Optimal Sensor Management Under Energy and Reliability Constraints

Mark A. Perillo, Wendi B. Heinzelman
Department of Electrical and Computer Engineering
University of Rochester
Rochester, NY 14627
(585) 275-{8078, 4053}
{perillo, wheinzel}@ece.rochester.edu

***Abstract*— In this work, we address the problem of maximizing the lifetime of an application that requires a minimum level of quality of service from a network of energy-constrained wireless sensors. The problem is formulated as a generalized maximum flow graph problem with additional constraints and an optimal solution is found through linear programming. The result of the optimization is a schedule that determines the mode that all sensors should operate in and allows redundant sensors to turn off and conserve energy whenever possible. We show through simulations of typical sensing applications that by intelligently managing sensors, application lifetime can be extended by up to a factor of 2 in some cases.**

I. Introduction

Wireless sensor networks are uniquely characterized by nodes constrained by limited battery lifetime. If proper consideration is not taken in managing the sensors in the network, the length of time that an application running over the network achieves its desired quality of service, or reliability, can be shortened to much less than its potential. Rotating active and inactive sensors in the network, some of which provide redundant information to the application, is one way that sensors can be intelligently managed to extend network lifetime.

To understand the problem at hand, consider an application that relies on data from a number of sensors in a network. The individual sensor data can be fused at a base station to provide more reliable information. One or more sensors may be used at any time to provide data to the application, but only certain subsets of available sensors may satisfy channel bandwidth and/or application quality of service constraints. The problem that we wish to solve is to determine which sensor sets should be used and for how long so that the lifetime of the network is maximized while the necessary quality of service is always maintained at the application. In this work, we interpret this problem as a generalized maximum flow problem and use a linear program to optimally solve it.

The rest of this paper is organized as follows. Section 2 formally presents the problem. Section 3 describes implications of the linear programming approach that can be used to solve the problem. Section 4 presents our simulation results with some analysis. Section 5 addresses related work. Section 6 concludes and details plans for future work.

II. Problem Statement

The problem we wish to solve can be modeled as a generalized maximum flow problem with some additional constraints. For a description of the generalized maximum flow problem, the reader is referred to [1].

A. Sensor Management Problem

We are trying to maximize the lifetime of a network containing N_S multi-mode sensors. We will refer to the complete set of sensors as $S = \{S_j, j \in \{1 \ldots N_S\}\}$. In general, we will assume that all sensors in the network are capable of operating in N_{m_j} active modes and additionally in sleep mode, where the sensor consumes zero to little power. An example of a sensor that is capable of operating in multiple active modes is a video camera that can send data at variable resolution or with a variable frame rate. All sensors can communicate directly with the data sink, which is unconstrained by energy.

In order to achieve the application's required reliability level, it may be possible to use a number of the sensors by themselves or in combination. In the latter case, the sensors' data would be fused to obtain more reliable information for the application. A great deal of research has been committed to multi-sensor data fusion [2]. We assume that the application knows how to optimally fuse the data and can determine the resulting level of reliability.

A sensor set is determined to be *feasible* if i) the total bandwidth necessary to support the set is below the capacity of the network and the traffic is schedulable and ii) the set provides the necessary reliability to the application. We will refer to the set of feasible sensor sets as $F = \{F_i, i \in \{1 \ldots N_F\}\}$. In order to describe the makeup of each feasible sensor set F_i, we use a variable a_{ijk}, which is equal to one if sensor S_j is being used in mode k in feasible sensor set F_i and equal to zero otherwise.

Finally, we must define a variable P_{jk}, which represents the power consumption (sensing and communication) by sensor S_j when used in mode k. It should be noted that we are treating the power consumption of a sensor as independent of the other sensors in the set, although this is not the case in all networks. When power consumption is dependent on the activity of other

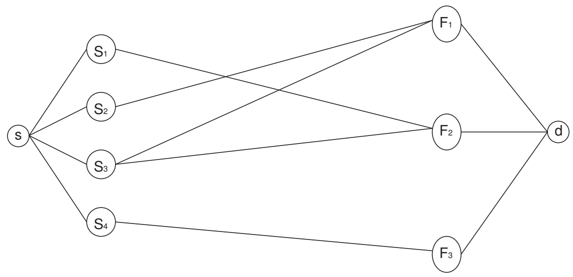

Fig. 1. Interpretation of the scheduling problem as a generalized maximum flow problem.

sensors, P_{jk} will also be a function of the set and should be designated as P_{ijk}.

