2000-10-17

Utility-Based Decision-Making in Wireless Sensor Networks

Utility-Based Decision-Making in Wireless Sensor Networks *

John Byers Gabriel Nasser
byers@cs.bu.edu comanche@cs.bu.edu

Computer Science Department
Boston University
Boston, MA 02215

October 17, 2000

Abstract

We consider challenges associated with application domains in which a large number of distributed, networked sensors must perform a sensing task repeatedly over time. For the tasks we consider, there are three significant challenges to address. First, nodes have resource constraints imposed by their finite power supply, which motivates computations that are energy-conserving. Second, for the applications we describe, the utility derived from a sensing task may vary depending on the placement and size of the set of nodes who participate, which often involves complex objective functions for nodes to target. Finally, nodes must attempt to realize these global objectives with only local information. We present a model for such applications, in which we define appropriate global objectives based on utility functions and specify a cost model for energy consumption. Then, for an important class of utility functions, we present distributed algorithms which attempt to maximize the utility derived from the sensor network over its lifetime. The algorithms and experimental results we present enable nodes to adaptively change their roles over time and use dynamic reconfiguration of routes to load balance energy consumption in the network.

*This work was partially supported by NSF research grant ANIR-9986397.

1 Introduction

Networks of small, inexpensive, low-power sensors are widely expected to provide cost-effective solutions for applications ranging from environment monitoring to collection of visual data for the purposes of scene reconstruction, to motion tracking and motion detection. Among the most significant challenges in gathering data from these wireless, ad-hoc sensor networks arises from the array of decisions that individual nodes must make with only local information. In particular, nodes are initially expected to form a connected ad-hoc network over which to communicate and are subsequently expected to run distributed algorithms to route around faults, while simultaneously performing sensing tasks and managing both local and global energy consumption.

For these reasons, there has been relatively limited discussion to date about applications which have firm guarantees about collection and delivery of sensory data over the sensor network. Instead, typically a *best-effort* service model is either explicitly or implicitly assumed, meaning that sensors are expected to perform sensing operations and route data to the destinations as best they can. Of course, issues such as node failure and network partition may temporarily or permanently short-circuit these plans. In this paper, we argue that for many interesting applications of wireless sensor networks, even a best-effort service model may in fact be too stringent. While having a node perform a sensory operation or route a message containing sensory data may be beneficial to the application, the benefit must be measured as a function of the cost. In economic terms, the *opportunity cost* associated with performing a particular activity may be large, which, if it outweighs the benefit of performing the activity, warrants perfoming another alternative activity. In the context of sensor networks, nodes may act either as sensors, or as routers, or both. Since both activities are beneficial, but energy-consuming, a node's choice of role is essential to achieving the objectives we specify. We argue for a model in which sensor nodes are allowed to make these choices, rather than adopt a best-effort mentality.

In the model we develop, the cost of an action is relatively easy to quantify, as we focus on managing energy consumption, and we measure costs in those terms. However, the distributed nature of nodes in our networks implies that they do not have global information. This further implies that it is unrealistic to expect nodes to accurately assess either the opportunity costs, or the relative benefits of a particular decision. Instead, we adopt a model in which nodes make heuristic assessments based on available information, which is often local in nature. This model is driven by objective functions which maximize the *utility* of a sensor network over the lifetime of the network.

To motivate the nature of the issues we consider more fully, we describe the underlying assumptions we make about the sensor domains we consider:

- The topology of the network connecting the sensor nodes can change frequently, either due to mobility, energy considerations, or permanent node failure.
- Sensory data must be routed to a wired base station adjoining the sensor network before it can be accessed by the end-user.
- Nodes are homogeneous and are powered by a finite, non-rechargeable energy supply.
- Remote deployment or cost considerations make human intervention infeasible; nodes are simply expected to continue collecting data until they exhaust their power supply.
- Conserving power to maximize the uptime of the network is essential.

In environments such as this, for the objectives we seek to address, computation in large-scale sensor networks will require scalable *coordination* amongst sensors to accomplish the desired tasks [6], In most circumstances, the sensors must coordinate to achieve one or more *global* objectives, but yet must do so with only *local* information. Distributed algorithms to achieve global objectives using local information have been widely studied in the context of classical networking problems (see for example [1, 2]). In this paper we consider global objective functions motivated by specific sensor network applications which are driven by utility functions, first studied in a networking context by Shenker [18]. Developing solutions which achieve these objectives are constrained in two primary ways: by the locality imposed by the distributed nature of the model, and by a resource constraint, namely the finite energy supply at sensor nodes. Our work develops a general model in which to study such problems and presents algorithmic results and experimental work in progress for a class of these problems.

The paper is organized as follows. We start by summarizing the related work in the context of power-aware routing and computation in sensor networks. Section 3 specifies the model which we advocate, from basic issues of the topological layout to a detailed description of the utility functions and objective functions which drive our work, to a description of the benefits of aggregation. Section 4 presents the power-aware routing algorithms and heuristics we prescribe for a class of relatively simple motivating applications. In Section 5, we present our experimental results for these algorithms, using performance analysis to assess the success of our algorithm with respect to load balancing, energy consumption and total utility. Finally, we conclude the paper in Section 6.

