Sensor Management Policies to Provide Application QoS

Mark A. Perillo *, Wendi B. Heinzelman

Department of Electrical and Computer Engineering
University of Rochester
Rochester, NY 14627

Abstract

Wireless sensor networks are uniquely characterized by tight energy and bandwidth constraints. These networks should be designed to provide enough data to their application so that a reliable description of the environment can be derived, while operating as energy-efficiently as possible and at the same time meeting bandwidth constraints. These goals are typically contradicting and must be balanced at the point where the application is best satisfied. In this paper, we address the problem of maximizing lifetime for a wireless sensor network while meeting a minimum level of application quality of service. This maximization is achieved by jointly scheduling active sensor sets and finding paths for data routing. Simulation results show that several heuristic policies can achieve near optimal network lifetime.

Key words: sensor management, energy-efficiency, application QoS

1 Introduction

Interest in the use of wireless sensor networks has blossomed over the last several years due to technological advances enabling smaller devices and the realization of the potential benefit of such networks in many applications. In some situations, sensor networks may consist of sensors with overlapping coverage areas that provide redundant information, giving an application a quality level that is more than necessary. Rather than provide this unnecessary redundant data, it may be desirable to reduce power consumption and

* Corresponding Author
Email addresses: perillo@ece.rochester.edu (Mark A. Perillo), wheinzel@ece.rochester.edu (Wendi B. Heinzelman).

conserve energy in these sensors to lengthen the lifetime of the network or minimize the rate at which the sensors must be replenished with energy. This energy conservation can be accomplished through a number of methods. For example, sensors' reporting rate or data resolution can be adjusted, or the sensors can be turned off completely for an extended period of time. Balancing the application quality with this goal of energy-efficiency essentially provides a type of application quality of service (QoS). To efficiently provide this QoS to the application, interaction with lower levels of the sensor network's protocol stack is required. Recently, efforts have been made to develop middleware providing this interaction while simplifying software development efforts [1] [2]. Here, we discuss the advantages of efficient sensor management when used in such a middleware system.

In this work, we show how the use of two strategies - turning off redundant sensors and using energy-efficient routing - can be used to extend network lifetime while meeting a required level of application quality. Recent research has focused on methods of in-network data aggregation to reduce the amount of communication in dense wireless sensor networks. In this case, low-level fusion is typically performed on data from neighboring sensors before being sent to a data sink. As an alternative to this approach, redundant sensors can be turned completely off for periods of time to save energy. Of course, there is a tradeoff between power consumption and overall quality of the data delivered by the network when choosing which approach to use. We consider the latter approach in this work but realize the benefits of the former. In our work, we also take careful consideration at the routing layer, calculating routes in conjunction with the sensor scheduling. We show in this paper how to jointly optimize sensor scheduling and data routing to extend the lifetime of sensor networks and we analyze the performance of some simple heuristic sensor management and routing policies.

The rest of this paper is organized as follows. Section 2 gives the problem formalization. Section 3 describes the simple policies that we use for scheduling and routing in our simulations. Section 4 provides simulation results. Section 5 provides context for where intelligent scheduling/routing can be used. Section 6 addresses related work. Section 7 concludes the paper.

2 Multihop Sensor Network Management Problem

If an application is able to perform at an acceptable level using data from a number of different sensor sets, we would like to schedule the sets so as to maximize the sum of the time that all sensor sets are used. Acknowledging the impact that route selection will have on network lifetime, we would like to determine route selection in conjunction with the sensor schedule. In general, the routes should be chosen so that nodes that are more critical for use as sensors are routed around as often as possible. Also, when determining the length of time for which a sensor set should be used, it is important to consider that the affected sensors are not only those that are active in the set, but also those being used in the chosen path(s) to the data sink. Obviously, it is wise to tightly couple the scheduling of sensor sets with the selection of routes. In this section, we formalize this problem and model it as a generalized maximum flow problem with additional constraints. For a description of the generalized maximum flow problem, the reader is referred to [3].

We assume that for the majority of the network lifetime, the sensors act in a vigilant state, observing a potential phenomenon in the environment being monitored. In this case, the state of the application remains constant over time. In applications such as object tracking where higher data quality/confidence is required in the current vicinity of the object and nearby sensors become more critical, the application state changes frequently and the problem becomes much more difficult to model. In this work, we assume the simpler model of a constant state application but discuss the implications that multi-state applications would have on our model in Section 2.3.

2.1 Problem Formalization

In previous work, we have shown how to maximize network lifetime via optimal scheduling in single hop wireless sensor networks [4]. Here, we extend the model to account for multihop networks. We consider a multihop network consisting of N_S multi-mode sensors and refer to the complete set of sensors as $S = \{S_j, j \in \{1 \dots N_S\}\}$. 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's power consumption is negligible. 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 an ECG system that can work with different numbers of leads.

