy during the night. Such an extended inactive period leads to a massive loss of utility.

3.2.2. Observation model

The scenarios we consider assume that the sensors are used to detect rare events - rather than monitoring continuous phenomena. In the latter case, for instance, for a sensor which collects temperature or humidity readings, the sensor has little leverage in managing its energy budget: it needs to wake up with some regularity to make measurements. In the case of rare events however, a period in which an event is observed normally comes with a larger energy expenditure compared to a quiet period, as the node needs to communicate additional information.

Another factor in the energy expenditure is the "don't lose the target" heuristic. This common sense principle requires that a node will not become inactive during the tracking of a target in its sensor range even if its active time scheduling would require it. This is justified by the rarity of target sightings and the high importance of the targets. All the algorithms we consider in this paper implement this heuristic. As a note, a more sophisticated system might allow a sensor node tracking the target to become inactive, provided that one or more sensor nodes have "acquired" the target. However, our algorithms do not implement these mechanisms.

The additional cost of observations require the active time scheduling algorithm to take into consideration the number of the observations a node is likely to make. A node which expects to track 100 targets during the day needs to budget differently compared to a node which expects to track 10 targets a day.

The observation model is naturally stochastic in nature and can be learned from historical data as well as a priori considerations. Perimeter control systems expect most days without intruders. On the other hand, illegal border crossings in many areas are frequent and subject to relatively predictable seasonal variations. Battlefield sensing involves a much higher uncertainty, but even here, certain assumptions can be made about the maximum number of targets.

4. Three Schemes For Active Time Scheduling

4.1. Static active time approach

In this case, the schedule contains a regular alteration of the active and inactive intervals for the complete duration of a day. During a day $T = 24h$, there will be n such cycles, where $n = T/(t_{inactive} + t_{active})$.

The total energy used by the node will be:

$$D(t) = \int_0^T d(t)\ dt = n \cdot (t_{inactive} \cdot P_{inactive} + t_{active} \cdot P_{active} + E_{wakeup})$$

The challenge is to determine the t_{active} and $t_{inactive}$ values. The resulting schedule needs to satisfy a series of conditions. First, assuming that the node starts the day with a certain energy reserve, E_{res}, we require that the remaining energy of the node should never be zero:

$$E_{res} - k \cdot (t_{inactive} \cdot P_{inactive} + t_{active} \cdot P_{active} + n \cdot E_{wakeup}) + \int_0^{k \cdot (t_{inactive} + t_{active})} r(t)\ dt > 0$$

Note that this can be achieved by a sufficiently high energy reserve at the beginning of the day. If this condition is not satisfied, the node will reach a point where it cannot turn itself on. If this happens, a long period of inactivity will result.

Second, the overall energy budget of the day should be non-negative:

$$-n \cdot (t_{inactive} \cdot P_{inactive} + t_{active} \cdot P_{active} + E_{wakeup}) + \int_0^T r(t)\ dt > 0$$

If this condition is not satisfied the node will start the next day with a smaller energy reserve. As the rechargeable sensor network is designed for infinite operation length, this will eventually lead to the node exhausting its energy reserve, and thus violating the previous equation, resulting in unplanned inactive periods.

As a note, in these formulas we assumed that the battery capacity is not a limiting factor. If a battery is fully charged, the harvest rate $r(t)$ is limited by the current consumption rate, and the available energy is limited by the battery capacity.

Everything else being equal, the faster alternation of active and inactive time increases observation quality. The main goal of the sensing is to detect targets in the sensing range. A node might miss a target if the target enters and exits the area during an inactive interval. With a shorter inactive interval, the sensor is more likely to catch at least a part of the target's presence in its sensor range. At the same time, a faster alternation of active and inactive times leads to a higher energy consumption, due to the fixed wake-up costs.

Figure 3 illustrates the static active scheduling over the course of one day, where three observations are made at 2am, 4am and 10pm (which trigger the "don't lose the target"). As a note, in Figures 3, 4 and 5, the size