2 Related Work

The main objective of our work is to motivate the design of algorithms for sensor networks which dynamically load-balances sensing and routing tasks to maximize the utility of the network in energy-constrained environments. While the objective functions and algorithms we propose are novel, they connect to a substantial body of work on ad-hoc routing protocols, fault tolerance, and energy conservation in sensor networks, which we survey here.

One aspect of our work leverages off of the considerable body of literature which has focused on improving adaptive routing protocols for communication in ad-hoc networks [20, 5, 10, 9, 19, 17]. In general, these protocols provide improved fault-tolerance and support for mobility, for example by establishing a routing backbone which can be updated dynamically by distributed algorithms which monitor the frequently changing network topology [20]. Another set of routing protocols specifically addresses the issue of power consumption [7, 21, 19] in ad-hoc sensor networks. These protocols concern themselves with issues of fault-tolerance and mobility, but also extend their models to address issues of power consumption in energy-constrained environments.

The concept of sensor fusion, or actively aggregating data from multiple heterogeneous sensory domains, has been used in [14, 8, 9], among others. Sensor fusion can also be applied in the context of aggregating homogeneous sensory information from multiple sensors. As an example, the Low-Energy Adaptive Clustering Hierarchy (*LEACH*) protocol [8] uses this form of sensor fusion to compress datasets within the network, reducing the energy dissipated during the resulting transmission. One application-specific example they describe is *beamforming* algorithms, which combine a set of acoustic signals into a single signal without loss of relevant information. Our work applies the same general principle in advocating application-specific data aggregation as a technique for conserving energy.

Finally, there are several sensor network efforts which attempt to load balance energy-consuming tasks in the network. *LEACH* [8] performs load balancing by randomized timesharing of the responsibility (and cost) of long-distance transmissions needed to offload sensory data to a remote base station. The applications we describe motivate a different form of load-balancing, which we achieve not by distributing transmission cost, but by distributing the sensing responsibilities. This approach to node specialization is somewhat similar to the idea of device modes in [7]. In that approach, the role of a node is determined as a function only of a node's current power levels, whereas we determine a node's role by considering the marginal cost and marginal benefit of a particular change in its assignment.

3 Model

In this section, we begin by motivating the various aspects of our model by outlining the details of problem requirements for an application from the realm of environment monitoring. Our description of the model follows in three subsections, the first describing the basic notation and assumptions about the network, the second describing constraints imposed by the sensory applications of interest, and the last describing the objective functions we seek to optimize.

3.1 A Motivating Example

Consider the problem of monitoring toxicity levels in an area in which hazardous materials are used and hazardous waste is produced. Due to the nature of the environment, the logistics and cost of deploying sensors, the deployment of sensors is a one-time operation; therefore, human intervention after the sensors have been deployed is not an option. To relay information off of the network, the sensors which are deployed are equipped with wireless communication devices with which they may communicate data to an adjoining base station. In the course of transmitting data, nodes can aggregate data collected at various sensors into summaries to reduce messaging overhead. In the best case, message size would be independent of the aggregation level.

Power consumption is perhaps the most significant consideration in this example, since we assume that the sensors do not have a renewable energy supply. Likewise, the sensors are collectively expected to monitor the environment for as long as possible, so they must preserve their energy reserves. We assume that both the sensing (or data gathering) operations and the transmission of data through the network incur non-negligible costs. As the number of nodes which relay a datum of sensory information increase, the total utility also increases, but we assume *diminishing marginal returns*, i.e. the benefit of an additional node's participation becomes less and less significant. The tension introduced between the goal of data gathering over large time intervals and that of recovering detailed, precise sensory information motivates a utility-driven approach to the distributed management of the sensor network.

3.2 Network Model

We start with the basic definitions and assumptions describing the networked environment in which our sensors operate. In many cases, these assumptions can be relaxed or altered without substantially changing the underlying model which we develop. We begin by assuming that all nodes communicate over a homogeneous wireless medium. Using the definitions provided in [20] and elsewhere, we define the *neighborhood* of a node u to be all the nodes that are both within u's transmission range and that are operational. We assume that communication among nodes is *commutative*, i.e. if node u can receive transmissions from node v, then node v also can receive transmissions from node u. All transmissions from any node v are *isotropic*, that is, they are omni-directional and they reach all nodes in the neighborhood of v. For simplicity, we also assume that transmissions are *perfectly scheduled*, so that no one node's transmission interferes with another, thereby avoiding collisions [4]. Furthermore, to conserve power, we assume that nodes power down temporarily when they overhear the beginning of a transmission for which they are not the intended receiver. While we will generally assume that the nodes forming our ad hoc network are stationary for the algorithms we develop, this assumption is not an inherent limitation of our model.