In order to achieve the application's required QoS, it may be possible to use a number of the sensors by themselves or in combination. A sensor set is determined to be *feasible* if i) the total bandwidth necessary to support the traffic of the set in any region is below the capacity of the network and the data is guaranteed to be schedulable and ii) the set provides the necessary reliability to the application. We will refer to the set of feasible sensor sets as $F = \{F_i, i \in \{1 \dots N_F\}\}$. 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 during feasible sensor set F_i and equal to zero otherwise. We must also define $P_{s,jk}$, which represents the power consumption of sensor

S_j when used in mode k. This power consumption includes both the sensing power and the processing power.

Now we will define a representation of the available routes from each node with variable $r_{j_1 j_2 l}$, which is equal to one if sensor S_{j_1} is included on sensor S_{j_2}'s lth distinct path to the data sink (routes are arbitrarily numbered from 1 to N_{P,j_2}, which represents the number of distinct paths from sensor S_{j_2} to the data sink). Since each sensor is a part of each of its paths,

$$r_{j_1 j_1 l} = 1 \qquad \forall (j_1, l \in \{1 \ldots N_{P,j_1}\}) \tag{1}$$

$P_{r,j_1 j_2 k}$ represents the power consumption of sensor S_{j_1} when routing sensor S_{j_2}'s data during mode k. For $j_1 = j_2$, this includes transmit power, while for $j_1 \neq j_2$, it includes receiving, processing, and transmit power.

We wish to develop a schedule that determines which sensor combinations should be used to monitor the environment and for how long and how the data from these sensors should be routed to the application. Let T_i represent the length of time that feasible sensor set F_i is being used in the schedule. T_i provides all the information that needs to be known about the schedule since the order of usage is of no consequence as long as startup costs of using the sets are negligible.

We also need to determine f_{jl}, the fraction of time that path l is used to route sensor S_j's data.

Sensor set lifetime is limited by the sensor battery levels E_j, which introduces our first constraint.

$$\begin{aligned}\sum_{i=1}^{N_F} \sum_{j_2=1}^{N_S} \sum_{k=1}^{N_{m,j_2}} \sum_{l=1}^{N_{P,j_2}} r_{j_1 j_2 l} P_{r,j_1 j_2 k} f_{j_2 l} a_{i j_2 k} T_i +& \\ \sum_{i=1}^{N_F} \sum_{k=1}^{N_{m,j_1}} a_{i j_1 k} P_{s,j_1 k} T_i \leq E_{j_1} \quad \forall j_1&\end{aligned} \tag{2}$$

This constraint says that the total amount of time any node can route other nodes' data and the total amount of time any node can be an active sensor are limited by that node's initial energy.

Another obvious constraint in this problem is that a sensor cannot realistically operate in multiple modes within a single sensor set, which means that

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

If a sensor is used in a feasible sensor set, its data needs to be forwarded to the data sink for the entire duration of each of its sensor sets' scheduled time. This is true for each feasible sensor set in which the sensor is used. This gives the constraint

$$\sum_{l=1}^{N_{P,j}} f_{jl} = 1 \qquad \forall j \tag{4}$$

The objective of the problem is to maximize

$$T_{total} = \sum_{i=1}^{N_F} T_i \tag{5}$$

2.2 Interpretation of Joint Scheduling and Routing Problem as a Generalized Maximum Flow Problem

The joint scheduling and routing 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 the flow x_{ij} along 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} on each arc $(s,\, S_j)$, such that

$$0 \leq x_{sS_j} \leq E_j \tag{6}$$

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.

If we wish to map the case of multihop networks to the 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_j (fourth column of nodes in Figure 1) represents the collective task of routing sensor S_j's data. Since the routing contributes to the sensor set, there are arcs drawn from these nodes to the sensor set during which they forward data. The number of these arcs is equal to the number of active sensors in F_i.

