Maximizing Data Gathering Capacity of Wireless Sensor Networks using Mobile Relays

Fatme El-Moukaddem Eric Torng Guoliang Xing Department of Computer Science and Engineering Michigan State University East Lansing, MI 48824, U.S.A. {elmoukad, torng, glxing}@cse.msu.edu

Abstract-Recently, the availability of numerous low-cost robotic units (e.g., Packbot, Robomote, and Khepera) has made it possible to massively deploy mobile sensors in a network and use them in a disposable manner. It has been shown that the controlled mobility offered by sensors can be exploited to improve the energy efficiency of a network. In this paper, we study a new problem called max-data mobile relay configuration (MMRC) that finds the positions of a set of mobile sensors, referred to as *relays*, that maximize the total amount of data gathered by the network during its lifetime. Different from previous controlled mobility approaches, we account for several characteristics of existing practical mobile sensing platforms including limited mobility and the high energy consumption of locomotion. We show that the MMRC problem is surprisingly complex even for a trivial network topology due to the joint consideration of the energy consumption of both wireless communication and mechanical locomotion. We present optimal MMRC algorithms and practical distributed implementations for several important network topologies. Our extensive simulations based on realistic energy models of existing mobile sensing platforms show that our approach can increase the data gathering capacity by a factor of at least 2 in most scenarios. Moreover, our distributed algorithms converge quickly and incur low messaging overhead.

I. INTRODUCTION

Recent years have seen the emergence of numerous lowcost mobile sensor prototypes such as Packbot [1], Robomote [2], Khepera [3], and FIRA [4]. The manufacturing cost of these platforms is comparable to that of typical static sensor platforms. As a result, they can be massively deployed in a network and used in a *disposable* manner. Mobility has been traditionally treated as a curse to wireless networks due to its potential disruption of network topology. However, recent research has shown that the controlled mobility of sensors can be exploited to improve the energy efficiency of a network. In particular, by relocating mobile sensors, the communication topology of a network can be dynamically configured to reduce transmission power consumption. Moreover, mobile sensors can physically carry large chunks of data to the destination resulting in less energy consumption in wireless transmissions [1].

Several different approaches have been proposed to exploit controlled mobility in WSNs. First, mobile base sta-

tions (BSs) or relays may move around the network and collect/forward the data from static nodes [5], [6], [7], [8], [9], [10]. By optimizing the motion paths of mobile nodes, the static bottleneck nodes are rotated, which leads to balanced energy consumption and improved system lifetime. Second, mobile nodes may also serve as "data mules" that travel between static nodes and BSs and mechanically transport data [11], [12]. However, the above approaches are not suitable for utilizing low-cost practical mobile sensors that only have limited mobility. Designed under tight cost and resource budgets, most existing mobile sensor platforms are only equipped with 8-bit CPUs and simple locomotion units. For instance, flip-based sensors [13] can only hop for a short distance in a given direction for each movement. The total number of hops is also small due to limited battery capacity. The slow motion speed, poor maneuverability, and low localization accuracy of such platforms make it extremely challenging, if not impossible, to conduct any complex motion planning typically required by mobile BSs or data mules [5], [6], [7], [8], [9], [10]. Another major design concern for practical mobile sensors is the high energy consumption of locomotion. For instance, Robomote [2] nodes carry small on-board batteries that only last for about 25 minutes in full-speed motion. However, the existing approaches often do not account for the energy consumption of movement. Instead, mobile nodes are assumed to be able to replenish their energy supplies, e.g., by periodic recharging at a fixed charging dock [9]. However, energy replenishment is not always feasible due to the constraints of the physical environment or the limited mobility of nodes.

In this paper, we explore the use of low-cost disposable mobile relays to maximize the data gathering capacity of WSNs. Our approach has several key differences with the existing controlled mobility approaches based on mobile BSs or data mules. First, mobile relays in our approach do not mechanically transport data. Instead, they move to different locations and then remain stationary to help static nodes to forward data to the BS. In contrast to the existing approaches (*e.g.*, mobile BSs or data mules) that often ignore the cost of locomotion, we account for the energy consumption of both locomotion and wireless communications in a unified framework for maximizing the total amount of data collected. Moreover, mobile relays only introduce a relocation delay in the initial configuration phase of the network. Second, different from the existing mobility approaches that typically assume only a few powerful mobile nodes, we exploit the availability of many low-cost mobile nodes to improve the data transport capability of a network. At the same time, our approach is practical as mobile sensors only need extremely simple motions, *i.e.*, one-shot relocation to designated positions after deployment.

We make the following contributions in this paper. (1) We present the formulation of a new problem called Max-Data Mobile Relay Configuration (MMRC) for hybrid WSNs composed of both static and mobile nodes. The objective of MMRC is to configure the positions of mobile relays to maximize the data gathering capacity of WSNs, which is defined as the total amount of data that can be transmitted to the BS during the lifetime of a network topology. Our formulation is based on realistic energy consumption models of both wireless communications and node movement, which is in contrast to existing mobility approaches that only account for the communication energy consumption. (2) We develop an optimal solution for a base case of the MMRC problem with one mobile and two static nodes. Despite the trivial network configuration, we show that the base case of MMRC has a surprisingly high complexity. In particular, a direct analytical solution requires finding the roots of a degree six bivariate polynomial. To the best of our knowledge, this result is the first to reveal the complexity of data gathering capacity maximization when both communication and mobility energy consumption are jointly considered. (3) Based on the solution to the base case of MMRC, we develop optimal algorithms for several important network topologies including lines, stars, and trees with and without data reduction techniques. We also present a practical distributed implementation for these algorithms. (4) We conduct extensive simulations based on energy models obtained from existing mobile and static sensor platforms. We show that our optimal algorithms can increase the amount of data gathered during the system lifetime by a factor of 2 or more. Moreover, the distributed protocol is shown to have fast convergence and low messaging overhead.

