A Framework for Battery-Aware Sensor Management

Sridhar Dasika, Sarma Vrudhula, Kaviraj Chopra, R.Srinivasan

Center for Low Power Electronics, ECE Department, University of Arizona

{suri,sarma,chopra,srini}@ece.arizona.edu

ABSTRACT

A distributed sensor network (DSN) designed to cover a given region R, is said to be *alive* if there is at least one subset of sensors that can collectively cover (sense) the region R. When no such subset exists, the network is said to be dead. A key challenge in the design of a DSN is to maximize the operational life of the network. Since sensors are typically powered by batteries, this requires maximizing the battery lifetime. One way to achieve this is to determine the optimal schedule for transitioning sets of sensors between active and inactive states while satisfying user specified performance constraints. This requires identification of feasible subsets (covers) of sensors and a scheme for switching between such subsets. We present an algorithmic solution to compute all the sensor covers in an implicit manner by formulating the problem as *unate covering problem* (UCP). The representation of all possible sensor sets is extremely efficient and can accommodate very large number of sensor covers. The representation and formulation makes it possible to consider the residual battery charge when switching between covers. We develop algorithms for switching between sensor covers aimed at maximizing the lifetime of the network. The algorithms take into account the transmission/reception costs of sensors, a user specified quality constraint and also utilize a novel battery model that accounts for the *rate-dependent capacity* effect and *charge recovery* during idle periods. Our simulation results show that lifetime improvement can be achieved by exploiting the charge recovery process. The work [1] presented here constitutes a *framework* for battery aware sensor management in which various types of constraints can be incorporated and a range of other communication protocols can be examined.

1. INTRODUCTION

Recent advances in MEMS, sensor technology and microelectronic fabrication have made it possible to design low cost micro-

---

sensors that can perform sensing, data processing and wireless communication. This capability has given rise to enormous research and development activity in the area of distributed sensor networks (DSN). Battery powered sensor units [8] are envisioned to be used for a wide range of applications due to their low cost, ease of deployment, maintainability and reliability. Consequently, a key challenge is to efficiently utilize the available finite energy resources to maximize the *lifetime of the network*. Energy management in a DSN must consider all three facets, namely, the network topology, the data processing scheme and the communication protocol, while satisfying the specified quality and reliability constraints. Although the total energy cost of a DSN includes all aspects of the sensor's actions, for many applications, communication consumes the greatest amount of energy [7].

In order to ensure connectivity, quality and reliability, sensor networks are designed to be redundant. Moreover, in many applications large numbers of sensors are deployed randomly. Consequently, many sensor management schemes exploit this redundancy to efficiently transition a subset of sensors into a sleep or inactive state while maintaining the network connectivity [2, 3, 4, 6, 9, 10, 13, 16]. In [3], the authors show that maximum lifetime can be achieved by balancing the energy consumption in the network. They propose algorithms for optimal routing of packets in sensor networks to maximise lifetime. In [2], the authors develop an upper bound on the lifetime of the network by assigning feasible roles (routers, data aggregators, etc.) to individual sensors. In [16], every grid point has several sensors, with only one active at a time. An event based approach is described in [13], where a communication path is set up by activating sensors upon the occurrence of an event. In [10], sets of sensors called the *feasible sets* are computed using a randomized approach. Since the complexity of computation of all feasible sets is exponential, only a small fraction of the feasible sets sensors (i.e. covers) can be maintained.In this paper, we present an algorithmic solution to compute all the feasible sensor sets (sensor covers) in an implicit manner by formulating the problem as unate covering problem (UCP). The representation of all possible sensor covers is very efficient and can accommodate a large number of sensor covers. In addition, an accurate battery model that accounts for the *rate-dependent capacity* effect and *charge recovery* during idle periods, is used todetermine the selection of sensor covers. The proposed sensor management scheme considers the battery nonlinear effects (charge recovery process), the transmission/reception costs of sensors, the network topology and specified quality constraints. We develop algorithms for switching between sensor covers aimed at maximizing the lifetime of the complete network. We assume that the sensors are static and homogeneous, and assume a direct transmission communication protocol, in which each sensor communicates directly with the basestation.

1

The rest of the paper is organized as follows. Section 2 has the problem formulation, the representation of covers and defines the covering problem. Section 3 describes the battery-aware sensor management strategies and algorithms. It also includes a brief description of the battery model. In section 4 the experimental setup and simulation results are described. Concluding remarks are made in Section 5.