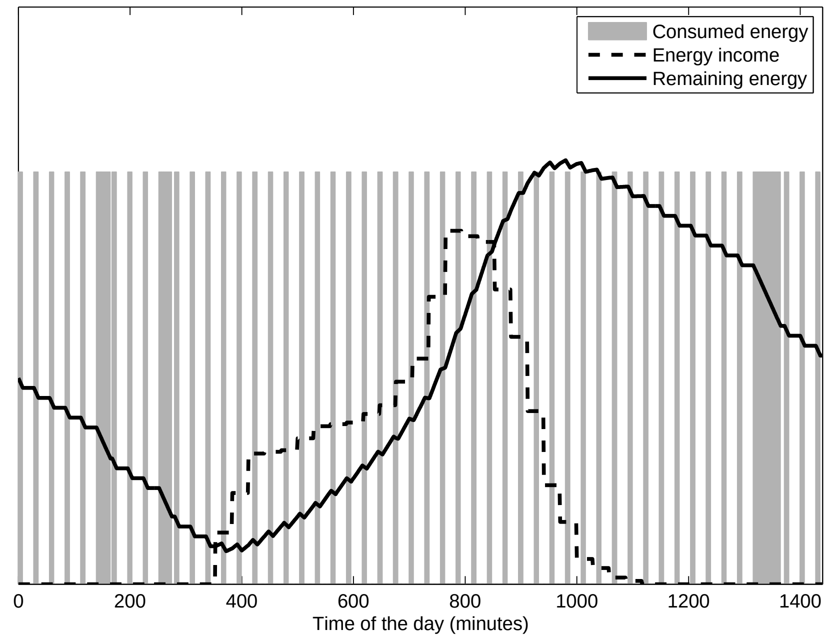

Figure 3: Static active time schedule over the course of a day (1440 minutes): the evolution of the energy income and available energy.

of the inactive time slot was set to 20 minutes to ensure a better readability of the figure. In a real deployment, the alternation of the active and inactive time slots would be faster, but the relative size of active and inactive slots will be the same. The three values represented in these graphs have different dimensionality, thus their absolute values should not be compared against each other. Plotting them on the same axis (time) helps illustrate the interrelationship between their trends.

4.2. Dynamic active time approach based on a multi-parameter heuristic

We have seen that the lack of dynamic adaptation to unexpected events is a significant drawback of the static active time approach. The natural alternative is to make the active time able to dynamically change in response to events by calculating the length of the next active time slot based on the sensor node's knowledge about the world: its energy harvest and consumption models, the observation models, and so on. Many of these models are probabilistic in nature. Unfortunately, sensor nodes are characterized by low computational power and memory; they can neither represent nor calculate overly complex probability distributions. In addition, the cost of such computation can be comparable with the energy consumption savings obtained by a better schedule.

To avoid the need for complex computations, we propose a multi-parameter heuristic, which achieves dynamic scheduling based on a simple, heuristically determined formula, which takes into account various components of the world knowledge of the sensor. These input parameters consist of the length of the previous active time slot, the amount of currently stored energy in a node, the probability of encountering a threat, and the number of one-hop neighbors. The formula for calculating the length of the next active time slot is:

$$T_{active}^{new} = T_{active}^{old} \frac{r^{new}(t)}{r^{old}(t) + r^{new}(t)} + \frac{E(t)}{\alpha \cdot d(t)} + C_{th} + C_n \tag{2}$$

The recharging rates $r^{new}(t)$ and $r^{old}(t)$ change with time to reflect the change in available energy for harvesting in the environment. $E(t)$ is the amount of available energy at time t, while $d(t)$ is the energy consumption rate at time t.

The parameter α depends on the available energy, and is a positive constant if the stored energy in a node is above fifty percent of total capacity, and a negative multiplier otherwise. The parameter C_{th} depends on the probability of encountering a threat. It increases gradually each time a threat is detected and decreases for every time interval during which no threats were observed. This reflects the intuition that the threats might be coming in clustered groups or teams. Finally, C_n is a parameter which increases with the number of one-hop neighbors of the node. Thus, the length of the active time slot will decrease when the node has little stored energy, and increase when the node has a lot of stored energy. The modifiers C_{th} and C_n further impact the length of the active time slot so that the active period is increased if the probability of encountering a threat is high, and decreased if there are many other sensor nodes nearby.

Figure 4 shows an illustration of the dynamic active time slot approach.

4.3. Utility-based active time scheduling

Let us now discuss some of the drawbacks of the previously proposed approaches. The static active time scheduling has the benefit of a consistent