The rest of the paper is organized as follows. Section II reviews related work. In Section III, we formally define our problem. Section IV presents several optimal algorithms for our problem. In Section V, we describe practical distributed implementations. Section VI describes our simulation results and Section VII concludes this paper.

II. RELATED WORK

Several different approaches have been proposed to exploit the controlled mobility of sensors to improve the energy efficiency of WSNs. They fall into three basic categories: mobile BSs, data mules, and mobile relays. In the first approach, a mobile BS moves around the network and collects data from the static nodes [7], [1], [14], [9], [10], [15], [16]. Such an approach can prolong system lifetime by rotating the bottleneck nodes close to the BS. In [1], [14], [9], the motion path of a BS is found to collect data from static nodes under constraints of buffer overflow or data delivery deadlines. In [10], [15], [16], several rendezvous-based data collection algorithms are proposed, where the mobile base station only visits a selected set of nodes referred to as rendezvous points within a deadline and the rendezvous points buffer the data from sources. However, the above approaches incur high latencies due to the low to moderate speed, *e.g.*, 0.1-1 m/s [2], [1], of mobile BSs.

Several studies [11], [12], [17] propose using mobile sensors as data mules that pick up data from the sensors and transport it to the BS. However, similar to the case of mobile BSs, the delay of data delivery is often large due to the intermittent contacts between data mules and other nodes. Moreover, the previous studies on mobile BSs and data mules are focused on minimizing total energy consumption of a network, which does not necessarily lead to maximum system lifetime.

In the third approach, the network consists of mobile relays and static nodes that collaborate to forward data to the BS. Instead of mechanically carrying data, mobile relays relocate to decrease the wireless transmission costs. In [6], an iterative mobility algorithm is proposed to move relay nodes to the midpoints of their neighbors, which is shown to converge on the optimal solution for a single routing path. However, these studies do not take into account the cost of moving the relay nodes. In [18], mobile nodes decide to move only when mobility costs are covered by the savings in transmission costs. However, only the cost of moving to the midpoint of neighbors is considered.

In our previous work [19], we developed two algorithms that iteratively refine the configuration of mobile relays and converge to the optimal solution. Although the approach in [19] minimizes the total energy consumption of both movement and wireless communications, it does not necessarily lead to the maximum amount of data reaching the sink. In particular, our previous work ignores the actual amount of energy stored at any node and thus may save energy at the wrong nodes. To the best of our knowledge, this paper is the first to maximize the total data collection capacity by accounting for the energy available to each node and the energy consumption of both movement and wireless communications in mobile relay configuration.

III. PROBLEM DEFINITION

The fundamental problem we address in this paper is maximizing the amount of data that can be gathered by the base station of hybrid WSNs composed of both static and mobile nodes. A key challenge of utilizing low cost mobile relay nodes is that they often are more unreliable than static nodes. For example, mobile relay nodes often experience mechanical failures when moving in rough terrain. To provide a balance between maximizing data gathering capacity and mitigating the impact of mobile sensor nodes on network reliability, we make the following assumptions. We first assume that a routing topology that consists only of static nodes has already been established. Second, we assume that each transmission link between two static nodes can be helped by at most one mobile relay and conversely, each mobile relay can help at most one link. These assumptions allow mobile relays to improve the system's data gathering capacity but minimize the dependence of the communication on mobile relay nodes.

We define the data gathering capacity of a network to be the total amount of data that can be transmitted from the sources to the sink during the lifetime of the current routing topology. Consistent with several other papers [20], [21], [22], the lifetime of the current topology ends when the first node dies. We use this definition because the death of any node entails a reconfiguration of the underlying routing topology of static nodes and a consequent reconfiguration of the mobile relay nodes. However, we do assume that data that is in transit from a source to the sink and still has a viable path to the sink will be sent along the remaining viable links to the sink. For example, suppose a node uexpends all its energy to transmit data to an intermediate node v. Since u has expended all its energy, the system lifetime has ended. However, we assume that node v and other nodes on the path to the sink can still send this data to the sink assuming they have enough energy. We use this data gathering capacity metric as an alternative to two other important but conflicting metrics, lifetime and throughput, since it directly measures the network's ability to perform its target functionality of transmitting data from the sources to the sinks. In one variant with a star routing topology where each source node is directly connected to the sink node, we simply measure the raw data gathering capacity without regard to system lifetime as no reconfiguration will be performed. That is, we simply measure how much data can be transmitted from the sources to the sink until the single sink node dies or all the source nodes die.

Because of our assumption that each transmission link can be helped by at most one relay node, our objective is to find a matching between mobile relay nodes and transmission links in the underlying transmission topology. Thus, all of our centralized algorithms employ a matching or assignment algorithm as a key step. It is possible that some mobile relay nodes may not be used in the matching, but this is unlikely in practice because the number of relay nodes will typically be much smaller than the number of static nodes. We also must find the optimal position of each mobile relay node. Different from previous work that only accounts for the energy required for transmission, we take into account the energy required to transmit and receive messages plus the energy required to move the mobile relay nodes into position. We first introduce the energy consumption models and then formally define our problem.

A. Energy Consumption Models

We adopt the same energy consumption models as used in [19]. Specifically, we develop models for the energy consumed by nodes during both transmissions and movement. This advances significantly upon previous work that considers only energy consumed during transmissions. We do not consider energy consumed in idle listening states as it can be significantly reduced through existing sleep scheduling schemes [23]. Alternatively, we can view our results as the limits of what can be achieved assuming a perfect sleep scheduling algorithm.