2. PROBLEM FORMULATION

2.1 Notation and Terminology

- $\mathcal{S}$ denotes the set of all sensors.

- R (*sensing region*) is a two dimensional grid.

- r (*sensing radius*) is the range of points a sensor can cover.

- *Sensing cover*: A set of sensors that cover R.

- $P_{x,y}$ is the set of sensors that cover point (x, y).

- A sensor $s \in \mathcal{S}$ is also represented by a Boolean variable s, where $s = 1(0)$ if s is on (off).

- $\lambda_{x,y} = 1$ if and only if there is at least one active sensor that covers (x, y). The covering constraint is given by

$$\lambda_{x,y} = \sum_{i \in P_{x,y}} s_i \qquad (1)$$

Note: $\sum$ denotes Boolean disjunction operation

- χ_R (*sensor cover function*) equals 1 if only if every point in R is covered by an active sensor. That is

$$\chi_R = \prod_{(x,y) \in R} (\lambda_{x,y}) \qquad (2)$$

Note: $\prod$ denotes Boolean conjunction operation

- A Reduced Ordered Binary Decision Diagram (ROBDD) [1] with respect to a given ordering of boolean variables is a rooted, directed, acyclic graph representing a Boolean function of the variables. It has two terminal nodes, (0) and (1). Each internal node u is labeled by a variable, and has two children, high(u) and low(u). The order in which the variables appear along a path is consistent with the ordering. The functions denoted by high(u) and low(u) are distinct and subgraphs rooted at any two nodes u and v are not isomorphic.

- Q_f (*quality factor*) is the minimum number of sensors that is required to cover every point in R.

2.2 Transmission Cost Model

The model for the energy cost of transmission follows [7]. Let $E_T(k,d)$ and $E_R(k)$ denote the total energy dissipated to transmit k bits over distance d and energy dissipated to receive k bits respectively. Let ϵ_{amp} be the transmitter amplification factor. Then,

$$E_{T_x}(k, d) = E_{elec} * k + \epsilon_{amp} * k * d^2 \qquad (3)$$
$$E_{R_x}(k) = E_{elec} * k. \qquad (4)$$

The energy consumed increases quadratically with the transmission distance from the base station. Hence, the current drawn from the battery is proportional to the square of the transmission distance of a sensor node. We assume that the transmission current is dominant. This might not true for other communication protocols.

2.3 Sensing Cover Problem (SCP)

Given a sensing region R, n sensors $s_1, s_2, \ldots, s_n$, and their respective sensing radii $r_1, r_2, \ldots, r_n$, the determination of all sensing covers, is equivalent to finding all n-tuples $< s_1, s_2, \ldots, s_n >$ such that χ_R is satisfied. This is identical to the classical unate covering problem [5]. To see this, consider a covering matrix, where the rows correspond to the points in R, and the columns correspond to the sensors. An entry of 1 in a $(row, col) = ((x, y), s)$ means that sensor s can cover grid point (x, y). Thus the sum (disjunction) of the all the variables in row (x, y) represents the conditions for covering point (x, y), i.e. $\lambda_{x,y}$. The conjunction of the row sums represents the conditions for covering every point, i.e. χ_R. As an example, consider a 4×4 R, with $n = 5$ sensors. Table 1 shows the covering table. The covering constraint associated with each point (x, y) is also shown. Taking the conjunction of all the covering constraints yields χ_R. For this example, we obtain (after simplification) $\chi_R = s_1 s_2 s_3 s_4 + s_0 s_1 s_2 s_3$.

Table 1: Covering matrix

(x,y)	s_0	s_1	s_2	s_3	s_4	$\lambda_{(x,y)}$
(0,0)	1	0	0	0	1	$s_0 + s_4$
(0,1)	1	0	0	1	1	$s_0 + s_3 + s_4$
(0,2)	1	0	0	1	1	$s_0 + s_3 + s_4$
(0,3)	0	0	0	1	0	s_3
(1,0)	1	1	0	0	1	$s_0 + s_1 + s_4$
(1,1)	1	1	1	1	1	$s_0 + s_1 + s_2 + s_3 + s_4$
(1,2)	1	1	1	1	1	$s_0 + s_1 + s_2 + s_3 + s_4$
(1,3)	0	0	1	1	0	$s_3 + s_2$
(2,0)	1	1	0	0	0	$s_0 + s_1$
(2,1)	1	1	1	1	1	$s_0 + s_1 + s_2 + s_3 + s_4$
(2,2)	1	1	1	1	1	$s_0 + s_1 + s_2 + s_3 + s_4$
(2,3)	0	0	1	1	0	$s_2 + s_3$
(3,0)	0	1	0	0	0	s_1
(3,1)	0	1	1	0	0	$s_1 + s_2$
(3,2)	0	0	1	0	0	s_2
(3,3)	0	0	1	0	0	s_2