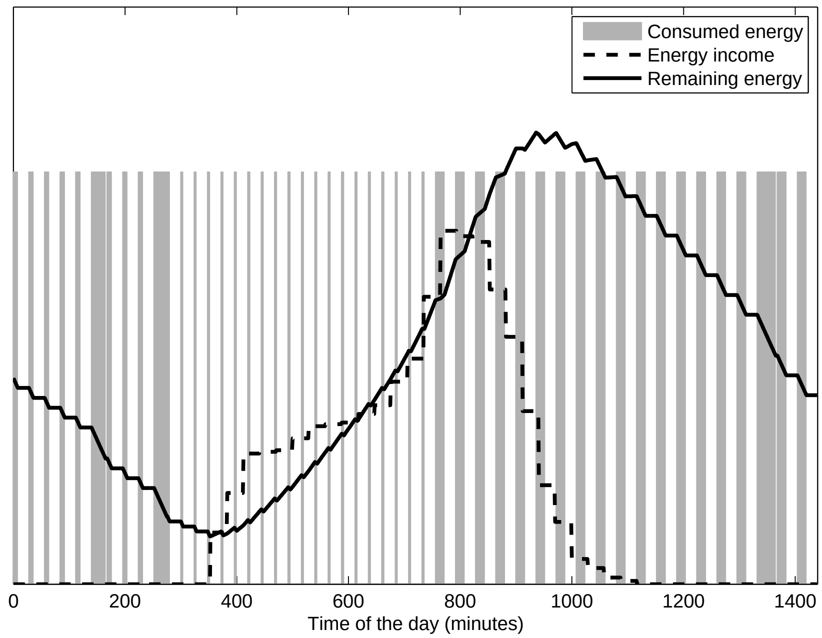

Figure 4: Dynamic active time schedule over the course of a day (1440 minutes): the evolution of the energy income and available energy.

observation schedule. However, if the number of observations are higher than expected, the static active time scheduling approach will consume more than its expected budget, and the node can run out of energy. To avoid this, the static schedule needs to budget its energy very conservatively assuming the largest number of observations. As a result, the static schedule will run an energy surplus almost all the time - but it will not provide the best possible observation ability.

The dynamic active time approach does not suffer from this problem, because if it overruns its energy budget, it will dynamically reduce the length of the of the active slot – thus it will not run out of energy. However, if the original assumptions were bad, it will be necessary to reduce the active time slot to very small values, which would also reduce the ability of the network to detect intruders (even if it is better than a complete shutdown).

In the following, we present a utility optimization based approach to active time scheduling. In the Uniform Sensing Protocol, the sensor nodes first calculate their energy budget at the beginning of each day and night cycle. Once this is accomplished, the length of the active time slot is estimated as

$$T_{active} = \begin{cases} \min\left(T_{active}^{max}, \frac{E(t)}{k(d(t)-2r(t))}\right) & \text{if } d(t) > 2r(t) \\ T_{active}^{max} & \text{otherwise} \end{cases} \tag{3}$$

where $E(t)$ represents the amount of currently stored energy in a sensor node, and $d(t)$ and $r(t)$ are the rates of energy consumption and gain respectively. The intuition behind the multiplier 2 for the $r(t)$ value is that the energy harvest happens both during active and inactive periods, while energy is consumed only during the active periods.

The parameter k represents the number of active and idle slots experienced by a sensor in a single day and night cycle. The number of active slots in a single cycle increases with increasing value of k. However, the length of a single active slot decreases with increasing k.

The parameter k has to be chosen a priori and has to satisfy the following condition:

$$kT_{active} \leq T_{cycle} \tag{4}$$

In case of equality, the node will maintain its energy at the end of the day provided that it never needs to extend its active period due to the presence of threat nodes. The value of k also depends on the probability that a sensor node will encounter a certain number of threats. If a sensor has high probability of encountering only a few threats, then the value of k can be reduced such that active periods are longer but less frequent. On the other hand, if a sensor node has a high probability of encountering many threats, then the value of k can be increased in order to have more active time slots. We assume that the number of threats the node will probably encounter is normally distributed around the average value of previous day and night cycles. We make the assumption that the number of the threats will not exceed the mean plus four standard deviations, which, if our assumptions are correct, will provide a 99.993% confidence.

Once the sensor nodes calculate their respective probabilities of encountering threats, as well as their energy budget, they compute corresponding T_{active} and begin sensing. If a sensor does not encounter any threats, then its T_{active} will remain unchanged throughout the entire day and night cycle. At the onset of a new cycle, each node will recompute its threat probability and energy budget, as well as set a new value for T_{active}.

However, once a sensor encounters a threat, it will extend the length of T_{active} for as long as it can sense that a threat is present (the "don't lose the target" heuristic). Once a threat moves out of the sensing range, the sensor node will recompute its energy budget, based on the currently available energy, and adjust the length of T_{active} such that uniform sensing can be continued for the remainder of the current cycle. This is illustrated in Figure 5. To illustrate the way in which the Uniform Sensing Protocol adjusts to the unexpected events, we have added several long sensing sessions during the middle of the day. We can see that the agent adjusts its active time, and terminates the day with the same energy reserve it began.

A major advantage of the Uniform Sensing Protocol over the static and dynamic active time slot approaches is the fact that uniform sensing does not leave any large openings in the network to be exploited by intruders. In other words, an intruder always has the same chance of being detected, regardless of when it attempts to infiltrate into the network area. This does not hold true for the static and dynamic active period schemes. On the contrary, an intruder has a much higher chance of passing through the network undetected towards the end of the cycle, when most of the sensors have run out of energy, or have drastically reduced their time spent in the active state.