Paraphrasing the models from [19], we base our mobility energy consumption models on two-wheeled sensor nodes where each wheel is controlled by independent engines such as Khepera [3], Robomote [2] and FIRA [4]. We adopt the distance proportional energy consumption model [24] where the energy $E_M(d)$ consumed by moving a distance d is $E_M(d) = kd$ where the parameter k depends on the speed of the node. In general, there is an optimal speed at which k is lowest. In [24], the authors discuss in detail the variation of the energy consumption with respect to the speed of the mote. When the node is running at optimal speed, k = 2 J/m [24]. In our simulations, we vary k between 1 and 4 to model different speeds and different terrains.

We develop our transmission energy model by analyzing empirical results obtained from two radios CC2420 [25] and CC1000 [26] that are widely used on sensor network platforms. For CC2420, the authors of [27] studied the power needed for transmitting packets reliably (e.g., above 95% packet reception ratio) over different distances. Let $E_T(d,m)$ be the energy consumed to transmit m bits reliably over distance d and $E_R(m)$ the energy consumed by receiving m bits. We have

$$E_T(d,m) = m(a' + bd^2)$$
$$E_R(m) = ma''$$

where a', a'' and b are constants depending on the environment. We assume that a node only receives data when it has enough energy to transmit that data to the next node. In this case, a node receiving m bits and transmitting it over a distance d consumes $m(a'+a''+bd^2) = m(a+bd^2)J$. We now discuss the instantiation of the above model for the CC2420 and CC1000 radio platforms. We used measurements in [27] on CC2420 obtained in an outdoor environment to determine the power level needed for different distances to get a packet reception ratio higher than 95%. The same measurements were collected for the CC1000 platform. Using these values along with current and voltage requirements from [25] [26], we computed the energy model parameters as summarized

in table I. These values are consistent with the theoretical analysis in [28]. We observe that a'' > a', but there is no dependence on distance d for reception whereas there is a dependence on d for transmission.

 Table I

 PARAMETER VALUES FOR ENERGY CONSUMPTION

	a' J/bit	a" J/bit	b Jm^{-2}/bit
CC2420	0.6×10^{-7}	1.4×10^{-7}	4.0×10^{-10}
CC1000	0.3×10^{-7}	2.6×10^{-7}	2.0×10^{-10}

The model parameters that we obtained fit the quadratic path loss model for distances up to 35m. For larger transmission distances, the energy consumption increased and diverged from the quadratic model and could be modeled by higher path loss coefficients (4 and 6) as discussed in [29]. To cope with higher path loss coefficients, the only modification we need is a new optimal point location algorithm to replace our algorithm in Section IV-A.

B. Max-Data Mobile Relay Configuration

We consider four variants of our problem that cover three different topologies and several different metrics.

(1) MMRC in a line: The static nodes form a line between the source and the sink. We want to maximize the total amount of data that can be relayed from the source to the sink before any node runs out of energy. This is equivalent to maximizing the bottleneck capacity of any transmission link on the line.

(2) MMRC in a star: The star transmission topology consists of n static sources and one static sink. Each of the n source nodes transmits data directly to the sink. We again want to maximize the total amount of data that can be sent from all n sources to the sink. The star topology is widely used in networks running clustering protocols and is the primary topology supported by the 802.15.4 standard [30]. For the star topology, we measure the total amount of data gathered until either the sink node dies or all the source nodes die since we will not reconfigure the topology when any source node dies.

(3) Fair MMRC in a tree with data reduction: The static nodes form a tree with n source nodes sending data through the tree to a single sink. When an event occurs, each source node sends one data unit to its parent. To ensure fairness, we require that each source's data must reach the sink. We assume perfect data aggregation where any internal node is able reduce the amount of data it sends to its parent to one data unit per event. Many natural sensor network applications use such an N-to-1 aggregation model. For instance, a user may periodically query the maximum or average value of temperature readings from all sensors deployed in a large biological habitat. Moreover, many in-network information processing techniques such as data fusion [31] can aggregate noisy measurements or local decisions from multiple sensors to improve the accuracy of sensing. In both applications described above, N sensor readings can be aggregated into one, which significantly reduces the energy consumption of the network. The net result is that we must maximize the bottleneck capacity of any link in the tree.



Figure 1. Two static tree topologies; each link is labeled with its capacity.

Consider the two examples in Figure 1. In the left tree, the data for 3 events can be sent to the sink before source c_2 runs out of energy; in the right tree, the data for 5 events can be sent to the sink before node c_3 runs out of energy. In both cases, the number of events reachable to the sink is limited by the edge with the smallest capacity.

(4) Fair MMRC in a tree without data reduction: this problem is identical to the previous problem except no data reduction is possible. That is, if an internal node receives one data unit from 3 children, it must send 3 data units to its parent. In both trees in Figure 1, after 2 events are processed, c_3 has only enough energy to send one data unit to the sink. With our fairness constraint, no more events can be processed as a third event would require sending 2 data units to the sink.

IV. OPTIMAL ALGORITHMS

The MMRC problem of maximizing system data gathering capacity is much more complex than the seemingly similar problem of minimizing the total energy required to transmit a fixed amount of data in a network [19] because we must account for the energies consumed by each individual node relative to its initial energy rather than worrying only about the total energy consumed by all nodes together. To the best of our knowledge, we are the first to optimize data gathering capacity taking into account the energy consumed by both movement and communication. We first present an optimal algorithm for the simplest possible "base case" of the MMRC problem: one source, one sink, and one mobile relay. Next, we discuss the assignment problem, a weighted matching problem that several of our application problems can be reduced to. We then present optimal algorithms for our four variants of MMRC.

A. Base Case of MMRC

Our base case is computing the optimal position r^* to move relay node r to that will maximize the total number of bits that can be sent from static source node s_1 to static sink node s_2 using r as a relay at position r^* given initial energies e_1 , e_2 , and e_r . An example scenario is given in Figure 2.