When a number of sensors sense the same event, the amount of noise in the sensed quantity is reduced, thus increasing the quality of the output. Any application of sensor networks needs to perform according to user specified quality constraints which are generally dynamic in nature. The *quality factor* Q_f specifies the minimum number of sensors required to cover each point in R. We now show how this additional constraint can be incorporated into the SCP. Let K be the set of all combinations, formed by choosing at least Q_f sensors from a total n sensors covering a point and let k_1, k_2, $k_3, \ldots, k_n$ be the elements of K. Then the covering constraint $\lambda_Q(x, y)$ and χ_R are defined as follows.

$$\lambda_Q(x, y) = \sum_{i=1}^{\binom{n}{Q_f}} k_i \qquad (5)$$

$$\chi_R = \prod_{(x,y) \in R} (\lambda_Q(x, y)) \qquad (6)$$

For example, the point $(0, 1)$ in Table 1 can be covered by sensors s_0, s_3 and s_4. If $Q_f = 2$, then $\lambda_Q(0, 1) = s_0 s_3 + s_3 s_4 + s_4 s_0$.

We use the method described in [15] to solve UCP. χ_R is represented by a ROBDD. Thus, the set of *1-paths* in the ROBDD forms the set of feasible solutions for the sensor cover function.

The ROBDD for χ_R obtained after simplification of the covering matrix Table 1 is shown in Figure 1. From the ROBDD, we observe that there exists 2 sensing covers.
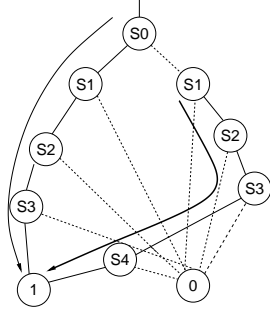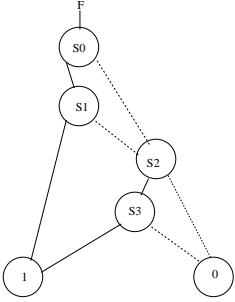

Figure 1: ROBDD of χ_R
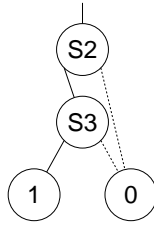

Figure 2: Before Quantification   **Figure 3: After Quantification**

3. BATTERY AWARE APPROACH

A energy based sensor management scheme requires accurate estimation of the residual charge of the battery. Each sensor node can be easily made to provide such an estimate to the base unit, either periodically or when explicitly queried. An alternative would be for the base unit to estimate the residual charge. This requires a battery model and it's discharge characteristics as a function of a time varying load profile. The battery model is discussed in Section 3.3.

- *Lifetime*:A given sensor network is alive if and only if, there exists at least one valid sensing cover.The earliest time at which no sensing cover exists is defined as the lifetime of the network.

- Let τ be the minimum time a sensing cover needs to be turned on. τ is a user specified parameter and depends on the type of application. The minimum value of τ is the transmission time for a single packet. The information content required by a specific application and the data rate of the system determine τ.

- Given I_s is the transmission current for sensor s, the *weight* denoted by $w_s(t)$, is it's residual lifetime at any instant t. $a_s(t)$ is the available charge at sensor s computed using battery model discussed in Section 3.3.

$$w_s(t) = a_s(t)/I_s \qquad (7)$$

- $S_C \subseteq \mathcal{S}$ is the valid *sensing cover* at time t if χ_R evaluated for S_C results in a tautology and $w_s(t) > \tau \quad \forall s \in S_C$.

- χ_{on} is a characteristic function (representing the set of sensor nodes) in the currently active sensing cover.

- At any time t, a sensor s is considered dead, if it's weight $w_s(t) < \tau$.

The pseudo code for updating the weight of all sensors is presented in Algorithm 1. The load profile of each sensor node is updated after a specified time depending on whether a given node is turned on or off. Using the updated load profile(discussed in 3.3) and the network on-time T, the available charge $a_j(T)$ at each sensor node is calculated using the battery model. The ratio of $a_j(T)$to I_j is the weight of the sensor node.