We represent our network as an undirected graph $G = (V, E)$, where V is the set of all nodes, including the base stations. E is the set of edges in the network defined as follows, where $d(u, v)$ is the distance between nodes u and v.

$$E = \{(u, v) \mid u, v \in V \text{ and } d(u, v) \leq R \text{ and } u, v \text{ are operational}\}$$

This graph is similar to a *unit graph* [3, 11, 21], in which all nodes' transmission ranges are equal. See Figure 1 for an example. The neighborhood of a node u is denoted by the set

$$N(u) = \{v \mid (u, v) \in E\}.$$

All nodes in our network are homogeneous[1] and have the following properties.

1. Each node i has a unique identifier, id_i, which serves as its address for all transmissions intended for it.

2. Each node i has a fixed, finite, and non-replenishable reserve of energy which we denote by p_i.

3. To achieve isotropy, each node has a fixed communication radius, R, which delimits its range of the transmission.

In our basic model, a single wired base station, through which all information is relayed off of the sensor network, has a different set of special properties. For clarity, we assume this node has $id = 0$, an infinite reserve of energy $p_0 = \infty$, and a fault rate $f_0 = 0$. With some additional notational complexity, our model and techniques can easily be generalized to scenarios in which multiple base stations are present. Figure 1 depicts a layout of nodes within the confines of the outer circle, with the communication range of two sensor nodes depicted by the two inner circles.

As nodes fail due to battery depletion, or permanent failure and restart after temporary failures, the corresponding network graph changes. In particular, V and E provide a dynamically changing reflection of the state of the nodes and the possible communication among pairs of nodes, respectively.

3.3 Sensing Model

We describe the sensing model we use in our network. We present the different costs associated with each operation. We explain our notion of *node specialization*, a node role-based mechanism that adapts to changes in the network, loosely similar to the idea of device modes in [7].

[1] Except in the case of base stations which we describe further on.

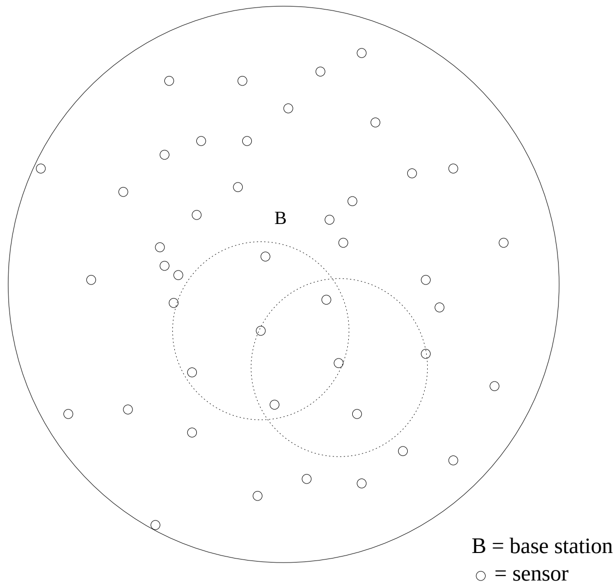

Figure 1: Sensors Distributed Uniformly at Random on a Unit Circle

3.3.1 Specialization

Network topology changes are frequent in ad-hoc networks. In our case of static networks, changes occur to the virtual topology as nodes fail terminally and deplete their energy. Such networks warrant the use of adaptive and fault-tolerant routing algorithms [5, 13, 17]. We propose to complement the role of adaptive algorithms with nodes that adapt their role, or *specialize*, as a response to changes not only in the virtual topology of the network but also to node power levels. In general, we classify the typical roles of a sensor node as follows.

- **Routing Only:** Due to large opportunity costs from a node's position in the topology or due to its current low energy levels, a node refrains from sensing in favor of routing data now and at future timesteps.
- **Sensing Only:** Nodes with no active children in the topology and who decide to sense operate in this mode.
- **Routing/Sensing:** Nodes whose energy reserves are sufficient to perform sensing as well as routing may do so.
- **Idle:** Nodes with no active children in the topology and who decide not to sense are idle and consume no energy.

3.3.2 Costs

We assign costs to the different operations performed by a node as follows.

- Let c_s be the cost associated with a single sensing operation. A sensing operation consists of a single probe of the environment in which a node collects experimental data amounting to a sensory task.

- Let c_t be the cost to transmit a fixed-size message containing the result of a sensory computation to a node's neighborhood.
- Let c_r be the reception cost, which is incurred when a node receives a fixed-size transmission for which it is the intended receiver. (We disregard the cost associated with "listening" for transmissions as it is substantially smaller than transmission and reception costs [7].)
- Let c_a be the fixed aggregation cost. That is, the cost of applying the computation to the sensed data and the collected data. In many contexts, this computation is denoted as sensor fusion.

In our model, the ratio $\frac{c_s}{c_t}$, or the sense-to-transmit ratio, plays a significant role as it captures the relative importance of the two most important roles of nodes in the network in terms of energy consumption.