Figure 2. Example scenario illustrating the base case. Using relay r at position r^* increases the amount of data that can be sent.

For cases where it may be better to not use r because r has low energy, we assume that s_1 sends data directly to s_2 . Note that we focus on maximizing the amount of data gathered by s_2 during the lifetime of the current network topology. Therefore, we do not allow s_1 to send data to r until r dies and then have s_1 send data directly to s_2 because this requires a topology reconfiguration. In the rest of this section, we study the problem of finding r^* when it is helpful to use r.

We now show that a direct analytical solution to the base case requires finding the roots of a degree 6 bivariate multimodal polynomial. For any candidate point r', we define two key values: $m(s_1, r')$ which is the amount that s_1 can send to r located at r', and $m(r', s_2)$ which is the amount that r can send to s_2 when r is located at r' taking into account the energy r loses by moving to r'. It follows that the maximum amount of data s_1 can send to s_2 using r as a relay at r', denoted m(r'), is the minimum of $m(s_1, r')$ and $m(r', s_2)$. Optimal positions r^* have one of the following properties: (O1) $m(s_1, r^*) = m(r^*, s_2)$ or (O2) $\forall r', m(s_1, r^*) \geq m(s_1, r') \text{ and } m(s_1, r') < m(r', s_2) \text{ or }$ (O3) $\forall r', m(r^*, s_2) \ge m(r', s_2)$ and $m(s_1, r') > m(r', s_2)$. Let's assume condition (O1) holds where r^* is located at (x, y) and (x_1, y_1) , (x_r, y_r) , and (x_2, y_2) are the initial coordinates of nodes s_1 , r and s_2 , respectively. Let d(u, v)be the distance between two nodes u and v. Finding point r^* is equivalent to solving the following equation.

$$m(s_1, r^*) = m(r^*, s_2) \tag{1}$$

$$\Leftrightarrow \frac{e_1}{1 + k + d(r, r^*)^2} = \frac{e_r - k_r d(r, r^*)}{1 + k + d(r, r^*)^2} \tag{2}$$

$$a_1 + b_1 a(s_1, r) \qquad a_r + b_r a(r_1, s_2)$$

$$cond + b_r a(r_1, s_2)$$

$$\Rightarrow B(x,y)^{2} A(x,y)^{2} - C(x,y)^{2} = 0$$
(4)

where

$$A(x, y) = k_r \left(a_1 + b_1 (x - x_1)^2 + b_1 (y - y_1)^2 \right)$$

$$B(x, y) = \sqrt{(x - x_r)^2 + (y - y_r)^2}$$

$$C(x, y) = e_1 \left(a_r + b_r (x - x_2)^2 + b_r (y - y_2)^2 \right)$$

$$- e_r \left(a_1 + b_1 (x - x_1)^2 + b_1 (y - y_1)^2 \right).$$

Because A(x, y) and $B(x, y)^2$ have degree 2, equation (4) has degree 6; this equation is also a nonconvex (multi-modal) polynomial. Thus, solving the base case using analytical methods is numerically complex.

We present an efficient algorithm that converges on an optimal solution by solving the special case of the problem

in which we are given the direction that r will move. Let α be any point on the line segment between s_1 and s_2 including s_1 and s_2 , let $pos_r(\alpha)$ be the optimal position for r when it moves towards α , and let $\overline{M}(\alpha)$ be the resulting amount of data that can be transmitted from s_1 to s_2 when r moves to $pos_r(\alpha)$. We develop an algorithm that computes $pos_r(\alpha)$ and thus $\overline{M}(\alpha)$ for any α between s_1 and s_2 , and we prove that $\overline{M}(\alpha)$ is unimodal on this range (this proof, which requires careful analysis of many cases, is omitted due to lack of space). Thus, we can converge on the global optimum by performing a golden ratio search for α in the range between s_1 and s_2 .

We now discuss the algorithm for computing $pos_r(\alpha)$ and $\overline{M}(\alpha)$. We first consider optimality conditions (O2) and (O3) by finding the unique points r_1 and r_2 on the line to α that maximize $m(s_1, r')$ and $m(r', s_2)$, respectively. Point r_1 is returned if $m(r_1, s_2) > m(s_1, r_1)$ as condition (O2) applies. Otherwise, if $m(r_2, s_2) < m(s_1, r_2)$, then we return r_2 as condition (O3) applies. Otherwise, it must be the case that condition (O1) applies. When restricting the movement to a given line (with slope s_α), the distance traveled becomes $\sqrt{1 + s_\alpha^2} |x - x_r|$ and equation (2) becomes

$$\frac{e_1}{a_1 + b_1(x - x_1)^2 + b_1(y - y_1)^2} = \frac{e_r - k_r \sqrt{1 + s_\alpha^2} |x - x_r|}{a_r + b_r(x - x_2)^2 + b_r(y - y_2)^2}$$

Identifying such points is equivalent to solving a degree 3 polynomial. Thus, there are at most 3 possible points, all of which can be computed in O(1) time. We return the point that maximizes the data sent. In summary, we find the optimal position along a given line by examining at most five well-defined points.

This base case algorithm only converges on the optimal r^* for energy models with path loss coefficient 2 or smaller. For other energy models with higher path loss coefficients, we need to use different techniques to find the optimal relocation point. We do not have efficient algorithms that do this for arbitrary energy models.