> **Algorithm 1:** Pseudo code of UPDATE_WEIGHTS
> UPDATE_WEIGHTS(S, χ_{on}, Δ_k, T, α, β)
> (1) **foreach** $s_j \in$ S
> (2)   **if** $s_j \in \chi_{on}$
> (3)   UPDATE_LOAD_PROFILE(I_{s_j},Δ_k);
> (4)   **else**
> (5)   UPDATE_LOAD_PROFILE($0,\Delta_k$);
> (6) $a_{s_j}(T) \leftarrow \alpha - \sigma_{s_j}(T, \alpha, \beta)$;
> (7) $w_{s_j}(T) \leftarrow a_{s_j}(T)/I_{s_j}$;
> (8) **return**

3.1 Max-Min Minimal Cover (M3C)

A sensing cover, or simply a cover, ceases to be a valid, if at least one sensor in the cover is dead. Therefore, the lifetime of a cover is limited by the lifetime of the weakest sensor in the cover. Maximizing the lifetime of the weakest sensors in a cover, delays the death of a cover. Moreover, every sensor is shared by a large number of covers. An optimal cover needs to be turned on among all the covers in such a way that the lifetime is maximized. This is accomplished, by selecting a cover whose minimum weighted node (min) is maximum (Max) over all existing covers. By keeping Max Min cover on, we avoid discharging all other covers whose weakest sensor node weights are smaller than the current weakest node, thus increasing the lifetime of the network. Additionally, we eliminate redundant sensors from this optimal cover by making it minimal. Such a cover is referred to as Max-Min Minimal Cover (M3C).

DEFINITION 3.1. *Let $F(x_1, \ldots, x_n)$ be a Boolean function. The universal quantification(Figure 2,3) of F with respect to x_i, denoted by $\forall(F, x_i)$,is defined as $\forall(F, x_i) = F(x_0, \ldots, x_i = 1, \ldots, x_n) \cdot F(x_0, \ldots, x_i = 0, \ldots, x_n)$*

DEFINITION 3.2. *At any given time t, $S \subseteq \mathcal{S}$ is an minimal cover if and only if, $\forall s \in S$, χ_R evaluated at $S - s$ is not a tautology.*

DEFINITION 3.3. *Let S_1 and S_2 be two minimal covers, with n_1 and n_2 sensors, respectively. Let $(s_{1,1}, s_{1,2}, \ldots, s_{1,n_1})$ and $(s_{2,1}, s_{2,2}, \ldots, s_{2,n_2})$ be the sensors in S_1 and S_2 respectively, arranged in increasing order of their weights. Let j be the smallest index such that $s_{1,j} \neq s_{2,j}$. Then the Max-Min Cover of S_1 and S_2 is equal to S_1 if $w_{s_{1,j}}(S_1) > w_{s_{2,j}}(S_2)$; and is equal to S_2 if $w_{s_{2,j}}(S_2) > w_{s_{1,j}}(S_1)$.*

DEFINITION 3.4. *A cover, which is both max-min and minimal, is called the max-min minimal Cover (M3C). The characteristic function of a max-min minimal cover is denoted by χ_{M3C}.*

The pseudo code to compute the χ_{M3C} is presented in Algorithm 2. The inputs are χ_R, the characteristic function describing the region R, χ_{on}, the characteristic function previous on cover, and $\mathcal{S}$, the

set of sensors. After initialization, sensors are sorted in ascending order by their weights in Step 2. Initially, when χ_{on} is empty, the index of the weakest sensor is assigned 0 and no sensor node is quantified from χ_{M3C} in step 7. If χ_{on} is not empty, then the index of the weakest sensor in the on cover is found in step 6. In step 7, all sensors weaker than the weakest sensor of χ_{on} are removed using universal quantification (Def: 3.1) over χ_{M3C} with respect to all weaker sensors. If χ_{M3C} is empty, χ_{on} is returned in step 9. Otherwise in step 10, χ_{M3C} is made minimal by removing redundant sensors and returned as a new M3C which will be the next on cover. This is further explained in the following section.