In addition, our protocol is able to adjust to various changes in the environment. For example, if the amount of energy available for harvesting changes due to events such as cloud cover or the nightfall, the sensor nodes will recompute their energy budget taking this into account. Since accurate intruder tracking is very important, the sensors can extend the length of their active time slot for as long as they sense the intruder's presence. However, once the threat moves out of the sensing range, the sensor nodes involved will once again adjust their T_{active} to provide uniform sensing. This is confirmed by our simulation results.

5. Simulation Study

In order to compare the proposed active time scheduling approaches, we have implemented an environment similar to the one described by Olariu et al. [15] in the YAES simulator [2]. In addition to the networking protocols, we have also implemented the energy consumption and harvesting model of the sensor nodes. The energy harvesting model was based on the assumption of a solar panel based energy harvesting. We implemented the

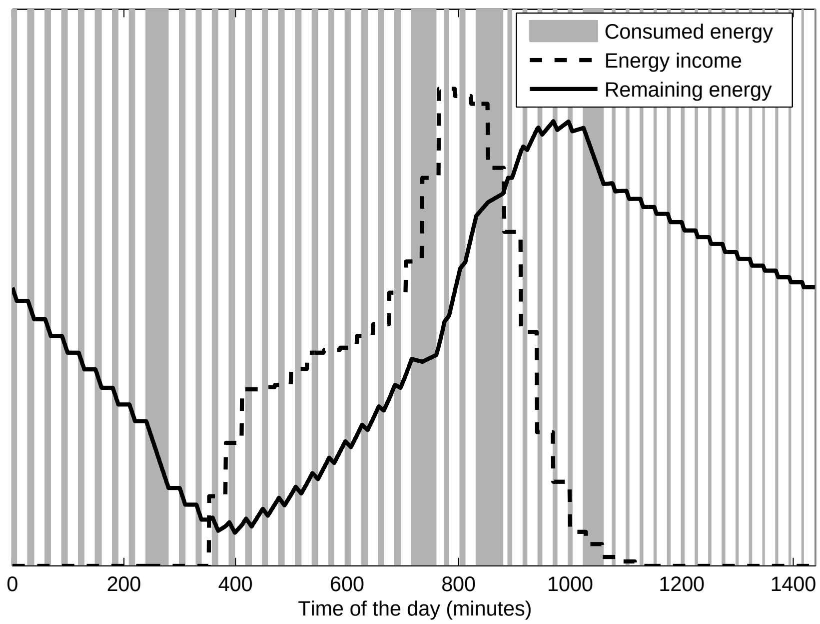

Figure 5: Utility-based active time schedule over the course of a day (1440 minutes): the evolution of the energy income and available energy.

environmental model based on the solar radiation values measured at the Orlando Internatinal Airport.

This scenario involves the movement of a trusted mobile node from a start to an end location. The node uses the information to avoid encounters with a number of threat nodes which move randomly in the environment. If the trusted node approaches a threat node closer than the threat's sensor range, the mission is considered a failure.

As a note, the behavior of the trusted node and the threat nodes is relatively simplistic in these settings: the threat nodes are following a random waypoint movement model, and do not actively search for nodes. When notified of the presence of a threat node, the trusted node takes a simple evasive maneuver by moving away from the threat.

We do not consider the time to reach the target location as a performance factor. As the threat node does not pursue the trusted node, virtually all the failures are due to the trusted node not having sufficient information about the location of the threat nodes. In effect, thus, this scenario measures the

Table 1: Simulation Parameters

Parameters	Value	Range
Common		
area	$900 \times 600(m^2)$	
number of mobile nodes	1	
number of AFNs (sinks)	6	
number of threat nodes	10	1-10
mobility of threat nodes	1 (m/s)	1-5
number of sensors	200	100-300
sensor transmission range	50 (m)	
sensor sensing range	25 (m)	
max battery capacity	1000 (units)	
discharge rate	5.0 (units/s)	
recharge rate	2.3 (units/s)	
single cycle time	3000 (s)	
Dynamic active time approach		
C_{th}	$10\times$ average threats	–
C_n	$10\times$ no. of neighbors	–
α	+4 or -4	–
Utility-based approach		
k	50	–

performance of the active time scheduling.

Table 1 shows a summary of our simulation parameters and their values. In the following results, whenever we omit the specification of a parameter, the default value specified in Table 1 is used.