B. The Assignment Problem

With our optimal base case algorithm, we now know where each relay node r should move if we know which transmission link r should help; that is, we need to match each mobile relay with a transmission link such that the application objective is optimized. This is essentially the assignment problem from combinatorial optimization where we typically have n tasks and n people, each task needs to be assigned to a single person, and the cost or efficiency of the *i*-th person performing the *j*-th task is given by c_{ij} . The goal is to assign people to tasks in order to maximize efficiency or minimize costs. The assignment problem is often formalized as a 0-1 integer program where the 0-1 variable x_{ij} takes on value 1 if person *i* is assigned to task *j* and 0 otherwise. Fortunately, we can relax this integer program to a linear program where x_{ij} can take on fractional values because the constraint matrix of the linear program is totally unimodular which means that an integral optimal solution is guaranteed. The linear programming formulation is the following:

$$\begin{cases} \max \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \\ \sum_{j=1}^{n} x_{ij} = 1 \quad (i = 1, 2, \dots, n) \\ \sum_{i=1}^{n} x_{ij} = 1 \quad (j = 1, 2, \dots, n) \\ x_{ij} \in \{0, 1\}, \quad (i = 1, \dots, n, \quad j = 1, \dots, n) \end{cases}$$

The first constraint guarantees that each person is assigned to one task, and the second constraint guarantees that each task is assigned to one person. The objective function attempts to maximize the sum of task efficiencies and can be solved in $O(n^3)$ time [32]. We use this formulation for MMRC in a star. We can create the maximum bottleneck assignment problem by changing the objective function from $\max \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$ to $\max \min_{1 \le j \le n} \sum_{i=1}^{n} c_{ij} x_{ij}$. This formulation can be solved in $O(n^{2.5})$ time [32]; we use it for MMRC in a line and for MMRC in a tree with data reduction.

C. MMRC in a Line, Star, and Tree with Data Reduction

Consider an arbitrary MMRC in a line instance. Let r be the number of mobile relay nodes, k be the number of static nodes on the transmission path, and n = r + k be the total number of nodes. Let $P = (s_1, \ldots, s_k)$ be the path of static nodes where s_1 is the source node and s_k is the sink node.

The basic idea in transforming this MMRC instance to a maximum bottleneck assignment problem instance is the following. The relay nodes take the place of people. The transmission links $l_j = (s_j, s_{j+1})$ take the role of jobs. To allow for the case where no relay nodes are used, we add k - 1 trivial "people" d_j to represent the case when node s_j transmits directly to s_{j+1} and r trivial jobs out_i to represent the case where relay node r_i is not used. For actual relays and transmission links, we use the base case algorithm in Section IV-A to compute c_{ij} , the size of the largest data chunk that can be transmitted from s_j to r_i to s_{j+1} . If relay node r_i cannot increase the size of data that s_j can transmit, then we set c_{ij} to 0. Let c_j be the size of data if the transmission is done directly from s_j to s_{j+1} . The efficiency matrix values would then be the following:

	r_1	 r_r	d_1		d_{k-1}
$l_1 \\ l_2$	$c_{11} \\ c_{12}$	 c_{r1} c_{r2}	$\begin{array}{c} c_1 \\ 0 \end{array}$		$\begin{array}{c} 0\\ 0\end{array}$
l_{k-1}	c_{1k-1}	 c_{rk-1}	0	 	c_{k-1}
out_1	∞	 ∞	∞		∞
out_r	∞	 ∞	∞	· · · · · · ·	∞

More formally,

$$A(l_j, r_i) = c_{ij} \qquad A(out_j, r_i) = \infty$$

$$A(l_j, d_j) = c_j \qquad A(out_j, d_i) = \infty$$

$$A(l_i, d_i) = 0 \text{ if } i \neq j$$

We use the exact same formulation for MMRC in a Tree with Data Reduction. For MMRC in a Star, we change the objective function to $\max \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$.

D. MMRC in a Tree without Data Reduction

The assignment based solution we described in Section IV-B can not be applied to this MMRC variant since the LP formulation does not capture the correct bottleneck edge which is no longer the edge with the smallest capacity; adding more constraints to solve this issue breaks the unimodularity property of the constraint matrix and subsequently the integer program can not be solved using linear programming techniques. Instead, we solve a decision version of this problem using unweighted bipartite matching algorithms. For the decision version, we add an input Mand the yes/no question is whether or not there exists an assignment of mobile relay nodes to transmission links such that every source node is able to send M events to the sink. We then converge on an optimal solution by using standard doubling and binary search techniques. We make an initial guess M and keep doubling this guess until it is infeasible. We then do a binary search between the last feasible value and the first infeasible value. If the initial guess M was not feasible, we binary search between 0 and M.

To solve the decision version, we first compute for each node the amount of data that it needs to send so that every M data units initiating at a source and passing through it can reach the sink by doing a bottom-up traveral of the tree. For each source, we set its value to M. For each internal node, we set its value to be the sum of the values of its children. Let M_i be the value that node s_i needs to transmit. Second, we eliminate any node that can send M_i bits without help. Third, we create a bipartite graph as follows. For each remaining static node s_i , we create an edge between s_i and relay node r_j if r_j can help s_i reach its M_i value. Finally, we search for a bipartite matching in the bipartite graph. If there is a bipartite matching such that all remaining static nodes are matched with some relay node, we return yes. Otherwise, we return no. Bipartite matchings can be found in $O(n^{2.5})$ time [33].