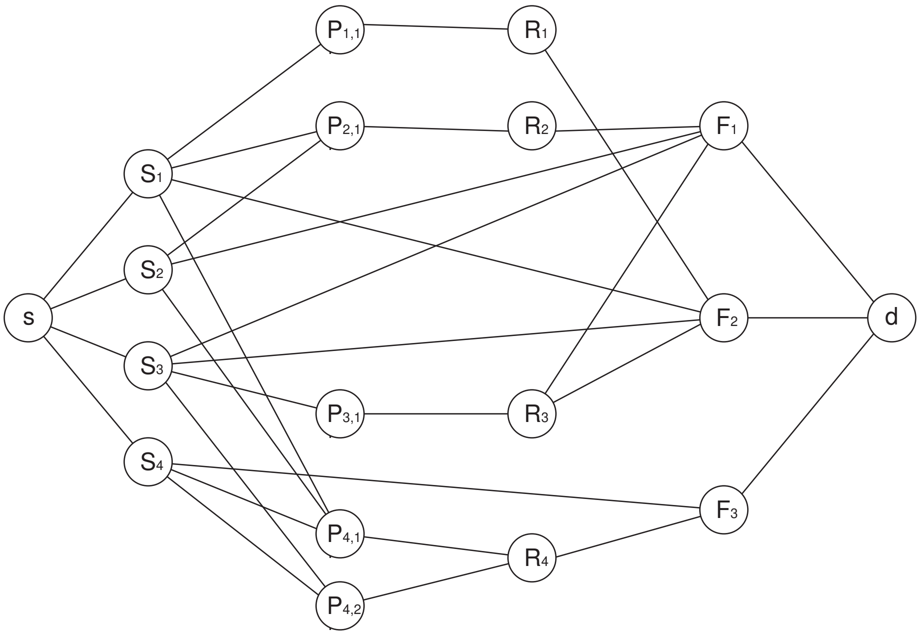


Fig. 1. Interpretation of the scheduling problem in a multihop network as a generalized maximum flow problem.

A node $P_{j,l}$ in the second additional set of nodes (third column of nodes in Figure 1) represents the lth distinct path used in forwarding S_j's data. If S_j is within communication range of the data sink, there will be one such path, consisting of just S_j and the data sink. 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 $P_{j,l}$ having an arc drawn to R_j is equal to the number of paths from sensor S_j to the data sink. Arcs are also drawn from each sensor on the path to the path node.

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 arcs $(S_j,\, F_i)$ and arcs $(S_j,\, P_{j,l})$, 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.

$$\gamma_{S_j F_i} = \frac{1}{\sum_{k=1}^{N_{m,j}} a_{ijk} P_{s,jk}} \tag{7}$$

The multiplier on $(S_{j_1},\, P_{j_2,l})$ is the inverse of the power consumption of sensor S_{j_1} when routing sensor S_{j_2}'s data.

$$\gamma_{S_{j_1} P_{j_2 il}} = \frac{1}{\sum_{k=1}^{N_{m,j_2}} a_{ij_2k} P_{r,j_1 j_2 k}} \tag{8}$$

Multipliers also need to be included on the arcs connecting the feasible sensor sets and the destination node. These need to be included because the time that set F_i is used is equal to the time that each of its active sensors is used, 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 sum of the number of active sensors in the set and the number of active routing tasks, which is equal to the number of active sensors.

$$\gamma_{F_i d} = \frac{1}{2 \times \sum_{j=1}^{N_S} \sum_{k=1}^{N_{m,j}} a_{ijk}} \tag{9}$$

The multiplier $\gamma_{P_{j,l} R_j}$ on each arc $(P_{j,l},\, R_j)$ must be equal to the inverse of the number of nodes on the lth path from sensor S_j to the data sink.

$$\gamma_{P_{j_1,l} R_{j_1}} = \frac{1}{\sum_{j_2=1}^{N_S} r_{j_2 j_1 l}} \tag{10}$$

Additional constraints are that the flows on each individual arc entering each feasible sensor set (both from nodes representing collective routing tasks as well as those representing sensors) must be equal. This is because of the fact that all sensors must be active for the entire duration of the sensor set's scheduled time in order for the set to be considered feasible from the application's perspective.

$$\gamma_{S_j F_i} x_{S_j F_i} = \gamma_{F_i d} x_{F_i d} \qquad \forall (i, j) \tag{11}$$

$$\gamma_{R_j F_i} x_{R_j F_i} = \gamma_{F_i d} x_{F_i d} \qquad \forall (i, j) \tag{12}$$

Also, the time that each node used in a path contributes to that path must be equal.

$$\gamma_{S_{j_1} P_{j_2,l}} x_{S_{j_1} P_{j_2,l}} = \gamma_{P_{j_2,l} R_{j_2}} x_{P_{j_2,l} R_{j_2}} \qquad \forall (j_1, j_2, l) \tag{13}$$

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

$$\text{Maximize} \qquad \sum_{i=1}^{N_F} \gamma_{F_i d} x_{F_i d} \tag{14}$$

Because of the constraints in Equations 11, 12, and 13, none of the algorithms that are commonly used to 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 [5].