Algorithm 2: Pseudo code to compute M3C

M3C($\chi_R, \chi_{on},$ S)
(1) $\chi_{M3C} \leftarrow \chi_R$;
(2) S $\leftarrow$ SORT_BY_WEIGHT(S);
(3) **if** χ_{on}=NULL
(4)   i$\leftarrow$ 0;
(5) **else**
(6)   i $\leftarrow$ FIND_WEAKEST_WEIGHTED_NODE_INDEX(χ_{on});
(7) $\chi_{M3C} \leftarrow \forall(\chi_{M3C}, s_1 \ldots s_i)$;
(8) **if** χ_{M3C} is ϕ
(9)   **return** χ_{on};
(10) **foreach** j $\leftarrow$ (i+1 to length of array S)
(11)   result $\leftarrow \forall(\chi_{M3C}, s_j)$
(12)   **if** result $\notin \phi$
(13) $\chi_{M3C} \leftarrow$ result;
(14) **return** χ_{M3C};

3.2 Sensor Management Schedules

Using M3C, we first present a simple scheduling scheme without considering the battery recovery effect as the baseline implementation. Since, in this scheme all the M3C's are discharged one after another, it is referred to as a sequential scheduling scheme.

The sequential schedule is computed using Algorithm 3. The algorithm uses χ_R, and $\mathcal{S}$ as it's inputs to iteratively generate the load profile for each sensor and to compute the lifetime T. At the beginning of each iteration, in Step 3, χ_{on} is computed by making an initial call to M3C. In the sequential scheme, the sensor cover is kept on until it ceases to be a valid cover. Therefore, in Step 4, the on time slot Δ_k, for every iteration is limited by the life time of the weakest sensor in the on cover χ_{on}. The lifetime of the weakest sensor is computed using the battery model by applying a constant transmission load. Likewise, in Step 5 the network lifetime T is incremented by Δ_k. In step 6,the weights of all sensors are updated by calling the function Update_Weights (Algorithm 1). A set of dead sensors S_d is computed in Steps 8, 9, and 10. All the dead sensors in S_d are eliminated from χ_R in step 12, thereby ensuring the set of invalid covers are pruned out from the search space. Finally, if χ_R is not empty, a new on-cover for the next iteration is again computed. Finally, if no sensor cover exists (i.e. χ_R is empty) the loop terminates.

The lifetime of the weakest sensor in the cover characterizes the lifetime of each cover. Hence, all the weakest sensors in each cover, can be treated as a set of multiple batteries. Recent research on discharging schemes of multiple batteries [12] has shown that lifetime of the multiple battery systems can be improved by appropriately switching between them. Based on this argument, we present the switching scheme aimed at maximizing the recovery effect at each sensor using the M3C. By switching between different M3Cs every τ units of time, we ensure that the sensors with the highest battery reserves bear the transmission load, giving greater opportunity for

Algorithm 3: Pseudo code of SEQ_LIFETIME

M3C_SEQ_LIFETIME($\chi_R, \mathcal{S}$)
(1) T $\leftarrow$ 0; $\chi_{on} \leftarrow$ NULL;
(2) **while** $\chi_R \notin \phi$
(3) $\chi_{on} \leftarrow$ M3C($\chi_R, \chi_{on}, \mathcal{S}$);
(4) $\Delta_k \leftarrow$ WEAKEST_NODE_LIFETIME(χ_{on});
(5)   T $\leftarrow$ T $+ \Delta_k$;
(6)   UPDATE_WEIGHTS($\mathcal{S}, \chi_{on}, \Delta_k, T$);
(7) $S_d \leftarrow \phi$;
(8)   **foreach** $s \in \mathcal{S}$
(9)   **if** WEIGHT(s) $< \tau$
(10) $S_d \leftarrow S_d \bigcup s$;
(11)   **if** $S_d \neq \phi$
(12) $\chi_R \leftarrow \forall(\chi_R, S_d)$;
(13) **return** T;

other sensors in the network to recover.

The switching schedule is computed in Algorithm 4. Given χ_R, and S as inputs, the algorithm computes the lifetime of the network. The iterative procedure starts by computing the M3C in Step 3. The selected M3C is turned on for τ units of time and the network on time is incremented in step 5. In Step 6, the weights of all sensors are updated. Steps 7, 8, 9, 10 describe a method to identify the dead sensors in the network. These sensors are removed from the χ_R using the universal quantification operation in Step 12. The steps from 2 to 13 are repeated until, no more valid covers exist and return the value of the lifetime.

In estimation of each sensor's battery charge, it is assumed that no transmission errors occur. If transmission errors do occur, the base station does not use the battery model to estimate each sensor's battery reserve, but relies on individual sensors transmitting their battery status after each iteration.