3.3.3 Aggregation

For many applications, it is not a requirement that the sensory data accumulated by the nodes of the network must be transmitted in full fidelity to the base station. In the monitoring example, nodes can transmit a minimal amount of information to convey the fact that very little has changed in their subtree since the last measurement and the toxicity levels all remain nominal. Or, when toxicity levels remain low, it might be sufficient to report an average measure over the subtree, rather than the value at each leaf. In these circumstances, the amount of work, in terms of messaging complexity, remains fixed at all levels of the tree, thus consumes substantially less power than full reporting. For these applications, we assume that data is aggregated at each node with a fixed aggregation cost before it is transmitted upstream in the network. That is, each node collects data from its children in the tree, performs its own sensing (when applicable), applies the computation to both sets of data before transmitting the result to its parent. Use of aggregation allows for better scalability, since it does not suffer from message implosion inherent in the monitoring example just described or in the case of deploying thousands of small sensors in disaster areas [6].

3.4 Utility Functions and Objective Functions

We associate each sensor domain with a monotonically non-decreasing utility function which maps the number of nodes participating in a sensory computation to a real value which quantitatively measures the utility derived from output from a subset of sensors of that size. Shenker [18] motivated the use of utility functions in quantitatively modeling a user's relative preference for a real-time stream encoded at varying levels of fidelity. Our motivation is similar – the user is the consumer of the output of the sensor network and the varying levels of fidelity correspond to increasingly detailed sensory output levels. Therefore we model the utility derived from a consumer of our sensor network resources by a monotone function

$$U : S \to [0, 1],$$

which, for a network graph $G = (V, E)$, maps the *sensing subset* $S \subseteq V$, the set of all nodes in the graph that are sensing, to a real-valued interval. It is worth noting that for most applications, not all subsets of sensors of a given size are created equal – in many circumstances, having a geographically distributed set of reporting sensors is essential. Addressing such spatial considerations is beyond the scope of the algorithmic and experimental work we present in this paper, but we will mention that these considerations could easily be modeled by a more general (and non-monotonic) utility function mapping all possible subsets of nodes to values:

$$U : S^* \to [0, 1],$$

where S^* denotes the power set of S.

We base our approach on the observation that not all nodes need to contribute data to the computation and, therefore, nodes can conserve their energy so that data aggregation may be performed over longer periods of time. As in [18], for simplicity, we will mainly concern ourselves with the qualitative aspects of our utility function. For many applications, it is not necessary to have the most highly optimized output at a given timestep, instead, we take advantage of opportunities to trade off the utility derived from a computation against the amount of power consumed. We discuss two general types of curves for modeling utility functions (also described in [18]) and as diagrammed in Figure 2. First consider the utility curve to be represented by the step function on the left hand side of Figure 2. In this all-or-nothing case, useful data fusion is only possible when and only when the number of nodes participating in the sensing operation is at least as large as the threshold set by the function. In a more forgiving scenario, our utility function might resemble the inelastic curve on the right hand side of Figure 2, where we can have some freedom in tuning the number of participating nodes to vary energy consumption in the network. This second curve has three regimes: when a very small number of nodes participate, the user derives little utility; at a certain threshold, the utility quickly increases dramatically; and then beyond a final inflection point, there are diminishing marginal returns and utility increases only very slowly. In both of these scenarios, ideally one would like to operate at the beginning of the third regime, just beyond the knee of the curve, to maximize utility relative to power consumption.

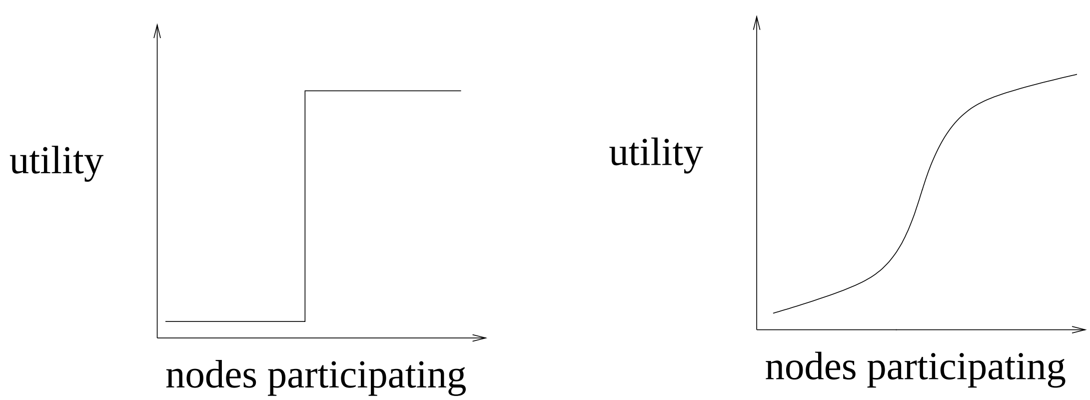

Figure 2: Utility functions: Step function and inelastic utility function

.

In many applications of ad hoc networks, the objective is not simply to perform a small number of high quality sensing operations, but rather to complete a large number of computations over longer timescales. In this context, fusing data from the maximum number of sensors is a short-sighted approach in a power-constrained environment. In this work, we motivate maximizing the utility of sensory operations over the duration of the network uptime.