We wish to develop a schedule that determines the length of time that each sensor set should be used to provide data to the application. Let T_i represent the length of time that feasible sensor set F_i is being used in the schedule. The objective of the problem is to maximize

$$T_{total} = \sum_{i=1}^{N_F} T_i \tag{1}$$

In addition to the bandwidth and reliability constraints that were considered when forming the set of feasible sensor sets F, we are constrained by the battery levels E_j of the sensors. This finite energy introduces our first constraint.

$$\sum_{i=1}^{N_F} \sum_{k=1}^{N_{m_j}} a_{ijk} T_i P_{jk} \leq E_j \qquad \forall j \tag{2}$$

Another obvious constraint in this problem is that a sensor cannot realistically operate in multiple modes within a single feasible set.

$$\sum_{k=1}^{N_{m_j}} a_{ijk} \leq 1 \qquad \forall (i, j) \tag{3}$$

B. Interpretation of Scheduling Problem as a Generalized Maximum Flow Problem

The scheduling problem that was formalized in the previous section can be modeled as a generalized maximum flow graph problem. Consider an energy bank represented by source node s in Figure 1. In our model, we will initially represent energy consumption as flow x_{ij} along the arc $(i,\ j)$. Nodes that represent sensors that are available for use by the application can be seen in the second column in Figure 1. Since the energy bank supplies the sensor nodes with their energy, arcs $(s,\ S_j)$ must be drawn from the source node s to each of the sensor nodes S_j. Each sensor can only be supplied with the energy that is contained locally, and so there is a capacity u_{sS_j} equal to E_j on each arc $(s,\ S_j)$, such that

$$0 \leq x_{sS_j} \leq u_{sS_j} \tag{4}$$

Additionally, there are nodes in the graph representing the feasible sensor sets F_i. An arc $(S_j,\ F_i)$ is drawn on the graph iff sensor S_j is included in feasible sensor set F_i. Once flow arrives at one of the nodes in F, it is more appropriate to consider the flow as time rather than energy. The arc multipliers to accomplish this will be described later.

A destination node d that represents the application is shown in the last column in Figure 1. Arcs $(F_i,\ d)$ are drawn from each feasible sensor set to the destination node. As in all generalized maximum flow problems, the objective is to maximize the total flow into the destination node.

Finally, we must define the multiplier values for each arc. On the arcs $(S_j,\ F_i)$ connecting sensors S_j and feasible sensor sets F_i, we wish to convert flow from units of energy to units of time. The multiplier on $(S_j,\ F_i)$ is the inverse of the power consumption of S_j when used in F_i.

$$\gamma_{S_j F_i} = \frac{1}{\sum_{k=1}^{N_{m_j}} a_{ijk} P_{jk}} \tag{5}$$

Multipliers also need to be included on the arcs connecting the feasible sets and the destination node. These need to be included because the time that set F_i operates is equal to the time that each of F_i's sensors operates in F_i, not the sum as the problem currently stands. The multiplier $\gamma_{F_i d}$ on each arc $(F_i,\ d)$ must be equal to the inverse of the size of the sensor set.

$$\gamma_{F_i d} = \frac{1}{\sum_{j=1}^{N_S} \sum_{k=1}^{N_{m_j}} a_{ijk}} \tag{6}$$

Since flow must be conserved at every node,

$$\sum_j x_{ij} - \sum_j \gamma_{ji} x_{ji} = 0 \qquad \forall (i \in S, F) \tag{7}$$

The final constraint is that the individual flows entering each feasible sensor set must be equal.

$$\frac{x_{S_{j_1} F_i}}{\sum_{k=1}^{N_{m_{j_1}}} a_{i j_1 k} P_{j_1 k}} = \frac{x_{S_{j_2} F_i}}{\sum_{k=1}^{N_{m_{j_2}}} a_{i j_2 k} P_{j_2 k}} \quad \forall (i, (j_1, j_2 \in F_i)) \tag{8}$$

Since we are attempting to maximize application lifetime, the objective of this graph problem is

$$\text{Maximize} \qquad \sum_{i=1}^{N_F} \gamma_{F_i d} x_{F_i d} \tag{9}$$

subject to constraints 4, 7, and 8.