Algorithm 4: Pseudo code of SWT_LIFETIME

M3C_SWT_LIFETIME($\chi_R,$ S)
(1) T $\leftarrow$ 0; $\chi_{on} \leftarrow$ NULL;
(2) **while** $\chi_R \notin \phi$
(3) $\chi_{on} \leftarrow$ M3C($\chi_R, \chi_{on},$ S);
(4) $\Delta_k \leftarrow \tau$;
(5)   T $\leftarrow$ T $+ \Delta_k$;
(6)   UPDATE_WEIGHTS($\mathcal{S}, \chi_{on}, \Delta_k, T$);
(7) $S_d \leftarrow \phi$;
(8)   **foreach** $s \in \mathcal{S}$
(9)   **if** WEIGHT(s) $< \tau$
(10) $S_d \leftarrow S_d \bigcup s$;
(11)   **if** $S_d \neq \phi$
(12) $\chi_R \leftarrow \forall(\chi_R, S_d)$;
(13) **return** T;

3.3 Battery Model

The capacity of a battery depends on the discharge current. The battery is less efficient at higher loads (rate dependent capacity effect). When a battery is disconnected from it's load, some of the charge that was *trapped* can be made available at the end of the idle period (recovery effect). A highly accurate and robust model of a battery that captures both these effects is presented in [11]. Consider a load profile given as a sequence of N constant current values $I_0, I_1, \ldots, I_{N-1}$, applied to the battery until the battery is fully discharged (up to time $t = L$). The load I_k starts at time t_k and is applied for a duration $\Delta_k = t_{k+1} - t_k$. Then the battery model

is defined by

$$\alpha = \sum_{k=0}^{N-1} I_k \Delta_k \qquad (8)$$
$$+ \quad 2 \sum_{k=1}^{N-1} I_k \sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 (L-t_k-\Delta_k)} - e^{-\beta^2 m^2 (L-t_k)}}{\beta^2 m^2}$$

Note: the parameters α and β are constants and characterize the battery. They are estimated by applying a set of constant load tests until the battery is fully discharged [11]. The parameter α represents the total charge in the battery when it is fully charged. β measures how fast the diffusion process can *keep up* with the rate of withdrawal of the charges. If β is moderately large, then the second term in (8) becomes negligible, and the total charge delivered to the application up to the time it is cutoff is the total charge available in the battery. A small value of β means that there can be some unused charge left in the battery at the time it was cutoff. Now consider an *actual* load profile $i_0, i_1, \ldots i_{n-1}$, applied to an (α, β) battery, where the duration of i_k is Δ_k. In general, i_k depends on the type of activity the sensor is performing, e.g. , data processing, transmitting, or receiving. Given $\{i_k\}$, we define a *cost function* $\sigma(t)$ which represents the *apparent* charge lost by the battery by time t, as follows.

$$\sigma(t) = \sum_{k=0}^{n-1} I_k \Delta_k \qquad (9)$$
$$+ \quad 2 \sum_{k=1}^{n-1} I_k \sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 (t-t_k-\Delta_k)} - e^{-\beta^2 m^2 (t-t_k)}}{\beta^2 m^2}$$

Comparing Equation (9) with (8), the lifetime L of the battery is the first instant t when $\sigma(t)$ equals α. Now, the first term in Equation (9) is the charge actually consumed by time t, while the second term is the charge that is unavailable at the electrode surface $\sigma(t)$ is then the *apparent charge lost* by the battery. The charge available at time t is $a(t) = \alpha - \sigma(t)$.

4. EXPERIMENTAL RESULTS

In this section we present the results of experiments to verify the approach and the improvement in the network lifetime due to battery recovery effect. The n sensors were randomly distributed on a grid of size 50×50. The results reported in all simulations were averaged over 20 randomly generated networks for each value of n. The programs were implemented using CUDD package [14] and executed 1.8GHz Pentium 4, with 512MB memory.

To examine the efficiency of the representation, we varied the network density by increasing the number of sensors from 50 to 100, with a sensor radius $r = 10$. Figure 4 shows that while the total number of covers increases significantly with respect to n, the size of the ROBDD representation increases at a much smaller rate. In fact, the ratio of the ROBDD size to the number of covers approaches zero as the number of covers increases (see Figure 5).