The objective function which we propose is one in which we maximize the sum, over the lifetime of the sensor network, of the utility of computations at intermediate timesteps. This objective reflects a natural goal – that of maximizing the total aggregated utility of the network over time. It would of course be possible to formulate other similar objectives in this framework. More formally, let us define those vertices which elect to perform a sensing operation at any time t as the sensing subset $S_t \subseteq V$. Similarly, we denote those vertices which elect to transmit data at any time t as the transmitting subset $R_t \subseteq V$. By our model, it follows that $S_t \subseteq R_t$, moreover, R_t must be established in such a way that enables all nodes in the sensing subset to route their data to the base station. With this formalization now in place, our objective function is the optimization problem:

$$\text{maximize } \sum_t U(S_t)$$

given the constraints:

$$\sum_t \sum_{i \in S_t} c_s + \sum_t \sum_{i \in R_t} c_t + (d_i - 1)c_r + c_a \leq p_i \tag{1}$$

$$\forall t : R_t \text{ induces a connected subgraph of } G \text{ spanning } S_t \text{ and } v_0 \tag{2}$$

The first constraint of this formulation uses the cost model defined earlier to ensure that nodes cannot consume more power than they have available, where d_i denotes the degree of node i in the subgraph induced by R_t. The second constraint ensures that the data collected from all nodes who get credit for participating in the sensing subset at time t actually gets routed to the base station.

This long-term strategy can be realized only through a combination of careful power management combined with distributed coordination on the part of the nodes in the sensor network in choosing their roles over time. We present algorithms for doing so in the next section.

4 Adaptive, Energy-Efficient Algorithms for Utility Maximization

Our algorithm exhibits several desirable properties for routing protocols as proposed by [6, 12].

- **Loop-freedom:** All routing and communication is performed over a logical spanning tree of the network
- **Localization:** Our algorithm is distributed and message exchanges among nodes are localized in that they take place within neighborhoods.
- **Non-proactivity:** Route computations are recomputed on an as-needed basis using a lazy evaluation approach.

The algorithm runs in two main stages, a setup stage in which the ad hoc network is established by distributedly building a spanning tree over the operational nodes; and a processing stage, in which the distributed algorithm performs its core duty of choosing roles, sensing and routing and handling faults by reconfiguring the tree (whenever possible).

Sensors in our algorithm take advantage primarily of local information, such as the network topology in their neighborhood, their remaining energy level and their depth in the spanning tree. As a consequence of computation on the tree, internal nodes which act as routers have the ability to learn a limited amount of information about their subtree. For example, we will assume that during the course of aggregation, a node can and will learn the magnitude of the set of nodes which are currently performing a sensing role in its subtree.

4.1 Notation and Terminology

We use a discrete time model to describe the operation of our algorithm in simple terms. The algorithm operates in a sequence of distributed rounds, where each round consists of a sequence of substeps performed at all nodes as defined below. The substeps themselves are performed asynchronously, in particular some nodes may be idle while others are actively engaged in computation. A node may remain idle during a substep and incurs no energy cost if its role dictates that it need not perform the operation associated with that substep. For the following discussion, *leaf* nodes are those nodes which have established the sensing role, but do not route data. The remaining nodes are *internal* nodes, who route data, and may undertake the role of sensing as well.

The substeps are defined as follows:

- **Sensing substep:** A node gathers sensory information from the environment and its energy reserve is depleted by the associated amount.[2]
- **Receiving substep:** Each internal node waits until it has received transmissions from each of its active children.
- **Aggregation substep:** Internal nodes aggregate the sensory data from their children, along with their own data, consuming a constant amount of power;
- **Transmission substep:** The raw sensory data collected by a leaf node, or the aggregated data produced at an internal node, is transmitted upstream, incurring a constant cost;
- **Feedback substep (Rare):** In the event that reconfiguration is necessary, a constant-size message may be broadcast down the tree.

We say that an internal node completes a *full* round when it does work during each of the substeps (i.e. specializes as a router/sensor). The amount of energy that node i consumes during such a round is

$$E^i_{full,n} = c_s + n \cdot c_r + c_a + c_t$$

where n is the number of children that sent transmissions to this node during the second substep. When a node's specialization is that of a *router* only then the amount of energy it consumes during a round is

$$E^i_{route,n} = n \cdot c_r + c_a + c_t$$

A node completes a *leaf* round when it only senses and transmits its own data, thus consuming

$$E^i_{leaf} = c_s + c_t$$

4.2 Network Initialization

The algorithm distributedly constructs a spanning tree rooted at the base station in the style of [15]. The algorithm is described as follows. Initially, all the nodes are idle except for the base station.

[2]This substep may consume a significant quantity of time, as sensory information may be gathered over much larger time scales than that required to route the data to the base station

Every newly-activated node transmits an active ping, which is a short message consisting of a pair of values (id, d), where id is the unique identifier of the transmitting node and d is the distance, in hops, of that node from the root node. For example, the base station always transmits the pair $(0, 0)$ since its id is always 0 and it is situated at a distance of 0 hops from the root (itself). Any idle node that is functioning properly and within range of the issuing node, registers the issuing node as its parent and notes its distance plus one. Once an idle node receives an active ping it becomes active. Any active node that receives subsequent active pings from any node, compares the new ping pair (id_{new}, d_{new}) with its current registered pair (id_{reg}, d_{reg}). If $d_{new} < d_{reg}$, then it discards its old pair and stores the new pair. Formally, given the representative graph $G = (V, E)$ on nodes reachable from the base station of the sensor network, the algorithm constructs a spanning tree[3] $T = (V', E')$, where $V' \subseteq V$ is a set that includes all the operational nodes that are on a path starting at the base station, where $E' \subseteq E$ is the set of all edges in the minimum spanning tree.