C. Extension to Multi-Hop Networks

It should be noted that this problem statement does not pertain to multi-hop networks, where the power consumed by nodes forwarding data must be considered as well. This makes the problem much more difficult since it introduces another set of free variables that we must solve in order to maximize network lifetime - the share of the routing load that individual sensors should bear.

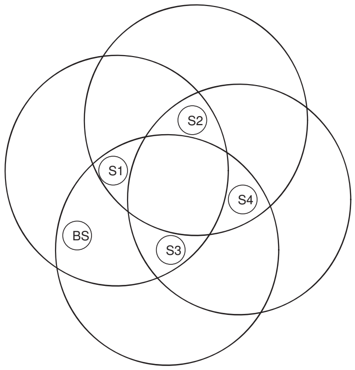

Fig. 2. Network for the multi-hop scheduling problem.

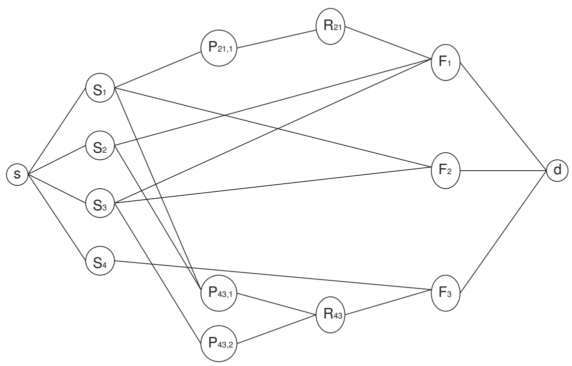

Fig. 3. Interpretation of the scheduling problem in a multi-hop network as a generalized maximum flow problem.

If we wish to map the multi-hop case to a graph problem, we must represent data forwarding on the graph. In order to do this, we introduce two additional sets of nodes to the graph. A node in the first set $R_{S_j F_i}$ (fourth column of nodes in Figure 3) represents the collective task of routing sensor S_j's data during the operation of feasible set F_i. Since the routing contributes to the feasible set, there are arcs drawn from these nodes to the feasible set during which they forward data. The number of these arcs is equal to the number of active sensors in F_i that are not within communication range of the data sink.

A node $P_{S_j F_i, l}$ in the second additional set of nodes (third column of nodes in Figure 3) represents an individual path used in forwarding S_j's data in F_i. In the case of multipath routing, several of these path nodes exist for each node described in the previous paragraph. The number of these nodes $P_{S_j F_i, l}$ having an arc drawn to $R_{S_j F_i}$ is equal to the number of distinct paths from sensor S_j to the data sink. Arcs are also drawn from each sensor on the path to the path node. Since all sensors must contribute equal time to the path, the flow along each arc incident on a path node must be equal. The multi-hop version of Figure 1, modified according to the network shown in Figure 2, is shown in Figure 3.

III. Linear Programming Approach

Because of the constraint in Equation 8, none of the algorithms that solve generalized maximum flow problems in polynomial time can be used for the sensor scheduling problem. Instead, we use a simple linear programming approach. It has been shown that linear programs are solvable in polynomial time [3]. The best implementation of Karmarkar's interior point method can solve linear programs with a worst case complexity of $O(n^3 L)$, where n is the number of variables and L is the variable resolution [4].

Our constraints and the size of the network determine the size of the linear program. The first set of equations that need to be used in the linear program regard conservation of energy at the sensor nodes. There are N_S such equations. Also, we need to add equations that allow each sensor in a feasible set to contribute equal time to that set. For each feasible sensor set node F_i, there are $\sum_{j=1}^{N_S} \sum_{k=1}^{N_{m_j}} a_{ijk} - 1$ of these equations, for a total of $\sum_{i=1}^{N_F} (\sum_{j=1}^{N_S} \sum_{k=1}^{N_{m_j}} a_{ijk} - 1)$ such equations in the problem. Finally, there are N_F equations needed for the conservation of flow at the nodes representing feasible sets, for a total of

$$N_{eq} = N_S + \sum_{i=1}^{N_F} \sum_{j=1}^{N_S} \sum_{k=1}^{N_{m_j}} a_{ijk} \tag{10}$$

The number of flows that need to be solved is equal to the number of arcs in the graph.

$$N_{arcs} = N_S + \sum_{i=1}^{N_F} \sum_{j=1}^{N_S} \sum_{k=1}^{N_{m_j}} a_{ijk} + N_F \tag{11}$$

The values of N_{eq} and N_{arcs}, as well as the variable resolution, give an indication of the size and running time of the linear program. The term $\sum_{j=1}^{N_S} \sum_{k=1}^{N_{m_j}} a_{ijk}$ is bounded by N_S. Thus, N_{arcs} is bounded as $O(N_F N_S)$. It can be seen that in the worst case, algorithms using the interior method can solve the sensor scheduling problem with $O(n^3 L) = O(N_F^3 N_S^3 L)$ complexity. In practice, the complexity will typically scale much better and will depend on N_{eq}.