Next, we varied the sensing radii of every sensor from 8 to 20, while keeping the $n = 75$. Figure 6 shows that the number of covers increases as expected with respect to the sensing radii, however, the size of ROBDD for χ_R initially increases for smaller radii and thereafter it decreases for larger radii. This is due the fact that for smaller radii, the cover matrix is sparsely filled. As a result, the ROBDD representations are more compact. Similarly for larger sensing radii, the cover matrix is very dense and this again results in a compact representation. However for a moderate radius, when
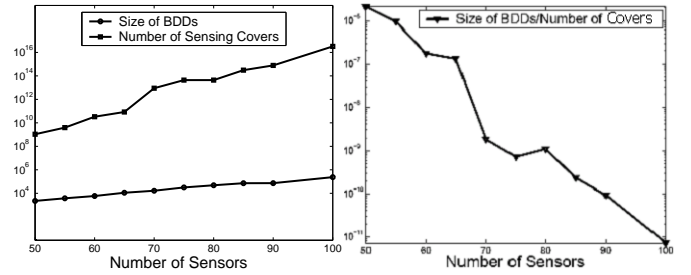

Figure 4: Impact of network density

Figure 5: Ratio of Size of BDDs and Number of Covers

the matrix is neither very sparse or dense, the ROBDD representation is larger.

Figure 7 shows the number of covers and the size of the ROBDD varies as the quality factor Q_f is varied from 1 to 4, with r and n being 10 and 90 respectively. As expected the number of covers and ROBDD size decreases with respect to increase in the Q_f again due to sparser covering matrix .
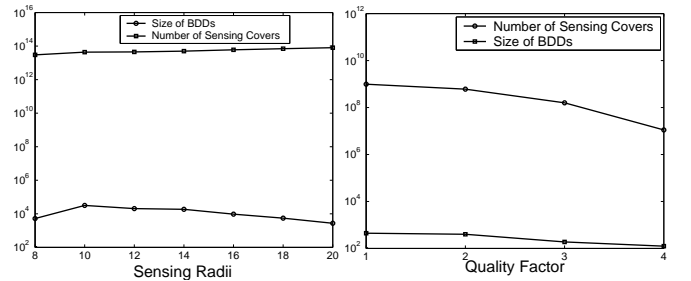
Figure 6: Increasing sensing radii

Figure 7: Increasing Q_f

It should be noted that even for a very sparse network with 50 sensors and sensing radii at 10, the number of covers is several orders of magnitude greater than those considered by any previous approaches[10]. The run-times observed to compute all covers varied from 2 to 426 seconds with increase in network density from 50 sensors to 100 sensors. Thus, the presented approach, provides an ideal platform for developing more efficient sensor management techniques.

The battery parameters α=40000 and β=0.22 were estimated by direct measurements of a 3V Lithium ion battery [11]. Data from the Berkeley Motes MICA2DOT [8] sensors was used for lifetime comparison between the sequential scheme and the proposed battery aware switching scheme. The MICA2DOT motes (MPR510CA) draw 25mA of current during transmission at maximum power and 8mA for reception. For the current set of simulation results, the parameter τ was set to 3 minutes.

The percentage improvement of M3C switching approach normalized with respect to the M3C sequential approach is shown in Figures 8 and 9. Results shown in Figure 8 were performed by varying n and keeping $Q_f = 1$, and $r = 10$, whereas those in Figure 9 were with $r = 10$ and $n = 90$ and varying Q_f. The results indicate up to 15% improvement in the sensor network life time is possible for the same M3C sensor management scheme by introducing switching of covers (Figure 8). Since the number of available covers decreases as Q_f increases, the percentage improvement in the network life time decreases. This is shown in Figure 9.