5.1. Simulation results

5.1.1. Average number of failures

In our scenario, the main goal of the network was to avoid node failures, that is, cases when the trusted mobile node cannot avoid being intercepted by a threat node. This happens when the node is not aware of the presence of the threat node (or it is notified too late). In our first series of experiments we measure the number of failures averaged over 100 runs with random initial conditions. As we have only one trusted mobile node, which either fails or not, the failure rate will be a number in the $[0, 1]$ range.

Figure 6 shows the results of the average number of failures versus the number of sensor nodes for all three approaches. As we expected, the number of failures are decreasing with the increase in the number of nodes, with a particularly sharp drop between 150 and 250 nodes. What we see is that the values are very high, in the 0.75-0.95 range for 100 nodes, while they are below 0.1 for all protocols at 300 nodes. As expected, the utility based active time slot scheduling provides the best result, followed by the the dynamic active time slot algorithm and the static active time slot approach.

To interpret the graph correctly, we need to emphasize that a properly working sensor network would not operate in the regime described at the left part of the graph, thus the differences between the approaches in that range lack practical importance. For instance, at 150 deployed sensor nodes, the failure rate for the utility based active time slot scheduling is about 0.5, while for the static time slot approach is 0.85 – a very big difference, but largely irrelevant, because even 0.5 is an unacceptably high number. On the other hand, it is important that at 275 nodes and above, the failure rate for dynamic active slot and utility based scheduling drops to virtually zero, while for static active slot remains at around 5%.

Another important factor in the failure rate is the mobility of the threat nodes. Intuitively, the faster the threat node moves, the harder it is to evade it, and an earlier notification is necessary for a successful evasive maneuver.

Figure 7 shows the average number of failures as a function of the mobility of the threat nodes. We see that the utility based active time scheduling approach provides the best performance and is virtually independent of the mobility of the threat nodes, while the other two approaches show a gradual increase.

5.1.2. Mean time to failure

Another way to look at the performance of the sensor network is by measuring the average time a mobile node can survive in an environment (the mean time to failure). Figure 8 shows the mean time to failure versus the number of active nodes. As expected, the rankings of the protocols are very similar to the ones for the average failures. The utility based active time slot scheduling provides the best result, followed by the the dynamic active time slot model and the static active time slot approach.

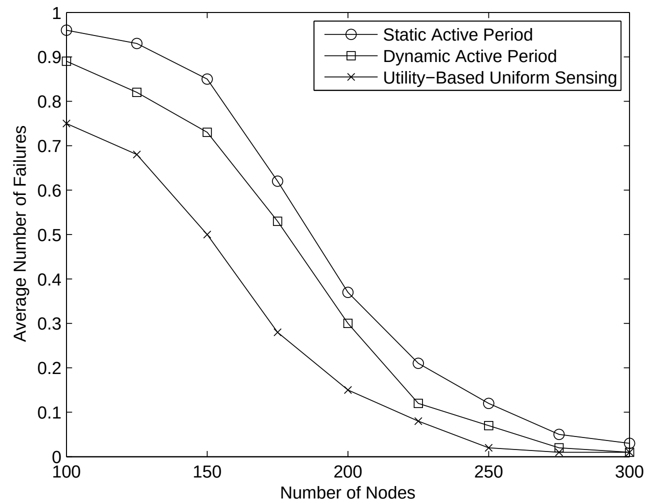

Figure 6: Average number of failures versus the number of sensor nodes.

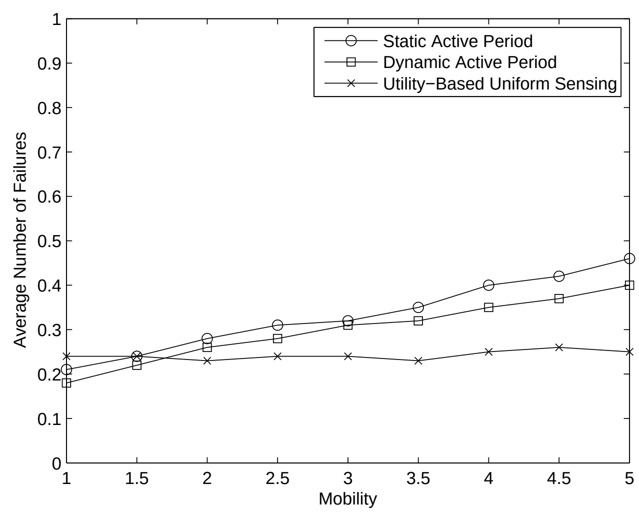

Figure 7: Average number of failures versus the mobility of the threat nodes.