4.3 Fault Tolerance: Inheriting Orphaned Subtrees

A node can fail permanently due to power depletion. When a node fails permanently it is no longer part of the network. If that node happens to be a *leaf* then other nodes will eventually adjust their roles appropriately in a manner we describe momentarily to restore an acceptable level of computation as dictated by the utility function. Otherwise, the failure of an *internal* node results in the partition of the network tree. In such a case, its children, themselves roots of other subtrees, become orphans. These roots need to reconnect their orphaned subtrees to the main network. Formally, let $G = (V, E)$ be the representative graph of a network, and $T = (V, E')$, $E' \subseteq E$ a spanning tree over G. When an internal node $u \in V$ fails, the network is partitioned into the primary partition $P \subset V$, which includes the base station $v_0 \in P$, and secondary partitions (also spanning trees) $S_i \subset V - P$, whose roots v_{S_i} are orphans. Each orphan v_{S_i} transmits a *search ping* message, indicating that it is looking for a parent. The message is received by all the nodes in the orphan's neighborhood, $N(v_{S_i})$. The set of prospective parents for the orphan can be described as

$$H(u) = \{v \mid v \in P \text{ and } v \in N(u)\},$$

where the prospective parent v must be in the neighborhood of u and must be in the primary partition P (and not in any secondary partition). Any parent who is able and willing to "adopt" the orphaned node will send back a message indicating so. The orphan then chooses the willing parent at minimum depth in the tree.

4.4 Selecting Roles: Maintaining a Sensing and Routing Invariant

With the spanning tree and fault tolerance mechanisms in place, we are ready to define the specialization mechanism whereby nodes select roles. As described earlier, nodes who have not depleted their energy supply choose one of four roles at any given timestep.[4] Nodes at the periphery of the network, who cannot communicate with the base station cost-effectively, are idle. Nodes are at the leaves of the communication tree established for a given timestep are sensing only. The remaining nodes act as routers, but may also decide to act as sensors. To streamline this decision process and to conserve energy near the base station, our sensor network maintains the following invariant:

[3] All trees constructed over unit graphs are minimum spanning trees

[4] Nodes which have depleted their energy are removed from the graph.

Invariant: In the communication tree spanning all active nodes, no node acting as a router and a sensor may be an ancestor of a node acting only as a router.

It is easy to see that a consequence of this invariant is that nodes form four concentric regions according to their roles as depicted in Figure 3.

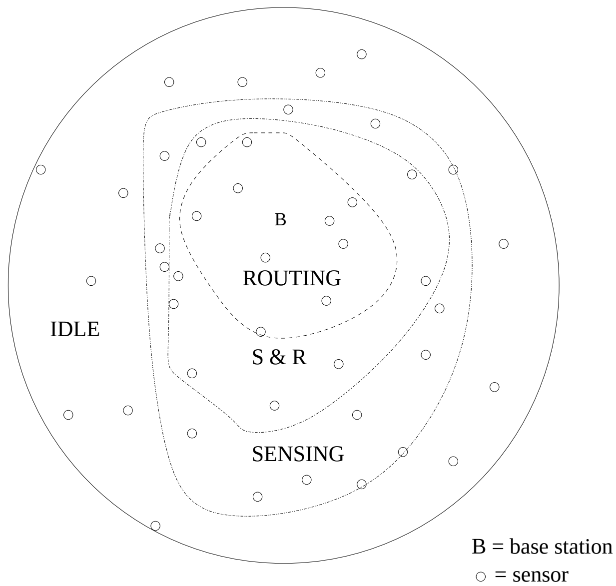

Figure 3: Graphical Depiction of the Specialization Invariant

The remaining question is for nodes to determine their role. In our approach, each node determines its role as follows. Node i computes a random value, $0 < r_i \leq 1$. The pseudo-code for the algorithm is given in Figure 4. N_{tot} is the threshold set by the utility function, and is known to all nodes. N_{curr} is the number of nodes that performed sensing in the previous round; if this value differs from N_{tot}, then it is propagated down the tree during the feedback substep. N_{idle} is the number of nodes that are part of the spanning tree but were idle in the previous round; this value is also propagated down the tree during the feedback step. If additional nodes need to sense, that is if $N_{tot} > N_{curr}$, then node i probabilistically chooses to complement its current role with a sensing operation, if and only if it does not violate the invariant. Similarly, if $N_{curr} > N_{tot}$ then node i probabilistically chooses to throttle back its role to just routing, if its current role was sensing and routing, or to idle, if its current role was to sense, without violating the invariant. The probabilistic computations made by the nodes are given in Figure 4.