**Note:** In the sequential scheme, when a sensor is exhausted, it

is removed from further consideration. Thus no battery recovery effect is present. In the switched scheme, each sensor is discharged for a fixed amount of time, and it can be become part of a cover in a subsequent iteration since during its idle period, some of its unavailable charge would become available. To demonstrate this, the percentage improvement in the network lifetime was computed as a function of the battery parameter β. Recall, that the large value of β indicates that the diffusion process can *track* the discharge and consequently, $\sigma(t)$ represents the actual charge consumed, and there is no unavailable charge to recover. Figure 10 clearly shows how the switching scheme exploits the charge recovery capability. For $n = 50$, the improvement shown in Figure 8 is approximately 8.5%. All of this improvement is due to a charge recovery as shown in Figure 10. As β increases, the improvement of the switching scheme over the sequential scheme decreases.
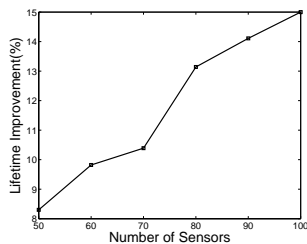
Figure 8: Improvement in Lifetime
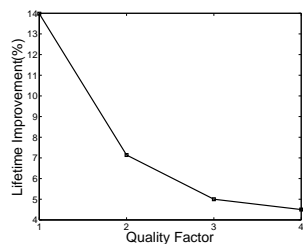
Figure 9: Effect of Q_f on lifetime for n=90,r=10
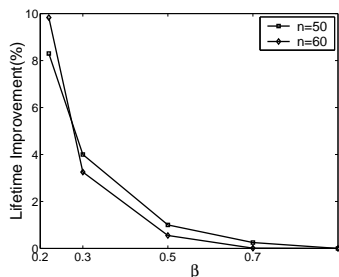
Figure 10: Effect of β on lifetime for n=50,r=10

5. CONCLUSIONS AND FUTURE WORK

Sensor management can be further improved by considering a larger set of covers. In this paper, we presented a formal method to compute all covers for the given network. Thus, it provides a ideal framework for evaluating network topologies and routing protocols. Additionally, the presented approach can be used to explore novel energy efficient sensor management techniques. The overhead to compute all sensor covers is a one time effort and can be performed at the base station which is not power constrained. Also, as demonstrated, the formulation naturally extends to the incorporation of additional constraints such as computing covers with specified quality factor Q_f. The simulation results show that by incorporating the switching to account battery recovery effect, the network lifetime can be further improved, provided that the switching overhead is negligible.

It will be interesting to extend the sensor covering paradigm to a distributed environment and develop a localized decision making process using energy information.

6. REFERENCES

[1] R. E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Trans. on Computers*, c-35(8):677–691, Aug. 1986.

[2] M.Bhardwaj, T.Garnett, and A.Chandrakasan. Upper Bounds on the Lifetime of Sensor Networks, in Proceedings of the IEEE International Conference on Commucniations, 2001.

[3] J-H Chang and L. Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *Proc. INFOCOM*, pp 22-31, 2000.

[4] B. Chen, K. Jamieson, H.Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *Annual International Conference on Mobile Computing and Networking*, 2000.

[5] O. Coudert and J. C. Madre. New ideas for solving covering problems. In *Proc. IEEE Design Automation Conf. (DAC)*, pages 641–646, 1995.

[6] Samir Goel and Tomasz Imielinski. Prediction-based monitoring in sensor networks: Taking lessons from mpeg. *CM Computer Communication Review*, 2001.

[7] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Application-specific protocol architectures for wireless networks. *IEEE Transactions on Wireless Communications*, 1(4), Oct. 2002.

[8] Berkeley Motes data sheet www.xbow.com/Products/Product_pdf_files/Wireless_pdf/6020-0043-01_A_MICA2DOT.pdf.

[9] P. Cheung Liu, L.Guibas, and F. Zhao. A dual-space approach to tracking and sensor. In *ACM International Workshop on Wireless Sensor Networks and Applications Workshop*, 2002.

[10] M. Perillo and W. Heinzelman. Optimal sensor management under energy and reliability constraints. In *IEEE WCNC*, 2003.

[11] Daler Rakhmatov and Sarma Vrudhula. Energy Management for Battery-Powered Embedded Systems. *ACM Transactions on Embedded Computing Systems*, 2(3):277–324 Aug. 2003.

[12] Ravishankar Rao,e.tal. Analysis of Discharge Techniques for Multiple Battery Systems. In *Proc. IEEE Int'l Symp. on Low Power Electronics and Design (ISLPED)*, pages 44–47, Aug. 2003.

[13] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava. Optimizing sensor networks in the energy-latency-density design space. *IEEE Transactions on Mobile Computing*, 2002.

[14] F. Somenzi. Cu decision diagram package, release 2.3.1. Technical report, ECE Department, University of Colorado, 1999.

[15] T. Villa, T. Kam, R. K. Brayton, and A. L. Sangiavanni-Vincentelli. Explicit and implicit algorithms for binate covering problems. *IEEE Trans. on Computer Aided Design*, 1997.

[16] Y. Xu, J. Heidemann, and D.Estrin. Geography-informed energy conservation for ad hoc routing. In *Proc. MOBICOM*, 2001.