2.3 Extension to Multi-State Applications

The previously described model applies specifically to applications where the set of sensor sets providing reliable data to the application remains fixed over time. In many cases, current environmental conditions may cause these sets to change over time. An example of this is a personal health monitoring system where abnormal conditions detected at the base station node require more intense monitoring of certain vital signs. Another extreme case of this is an object tracking application. For these types of applications, more sensors may be required to be active in the current vicinity of the object being tracked. These applications can have a huge number of possible states. To incorporate this into the graph problem, one would simply need to list all feasible sets for all possible states in the column where feasible sets are currently listed. The arcs from the feasible sets to the destination node would be replaced by arcs to a new set of nodes representing the possible application states. Arcs would be drawn from nodes representing feasible sensor sets to each state in which the set provides the acceptable application QoS. Finally, arcs would be drawn from these state nodes to the destination node. Further constraints would need to be placed on the problem so that the flows on the arcs from the state nodes to the destination are proportional to the fraction of time that the system is expected to be in the respective states. Depending on the application, building a probabilistic model to find these fractions may be simple or very complex. Also, for such applications, it may be necessary to rerun the optimization program periodically to reflect deviations from the expected behavior of the application state over time.

3 Scheduling/Routing Policies

In the previous section, we have shown how to find the upper bounds on sensor network lifetime. However, calculating the optimal schedule and routes for large networks may become very complex and heuristic methods may be desirable. In this section, we describe some simple heuristic policies for sensor scheduling and routing. We will compare the performance of these approaches with the optimized scheduling/routing approach in Section 4. The first strategy that we considered was the use of an energy-based policy in which the scheduling and routing were chosen so that nodes with low residual energy were avoided. In the first energy-based policy, we chose the feasible sensor set in which the minimum expected lifetime among sensors in the set was maximized. Following the selection of the feasible sensor set, routes were calculated based on shortest cost to the data sink, where costs of using sensors as routers were assigned as the inverse of their residual energy. After a round of predetermined length, feasible sets and routes were recalculated based on the sensors' new residual energies. A shortcoming of this approach is that it does not consider the fact that choosing a feasible sensor set affects not only the sensors in the set, but also the sensors chosen as routers. Realizing this, we also used a policy in which shortest cost paths were pre-computed for every sensor in the network and the current feasible set was chosen so that the minimum lifetime among sensors *and routers* in each feasible set was maximized.

In addition to the energy-based policies, we also considered some power-based policies. In the first power-based policy, the feasible set that consumed the least total power among active sensors (fewest sensors) was chosen, and shortest path routes to the data sink were calculated. It should be noted that we have made the assumption that sensors are homogenous and consume equal power when routing data. If this is not the case (e.g., if adaptive power control is used or if the network consists of heterogenous nodes), it would be more appropriate to find shortest cost paths, where cost is a function of power consumption. As with the energy-based policies, new feasible sets and routes were chosen at the end of rounds. We also considered a policy in which shortest path routes were pre-computed for each sensor in the network and the set which consumed the least power among all sensors and routers was chosen.

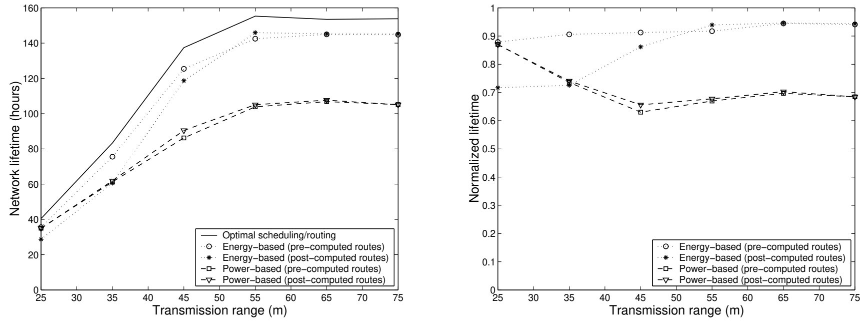
4 Simulations and Analysis

In order to compare the performance of the simple sensor management policies outlined in Section 3 with the optimal scheduling/routing, we ran simulations of applications that require a minimum sensor coverage area over an environment where a potential phenomenon is expected to occur. It should be noted that our optimization and simple policies use global information. In larger networks, this is not realistic as the cost of gathering this information and the complexity of solving the optimization problem grows quickly. With larger networks, an alternative approach might be to solve several smaller optimization problems instead or implement one of our policies on several smaller clusters of nodes. Also, in all simulations, we did not consider the overhead of setting up and tearing down routes, although we acknowledge that in real scenarios, this could in fact impact power consumption and network lifetime.