Alternatively, heuristical approaches such as evolutionary algorithms and simulated annealing could be used to solve this problem. These methods may in fact be preferable in large networks, where the optimality of the solution may be traded off in return for smaller solution complexity.

IV. Simulations and Analysis

We ran simulations of random sensor-based applications as well as typical sensor-based applications to demonstrate the benefits of using an intelligent schedule to maximize application lifetime. We contrasted our results with the results under identical situations when the sensor management is done without regard to future implications of the immediate choice of sensor set. Alternative sensor management schemes include the Minimum Power Scheme (MPS) - choosing the sensor set among those whose nodes all have positive energy that consumes the least total power - and the Maximum Lifetime Scheme (MLS) - choosing the set that has the longest minimum lifetime among the sensors in the set. Our results

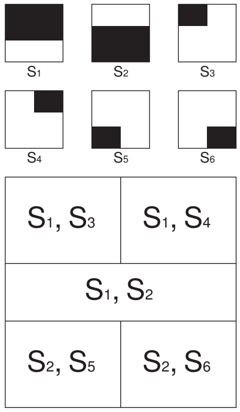

Fig. 4. Example sensor coverage situation.

F_1	S_1, S_2
F_2	S_1, S_5, S_6
F_3	S_2, S_3, S_4

TABLE I

FEASIBLE SETS FOR EXAMPLE SENSOR COVERAGE SITUATION.

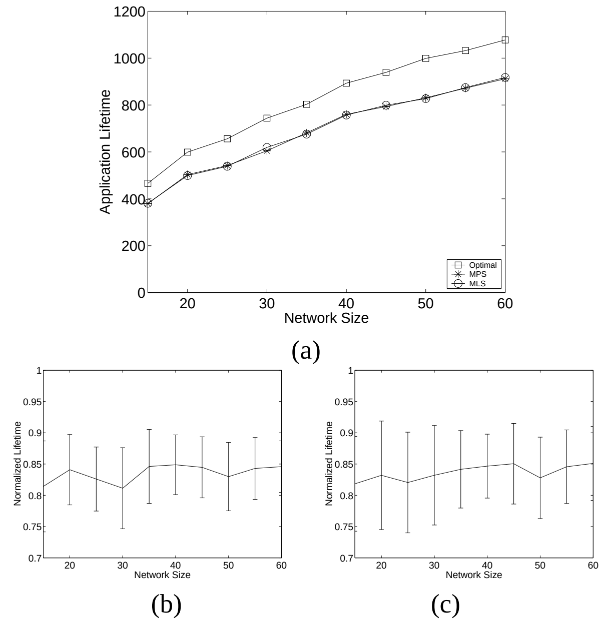

Fig. 5. Lifetime of the application with variable network size (a). Lifetime using MPS (b) and MLS (c) normalized to the lifetime using optimal scheduling.

show that depending on the situation, improvement can vary from nothing up to a factor of 2.

We first illustrate a simple example. Consider a network of 6 sensors, each of which is capable of operating in a single mode. The power consumption and initial energy among the sensors in this case is identical (1 μW and 1 μW-hour, respectively). Each sensor has the ability to monitor a partition of the observation space with 100% reliability, as shown in Figure 4. If we wish to detect the presence of phenomena anywhere in the observation space with minimal redundancy, there are a number of feasible sensor sets, shown in Table I, that can be used. When deciding which feasible set to use, the MPS approach chooses the sensor set with the least power consumption (F_1), giving the application a lifetime of 1 hour. In the MLS approach, all sets would be considered equal and a tiebreaker would need to be used. If a more intelligent approach were used, the application would have chosen F_2 and F_3 to be used subsequently, allowing the application to run twice as long as if F_1 was used.

In the first simulations, we randomly generated feasible sets, while independently varying the number of sensors, the number of feasible sets, and the maximum number of modes a sensor could operate in. In each of these simulations, the default parameters used were as follows. Initial energy was chosen randomly from 1 to 5 J. Power consumption was 10 μW in the first mode and increased linearly with mode number. The average number of sensors in the feasible sets was 5. We ran 100 trials and calculated the mean and standard deviation for each situation.