V. DISTRIBUTED IMPLEMENTATION

Before describing our distributed implementation, we propose three centralized greedy algorithms that provide insight to the distributed approach; these greedy strategies differ only in their link priority strategies which determine which link to process first. The three algorithms proceed as follows. First, for each link l between two static nodes s_i and s_j , we compute the maximum amount of data c(l) that s_i can

```
procedure STATICRUN
   ▷ Initialize empty priority queue
    offers \leftarrow \emptyset;
    committed \leftarrow false:
    repeat
       > Listen to mobile nodes in range and collect offers presented
        repeat
           RECEIVE(mobile, type, data);
           if type = MOBILE_IN_RANGE then
               SEND(mobile, META_DATA, info);
           else if type = OFFER then
               offers.add(data);
           end if
        until timeout
       ▷ Process offers and pick best
        if offers \neq \emptyset then
           bestOffer \leftarrow offers.dequeue();
           SEND(bestOffer.sender, ACCEPT_OFFER);
           committed \leftarrow true:
        end if
        while offers \neq \emptyset do
           offer \leftarrow candidates.dequeue();
           SEND(offer.sender, REJECT_OFFER);
        end while
   until committed
end procedure
```

Figure 3. Local algorithm executed by each static node

transmit to s_j . Second, for each mobile node r and for each link l, we compute the total amount of data c(r, l)that can be transmitted from s_i to r to s_j using the optimal algorithm for the base case (section IV-A). If s_i can send more data directly to s_j than going through r, we set c(r, l)to zero. We now choose the link with the highest priority using three different link priority schemes: (1) the highest value c(r, l), (2) the largest improvement c(r, l) - c(l) and (3) the smallest (bottleneck) direct value c(l). In case of a tie for the third metric, we pick the link with the highest c(r, l). In all three cases, we match the selected link to the corresponding mobile relay r and then repeat the greedy selection process until all the mobile nodes are matched to links or cannot help the remaining links.

We now propose a distributed approach that mimics our greedy algorithms. For MMRC in a line or tree, we use the bottleneck link priority scheme. For MMRC in a star, we use the largest improvement link priority scheme. The algorithm begins with each mobile relay node broadcasting its existence (a MOBILE_IN_RANGE message) to nodes within its broadcast range. Each static node that receives such a message responds by sending its location, its remaining energy, and the location of its destination static node (a META_DATA message) to each mobile relay node that sends it a broadcast message. Each mobile relay node r then calculates for each responding static node s how much that s can transmit now and how much r can improve s's transmission capacity. It then sends a message (of type OFFER) to the static relay node with the highest priority saying how much it can help it and waits for a response from that static node. Each static node that receives an offer accepts the best offer it receives, sending an acceptance message to the one relay node it accepts and rejection messages to the other relay nodes. Rejected relay nodes then send messages to the next node on their priority list. If a static node that is already committed receives a late offer from a relay node, it automatically rejects the offer even if that relay node can provide more help. We show the pseudocode for the local algorithm performed by the static nodes in Figure 3; we omit the pseudocode for the mobile nodes due to lack of space.

VI. SIMULATION RESULTS

We evaluated our algorithms using simulations. For a given simulated network, we varied the number of mobile relay nodes by 5 between 5 and 30. For the star and tree topologies, we varied the number of source nodes between 5 and 30 by 5. For each number of source nodes and number of mobile relay nodes, we generated 20 random networks (120 for the line, 720 for the tree and star) consisting of 100 nodes in a 150m by 150m area. To ensure high quality transmissions based on empirical data using Tmotes [27], we set the maximum communication distance to 35m. We set the initial battery energy of nodes uniformly at random between half charged and fully charged. We modeled different platforms and different settings by having 16 possible choices for node energy consumption parameters (4 values for a and b and 4 values for k). To build the communication trees between the sources and sink, we used greedy geographic forwarding in which each node forwards the packet to the neighbor closest to the sink. We note that our solutions only require the establishment of a routing topology before mobile relay configuration, and hence can work with other routing algorithms as well. Finally, we use the quadratic path loss energy model.

For all networks that we generated, we evaluated our optimal algorithm as well as seven different greedy strategies. Six of our greedy strategies differed in two main ways: point selection and link priority. We consider two different point selection strategies. The first is to move the relay node to the midpoint of the two static nodes defining the link that will be helped; we label such greedy strategies with the suffix -MP. The other option is to compute the optimal position using our optimal base case algorithm; these strategies are labeled with the suffix -OPT. We consider the three link priority schemes we described in the distributed implementation section: the highest value, the largest improvement, and the bottleneck value with ties broken by largest improvement. The net result is six greedy strategies. We also consider a seventh greedy algorithm that uses matching to minimize the total travel distance of all mobile relay nodes. Finally, we evaluated our distributed implementations, one for each of the three link priority strategies. To model the asynchronicity of our protocol, we vary the fraction of mobile relay nodes that participate in the protocol at various percentages between



Figure 4. Improvement in data gathering capacity of our algorithms compared to the baseline topology as a function of the number of mobile relay nodes.

25% and 100%. We did not implement existing algorithms because their fundamental assumptions do not match ours making meaningful comparisons difficult if not impossible.

For a network N and an algorithm A, let A(N) be the data gathering capacity of the resulting network N' computed by algorithm A. Let B denote the baseline algorithm that uses no mobile relay nodes; thus, the resulting network N' is the original network of static nodes. We measure the performance of algorithm A on network N by its improvement ratio which is A(N)/B(N). When reporting results for an algorithm A, we report the average of A(N)/B(N) for all N that have the same number of mobile relay nodes. For line topologies, this will be an average of 120 networks; for tree and star topologies, this will be an average of source nodes). We also report results for where we average over all networks for the given application. We now present a summary of our simulation results for each application.

A. MMRC in a Line

We highlight some of our key results for MMRC in a Line in Figure 4(a). The average improvement ratio for our optimal algorithm over all 120 networks is 232%, and the improvement ratio increases from 180% to 285% as we increase the number of mobile relay nodes. The best greedy strategy, Bottleneck-OPT, also has an average improvement ratio of 232%. TAs we can see from Figure 4(a), its improvement ratio is very close to that of the optimal algorithm for all numbers of mobile relay nodes. This implies that we can use a simple greedy strategy to perform the matching rather than using more sophisticated weighted matching algorithms. The remaining greedy strategies perform significantly worse with average improvement ratios ranging from 165% to 210%. We also observe the importance of optimal point selection; the Bottleneck-MP greedy strategy achieves an average improvement ratio of 175% which is much worse than that of Bottleneck-OPT. In fact, for each link priority scheme, the midpoint point selection strategy has a significantly lower improvement ratio than its optimal point selection counterpart.