We placed a number of sensors, each capable of monitoring the environment within a constant sensing range, at random locations in a rectangular environment. The feasible sensor sets were found by determining which combinations of sensors would allow 100% of a predetermined portion of the area to be monitored. The choice of this method for specifying feasible sets is not necessarily important; any method that provides the possible sets of sensors that provide acceptable application QoS would be able to map to our framework. Because the complexity of finding every possible feasible set increases exponentially with the number of sensors used, we found 100 feasible sensor sets that represent a subset of F for each simulation. The way that these feasible sensor sets were chosen was as follows. Starting with a randomly chosen sensor, additional sensors were iteratively added to the set until the 100% coverage threshold was met. Additional sensors were chosen randomly with probability proportional to the added coverage that they would potentially provide.

In all simulations, we set each node's initial sensor energy to 1 J $\pm$ 0.2 J. The energy to transmit a packet was 15 μJ and the energy to receive a packet was 10 μJ. The energy to forward a packet was equal to the sum of the energy to transmit a packet and the energy to receive a packet (sensing power was ignored, although this could easily be incorporated into the model). Each sensor was capable of operating in a single mode in which it sent packets at a rate of 1 packet/second. Assuming that we were working well under the point where network congestion becomes an issue, all sets were considered feasible from the network perspective. Each data point is averaged over 10 scenarios. For each scenario, we found the optimal scheduling/routing and ran 10 trials using the simple policies, since there were several random factors involved. All simulations for a single scenario used identical sensor locations and feasible sensor sets.

In the first simulations, we kept the number of nodes constant at 100 and varied the nodes' transmission range to observe the effect of path length on the maximum achievable network lifetime and the relative performance of the heuristic policies. The size of the environment was fixed at 100 m x 100 m and the sensing range was fixed at 25 m. The sink to which all data was routed was located in the center of the environment. A plot of application lifetime using the optimal scheduling and routing and using the simple policies that we considered are shown in Figure 2a. A plot of the lifetime achieved through the simple policies normalized to the optimal solution's lifetime is shown Figure 2b. Both power-based policies achieve lifetime closer to the optimal solution for smaller transmission ranges. The energy-based policy using pre-computed routes consistently performs very well both for large and small transmission ranges, but the energy-based policy with post-computed routes performs worse as routing becomes more of a factor. We have also plotted the network lifetime against the average shortest path length from the sensors to the data sink in Figure 3. As expected, for all policies, lifetime decreases as the average number of hops to the data sink increases.



(a) (b)

Fig. 2. Absolute lifetime (a) and lifetime normalized to optimal solution (b) of the sensor application, as a function of transmission range.

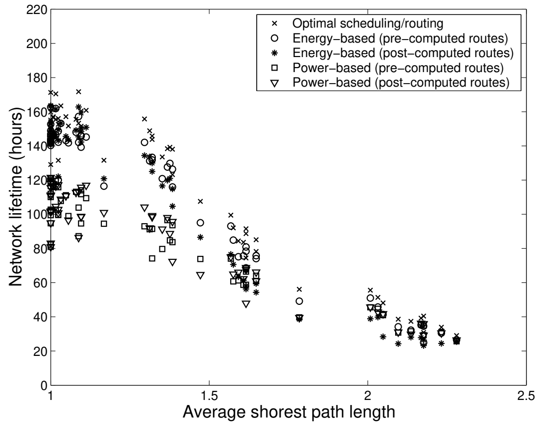
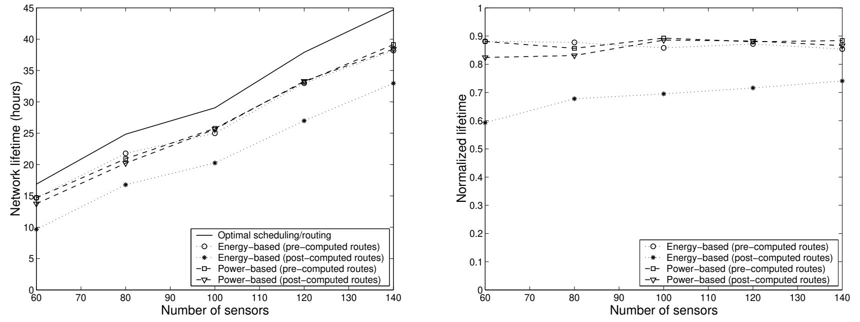


Fig. 3. Lifetime of the sensor application as a function of average shortest path length.