In the simulations where the number of sensors was varied, we kept the number of feasible sets constant at 50 and the maximum number of active modes constant at 1. Figure 5a shows that the average application lifetime increases approximately linearly with network size using all three scheduling approaches. This is expected as more aggregate energy is distributed throughout the network for use in the sensing task. In Figures 5b and 5c, it can be seen that the average lifetimes using the MPS and MLS approaches, respectively, are consistently about 15% below the potential lifetime that can be achieved through optimization.

Next, we kept the number of sensors in the network constant at 50 and the maximum number of active modes constant at 1, while varying the number of feasible sets. Figure 6a shows that for all scheduling methods tested, the average application lifetime increases with the number of feasible sets. This matches our expectation that as the number of options for sensor sets increases, application lifetime should increase as well. Figures 6b and 6c show that the benefit of schedule optimization is approximately constant at 15% and does not seem to depend on the number of feasible sensor sets.

To observe the effect of using multi-mode sensors, we kept the number of sensors in the network constant at 50 and the number of feasible sets constant at 50 as we varied the maximum number of active modes that the sensors could operate in. Figures 7a and 7b show the lifetime using the MPS and MLS approaches, respectively, normalized to the lifetime using the optimal schedule. The results of these simulations show that the benefit of optimizing the schedule is effectively independent of the maximum number of modes in which a sensor can operate and typically gives around 15% average improvement over both MPS and MLS.

In addition to these random applications, we ran simulations for some typical sensor-based applications. We placed

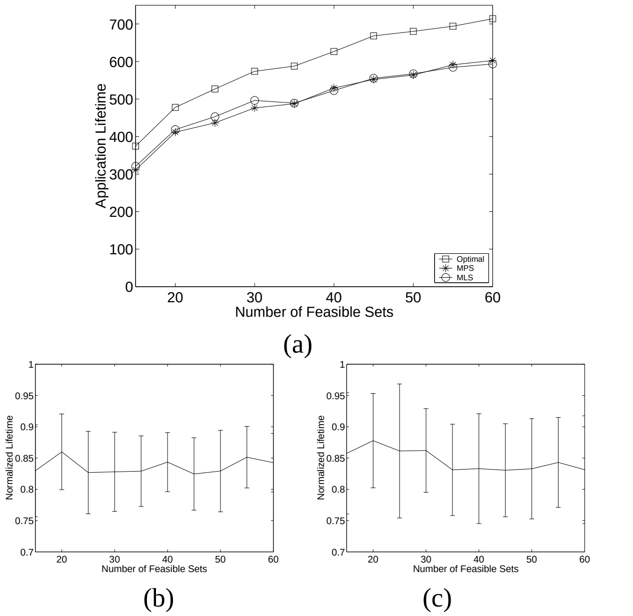

Fig. 6. Lifetime of the application with variable number of feasible sets (a). Lifetime using MPS (b) and MLS (c) normalized to the lifetime using optimal scheduling.

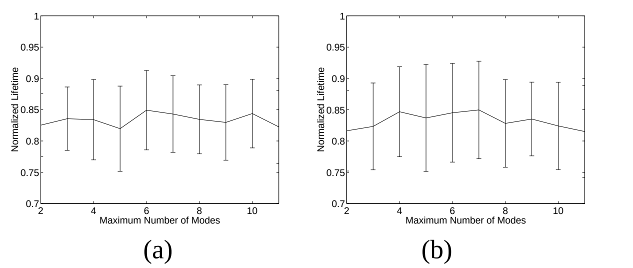

Fig. 7. Lifetime of the application with variable number of sensor modes using MPS (a) and MLS (b) normalized to the lifetime using optimal scheduling.