5.1.3. Detected threats

A related, but different measure of performance is concerned with the number of threat nodes detected by the network and the percentage of time when these nodes are kept under observation by at least one sensor node

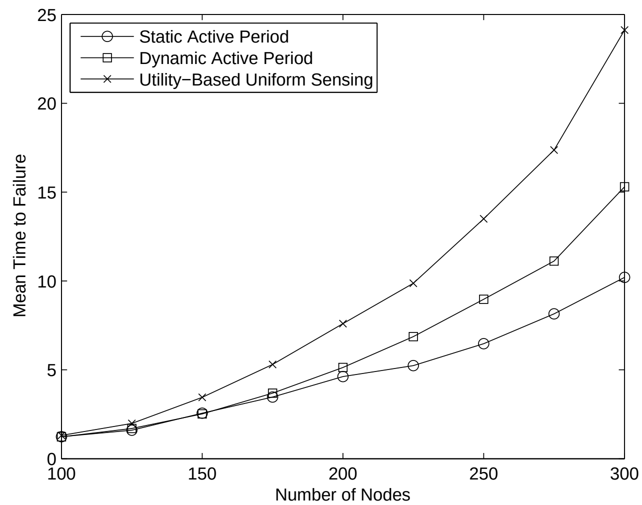

Figure 8: Mean time to failure versus the number of sensor nodes.

during their path in the system. In contrast to the previous metrics, this metric does not depend on the trusted mobile node. A threat might be entering the network and leaving harmlessly without being detected. Such a missed target would not affect a failure metric, but it is a potential danger for the overall system. Figure 9 shows the ratio of the detected threat nodes versus the threat nodes leaving the system without being detected. Figure 10 shows the average percentage of time the threat nodes were under observation by at least one sensor node.

5.1.4. Energy consumption

In non-rechargeable sensor nodes the goal of energy management is clear: we are trying to use as little energy as possible, while maintaining the detection performance at an acceptable level. For a rechargeable sensor, the goal is more complicated: the node needs to keep its energy consumption below the energy harvest, to maintain indefinite operation. At the same time, it needs to consume as much of the energy harvest as possible to attain the best performance, without expanding it. This leads to a delicate balancing act: consuming a share of 0.99 of the energy harvest is better than 0.95, but 1.01 is not acceptable! Naturally, such tight bounds are difficult to achieve.

Figure 11 illustrates the energy consumption of the three approaches as a function of the number of nodes, while Figure 12 as a function of the mobility

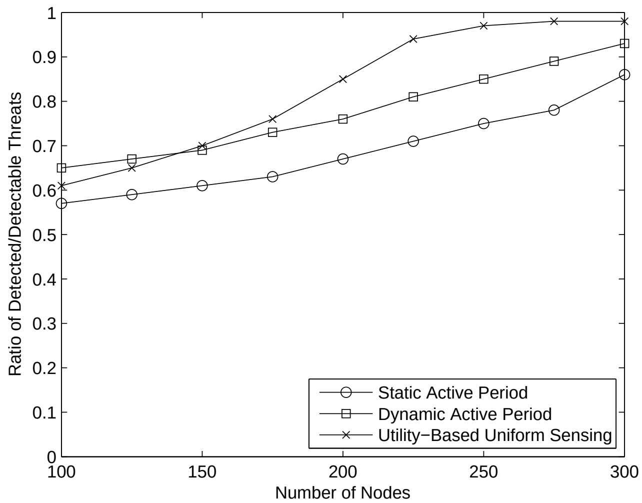

Figure 9: Ratio of detected threats to the number of present threats.

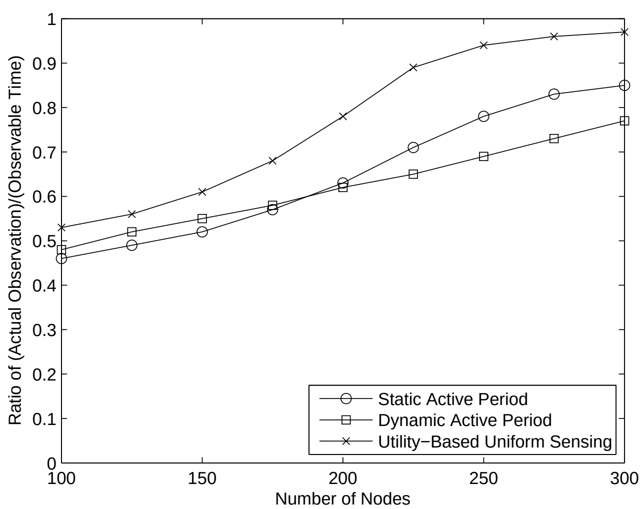

Figure 10: Percentage of time the threat nodes spent under observation.

of the threat node. For all the graphs, the energy consumed is expressed as a fraction of the harvested energy. For all approaches, the utility based scheduling showed the highest energy consumption, at around 0.9-0.95 of the harvested energy, followed by the dynamic and static active time scheduling approaches.