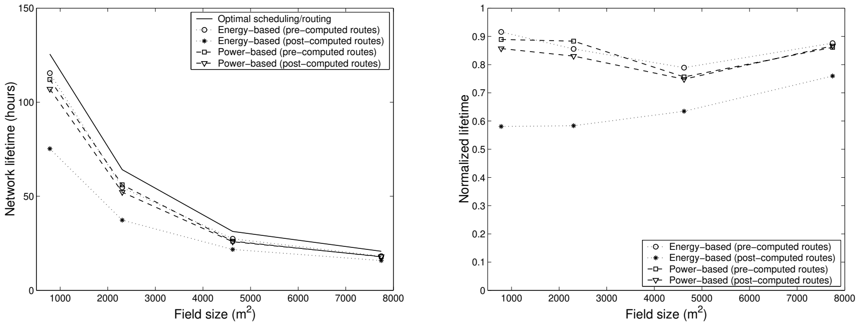
Next, we varied the number of sensors while keeping the transmission radius fixed at 25 m and the size of the environment fixed at 100 m x 100 m to observe the effect of sensor node density. A plot of application lifetime using the optimal scheduling/routing and using our policies is shown in Figure 4a. As more energy is distributed throughout the network, network lifetime is extended for all scheduling/routing policies. The relative performance of our policies compared to the optimized solution is shown in Figure 4b. Sensor node density seems to have a small effect on the performance of our policies relative to the optimal solution. All policies except for the energy-based policy using post-computed routes perform consistently well, around 90% of the optimal lifetime.

Finally, we kept the node density constant at 0.01 node/m^2 and varied the size of the environment to observe the effect of the size of the problem. If the network were consistently single hop and sensors were distributed very uniformly throughout the environment (e.g., in grid-like fashion), we would expect the absolute lifetime to stay relatively constant. Since sensor location is random, however, the possibility of a lightly covered area in the environment,



(a) (b)

Fig. 4. Absolute lifetime (a) and lifetime normalized to optimal solution (b) of the sensor application, as a function of the number of sensor nodes.



(a) (b)

Fig. 5. Absolute lifetime (a) and lifetime normalized to optimal solution (b) of the sensor application, as a function of the field size.

limiting network lifetime, increases. Also, the average power consumption in the network should increase as the sensor data needs to be forwarded along more hops on average. These two factors should cause network lifetime to decrease as network size increases. Figure 5a verifies this intuition and shows that network lifetime decreases with an increase in the size of the environment using all policies. Figure 5b shows the lifetime of our policies normalized to the optimal solution.

5 Motivation

We have begun development of a middleware for sensor networks whose purpose is to optimally balance the tradeoff between application performance, power consumption, and network lifetime. MiLAN (Middleware Linking Applications and Networks), an ongoing project at University of Rochester's Center for Future Health [6], will hide all network configuration information and decision-making from the application, simplifying program development efforts [1]. The main idea behind MiLAN is that sensor networks often consist of sensors that provide redundant information and are capable of providing more information than what the underlying application requires. MiLAN should decide the states of each sensor and be able to dynamically modify these states in response to changes in network topology and application state. Initially, we are developing MiLAN as a centralized middleware, where a central base station running the application makes all decisions regarding network configuration. Eventually, we plan to develop a distributed version of MiLAN, where sensors can either receive commands from local administrators instructing them about the ideal state that they should be in at a given time or actually make decisions by themselves based on local information.

There are many ways that an energy-performance tradeoff can be balanced. In some detection problems, the reporting rate of sensors can be varied so that a low delay between the time of event occurrence and the time that the application is notified can be given up for lower power consumption. Also, sensors can be turned off completely if their data is not important to the application. In that case, one might question why these additional sensors were deployed in the first place. The additional sensor deployment is justified because the state of the application or environment being monitored can change over time, and some sensors' data might be more critical during one state than in others. Also, it may be wise to deploy additional sensors for redundancy in the sense that if one sensor dies or malfunctions, it can be easily replaced by another.

Sensors can have diversity or similarity either spatially or in the type of information that they provide to the application. Activating sensors that are similar in either sense may be unnecessary and even undesirable when power consumption is a critical issue, as is often the case. For example, consider a heart-monitoring application that we are developing as the testbed for MiLAN. One type of sensor that we are using in our network is a blood pressure monitor. The application may want to monitor blood pressure with a number of sensors located at different places to see if there are blood pressure abnormalities anywhere on the body. The correlation of the data being sent from these sensors depends on the spatial diversity of the sensors. Using a sensor whose coverage area has large overlap with another sensor that is already being used is not wise since the marginal information that it provides to the application is minimal. To illustrate diversity (or similarity) in the type of information that a sensor provides, consider the use of an ECG system in conjunction with a simple direct heart rate monitor. While the heart rate monitor provides useful information in the absence of the ECG system, it does not provide much additional information when the ECG is already being used, and its energy could be conserved by turning the sensor off with no penalty in application performance. More importantly, the bandwidth consumed by such a sensor could prevent a sensor with more diverse information, such as a pulse oxymeter or gait analyzer, from joining the network if network congestion is an issue.