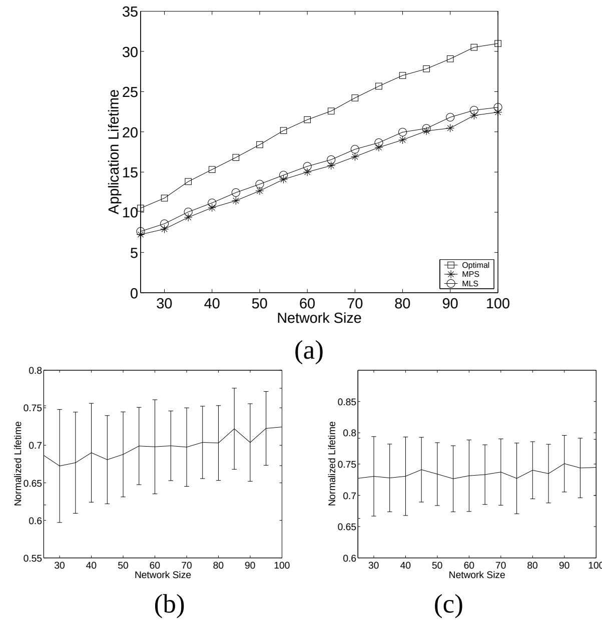

Fig. 8. Lifetime of the multi-sensor (equal coverage) detection application (a). Lifetime using MPS (b) and MLS (c) normalized to the lifetime using optimal scheduling.

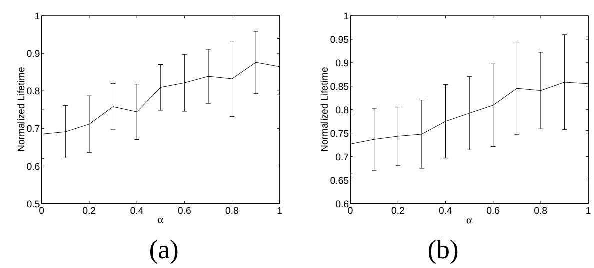

Fig. 9. Lifetime of the multi-sensor (varied coverage) detection application using MPS (a) and MLS (b) normalized to the lifetime using optimal scheduling.

a number of sensors, each capable of monitoring a circle of constant area, at random locations on a rectangular grid. The feasible sets were found by determining which combinations of sensors would allow a certain percentage of grid (90%) to be monitored. Finding the feasible sets becomes a large problem in itself, as the number of possible sensor combinations grows exponentially with network size. Because of the difficulty in finding every possible feasible set, we found 100 "good" feasible sets that represent a subset of F. It should be noted that our optimization may not give the same result as the true absolute optimal schedule since some feasible sets are not considered. We ran simulations to find the improvement of optimizing the schedule over the MPS and MLS approaches while varying the network size from 25 to 100 sensors. For these simulations, we set initial sensor energy to a random value uniformly distributed from 1 to 5 J and allowed all sensors to operate in a single mode consuming 10 μW of power. The application lifetime for all scheduling methods is shown in Figure 8a. The lifetime using the MPS and MLS approaches, normalized to the optimal schedule lifetime, is shown in Figures 8b and 8c, respectively. The size of the benefit of optimizing the schedule is again independent of network size.

Next, we ran simulations with sensors having random sensing areas (power consumption increases linearly with sensing area). We kept the network size constant at 50 sensors and varied the "randomness" of the sensing area (setting the maximum sensing distance of the sensors to a uniformly distributed value on $[d_{max,avg}(1-\alpha), d_{max,avg}(1+\alpha)]$, where α is the parameter that we varied) and observed the improvement that could be achieved over the MPS and MLS schemes through schedule optimization. The average lifetime of the MPS and MLS methods, normalized to the optimal schedule lifetime, is shown in Figures 9a and 9b, respectively. The benefit of schedule optimization seems to be greatest when sensors have equal sensing area (homogeneous sensors).

Finally, we ran simulations of the application illustrated in Figure 4, varying initial node energies. Initial energies

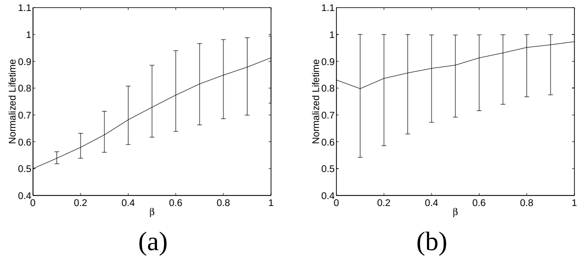

Fig. 10. Lifetime of the application in Figure 4 using MPS (a) and MLS (b) normalized to the lifetime using optimal scheduling.

were assigned as uniformly distributed random variables on $[E_i(1-\beta), E_i(1+\beta)]$ J, where β represents the parameter that was varied. The plots of the lifetime using the MPS and MLS methods, normalized to the optimal schedule lifetime, are shown in Figures 10a and 10b, respectively. The results show that as the energy distribution varies more, the benefit of optimizing the schedule seems to drop off.