5 Experimental Results

We now describe preliminary results obtained in a discrete event simulation of the algorithms described in the preceding section. In this simulation, we uniformly generate N nodes on a unit-radius circle at random and we position the base station in the center of the circle, as was depicted in Figure 1. We then fix the relative costs of sensing, transmitting and receiving relative to the initial power at each node. We also fix the transmission range $R = 0.2$. In our experiments, we consider the 0/1 utility step function, setting the threshold as a function of N, i.e. a setting of $N/4$ implies that full utility is derived when and only when at least $N/4$ nodes act as sensors. Future

```
Routing-Alg()
   role := idle;
   repeat
      generate r_i;
      if N_tot > N_curr then          /* Need to increase sensors */
         ε := (N_tot − N_curr) / N_idle;
         if r_i < ε then
            if leaf then
               role := sense;
         /* Maintain the invariant */
            else if all children are sensing then
               role := sense_and_route;
      else                /* Need to reduce sensors */
         ε := (N_curr − N_tot) / N_curr;
         if r_i < ε then
            if leaf then
               role := idle;
            else
               role := route;
   until power is depleted
```

Figure 4: The Adaptive Routing Algorithm at Node i

work considers the effects of weakening the stringent requirement imposed by a step function, for example as depicted in the utility function on the right hand side of Figure 2. The utility function is known to all nodes, and we set the algorithm's target number of participating nodes T to exactly the threshold value specified above.[5]

5.1 Achieving the Objective

In Figure 5, we present a histogram of the number of nodes participating over time using the algorithms described in the preceding section performance averaging over 100 trial topologies with $N = 200$, a target $N/4 = 50$, a sense-to-transmit ratio $\frac{c_s}{c_t} = 4$, a transmit-to-receive ratio $\frac{c_t}{c_r} = 2$ and a power supply p at each node capable of 125 sensory operations. As the histogram indicates, the overwhelming majority of timesteps had participation levels exactly at the target level. The exceptions arose during initialization, in which large numbers of nodes may participate unnecessarily, during reconfiguration, when we may briefly slip below the threshold, and during termination, when an insufficient number of nodes participate.

[5]One could also consider introducing a margin of error between the utility threshold and the target value, but our preliminary results suggest this does not introduce a significant effect.

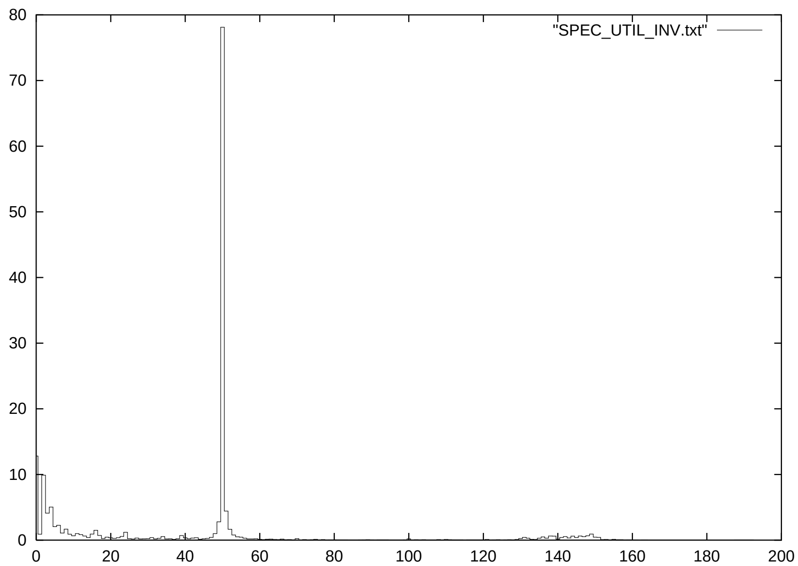

Figure 5: Distribution of Node Participation Over Time

Heuristic	Lifetime	Total Utility	Total Energy Consumed	Utility/Energy
NAIVE	92.98	62.3	22039.3	.002827
SIMPLE_UTIL	349.633	64.855	15418.01	.004206
SPEC_INV	216	104.87	14505.54	.00723

Figure 6: Comparison of Heuristics

5.2 Success Relative to Alternative Approaches

In Figure 6, we present a comparison between the performance of three algorithms averaging over 150 trial topologies for $N = 200$, a sense-to-transmit ratio $\frac{c_s}{c_t} = 4$, a transmit-to-receive ratio $\frac{c_t}{c_r} = 2$ and a power supply p at each node capable of 125 sensory operations. **NAIVE** is a simple naive, best-effort algorithm. The second algorithm, **SIMPLE_UTIL**, is utility-aware and always tries to operate at the threshold set by the utility function[6]. Finally, **SPEC_INV** is the full algorithm described in the previous section. Over these trials, we measure the average lifetime of the network, the average total utility derived, the average total energy consumed, and the cost of utility per unit energy. Algorithms which are successful in this context are those which derive the highest utility while consuming the minimum amount of energy; Table 6 shows that **SPEC_INV**, the full algorithm, achieves the highest such ratio.