In the simulations included in this paper, we have considered sensor manage-

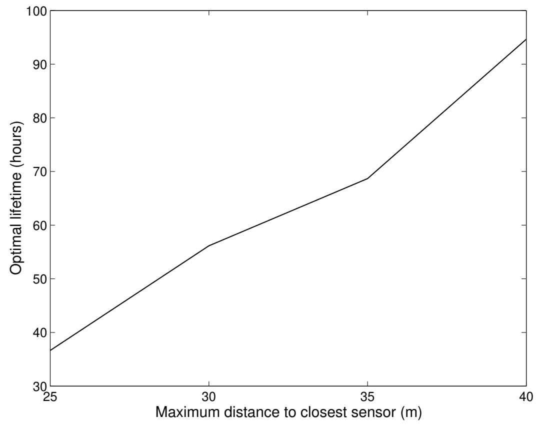


Fig. 6. Coverage quality/lifetime tradeoff for sensor application.

ment for networks of spatially diverse sensors. Specifically, for a given level of application reliability (in our simulations, we consider reliability to be the fraction of area covered by the sensor network), we show how to optimally schedule sensors to extend the system lifetime to its maximum. In addition to the blood pressure monitor that we have mentioned, it is obvious that this strategy will have benefits in many other types of sensor networks, such as fire detection in forests and chemical detection in disaster situations.

In some network architectures, the entire set of spatially diverse sensors can be fused and interpreted as a single "virtual sensor." This "virtual sensor" would be capable of operating in multiple modes, varying from large coverage/high confidence and short lifetime/large power consumption to little coverage/low confidence and long lifetime/little power consumption, depending on what level of reliability is met. For an application where quality is determined by the minimum distance from any point in the field to the nearest sensor, an example of this tradeoff curve is shown in Figure 6. Before determining the sensor set scheduling and routing, the application could look at one of these curves to determine its available options. If the application absolutely needs data of a certain level of quality, it would set a threshold on the horizontal axis and determine the upper bound on network lifetime. If on the other hand, the application sets a hard constraint on network lifetime, it would set a threshold on the vertical axis, determining the quality that should be obtained before running the optimization or implementing one of the heuristic policies.

Running these optimization programs may in fact be too computationally intense for some networks, especially in those where the base station itself is an energy-constrained node. One option that can be used in situations with spatially determined feasible sets is to divide the problem into several local optimizations. This could simplify and distribute the optimization computation significantly.

6 Related Work

A great deal of research has focused on developing energy/power-aware routing algorithms to be used to extend the lifetime of wireless networks. Singh *et al.* argued for the use of new power-oriented metrics to be used when determining paths in routing protocols [7]. In Li's $max - minxP_{min}$ method, paths are chosen by maximizing the minimum residual energy of all nodes on the chosen path while bounding the total power consumption [8]. Chang *et al.* found local algorithms that provide routing solutions allowing network lifetime to be extended near or equal to the optimal off-line calculated solution [9]. Many others have written about ways to achieve energy-efficiency in routing protocols [10] [11] [12].

Many researchers have considered sensor management from a variety of perspectives [13] [14] [15] [16]. 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.* [17], and SPAN, proposed by Chen *et al.* [18], 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.* [19], 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 has attempted to minimize power consumption by reducing communication, not much has been done to modify sensor modes (and in turn reduce the amount of traffic generated in the network) when the amount of data being collected is more than necessary. Tilak *et al.* have shown the advantages of modifying the sensor mode (reporting rate) so that an efficient point on the reliability curve is met [13]. In their work, the limiting factor being considered was network congestion rather than sensor energy.

The work of Bhardwaj *et al.* is the most similar to that described in this paper [20]. Bhardwaj solves a similar sensor management optimization problem from a slightly different perspective. Our work considers the problem from a high-level perspective, and is flexible in terms of finding feasible sensor sets for a variety of sensor network applications. The model described in this paper was intended for applications which spend the most of their time in a vigilant state, with alarm states occurring infrequently and for short periods of time. Meanwhile, the work of Bhardwaj *et al.* is more focused on efficiently sending event-triggered sensor data to the data sink, and the model is aimed more toward tracking-type applications.

7 Conclusions and Future Work

Intelligent sensor management is one way to provide quality of service to an application. We have formalized a sensor management problem and shown through simulations that for applications using spatially diverse sensors, several heuristic scheduling/routing approach can achieve near optimal lifetime.