V. Related Work

Many researchers have considered sensor management from a variety of perspectives [5] [6] [7] [8]. Limited energy resources are typically considered the primary design constraint for wireless sensor networks. Several sensor management schemes have focused on providing energy-efficiency in wireless sensor networks. STEM, proposed by Schurgers *et al.* [9], and SPAN, proposed by Chen *et al.* [10], allow smart sensors to be turned off whenever they are not being used as a traffic source or in a vital role in packet forwarding. Other cluster-based protocols such as LEACH, developed by Heinzelman *et al.* [11], take advantage of the redundant nature of data from densely populated wireless sensor networks by aggregating sensor data before forwarding to the data sinks, greatly reducing the communication involved. While most of the research in this area attempts to minimize power consumption by reducing communication, not much has been done to reduce the amount of traffic that is generated on the network when the amount of data being collected is more than necessary. This work does not present such a protocol completely, but it demonstrates the advantages of doing so.

Bhardwaj *et al.* have found the upper bounds of the lifetime of multi-hop wireless sensor networks [12]. Where we consider the problem from an application perspective and ignore data routing and implications of protocols below the application level on the network stack, Bhardwaj *et al.* considered the problem from a lower level perspective.

VI. Conclusions and Future Work

We have formalized a sensor management problem and used linear programming to maximize the lifetime of sensor-based applications. An improvement as large as a factor of 2 can be obtained using an optimal schedule, but the price that is paid for this lifetime maximization is the computational overhead of calculating the optimal schedule. However, this task has a one-time cost and can be carried out by an unconstrained data sink, or base station. Also, while our approach always guarantees the maximum network lifetime, it does not guarantee the simplest way to achieve it. In other words, several schedules may achieve the same network lifetime, and while a single schedule might use the minimum number of feasible sets, minimizing communication overhead, our method does not always choose this schedule. Another related area that we would like to explore is the optimization of a multi-hop network, where the sensor schedule and routing would be designed in conjunction, as detailed in Section II-C.

In the near future, our research will be directed toward the design of a middleware for use in sensor network applications such as those being developed for a Smart Medical Home at the University of Rochester's Center for Future Health [13]. The goal of this middleware is to allow the network to adapt itself to the requirements of an application so that a minimum level of reliability is maintained in a network whose topology is potentially very dynamic.

Acknowledgement

The authors would like to thank Dr. Greg Dobson for his helpful comments.

Mark Perillo is supported by the Office of Naval Research through a NDSEG fellowship.

References

[1] R. Ahuja, T. Magnanti, and J. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Inc., 1993.

[2] D. Hall and J. Llinas, "An Introduction to Multisensor Data Fusion," *Proceedings of the IEEE*, vol. 85, no. 1, January 1997.

[3] L. Khachiyan, "A Polynomial Algorithm in Linear Programming (in Russian)," *Soviet Mathematics Doklady*, vol. 20, 1979.

[4] N. Karmarkar, "A New Polynomial Time Algorithm for Linear Programming," *Combinatorica*, vol. 4, no. 4, 1984.

[5] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, "Infrastructure Tradeoffs for Sensor Networks," in *Proceedings of the ACM International Workshop on Wireless Sensor Networks and Applications Workshop*, 2002.

[6] M. Chu, H. Haussecker, and F. Zhao, "Scalable Information-Driven Sensor Querying and Routing for ad hoc Heterogeneous Sensor Networks," *International Journal of High Performance Computing Applications*, 2002.

[7] J. Liu, P. Cheung, L. Guibas, and F. Zhao, "A Dual-Space Approach to Tracking and Sensor Management in Wireless Sensor Networks ," in *Proceedings of the ACM International Workshop on Wireless Sensor Networks and Applications Workshop*, 2002.

[8] S. Meguerdichian, S. Slijepcevic, V. Karayan, and M. Potkonjak, "Localized Algorithms in Wireless Ad-Hoc Networks: Location Discovery and Sensor Exposure," in *Proceedings of MobiHOC*, 2001.

[9] 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, January 2002.

[10] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," in *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking*, 2000.

[11] Heinzelman, W. and Chandrakasan, A. and Balakrishnan, H., "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, October 2002.

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

[13] "University of Rochester Center For Future Health," http://www.futurehealth.rochester.edu/, 2002.