6 Conclusion

We have explored objective functions for computations over wireless sensor networks in which it is neither a requirement, nor a desired goal, to have all nodes participate in the capacity of both a sensor and a router at all timesteps. Economic considerations, especially resource constraints

[6] We assume that this algorithm knows how to exactly achieve the threshold by as we make the number of nodes sensing available to all the participants in the tree

and the prevalence of network faults, motivate objective functions which give substantial freedom in letting nodes choose their role over time. In a sense, this is a departure from the traditional best-effort service model that underlies much of the design philosophy of standard internetworking protocols. In a best-effort model, nodes attempt to optimize the utilization of resources in the present without regard to future cost. But in sensor networks, energy considerations force nodes to take a longer term view, optimizing their resource utilization over an uncertain future.

In our work, we model application-level flexibility with utility functions, which are capable of succinctly and quantitatively expressing a measure of service that the sensor network provides. With the goal of optimizing the total utility derived over time, the distributed algorithms we present for this model enable nodes to successfully discount current gains in lieu of future rewards; thereby highly optimizing their consumption of energy over time. Our work in progress explores the numerous directions and questions raised by this study.

References

[1] B. Awerbuch and Y. Azar. Local Optimization of Global Objectives: Competitive Distributed Deadlock Resolution and Resource Allocation. In *Proc. of the 35th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 240–9, October 1994.

[2] Y. Bartal, J. Byers, and D. Raz. Achieving Global Objectives using Local Information with Applications to Flow Control. In *Proc. of the 38th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, Miami Beach, FL, October 1997.

[3] S. Basagni, I. Chlamtac, V.R. Syrotiuk, and B.A. Woodward. A Distance Routing Effect Algorithm for Mobility (DREAM). In *Proc. of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 76–84, Dallas, TX, October 1998.

[4] V. Bharghavan and B. Das. Routing in Ad-hoc Networks Using Minimum Connected Dominating Sets. In *International Conference on Communications '97*, Montreal, Canada, June 1997.

[5] M. S. Corson and A. Ephremides. A Distributed Routing Algorithm for Mobile Wireless Networks. *ACM Journal for Wireless Networks*, pages 61–81, February 1995.

[6] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next Century Challenges: Scalable Coordination in Wireless Networks. In *Proc. of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 263–270, Seattle, WA, 1999.

[7] J. Gomez, A. T. Campbell, M. Naghshineh, and C. Bisdikian. Power-Aware Routing in Wireless Packet Networks. In *Proc. of Sixth IEEE International Workshop on Mobile Multimedia Communications*, San Diego, CA, November 1999.

[8] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocols for Wireless Microsensor Networks. In *Proc. of Hawaiian Int'l Conf. on Systems Science*, January 2000.

[9] W. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive Protocols for Information Dissemination in Wireless Sensor Networks. In *Proc. of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 174–185, Seattle, WA, 1999.

[10] D. Johnson and D. Maltz. *Dynamic Source Routing in Ad Hoc Wireless Networks*, chapter 5, pages 153–181. Kluwer Academic Publishers, 1996.

[11] X. Lin and I. Stojmenovic. GPS Based Distributed Routing Algorithms for Wireless Networks. Unpublished manuscript.

[12] J. Macker and M. Corson. Mobile Ad-hoc Networking and the IETF. *Mobile Computing and Communications Review*, 1998.

[13] P. Merlin and A. Segall. A Failsafe Distributed Routing Protocol. *IEEE Transactions on Communications*, 27, September 1979.

[14] MIT Microsystems Technology Laboratory. The Ultra Low Power Wireless Sensors Project. URL = **http://www-mtl.mit.edu/~jimg/project_top.html**.

[15] R. Perlman. An Algorithm for Distributed Computation of a Spanning Tree in an Extended LAN. In *Proc. of the Ninth Symposium on Data Communications*, pages 44–53, 1985.

[16] R. Perlman and G. Varghese. Pitfalls in the Design of Distributed Routing Algorithms. In *Proc. of the Symposium on Communications Architectures and Protocols*, pages 43–54, 1988.

[17] B. Rajagopolan and M. Faiman. A New Responsive Distributed Shortest-Path Routing Algorithm. In *ACM Computer Communications Review*, 1989.

[18] S. Shenker. Fundamental Design Issues for the Future Internet. *IEEE Journal on Selected Areas in Communications*, 13:1176–88, September 1995.

[19] S. Singh, M. Woo, and C. Raghavendra. Power-aware Routing in Mobile Ad Hoc Networks. In *Proc. of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, Dallas, TX, 1998.

[20] P. Sinha, R. Sivakumar, and V. Bharghavan. CEDAR: a Core-Extraction Distributed Ad hoc Routing Algorithm. In *IEEE INFOCOM '99*, New York, NY, March 1999.

[21] I. Stojmenovic and X. Lin. Power-Aware Localized Routing in Wireless Networks. In *Proc. of IEEE International Parallel and Distributed Processing Symposium*, Cancun, Mexico, May 2000.