In our model, average power consumption of a node to route data is proportional to the routing load. In real networks, this power cost may actually depend on the state of the node. For example, in an IEEE 802.11 network, the marginal power cost to route data for a node already being used as an active sensor in the current set is significantly less than for one that is otherwise idle. For the active sensor, the energy cost is approximately proportional to the difference between idle power and the average of the receive power and transmit power. Meanwhile, a sensor that is not being used in the current sensor set would otherwise have its radio turned off, and the marginal cost is proportional to the sum of the receive power and transmit power. Similarly, if a node is being used to route data in multiple paths, its energy cost per forwarded packet is significantly less. When modeling such types of networks, the problem can no longer be modeled as a generalized maximum flow problem and becomes NP-hard, requiring a heuristical approach to solve.

While not representative of all networks, we believe that our model does represent some typical networks that are likely to be used in sensor networks. Many TDMA-based networks can be modeled as having power dissipation that is nearly proportional with the routing load, as we have modeled. Such networks include Bluetooth networks when an intelligent scheduling method is used. Still, even in these TDMA-based networks, traffic schedules may need to be set up to allow for this efficiency, and this overhead is not accounted for in our model.

In all simulations, we did not consider the overhead of setting up and tearing down routes. We acknowledge that in real situations this could in fact impact average power consumption and network lifetime and our model would need to account for these additional factors before running the optimization program. It should also be noted that these simulations were simplified and that quirks of specific routing protocols and other factors could cause discrepancies in performance measurements. Our approach also requires global information about the neighborhoods of each node. For small networks, this may not be a problem, but requiring this information to be propagated back to the base station would not scale well for larger networks.

The motivation for this work was a larger project in which we are developing a middleware system for use in wireless sensor networks. We will incorporate the ideas and strategies presented in this paper during the development of this middleware. We are also developing distributed algorithms in which sensors use information from neighboring nodes, gathered through eavesdropping, to determine what their current network role should be. We hope that these algorithms will achieve lifetime results that are near the upper bounds that can be found by the methods described in this paper.

Acknowledgements

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

References

[1] A. Murphy, W. Heinzelman, Milan: Middleware linking applications and networks, TR-795, University of Rochester, Computer Science (November 2002).

[2] K. Nahrstedt, D. Xu, Qos-aware middleware for ubiquitous and heterogeneous environments, IEEE Communications Magazine 39 (11).

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

[4] M. Perillo, W. Heinzelman, Optimal sensor management under energy and reliability constraints, in: Proceedings of the IEEE Wireless Communications and Networking Conference, 2003.

[5] L. Khachiyan, A polynomial algorithm in linear programming (in russian), Soviet Mathematics Doklady 20.

[6] University of rochester center for future health, http://www.futurehealth.rochester.edu (2002).

[7] S. Singh, M. Wu, S. Raghavendra, Power aware routing in mobile ad hoc networks, in: Proceedings of the Fourth Annual International Conference on Mobile Computing and Networking, 1998.

[8] Q. Li, J. Aslam, D. Rus, Online power aware routing in wireless ad hoc networks, in: Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking, 2001.

[9] J. Chang, L. Tassiulas, Energy conserving routing in wireless ad hoc networks, in: Proceedings of INFOCOM, 2000.

[10] D. Braginsky, D. Estrin, Rumor routing algorithm for sensor networks, in: Proceedings of the International Conference on Distributed Computing Systems (ICDCS-22), 2001.

[11] C. Intanagonwiwat, R. Govindan, D. Estrin, Directed diffusion: A scalable and robust communication paradigm for sensor networks, in: Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking, 2000.

[12] Y. Xu, J. Heidemann, D. Estrin, Geography-informed energy conservation for ad hoc routing, in: Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking, 2001.

[13] S. Tilak, N. B. Abu-Ghazaleh, W. Heinzelman, Infrastructure tradeoffs for sensor networks, in: Proceedings of the ACM International Workshop on Wireless Sensor Networks and Applications Workshop, 2002.

[14] M. Chu, H. Haussecker, F. Zhao, Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks, International Journal of High Performance Computing Applications .

[15] J. Liu, P. Cheung, L. Guibas, 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.

[16] S. Meguerdichian, S. Slijepcevic, V. Karayan, M. Potkonjak, Localized algorithms in wireless ad-hoc networks: Location discovery and sensor exposure, in: Proceedings of MobiHOC, 2001.

[17] C. Schurgers, V. Tsiatsis, S. Ganeriwal, M. Srivastava, Optimizing sensor networks in the energy-latency-density design space, IEEE Transactions on Mobile Computing 1 (1).

[18] B. Chen, K. Jamieson, H. Balakrishnan, 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.

[19] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, IEEE Transactions on Wireless Communications 1 (4).

[20] M. Bhardwaj, A. Chandrakasan, Bounding the lifetime of sensor networks via optimal role assignments, in: Proceedings of INFOCOM, 2002.