This difference in the consumed energy is the major reason behind the improved sensing performance of the utility based scheduling. We need to consider the case of static scheduling: naturally, the energy consumption can be set higher or lower by adjusting the constant active period. The energy budget however, is affected by the observations (through the "don't lose the target" heuristics) and the uncertainty of the energy income. The static scheduling method needs to start out with a more cautious approach than the other ones because it does not have the ability to adjust to the changing conditions during the day.

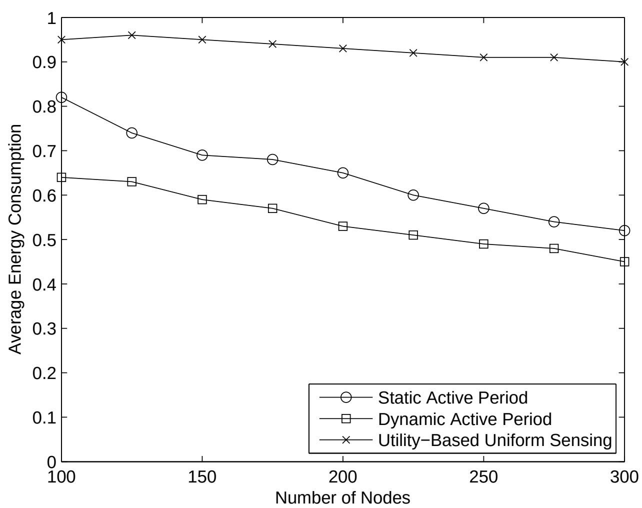

Figure 11: Average energy consumption versus the number of sensor nodes.

6. Conclusions

In this paper we proposed and compared three active time scheduling schemes for wireless sensor networks with rechargeable nodes. The simulation results show that both the multi-heuristic dynamic active time approach and the utility-based active time scheduling significantly outperforms the static active period protocol. As the performance strongly depends on the quality of the probabilistic weather and target models, one direction of future research involves developing models which can learn in the field the specifics

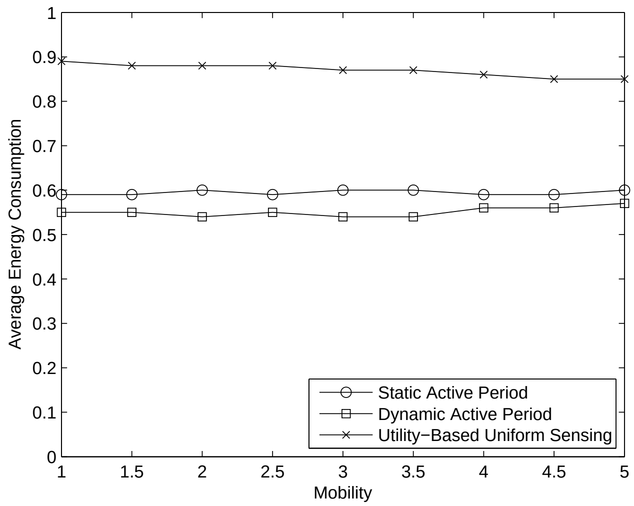

Figure 12: Average energy consumption versus mobility.

of their deployment environment. Another direction of our future research is directed towards distributed algorithms in which nodes can trade their responsibilities depending on their current and predicted energy budget as well as the relative importance of their deployment position in the early detection of threat nodes.

References

[1] F. Bian, D. Kempe, and R. Govindan. Utility based sensor selection. In *IPSN '06: Proceedings of the 5th International Conference on Information Processing in Sensor Networks*, pages 11–18, April 2006.

[2] L. Bölöni and D. Turgut. YAES - a modular simulator for mobile networks. In *MSWiM '05: Proceedings of the 8th ACM Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 169–173, October 2005.

[3] J. Byers and G. Nasser. Utility-based decision-making in wireless sensor networks. In *MobiHoc '00: Proceedings of the 1st ACM International Symposium on Mobile Ad hoc Networking & Computing*, pages 143–144, October 2000.

[4] Y. Chen and E. Fleury. A distributed policy scheduling for wireless sensor networks. In *INFOCOM '07: Proceedings of the 26th IEEE International Conference on Computer Communications*, pages 1559–1567, May 2007.

[5] J. Hsu, S. Zahedi, A. Kansal, M. B. Srivastava, and V. Raghunathan. Adaptive duty cycling for energy harvesting systems. In *ISLPED '06: Proceedings of the 2006 International Symposium on Low Power Electronics and Design*, pages 180–185, October 2006.