Our distributed implementation of the Bottleneck-OPT greedy strategy has an average improvement ratio of 216% which is a bit lower than both the optimal algorithm and Bottleneck-OPT. There are two possible explanations for the slight decrease in performance. The first is that a relay node may not be matched to the correct link because it cannot reach the source node of the link with its transmission (e.g. it is farther than 35m away from the source node of the link); the second is that a relay node may not be matched to the correct link because its offer of help arrives after the source node of the link has already accepted the help of a different relay node. Our results indicate that the 35m transmission distance limitation is the more important factor, particularly as the number of relay nodes increases. Specifically, we see little difference between the distributed implementation with 33% of the relay nodes participating and 100% of the relay nodes participating for any number of relay nodes; this also holds for all values between 33% and 100%. We also see that the distributed implementation's average improvement ratio approaches that of the optimal algorithm as the number of relay nodes increases. As the number of relay nodes increases, each transmission link is likely to receive an offer from a good if not optimal relay node.

B. MMRC in a Star

We highlight some of our key results for MMRC in a Star in Figure 4(b). The average improvement ratio for our optimal algorithm over all 720 networks is 155%, and the improvement ratio increases from 130% to 170% as we increase the number of mobile relay nodes. The best greedy strategy, Improvement-OPT, has an average improvement ratio of 153% which is nearly identical to the optimal algorithm for all numbers of relay nodes. The remaining greedy strategies perform significantly worse with improvement ratios approximately 20% lower. We again observe the importance of optimal point selection; the greedy strategies that use optimal point selection including Improvement-OPT have improvement ratios roughly 17% better than their midpoint point selection counterparts.

Our distributed implementation of the Improvement-OPT greedy strategy has an improvement ratio of roughly 130%. We again observe that the main cause of the performance loss is the 35m transmission distance limit as our distributed algorithms perform essentially identically once at least 33% of the relay nodes participate in the protocol. We again see the distributed implementation's performance increase significantly as we increase the number of mobile relay nodes.

C. Fair MMRC in Trees with Data Reduction

We highlight some of our key results for MMRC in trees with data reduction in Figure 4(c). The average improvement ratio for our optimal algorithm over all 720 networks is 270%, and the improvement ratio increases from 215% to 300% as we increase the number of mobile relay nodes. The best greedy strategy, Bottleneck-OPT, has an average improvement ratio of 268% which is nearly identical to the optimal algorithm in all cases. The remaining greedy strategies perform significantly worse with average improvement ratios of at most 240%. We again observe the importance of optimal point selection; the greedy strategies that use optimal point selection including Bottleneck-OPT have improvement ratios roughly 65% higher than their midpoint point selection counterparts. For Bottleneck-OPT and Bottleneck-MP, the divergence in performance increases as the number of mobile relay nodes increases.

Our distributed implementation of the Bottleneck-OPT greedy strategy has an improvement ratio of roughly 225%. We again observe that the main cause of the performance loss is the 35m transmission distance limit as our distributed algorithms perform essentially identically once at least 33% of the relay nodes participate in the protocol. We again see the distributed implementation's performance approaches that of the optimal algorihtm as we increase the number of mobile relay nodes. We also study how many communication rounds are required before all the mobile nodes commit to some edge or remove themselves from consideration. In all cases, the number of rounds grows very slowly with the number of mobile nodes. On average, only 4 to 6 rounds are needed. Even when the number of relays was high (\approx 30), the number of rounds never exceeded 8. These results also hold for MMRC in trees without data reduction.

D. Fair MMRC in Trees without data reduction

We highlight some of our key results for MMRC in trees without data reduction in Figure 4(c). The average improvement ratio for our optimal algorithm over all 720 networks is 230%, and the improvement ratio increases from 210% to 250% as we increase the number of mobile relay nodes. The best greedy strategy, Bottleneck-OPT, has an average improvement ratio of 220% and it approaches that of the optimal algorithm as we increase the number of mobile relay nodes. Improvement-OPT performs almost as well as



Figure 5. Improvement ratios in trees without data reduction

Bottleneck-OPT, particularly for many mobile relay nodes. The remaining greedy strategies perform significantly worse with improvement ratios of at most 175%. We again observe the importance of optimal point selection; the greedy strategies that use optimal point selection including Bottleneck-OPT have improvement ratios roughly 40% higher than their midpoint point selection counterparts. For Bottleneck-OPT and Bottleneck-MP, the divergence in performance increases as the number of mobile relay nodes increases.

Our distributed implementation based on Bottleneck-OPT again performs well with an average improvement ratio of 200%. We also show that the average number of messages generated per mobile relay node grows slowly and linearly with number of mobile nodes. For example, when a static node receives offers from 33% of the mobile relay nodes in its neighborhood, the average number of messages sent by each relay node is only 2.1. Even with large number of mobiles nodes (30) and high communication ratio (100%), the number of messages generated by each mobile node is only 4.

Finally, in our simulations for all four applications, the approach that matches mobile relays based on minimizing the total distance traveled produces noticeably inferior results to the optimal approach and, in most cases, the other greedy approaches. This shows that optimizing only to minimize movement without regard to transmission is not helpful. In particular, it is helpful to spend a little more energy on movement if this will significantly decrease energy consumed by transmissions.

VII. CONCLUSION

In this paper, we study a new problem, MMRC, maximizing the data gathering capacity of hybrid wireless sensor networks consisting of both mobile and static nodes. We presented optimal solutions to four variants of MMRC. Unlike most previous work that exploits controlled mobility, we consider both the energy consumed during the transmission process as well as the energy consumed by mechanical locomotion. In most variants, our optimal algorithm improved system lifetime by a factor of 2 or more. For the star topology, it increased system lifetime by a factor of 1.5. Our distributed protocols were almost as effective at improving data gathering capacity, particularly as the number of mobile relay nodes increases. Our simulations also showed that these protocols quickly converge on a final solution with little messaging overhead. For future work, we plan on finding optimal algorithms for more general topologies.

ACKNOWLEDGEMENT

This work is supported by National Science Foundation under a CAREER award CNS-0954039.

REFERENCES

- A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling with dynamic deadlines," *IEEE Transactions on Mobile Computing*, vol. 6, no. 4, 2007.
- [2] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. S. Sukhatme, "Robomote: enabling mobility in sensor networks," in *IPSN*, 2005.
- [3] http://www.k-team.com/robots/khepera/index.html.
- [4] J.-H. Kim, D.-H. Kim, Y.-J. Kim, and K.-T. Seow, Soccer Robotics. Springer, 2004.
- [5] W. Wang, V. Srinivasan, and K.-C. Chua, "Using mobile relays to prolong the lifetime of wireless sensor networks," in *MobiCom*, 2005.
- [6] D. K. Goldenberg, J. Lin, and A. S. Morse, "Towards mobility as a network control primitive," in *MobiHoc*, 2004, pp. 163– 174.
- [7] J. Luo and J.-P. Hubaux, "Joint mobility and routing for lifetime elongation in wireless sensor networks," in *INFOCOM*, 2005.
- [8] Z. M. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli, "Exploiting sink mobility for maximizing sensor networks lifetime," in *HICSS*, 2005.
- [9] A. Kansal, D. D. Jea, D. Estrin, and M. B. Srivastava, "Controllably mobile infrastructure for low energy embedded networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 8, 2006.
- [10] G. Xing, T. Wang, W. Jia, and M. Li, "Rendezvous design algorithms for wireless sensor networks with a mobile base station," in *MobiHoc*, 2008, pp. 231–240.
- [11] D. Jea, A. A. Somasundara, and M. B. Srivastava, "Multiple controlled mobile elements (data mules) for data collection in sensor networks," in *DCOSS*, 2005.
- [12] S. Jain, R. Shah, W. Brunette, G. Borriello, and S. Roy, "Exploiting mobility for energy efficient data collection in wireless sensor networks," *MONET*, vol. 11, no. 3, pp. 327– 339, 2006.
- [13] S. Chellappan, X. Bai, B. Ma, D. Xuan, and C. Xu, "Mobility limited flip-based sensor networks deployment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 2, pp. 199–211, 2007.
- [14] Y. Gu, D. Bozdag, and E. Ekici, "Mobile element based differentiated message delivery in wireless sensor networks," in *WoWMoM*, 2006.

- [15] G. Xing, T. Wang, Z. Xie, and W. Jia, "Rendezvous planning in wireless sensor networks with mobile elements," *IEEE Transactions on Mobile Computing*, vol. 7, no. 12, pp. 1430– 1443, 2008.
- [16] —, "Rendezvous planning in mobility-assisted wireless sensor networks," in RTSS '07: Proceedings of the 28th IEEE International Real-Time Systems Symposium, 2007, pp. 311– 320.
- [17] C.-C. Ooi and C. Schindelhauer, "Minimal energy path planning for wireless robots," in *ROBOCOMM*, 2007, p. 2.
- [18] C. Tang and P. K. McKinley, "Energy optimization under informed mobility," *IEEE Trans. Parallel Distrib. Syst.*, vol. 17, no. 9, pp. 947–962, 2006.
- [19] F. El-Moukaddem, E. Torng, G. Xing, and S. Kulkarni, "Mobile relay configuration in data-intensive wireless sensor networks," in *IEEE MASS*, 2009.
- [20] S. Singh, M. Woo, and C. S. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *MOBICOM*, 1998, pp. 181–190.
- [21] J.-H. Chang and L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks," in *INFOCOM*, 2000, pp. 22–31.
- [22] P. Santi, D. M. Blough, and F. S. Vainstein, "A probabilistic analysis for the range assignment problem in ad hoc networks," in *MobiHoc*, 2001, pp. 212–220.
- [23] L. Wang and Y. Xiao, "A survey of energy-efficient scheduling mechanisms in sensor networks," *Mob. Netw. Appl.*, vol. 11, no. 5, 2006.
- [24] G. Wang, M. J. Irwin, P. Berman, H. Fu, and T. F. L. Porta, "Optimizing sensor movement planning for energy efficiency," in *ISLPED*, 2005, pp. 215–220.
- [25] "Cc2420 datasheet," http://inst.eecs.berkeley.edu/cs150/Documents/CC2420.pdf.
- [26] "Cc1000 single chip very low power rf transceiver." http://focus.ti.com/lit/ds/symlink/cc1000.pdf.
- [27] M. Sha, G. Xing, G. Zhou, S. Liu, and X. Wang, "C-mac: Model-driven concurrent medium access control for wireless sensor networks," in *INFOCOM*, 2009.
- [28] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *HICSS*, 2000.
- [29] Q. Wang, M. Hempstead, and W. Yang, "A realistic power consumption model for wireless sensor network devices," in *IEEE SECON*, Reston, VA, September 2006.
- [30] IEEE, "Wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (lr-wpans)," in *IEEE Standard* 15.4, 2003.
- [31] P. Varshney, *Distributed Detection and Data Fusion*. New York, NY: Springer-Verlag, 1996.
- [32] D. Optimierung, R. E. Burkard, and R. E. Burkard, "Selected topics on assignment problems," 1999.
- [33] J. E. Hopcroft and R. M. Karp, "An n^{5/2} algorithm for maximum matchings in bipartite graphs," *SIAM J. Comput.*, vol. 2, no. 4, pp. 225–231, 1973.