[6] X. Jiang, J. Polastre, and D. Culler. Perpetual environmentally powered sensor networks. In *IPSN '05: Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, pages 463–468, April 2005.

[7] A. Kansal, J. Hsu, M. B. Srivastava, and V. Raghunathan. Harvesting aware power management for sensor networks. In *DAC '06: Proceedings of the 43rd Annual Conference on Design Automation*, pages 651–656, July 2006.

[8] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava. Power management in energy harvesting sensor networks. *ACM Transactions on Embedded Computing Systems*, 6(4):32, September 2007.

[9] A. Kansal, D. Potter, and M. B. Srivastava. Performance aware tasking for environmentally powered sensor networks. *SIGMETRICS Performance Evaluation Review*, 32(1):223–234, June 2004.

[10] A. Kansal and M. B. Srivastava. Energy harvesting aware power management. In N. Balusu and S. Jha, editors, *Wireless sensor networks: A systems perspective*. Artech House, April 2005.

[11] K. Kar, A. Krishnamurthy, and N. Jaggi. Dynamic node activation in networks of rechargeable sensors. *IEEE/ACM Transactions on Networking*, 14(1):15–26, February 2006.

[12] C. Moser, D. Brunelli, L. Thiele, and L. Benini. Lazy scheduling for energy harvesting sensor nodes. In *From Model-Driven Design to Resource Management for Distributed Embedded Systems*, pages 125–134. Springer Boston, 2006.

[13] A. Nahapetian, P. Lombardo, A. Acquaviva, L. Benini, and M. Sarrafzadeh. Dynamic reconfiguration in sensor networks with regenerative energy sources. In *DATE '07: Proceedings of the 2007 Conference on Design, Automation and Test in Europe*, pages 1054–1059, April 2007.

[14] H. Nama, M. Chiang, and N. Mandayam. Utility-lifetime trade-off in self-regulating wireless sensor networks: A cross-layer design approach. In *ICC '06: Proceedings of the IEEE International Conference on Communications*, pages 3511–3516, June 2006.

[15] S. Olariu, M. Eltoweissy, and M. Younis. ANSWER: AutoNomouS netWorked sEnsoR system. *Journal of Parallel and Distributed Computing*, 67(1):111–124, January 2007.

[16] S. Olariu, A. Wadaa, L. Wilson, and M. Eltoweissy. Wireless sensor networks: leveraging the virtual infrastructure. In *IEEE Network*, pages 51–56, July-August 2004.

[17] S. Olariu, Q. Xu, and A. Y. Zomaya. An energy-efficient self-organization protocol for wireless sensor networks. In *ISSNIP'04: Proceedings of the 2004 Intelligent Sensors, Sensor Networks and Information Processing Conference*, pages 55–60, December 2004.

[18] P. Padhy, R. K. Dash, K. Martinez, and N. R. Jennings. A utility-based sensing and communication model for a glacial sensor network. In *AAMAS '06: Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1353–1360, May 2006.

[19] J. A. Paradiso and T. Starner. Energy scavenging for mobile and wireless electronics. *IEEE Pervasive Computing*, 4(1):18–27, January-March 2005.

[20] K. Premkumar and A. Kumar. Optimal sleep-wake scheduling for quickest intrusion detection using wireless sensor networks. In *INFOCOM '08: Proceedings of the 27th IEEE International Conference on Computer Communications*, pages 1400–1408, April 2008.

[21] V. Raghunathan and P. H. Chou. Design and power management of energy harvesting embedded systems. In *ISLPED '06: Proceedings of the*

2006 International Symposium on Low Power Electronics and Design, pages 369–374, October 2006.

[22] A. Wadaa, S. Olariu, L. Wilson, M. Eltoweissy, and K. Jones. Training a wireless sensor network. *Mobile Networks and Applications*, 10(1-2):151–168, February 2005.

[23] H. Zafar and D. Corkill. Simplifying solar harvesting model-development in situated agents using pre-deployment learning and information sharing. In *ATSN'08: Proceedings of 2nd International Workshop on Agent Technology for Sensor Networks*, pages 41–48, May 2008.

[24] Y. Zhu, Y. Liu, L. Ni, and Z. Zhang. Low-power distributed event detection in wireless sensor networks. In *INFOCOM '07: Proceedings of the 26th IEEE International Conference on Computer Communications*, pages 2401–2405, May 2007.

[25] Y. Zhu and L. M. Ni. Probabilistic wakeup: adaptive duty cycling for energy-efficient event detection. In *MSWiM '07: Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*, pages 360–367, October 2007.

[26] National solar radiation data base. http://rredc.nrel.gov/solar /old_data/nsrdb/.